

TTÜ Robotiklubi



TTÜ Robotiklubi
Tallinn University of Technology Robotics Club

Arduino progemine

ROBOT IGAÜHELE 2014

Kava

- Üldine teooria
- Arduino programmeerimisest lähemalt
- Praktilised ülesanded



Kahendsüsteem

Kahendsüsteem

- Kümnersüsteem – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Kahendsüsteem – 0, 1
- 0 – false (väär)
- 1 – true (tõene)

Kahendsüsteem

- Kümne süsteemi olemus
- Numbrikohtade arv näitab maksimaalset võimalikku väärtust
 - 1 (0-9) – 10^1
 - 2 (0-99) – 10^2
 - 3 (0-999) – 10^3
 - Jne...
- Numbrikohad näitavad vastava kümne astme kordajat
 - $1 = 1 * 10^0$
 - $45 = 4 * 10^1 + 5 * 10^0$
 - $5847,21 = 5 * 10^3 + 8 * 10^2 + 4 * 10^1 + 7 * 10^0 + 2 * 10^{-1} + 1 * 10^{-2}$

Kahendsüsteem

- Teisendamine kahendsüsteemist kümnendsüsteemi

- $10_2 = 1 \cdot 2^1 + 0 \cdot 2^0 = 2_{10}$

- $1111100111_2 = 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 999_{10}$

- Teisendamine kümnendsüsteemist kahendsüsteemi

- Arvu tuleb kahega täisarvuliselt jagada ning saadav arv moodustub jääkidest

- $18_{10} = 10010_2$

- $18 / 2 = 9$, jääk 0

- $9 / 2 = 4$, jääk 1

- $4 / 2 = 2$, jääk 0

- $2 / 2 = 1$, jääk 0

- $1 / 2 = 0$, jääk 1

Kahendsüsteem

- Numbrikohtade arv näitab maksimaalset võimalikku väärtust
 - 1 (0-1) = $2^1 = 2$ (0-1)
 - 2 (0-11) = $2^2 = 4$ (0-3)
 - 3 (0-111) = $2^3 = 8$ (0-7)
 - 4 (0-1111) = $2^4 = 16$ (0-15)
 - 5 (0-11111) = $2^5 = 32$ (0-31)
 - 6 (0-111111) = $2^6 = 64$ (0-63)
 - 7 (0-1111111) = $2^7 = 128$ (0-127)
 - 8 (0-11111111) = $2^8 = 256$ (0-255)
 - Jne...

Kahendsüsteem

- There are 10 types of people in the world: those who understand binary and those who don't.

```
decodeMessage(
    0; i < MAX_RES
    i = 0;
    s.length) {
    buf[loc
    RES_LEN)
} (
    buf[i] = 0;
    t(1000 + i);
```

Programmerimine

Programmeerimine

- IDE - Integrated Development Environment – integreeritud programmeerimiskeskond
 - Kasutajaliides/tekstiredaktor - user interface
 - Kompilaator - compiler
 - Silur - debugger
- Kood
 - Programmeerimiskeeled
 - Arduino, C, C++, C#, Python, Java...
 - Masinakood

Programmeerimine

- Keele struktuur
 - Muutuja
 - Funktsioon
 - Kontrollstruktuurid

Muuttuja

- Muuttuja
 - Nimi
 - Mäluaadress
 - Väärtus
 - Andmetüüp

Muutuja

- Nime järgi saab muutujale viidata koodis
- Üldiselt nimed algavad väiketähega
- Kasutada ainult inglise tähestiku tähti ja numbreid
- Analoogiad
 - Matemaatikas $x = 1$
 - Faili nimi

Muutuja

- Arvuti muutmälu koosneb üksikutest bittidest - 1/0
- Mäluaadress on otsene viide muutuja esimese biti füüsilisele asukohale arvuti muutmälus(RAM)
- Analoogiad
 - Ruut su vihikus (1, 2, 3, 4, ...)
 - Kaust/kettaseade kus fail asub

Muutuja

- Väärtus vastavalt andmetüübile
- Väärtus saab muutuda programmi töö käigus
- Analoogiad
 - Matemaatikas $x = 3$
 - Faili sisu (pilt, tekst, video, ...)

Muutuja

- Andmetüüp näitab muutuja sisu tüüpi
- Andmetüüp peab olema fikseeritud muutuja deklareerimisel
- Analoogiad
 - Matemaatikas $x = 3$, $\alpha = (1; 2)$, $P(3; 4)$
 - Faililaiend (.exe, .jpg, .txt)

Muutuja

- Andmetüübid

- int

- Täisarv (integer)
 - 2 baiti – 16 bitti – $2^{16} = 65536$
 - -32768 kuni 32767
 - unsigned 0 kuni 65535

- char

- Tähemärk (character)
 - 1 bait – 8 bitti - $2^8 = 256$
 - -128 kuni 127
 - ASCII tabel (American Standard Code for Information Interchange)

Muutuja

- Andmetüübid
 - boolean
 - Tõeväärtus (George Bool) – Inglise matemaatik
 - true false
 - 1 bait (vt. int)
 - long
 - „Pikad numbrid“
 - 4 baiti – 32 bitti – $2^{32} = 4294967296$
 - -2147483648 kuni 2147483647

Muutuja

- Andmetüübid

- float

- Ujukomaga arv(floating point number)
 - 4 baiti – 32 bitti
 - Väärtus vahemikus $-3.4028235 * 10^{38}$ kuni $3.4028235 * 10^{38}$
 - Täpsus 6-7 kohta
 - 123456700000000000
 - 0.00000000001234567
 - Arvutused vajavad palju jõudlust – kasutada mikrokontrollerite puhul ainult äärmisel vajadusel

- void

- Puuduv
 - Ainult funktsioonide puhul

Funktsioon

- Funktsioon
 - Nimi
 - $f(x)$, $g(x)$
 - Tagastatava väärtuse andmetüüp (return type)
 - Peab olema määratud
 - Argumendid
 - Deklareeritakse nagu muutujad.
 - Valikulised. Ei sõltu tagastatava väärtuse andmetüübist.

Funktsioon

- Mõned näited matemaatikast

- $y = 1$
 - Konstantne funktsioon
- $y = 2x$
 - Argument ja väärtus sama tüüpi
- $y = \sin(x)$
 - Argument ja väärtus eri tüüpi – nurk ja reaalarv
 - Funktsioonist on mitu eri versiooni radiaanid, kraadid, kümnendkraadid jne.

Kontrollstruktuurid

- Programmid toimivad step-by-step põhimõttel
- Kontrollstruktuurid juhivad programmi kulgu – reguleerivad mis on järgmiseks sammuks
 - if
 - if...else
 - if...else if
 - for
 - while
 - do...while
 - switch case
 - break
 - continue
 - return

Kontrollstruktuurid

- if
 - Ainult siis, kui tingimus on täidetud, käivitatakse järgnev programmi plokk
- if...else
 - Kui tingimus on täidetud käivitatakse (if) järgnev programmi plokk, vastasel juhul (else) järgnev
- if...else if
 - Kui tingimus on täidetud käivitatakse (if) järgnev programmi plokk, vastasel juhul testitakse (else if) tingimust ja kui see on täidetud, siis käivitatakse (else if) järgnev programmi plokk
 - (else if) tingimusi võib olla mitu järjest, iga kord testitakse
 - Kui esimene (if) või (else if) tingimus on täidetud, siis edasi ei testita tingimusi
- for
 - Sarnane (while) struktuuriga
 - Korratakse programmi plokki seni, kuni tingimus on täidetud
 - Üldiselt on tingimuseks arvu väiksem olek konstandist, kusjuures arvu suurendatakse iga kord 1 võrra – korratakse kindel arv kordi

Kontrollstruktuurid

- switch case
 - Käivitatakse vastavalt muutuja väärtusele plokk
 - Iga plokk peab olema lõpetatud (break) struktuuriga
- while
 - Sarnane (for) struktuuriga
 - Korratakse kuni tingimus on täidetud, korduste arv sõltub üldjuhul olukorrast ja ei ole fikseeritud
- do...while
 - Käivitatakse plokk ja peale seda kontrollitakse (while) tingimust, mille järgi otsustatakse, kas tuleb plokki korrata
 - Plokk käivitatakse vähemalt ühe korra

Kontrollstruktuurid

- break
 - „murtakse“ välja koodi plokist
- continue
 - Minnakse korratava (for, while) struktuuri ploki algusesse tagasi ja testitakse uuesti tingimust
- return
 - Väljutakse funktsioonist
 - Kui funtsiooni andmetüüp ei ole (void), saadetakse ka mingi muutuja väärtus tagasi, kust funktsioon käivitati

Handwritten text in a script, likely a form of shorthand or a specific dialect, written on aged, torn paper. The text is arranged in four horizontal lines, with some characters appearing to be numbers or symbols. The paper is heavily creased and has irregular edges.

Arduino keel

Arduino

- Arduino keel põhineb C keelel
- C ja C++ on väga mitmekülgsed ja rohkete võimalustega ning annavad väga tugeva põhja programmeerimise õppimisel

Arduino

- Arduino/C keeles:

- Ei ole oluline teksti joondus
- Eraldatakse koodi plokid loogeliste sulgudega { }
- Üherealise ploki puhul pole sulud vajalikud
- Lõpetatakse laused semikooloniga ;
- Nimed on tõstutundlikud (case sensitive)
- Kommentaari rida algab kahe kaldkriipsuga //
- Kommentaari lõik algab kaldkriipsu ja tärniga /* ja lõpeb täрни ja kaldkriipsuga */

Arduino

- Muuttuja deklareerimine

```
int x;
```

```
int x = 1;
```

```
char t = "a";
```

```
char t = 97;
```

Arduino

- Funktsioon
 - Liidetakse kaks muutujat mis deklareeritakse funktsiooni sees

```
void liida() {  
    int a = 1;  
    int b = 3;  
    int c;  
    c = a + b;  
}
```

Arduino

- Funktsioon
 - Liidetakse kaks muutajat a ja b, mis antakse argumentidena funktsioonile ja tagastatakse kolmanda muutuja C väärtus

```
int liida(int a, int b) {  
    int c;  
    c = a + b;  
    return c;  
}
```


Arduino

- Funktsioon
 - Sama, mis eelmine funktsioon, aga tehe sooritatakse muutuja deklareerimisel

```
int liida(int a, int b) {  
    int c = a + b;  
    return c;  
}
```

Arduino

- Funktsioon
 - Tulemus sama, mis eelmistel funktsioonidel, aga uut muutujat ei deklareerita ja saadetakse tagasi kohe argumentide summa väärtus

```
int liida(int a, int b) {  
    return a + b;  
}
```

Arduino

- Funktsioon
 - Funktsioonis mingi Funktsioon() luuakse muutuja c ja seejärel omistatakse talle väärtus mille tagastab funktsioon liida()

```
int liida(int a, int b) {  
    return a + b;  
}
```

```
void peamineFunktsioon() {  
    int c;  
    c = liida(1, 3);  
}
```

Arduino

- if...else struktuur
 - Funktsioon max() leiab kahest arvust suurima

```
int max(int a, int b) {  
    if (a < b) {  
        return b;  
    } else {  
        return a;  
    }  
}
```

Arduino

- if...else struktuur
 - Teistsugune vormindus – funktsionaalsus sama

```
int max(int a, int b)
{
    if (a < b)
    {
        return b;
    }
    else
    {
        return a;
    }
}
```

Arduino

- if...else struktuur
 - Üherealise ploki puhul pole loogelisi sulge vaja

```
int max(int a, int b) {  
    if (a < b) return b;  
    else return a;  
}
```

Arduino

- if...else if...else struktuur
 - Kui $a < b$ tagastatakse b , mis on suurem, kui $a > b$, tagastatakse a , mis on suurem
 - Kui kumbki tingimus pole täidetud tagastatakse false ehk 0

```
int max(int a, int b) {  
    if (a < b) {  
        return b;  
    } else if (a > b) {  
        return a;  
    } else {  
        return false;  
    }  
}
```

Arduino

- Enne esimest korda plokki sisenemist luuakse muutuja `int i = 0`
- Iga kord kui plokki korratakse kontrollitakse, kas tingimus `i < 256` on täidetud
- Iga kord kui plokk lõpetatakse sooritatakse tehe `i = i + 1`

```
for (int i = 0; i < 256; i = i + 1) {  
    analogWrite(PWMPin, i);  
    delay(10);  
}
```


Arduino

- Enne ploki sisenemist luuakse muutuja `int i = 0`
- Ploki korratakse seni, kuni tingimus `i < 256` on täidetud
- Ploki lõpus suurendatakse muutujat `i` ühe võrra

```
int i = 0;
while (i < 256) {
    analogWrite(PWMPin, i);
    delay(10);
    i++;
}
```

Arduino

- Enne ploki sisenemist luuakse muutuja `int i = 0`
- Esimest korda käivitatakse plokk ilma tingimust kontrollimata
- Plokk korratakse seni, kuni tingimus `i < 256` on täidetud
- Ploki lõpus suurendatakse muutujat `i` ühe võrra

```
int i = 0;
do {
    analogWrite(PWMPin, i);
    delay(10);
    i++;
} while (i < 256)
```

Arduino

- `switch` struktuur valib vastavalt muutuja väärtusele koodi ploki, mis käivitatakse
- Kui muutuja väärtus ei sobi ühegi `case` puhul, siis käivitatakse `default`
- `default` on valikuline

```
switch (muutuja) {  
    case 1:  
        //tee midagi kui muutuja = 1  
        break;  
    case 2:  
        //tee midagi kui muutuja = 2  
        break;  
    default:  
        //kui muutuja väärtus on midagi muud siis tee seda  
}  
}
```

Arduino

- Kui $x > 40$ ja $x < 120$, siis käsk `continue` viib programmi automaatselt ploki lõppu, st alustatakse uut ploki ilma viimast lõpuni tegemata

```
for (x = 0; x < 255; x++) {  
    if (x > 40 && x < 120){  
        continue;  
    }  
    digitalWrite(PWMPin, x);  
    delay(50);  
}
```

Arduino

- Väärtuste nimetamine
 - `#define` annab konstantsele väärtusele nime, mis alati enne programmi kompileerimist asendatakse väärtusega
 - Lõpus pole semikoolonit, kuna see pole osa programmist, vaid instruksioon kompilaatorile

```
#define LEDpin 13
```

```
#define vasakServoseisab 90
```

Arduino

- Teekide (library) lisamine
 - `#include` käseb kompilaatoril kompileerida lisaks meie failile veel ka teegi faili(d)
 - Teek on kollektsioon funktsioone, klasse vms komponente, mis on mõeldud korduvkasutuseks programmides
 - Lõpus pole semikoolonit, kuna see pole osa programmist, vaid instruktsioon kompilaatorile

```
#include <Servo.h>
```

```
#include <Ultrasonic.h>
```

Arduino

- Aritmeetilised operaatorid

- = Määramisoperaator, märgist vasakul olevale muutujale määratakse väärtuseks märgist paremal oleva väärtus
- + Liitmine
- - Lahutamine
- * Korrutamine
- / Jagamine
- % Jagamise jääk
 - $5 \% 2 = 1$
 - $9 \% 5 = 4$
 - $4 \% 5 = 4$

Arduino

- Võrdlemise operaatorid
 - == võrdub
 - != ei võrdu
 - < väiksem
 - > suurem
 - <= väiksemvõrdne
 - >= suuremvõrdne

Arduino

- Tõeväärtuse operaatorid
 - && Ja (and)
 - || Või (or)
 - ! Mitte (not)

```
if ((a == b) && (b < c) || !( c >= (40 % 7))) {  
    //siis tehakse midagi  
}
```

a = 4, b = 4, c = 3 ?

Arduino

- Liitoperaatorid

- $x++$

- $x = x + 1$

- $x--$

- $x = x - 1$

- $x += 2$

- $x = x + 2$

- $x -= 2$

- $x = x - 2$

- $x *= 2$

- $x = x * 2$

- $x /= 2$

- $x = x / 2$

Arduino

- Massiiv (array)
 - Massiiv on andmestruktuur, mis koosneb sama tüüpi elementidest ja on indekseeritav
 - Massiivid võivad olla ühe või mitmemõõtmelised

```
int x[5];
```

```
x[0] = 33;
```

```
x[1] = 56;
```

```
x[2] = 34;
```

```
x[3] = 34;
```

```
x[4] = 7;
```

```
int x[5] = {33, 56, 34, 34, 7};
```

Arduino

- Mitmemõõtmelised massiivid

```
int maleLaud[8][8];
```

```
int ruubikuKuubik[3][3][3];
```

```
int ruubikuHyperkuup[3][3][3][3];
```

```
maleLaud[0][0] = valgevanker;
```

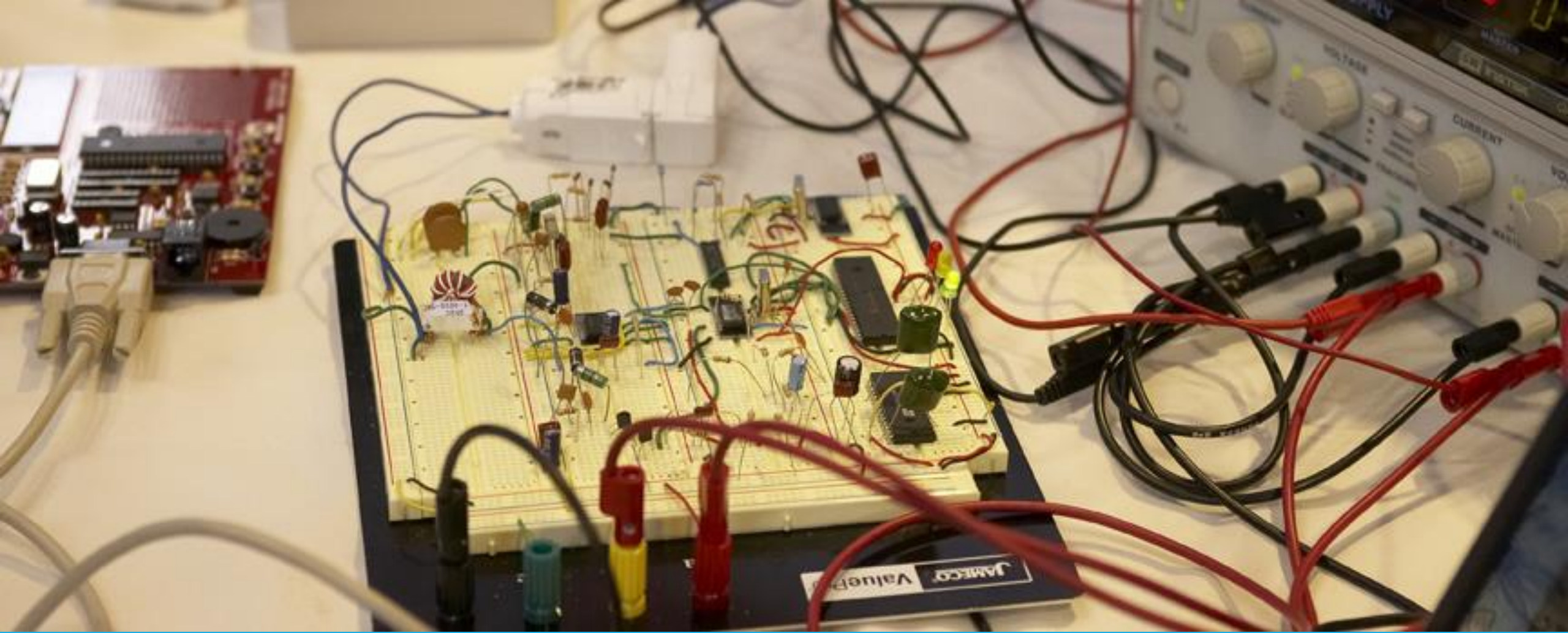
```
maleLaud[7][7] = mustvanker;
```

Arduino

- Arduino programmi kohustuslikud elemendid

```
void setup() {  
    // See funktsioon käivitatakse esimesena  
  
}
```

```
void loop() {  
    // Seda funktsiooni korratakse lõputult  
  
}
```

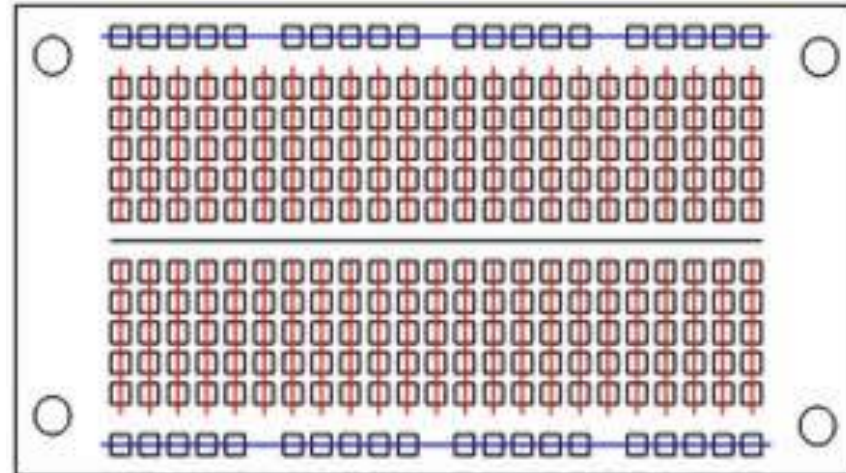


FUN FUN FUN

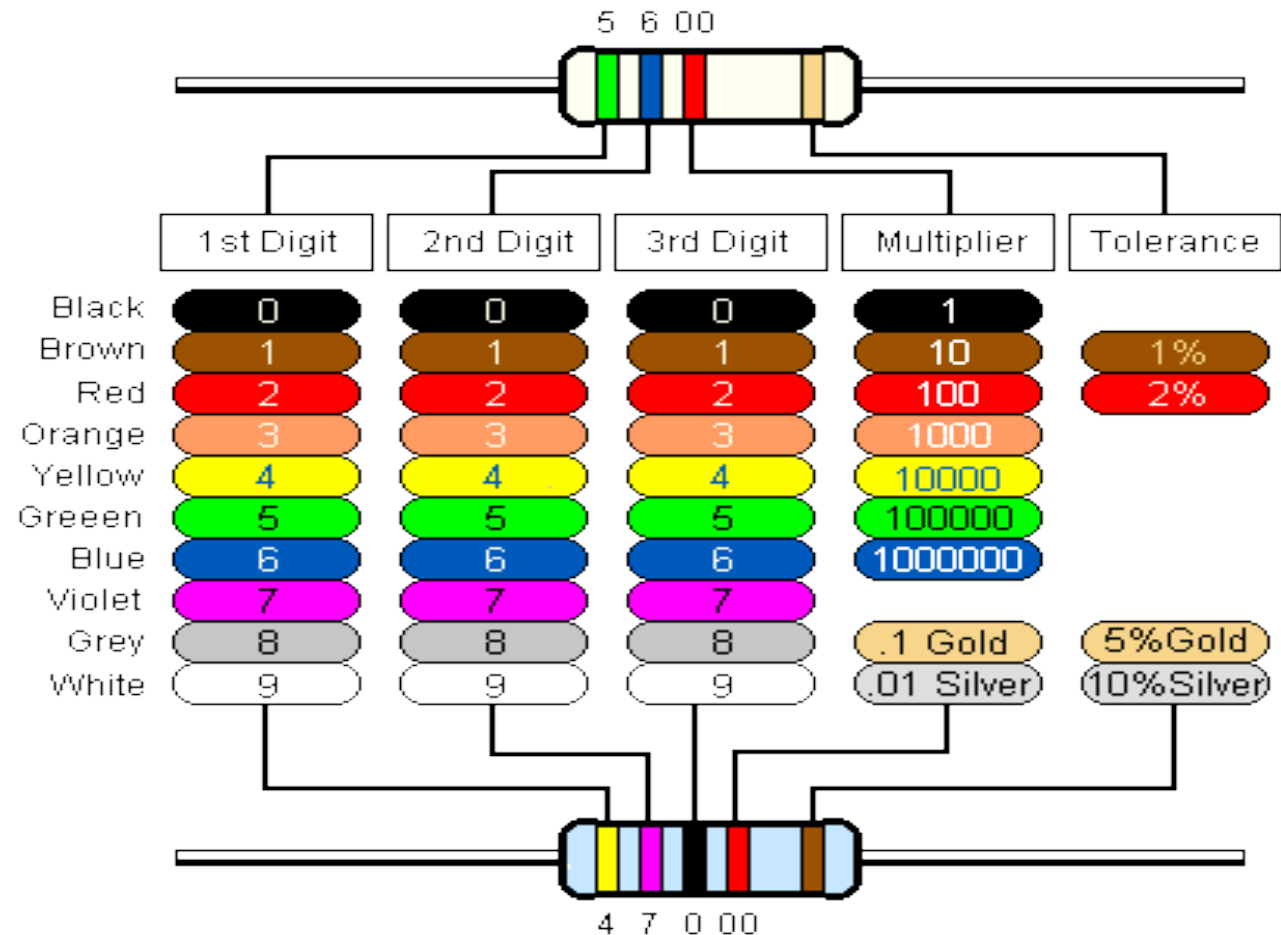
Praktilised tööd

- LED põlema
- LED vilkuma
- Servo tööle
- Servo tööle potekaga
- Ultraheli anduri lugem serialisse
- Ultraheli anduri järgi PWM-ga LED-i heledus
- Lõputöö

Protoplaat

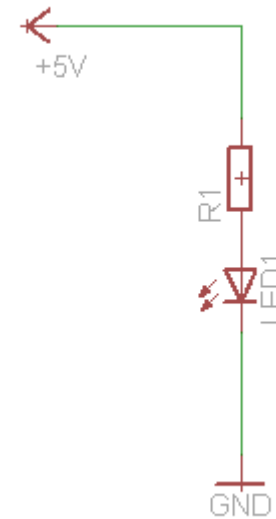
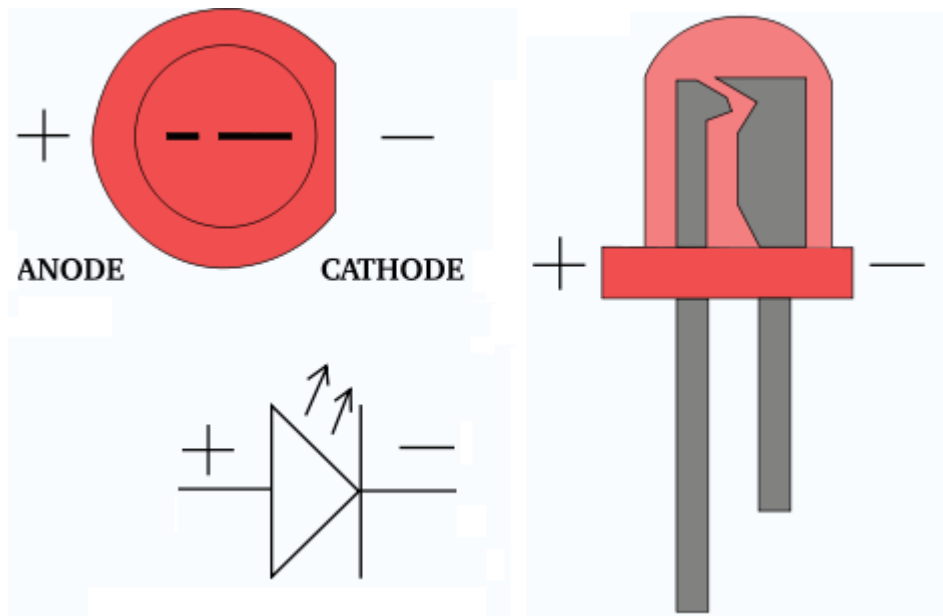


Takistite värvikoodid



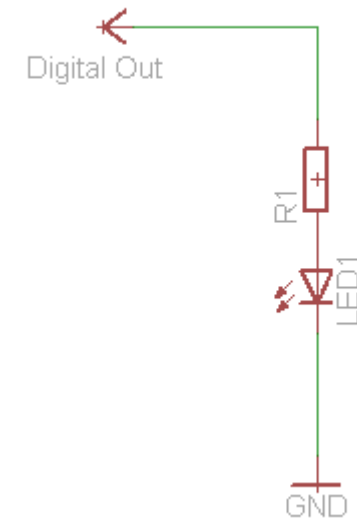
LED põlema

- Arvutada takisti väärtus
- Ühendada elektriskeem



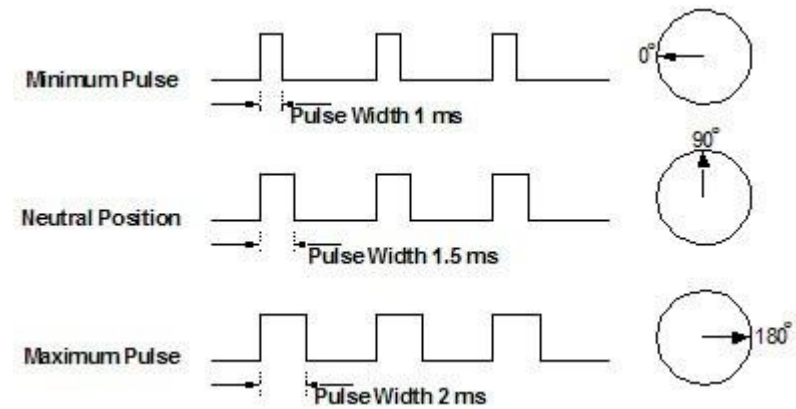
LED vilkuma

- Kirjutada kood
- Vahetada skeemi sisend



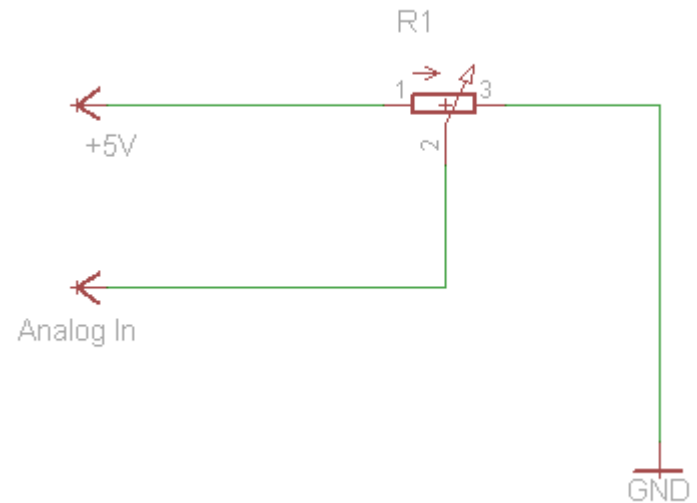
Servo tööle

- Kirjutada kood
- Ühendada servo



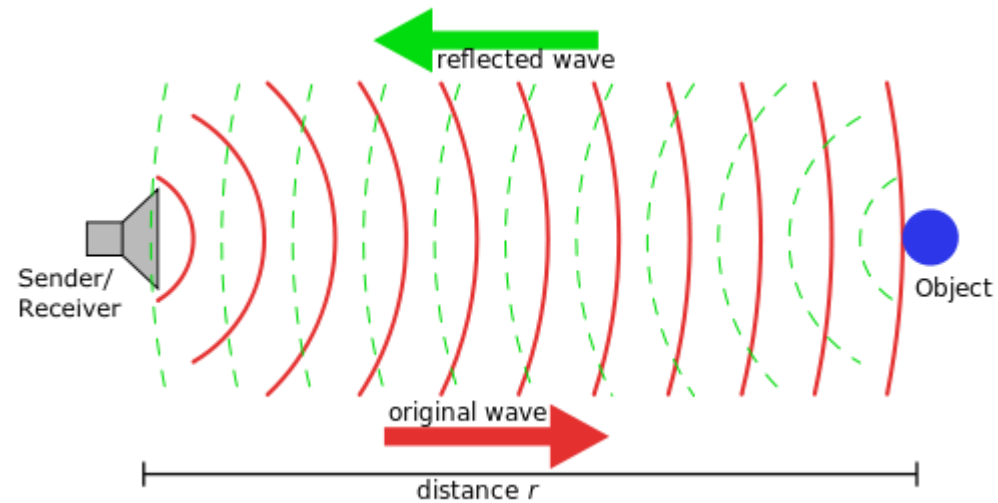
Servo tööle potekaga

- Kirjutada kood
- Ühendada servo
- Ühendada skeem



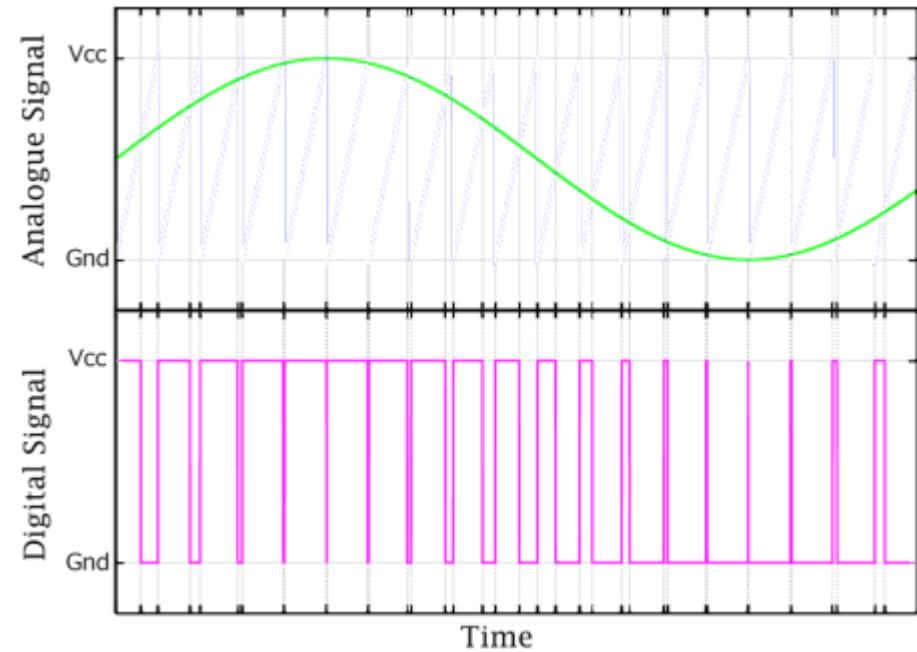
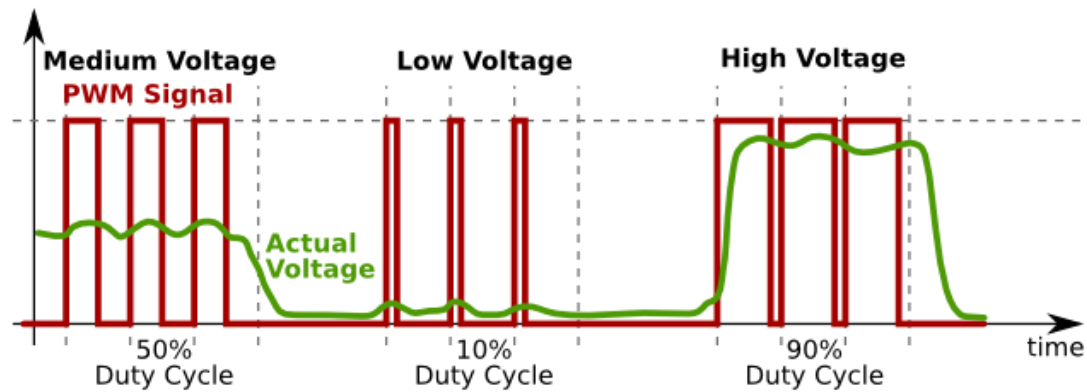
Ultraheli andur serialiga

- Kirjutada kood
- Ühendada andur



PWM LED ultraheli juhtimisega

- Kirjutada kood
- Ühendada andur



Lõputöö

- Ultraheli andur loeb kaugust
 - Vähemalt 3 LED-i annavad tagasisidet selle kauguse kohta
 - Potentsiomeeter häälestab seadme tundlikkust
 - Servo liigub vastavalt kaugusele
-
- Kui on häid ideid võib ka midagi muud teha

Viited

- <https://www.google.ee/>
- <http://arduino.cc/en/Reference/HomePage>
- <http://www.cplusplus.com/>
- <http://www.youtube.com/playlist?list=PL1D10C030FDCE7CE0>
- <http://www.electronics-tutorials.ws/index.html>



Ingvar Lond
Rain Ellermaa

e-mail: rain.ellerm@robotiklubi.ee

tel. +372 53408660