Tallinna Tehnikaülikooli, Tartu Ülikooli ja Maaülikooli õppejõud hakkasid arutlema selle üle, kumb on parem - naine või armuke.

"Naine peab olema", ütleb Maaülikooli õppejõud , "Töötad terve päeva põllul, pärast naine annab süüa, veidi kangematki ja hea voodis külje alla pugeda."

"Mis naine - armukesed" , ütleb TÜ noor õppejõud. "Endal eluaset pole eriti vajagi, ühikas naistudengeid täis. iga õhtu jaoks on keegi. Saab hea kõhutäie ja voodirõõmud kauba peale."

TTÜ oma muigas selle peale " ehh teid... Mõlemad peavad olema. Armukesele ütled, et oled naise juures, naisele ütled, et oled armukese juures, ise istud TTÜs ja aina lahendad ja lahendad kõrgema matemaatika ülesandeid…."

EEM3010 SISSEJUHATUS MEHHATROONIKASSE

Sügis 2024

Algoritm, programm ja Arduino algajatele (osa 1)

Martin Jaanus NRG-308
martin.jaanus@ttu.ee 620 2110, 56 91 31 93

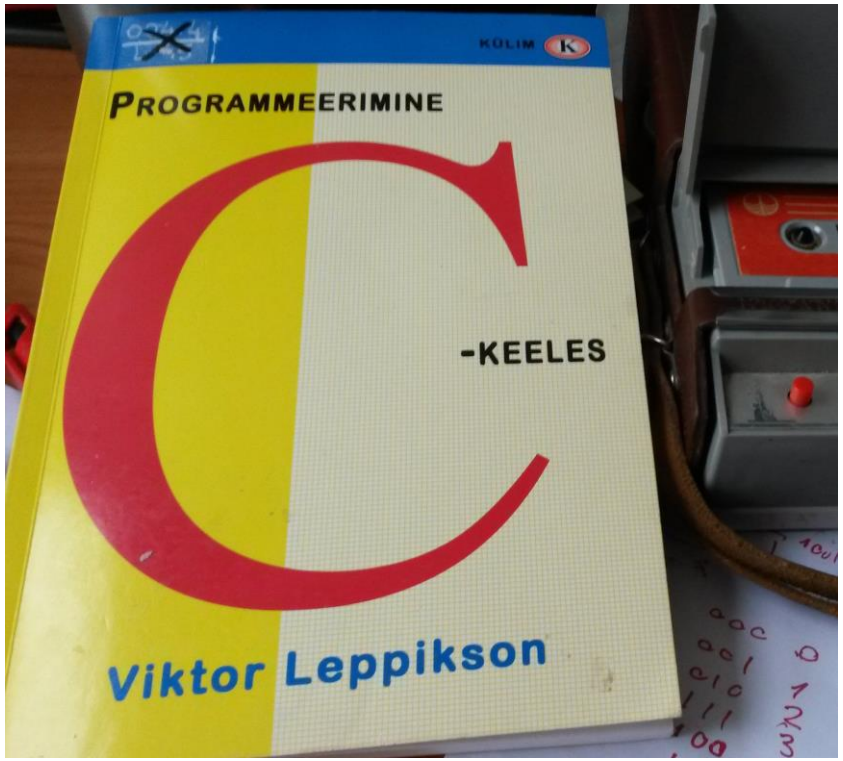
Õppetöö : http://isc.ttu.ee
Õppematerjalid : http://isc.ttu.ee/martin

Teemad

- Arduino tutvustus

- Algoritmist programmini
- Arduino programmeerimise eripära
- Slaididel on palju näiteid. Proovige need iseseisvalt järgi !
- Programmeerimisest madala taseme keeltes (C,ASM)

Soovitan hankida kui võimalik ! →



Algoritm

- **Algoritm** on astmeline tegevusjuhis, juhend või eeskiri mingi tegevuse sooritamiseks või eesmärgi saavutamiseks. Kõige sagedamini kasutatakse seda terminit matemaatilise ülesande lahendamiseks mõeldud eeskirja kohta. Algoritmi esitust mingis formaalses keeles, tavaliselt programmeerimiskeeles või masinakoodis, nimetatakse **arvutiprogrammiks.** (Wikipedia)

- **Arvuti teeb seda,mida me käsime,mitte seda,mida me tahame !!!!!!!** (idee poolest võiks see ju nii olla, tänapäeval teeb inimene pigem seda, mida arvuti käsib)

- Enne programmi kirjutamist mõelge läbi tegevuste algoritm ja veelgi parem, pange see kirja !
- Samuti kommenteerige enda loodud programmi - mis otstarbel on mingi osa loodud.

Teil endil on pärast lihtsam (pea ei ole prügikast).

Kui kirjutate mingi programmi ja see ei täida teie ootusi, mõelge uuesti selle lause peale.

Algoritm

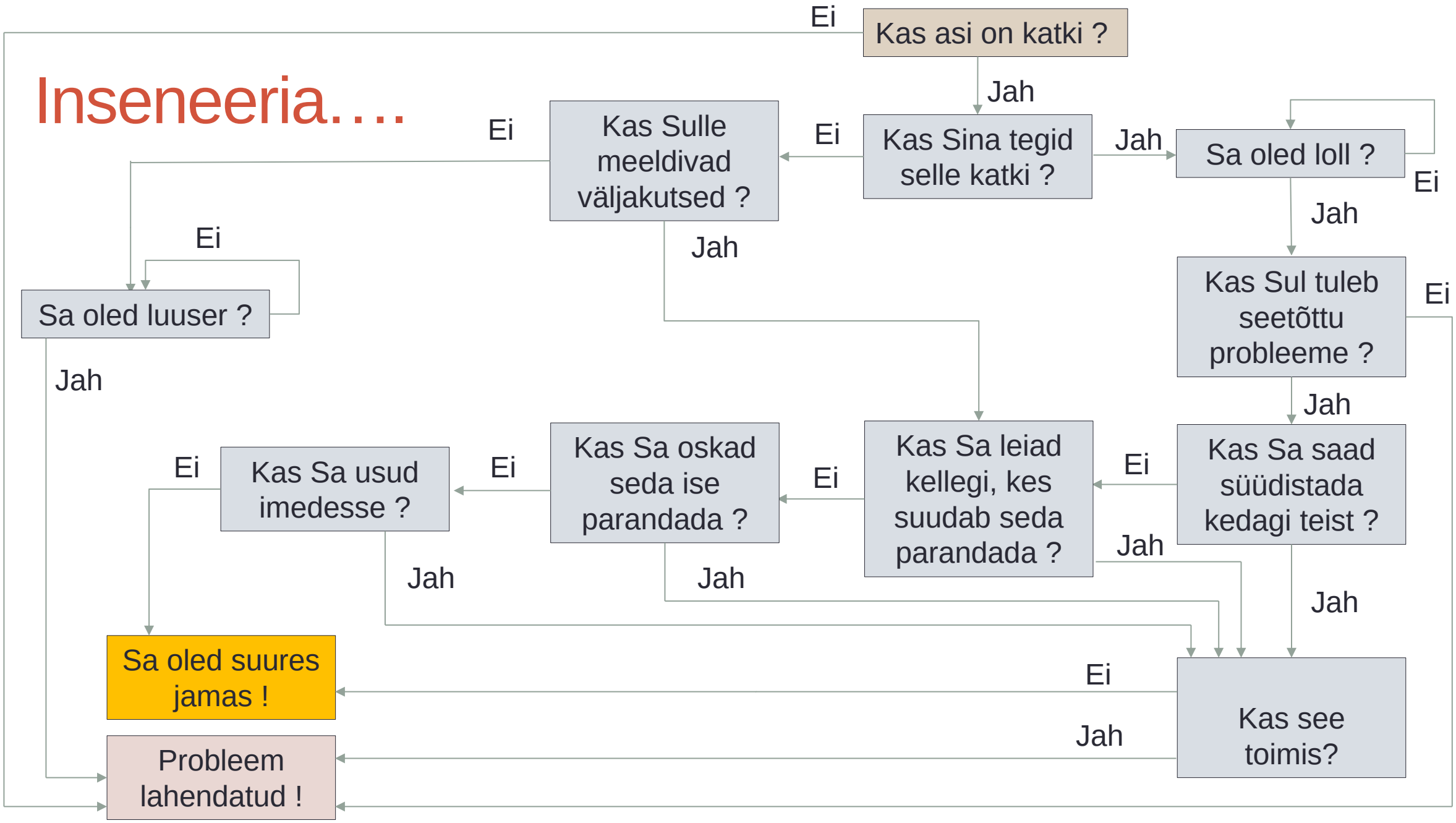
- Reaaleluline IT huumor

Naine saadab doktorikraadiga programmeerijast mehe poodi:
“Nii. Osta üks leib , kaks piima , ja noh,
osta endale siis see neetud üks õlu ka.
…..Ahjaa – kui poes on mune, osta 12 .”

Veidi aja pärast võtab naine teda “lahkelt” vastu.
- “Ma ju ütlesin – üks õlu !!!!!”
- “Aga poes oli mune …”

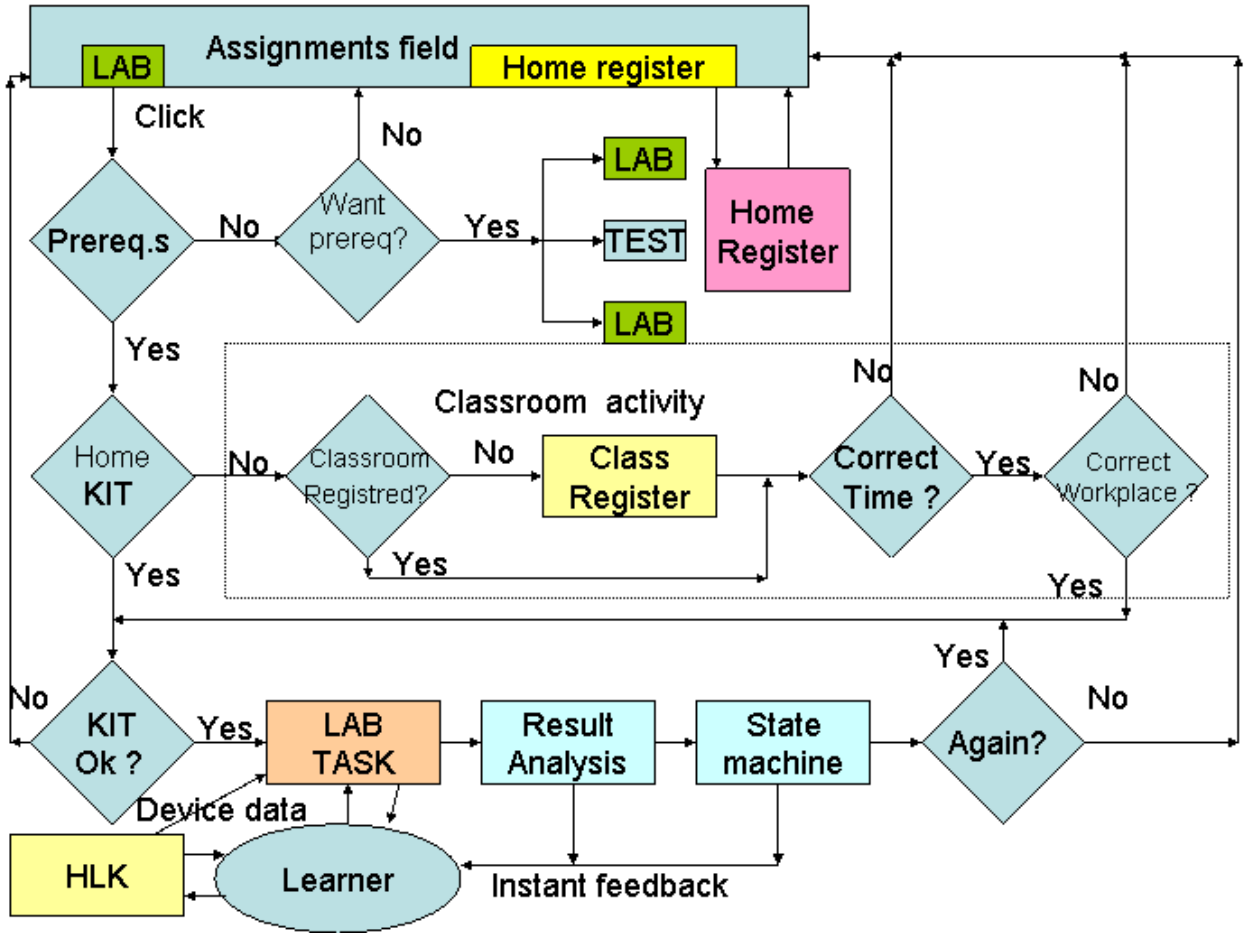
- Arvuti ei oska lugeda “ridade vahelt”
- Algoritm tuleb koostada selliselt, et arvuti sellest aru saaks !!!

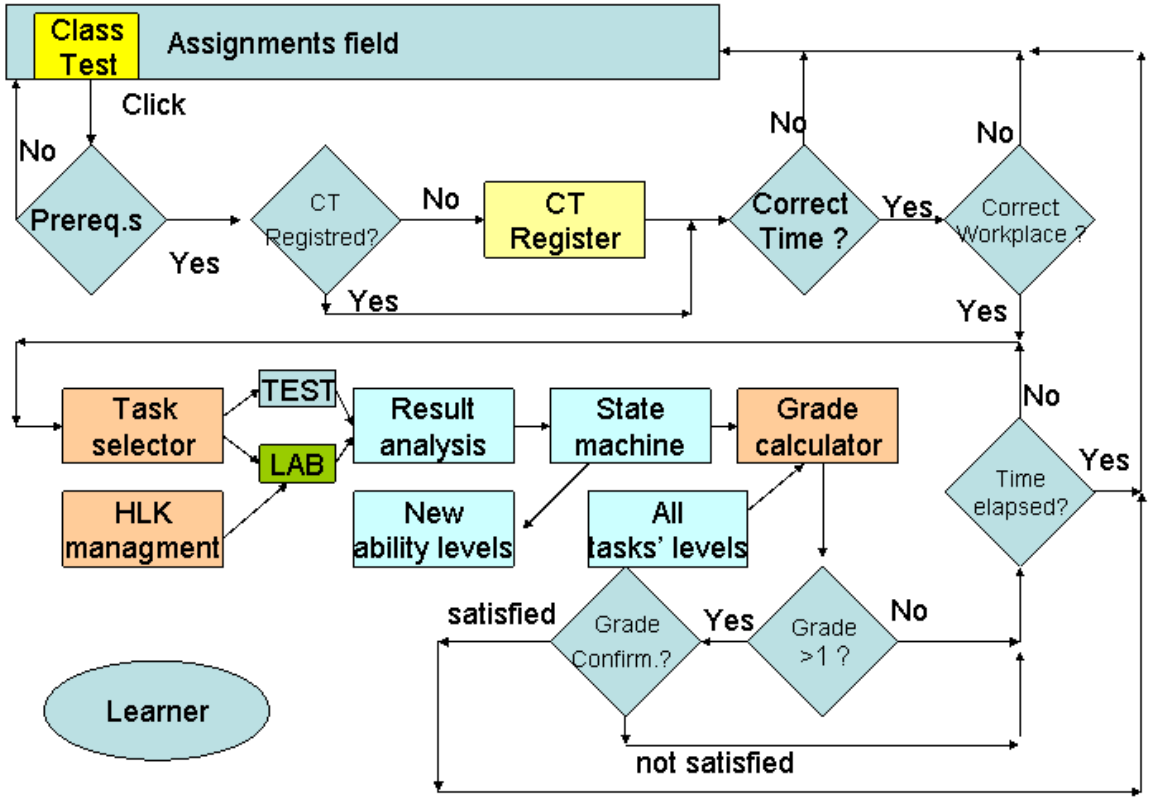
Inseneeria….

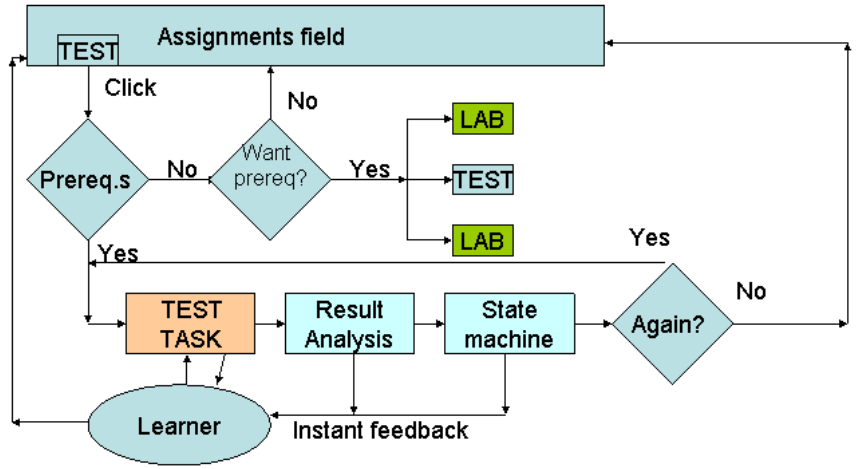

Kas asi on katki ?
Ei
Jah
Kas Sina tegid selle katki ?
Ei
Jah
Sa oled loll ?
Ei
Jah
Kas Sulle meeldivad väljakutsed ?
Ei
Jah
Sa oled luuser ?
Ei
Jah
Kas Sul tuleb seetõttu probleeme ?
Ei
Jah
Kas Sa saad süüdistada kedagi teist ?
Ei
Jah
Kas Sa leiad kellegi, kes suudab seda parandada ?
Ei
Jah
Kas Sa oskad seda ise parandada ?
Ei
Jah
Kas Sa usud imedesse ?
Ei
Jah
Sa oled suures jamas !
Kas see toimis?
Ei
Jah
Probleem lahendatud !

ISC veebikeskkond

- M.J.2011



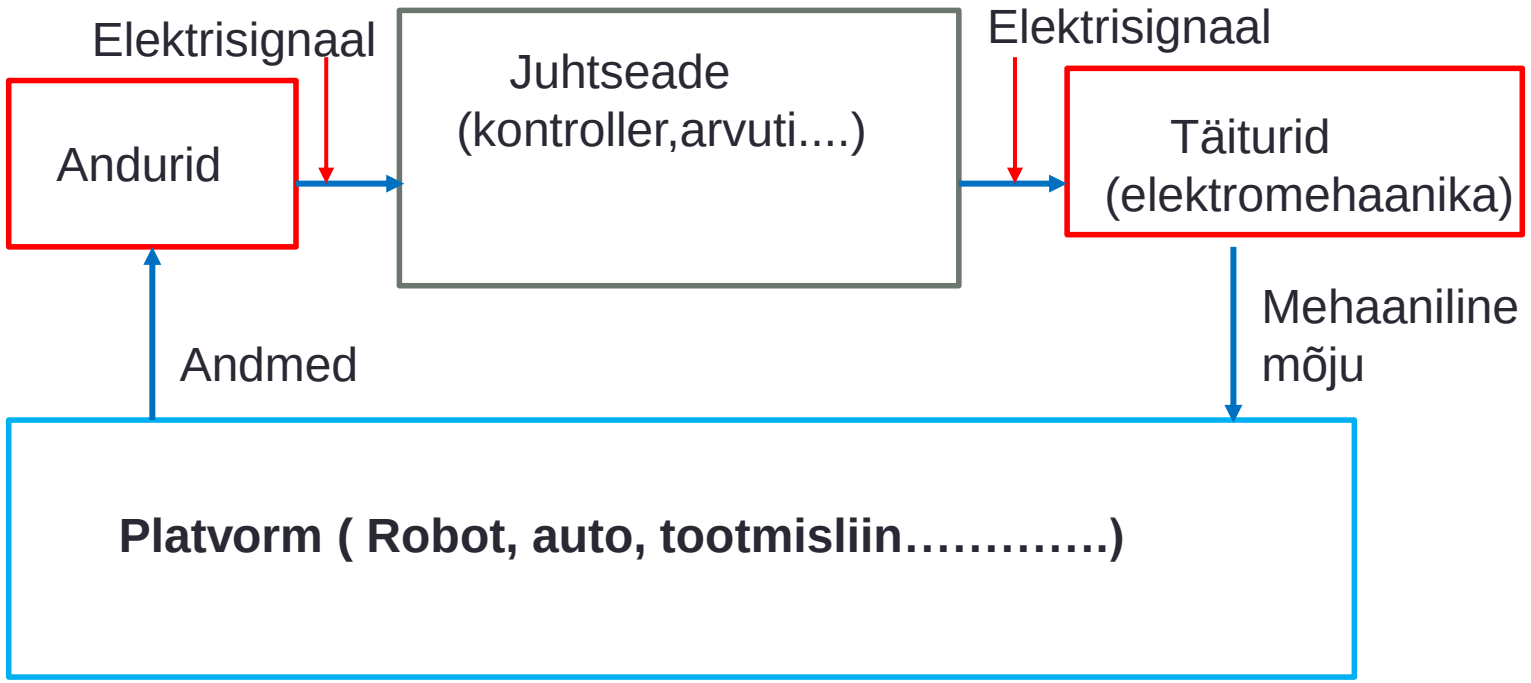




Algoritm ja mehhatroonika

- Teeme joonejärgija

Robotplatvormi struktuur:
NB ! **Tagasisidestatud** süsteem !



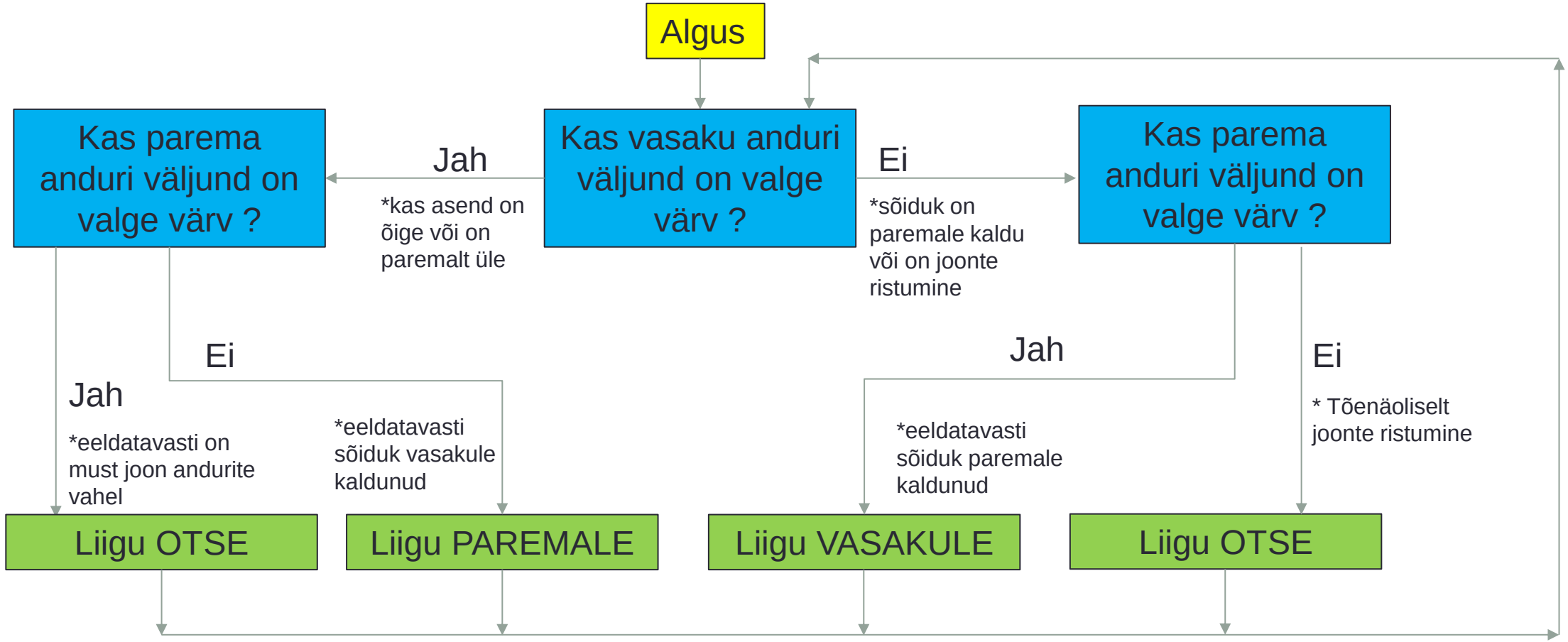
Eesmärk – robotplatvorm liigub edasi mööda mahatõmmatud musta joont, mis on valgel taustal

Andurid:
- Kaks joonejärgmisandurit (vasakul ja paremal)

Täiturid:
- Kaks elektrimootorit (vasakul ja paremal)

Juhtseade:
- Arduino Uno arendusplaat

Joonejärgija algoritm



Kas lõppu ei olegi ? – tsükliline programm

Joonejärgija algoritm

- Mida tähendab – värv on valge , värv on must ? Joonejärgimisanduri väljundis pinget kas 0 V või ligikaudu 5 V (sõltub andurist)

- Mida tähendab – liigu otse, paremale, vasakule... ? Sõiduki mootoritele rakendatakse pinge 6 V .
- **Otse** : töötavad mõlemad mootorid
- **Vasakule** : töötab parem mootor
- **Paremale** : töötab vasak mootor
- **Seis**: mõlemad mootorid seisavad

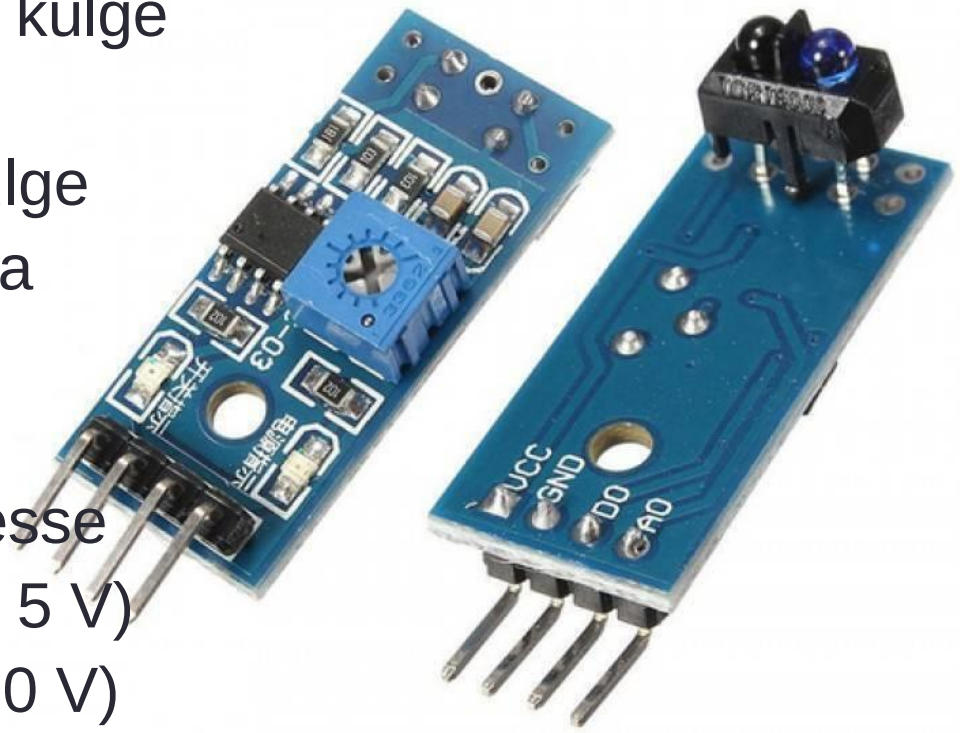
(lihtsuse mõttes ei ole hetkel realiseeritud reversit, ehk tagasiliikumist)

- Vaja realiseerida elektriskeem (otse Arduinoga mootoreid ei saa juhtida) ja nende juhtimisalgoritm

```
// algoritmi algus
 if (vasak_varv==valge)
 {
  if (parem_varv==valge)
  {
    Liigu_OTSE();
    }
    else
    {
      // parem värv on must
      Liigu_PAREMALE();
      }
 }
 else
 {
   // vasak värv on must
  if (parem_varv==valge)
 {
   Liigu_VASAKULE();
   }
   else
   {
     // parem värv on must, võimalik, et ristumiskoht
     Liigu_OTSE();
     }
   }
// algoritmi lõpp
```

Joonejärgija elektroonika

- Andurid – Hiinas valmistatud valmismoodulid – pilt Aliexpress
- 4 ühendusklemmi , saab otse ühendada Arduino külge
- GND ehk “maa” , ühendatakse üldjuhtme külge
- VCC ehk “toide” , ühendatakse 5 V toiteallika külge
- AO ehk analoogväljund – võimaldab tuvastada ka Intensiivsust



- DO ehk digitaalväljund - info on kodeeritud pingesse

1 - kui valgus ei peegeldu tagasi (pinge lähedane 5 V)

0 - kui valgus peegeldub tagasi (pinge lähedane 0 V)

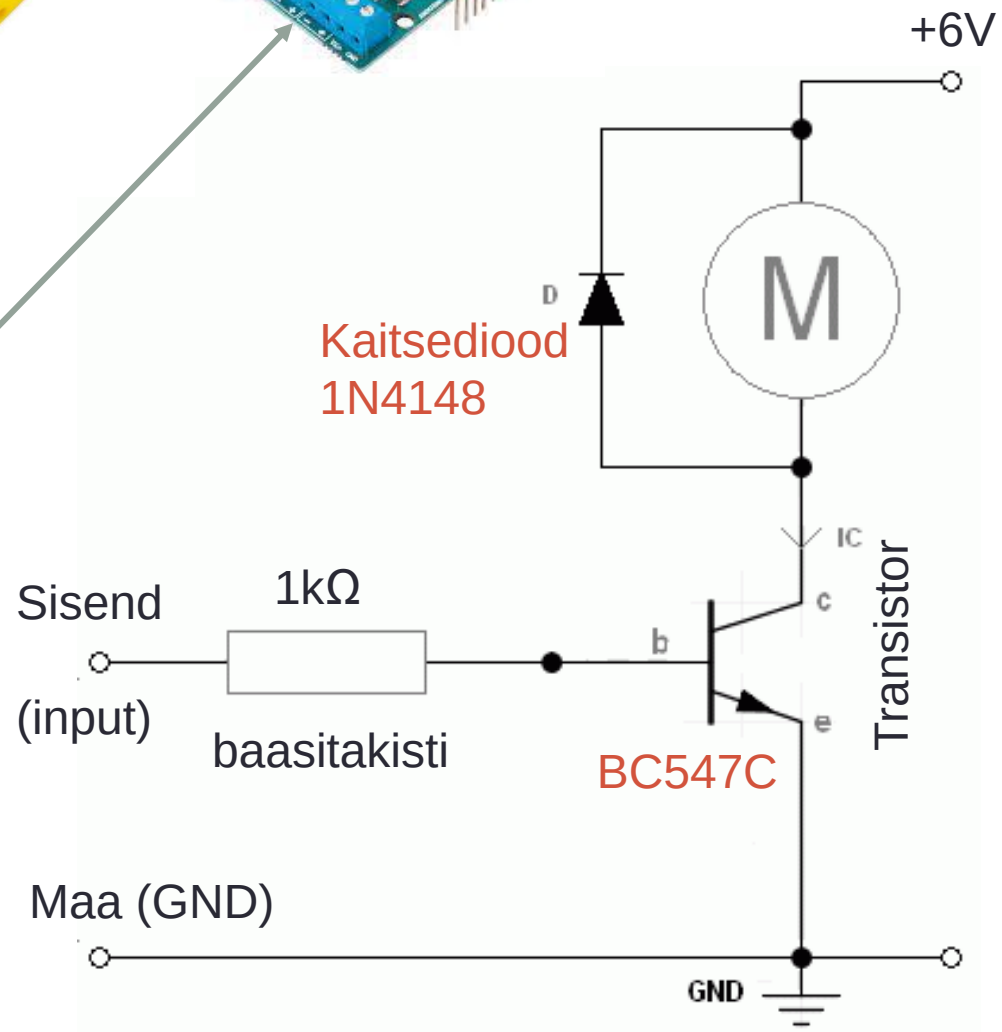
Reageerib ka välisele valgusele !

NB !! Otsige internetist kasutusnäiteid (kehtib iga “Hiina” mooduli kotha) !

Joonejärgija elektroonika

- Täiturid – reduktorajamiga alalisvoolu kollektormootorid (pilt Aliexpress)
- Kaks ühendusklemmi, millele rakendatakse pinge – pöörlemissuund sõltub polaarsusest
- Ei saa otse ühendada Arduinoga, vajab lisaelektroonikat (guugeldage Arduino Motor Shield näiteks) – saab poest osta/Hiinast tellida .
- Praeguses näites kasutame juhtimiseks transistori , mida omakorda juhib Arduino
- **Sisend:**

- 0 (nullilähedane pinge – mootor seisab)
- 1 (ligikaudu 5 V pinge – mootor töötab)
- See skeem ei võimalda pöörlemissuunda vahetada !



Arduino

- Arduino on arendus- ja prototüüpimiskeskkond ,mis sisaldab:

1. Mikrokontrollerit koos seda toetava elektroonikaga arendusplaadil (väga palju versioone)

2.Tarkvarakeskkondasid , sealhulgas valmisfunktsioone riistvaraga suhlemiseks

(NB ! Neid kahte saab kasutada ka teineteisest sõltumatult)

- Algaja saab Arduino platvormi vaadata pikka aega päris edukalt kui “musta kasti”, mis teeb seda, mida me sellel käsime .

Arduino UNO rev 3

- 19 üldotstarbelist digitaalset sisend-väljundklemmi
- Paljudel klemmidel lisafunktsioonid (analoogsisend, taimeriväljundid jne)

Klemmifunktsioonid tuleb defineerida , ühel klemmil võib olla mitu funktsiooni.

Programmeerimine on C keele baasil.

C keel

- 1958 kõrgkeel ALGOL (tänapäevaste kõrgkeelte eelkäija)
- 1969 Bell Laboratories , keel B
- Programeerimiskeel, mis oleks platvormist sõltumatu aga samas kiire.
- Kõrgkeeled ei suuda garanteerida ajakriitilistes kohtades kiirust ning kompaktsust
- 1973 programmeerimiskeel C (B täiustus), standardiseeriti 1983
- C++ 1989 (objektorienteeritud programmeerimine)
- C# 2002 (.net framework)
- Väga palju sugulaskeeli (sh php, java)
- Peavad olema üsna head eelteadmised riistvarast.
- Vahendid võivad tunduda primitiivsed, aga tegelikult on väga võimekad.

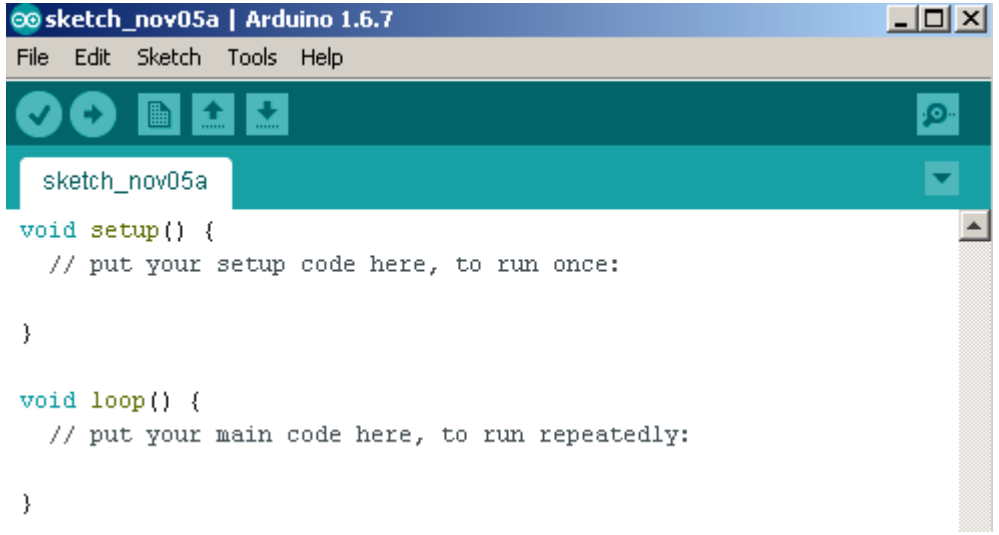
Arduino programmeerimise eripärad

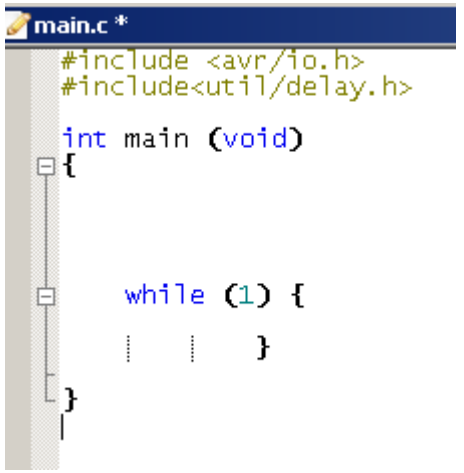
- Arduino arenduskeskkond kasutab olemuselt C++ kompilaatorit, ehk kehtivad C/C++ keele reeglid.
- Lisaks eelnevale on olemas „mugavus“täiendused ning abifunktsioonid, mida C keele maailmas ei tunta. Lisatud definitsioonid liidetakse programmi kompileerimisel. Kui on plaanis tulevikus hakata reaalses C/C++ keeles programmeerima, tasuks nendega äraharjumist vältida !
- Reeglina värvitud need keskkonnas oranžiks (aga ei pruugi)

Arduino programm (sketš)

Arduino koodis peavad olema ilmtingimata funktsioonid setup () , ehk kood täidetakse ühe korra ning loop () , kus koodi täidetakse tsüklis . Vastasel juhul tuleb kompileerimisel veateade !

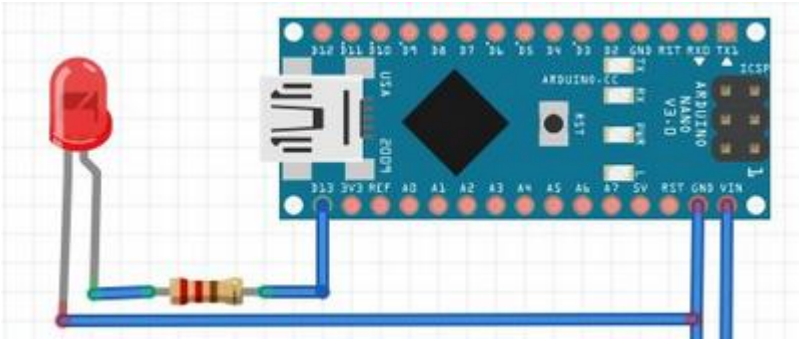
Arduino „keele“ kirjeldus : https://www.arduino.cc/reference/en/





C keelne „tühi“ programm

Arduinio koodinäide



- Vilgutame valgusdioodi

Blink | Arduino 1.6.7

File Edit Sketch Tools Help

Blink §

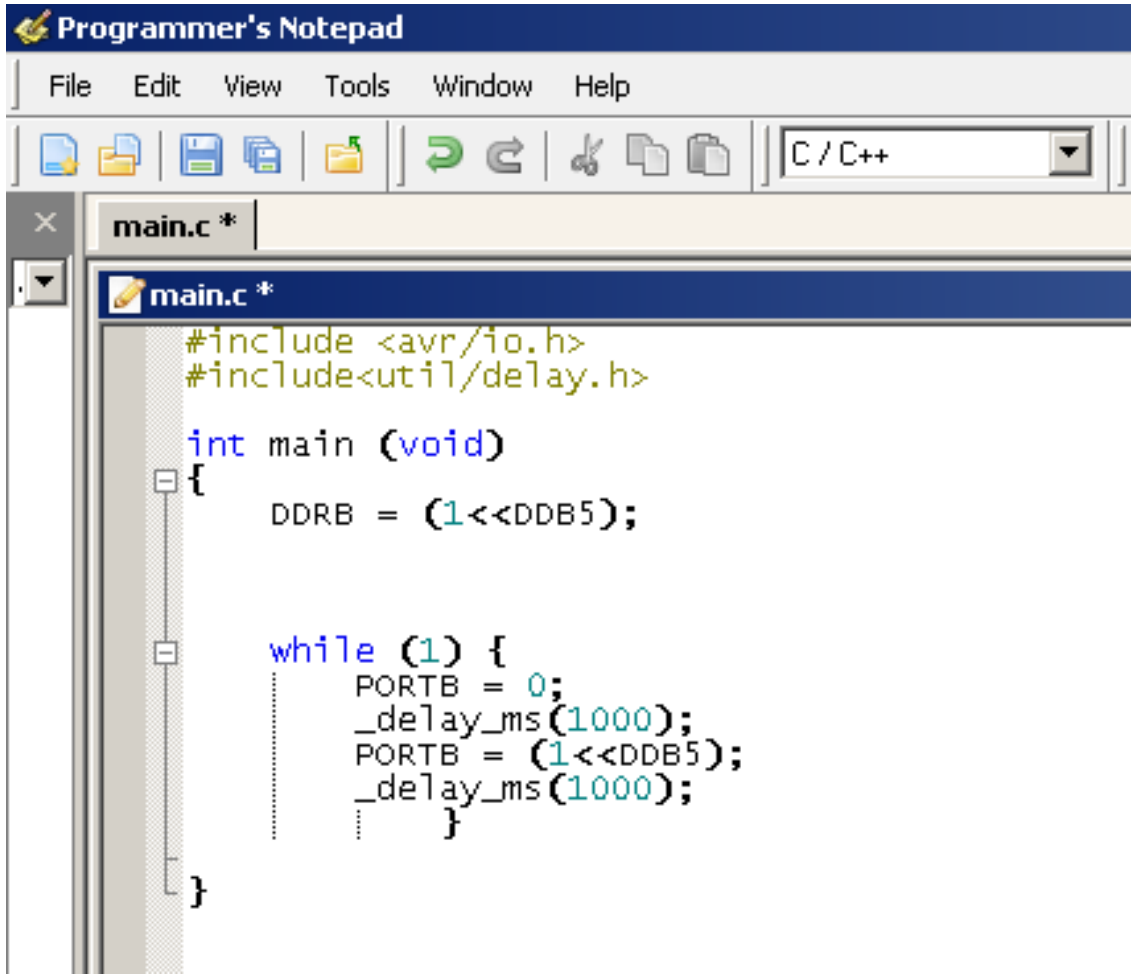
```
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

See kood eeldab töötamiseks ka õigesti koostatud elektriskeemi !

Kui ühendame nt valgusdioodi valepidi või mõne teise klemmi külge, siis see ei vilgu !

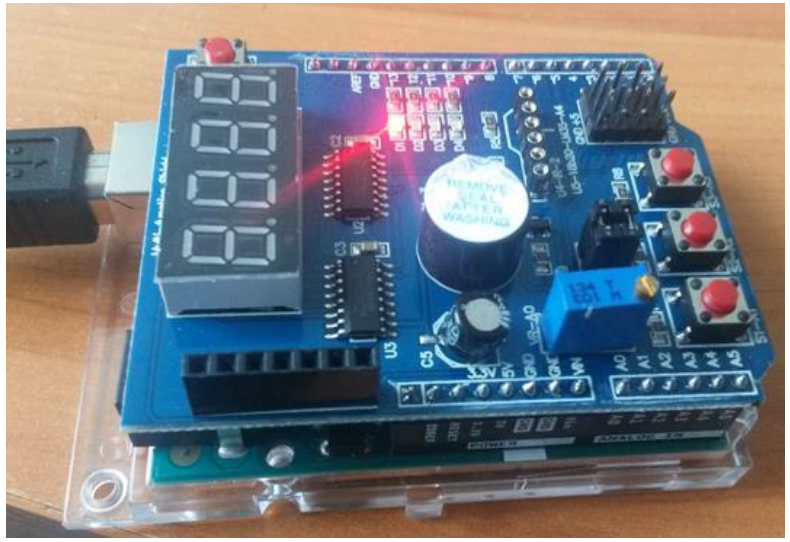
Tegu on nn. sardsüsteemiga kus kontroller programmeeritakse konkreetsest rakendusest lähtuvalt. (nt ,mingis rakenduses võib olla valgusdiood ka klemmi 11. küljes) , aga siis peab ka programm seda toetama !

C koodinäide, sama mikrokontroller



Assembleri koodinäide, sama mikrokontroller

- Näide tehtud õppeaine „Mikroprotsessorsüsteemid“ jaoks
- See on vaid funktsooni väljakutsumine ja kohustuslik programmi „kest“



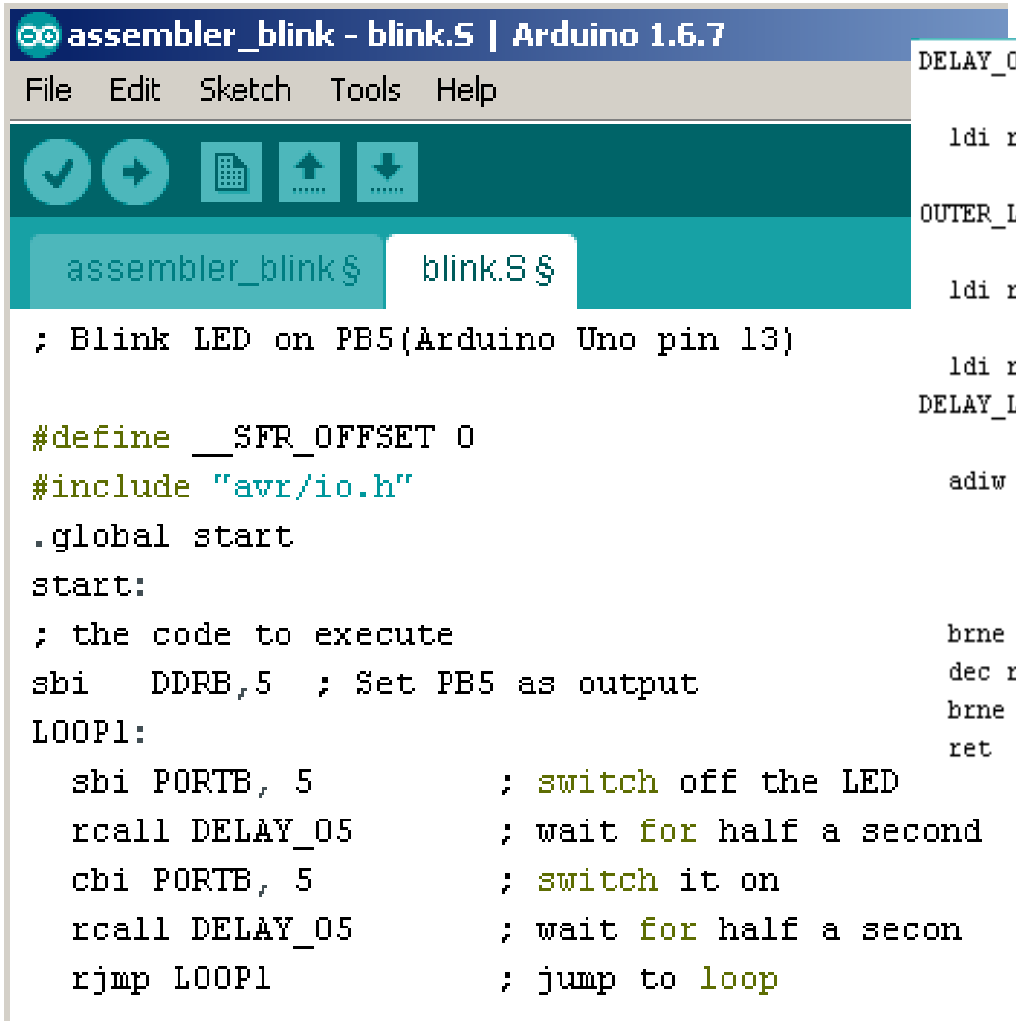
assembler_blink | Arduino 1.6.7

File Edit Sketch Tools Help

assembler_blink§ blink.S

```
extern "C" {
  // function prototypes
  void start();
;
}

void setup() {
  start();

}
void loop() {

}
```

Assembleri koodinäide, sama mikrokontroller




```
; Blink LED on PB5(Arduino Uno pin 13)

#define __SFR_OFFSET 0
#include "avr/io.h"
.global start
start:
; the code to execute
sbi   DDRB,5   ; Set PB5 as output
LOOP1:
  sbi PORTB, 5          ; switch off the LED
  rcall DELAY_05        ; wait for half a second
  cbi PORTB, 5          ; switch it on
  rcall DELAY_05        ; wait for half a secon
  rjmp LOOP1            ; jump to loop
```

```
DELAY_05:               ; the subroutine:

  ldi r16, 31           ; load r16 with 31

OUTER_LOOP:             ; outer loop label

  ldi r24,0    ; load registers r24:r25 with 1021, our new
                        ; init value
  ldi r25, 0          ; the loop label
DELAY_LOOP:             ; "add immediate to word": r24:r25 are
                        ; incremented
  adiw r24, 1           ; if no overflow ("branch if not equal"), go

                        ; back to "delay_loop"

  brne DELAY_LOOP
  dec r16               ; decrement r16
  brne OUTER_LOOP       ; and loop if outer loop not finished
  ret                   ; return from subroutine
```

Arduino klemmi seadistamine

- Käsk pinMode (klemm,parameter);
- Toimib kõikide sisend-väljundklemmidega.
- Enne konkreetse klemmi poole pöördumist on see käsk kohustuslik !

sketch_nov17a §

```
void setup() {

pinMode (11,OUTPUT); //Klemm 11 väljundiks
pinMode (12,INPUT); // Klemm 12 sisendiks
pinMode (13,INPUT_PULLUP); // kelmm 13 sisendiks
//ülestõmbava takistiga
}
```

Arduino digitaalsisend

- Eeldusel, et toitepinge Vcc=5V
- Muutuja a omandab väärtuse “TRUE” kui pinge klemmil 12 ületab 2 V enne vastava käsu täitmist !
- Muutuja b omandab väärtuse “TRUE” kui pinge klemmil 13 ületab 2 V enne vastava käsu täitmist !

NB ! Muutuja a väärtustatakse vaid siin !
Kui sisend muul ajal muutub, a väätus ei muutu !

Arduino digitaalväljund

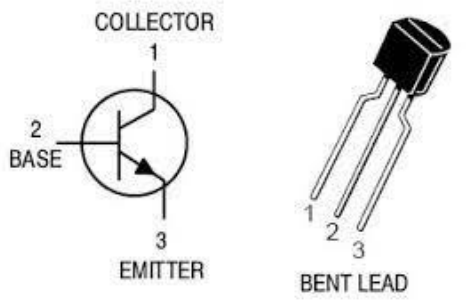
sketch_nov17a §

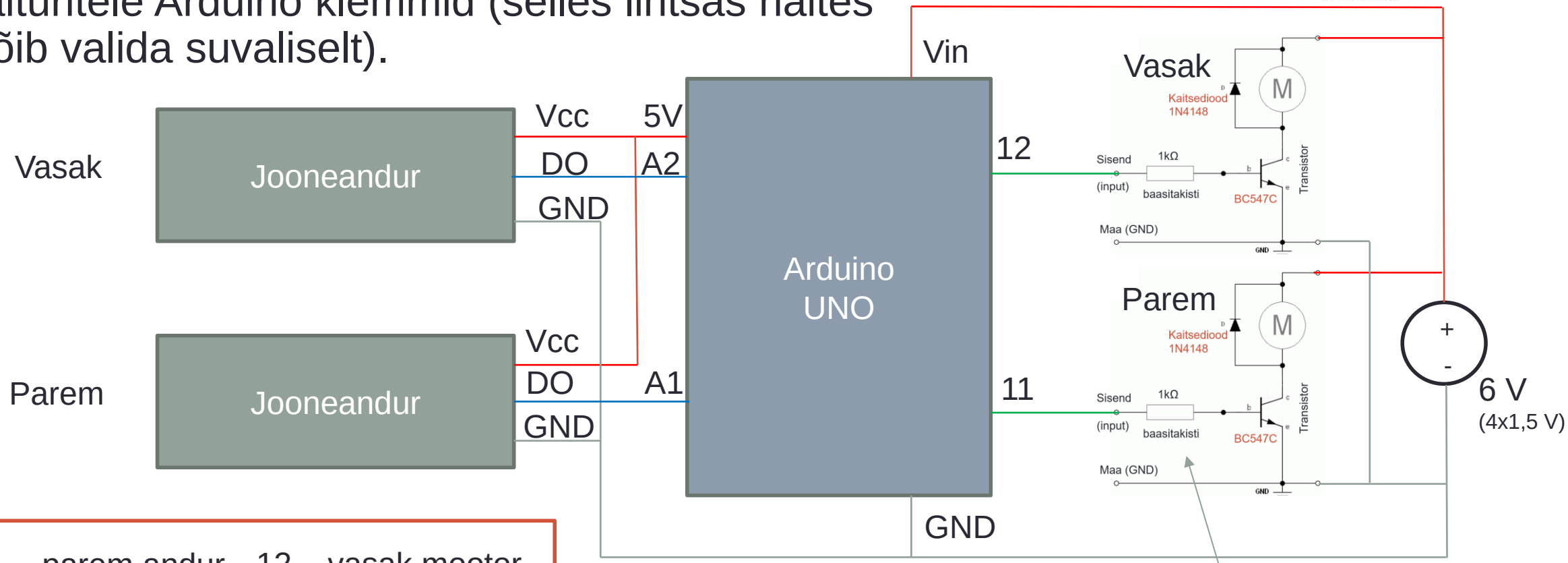
```
digitalWrite (11,HIGH);// annab kemmile 11 ligikaudu
// toitepinge
digitalWrite (10,LOW);// annab kemmile 10 ligikaudu
// 0 V pinge
```

Enne on vaja vastav kontrolleriklemm konfigureerida väljundiks (siin näidatud pole).
Vastasel juhul koodi täitmine ei anna ei veateadet ega loodetud tulemust !

Joonejärgija elektroonika

- Koostame elektriskeemi , valides anduritele, täituritele Arduino klemmid (selles lihtsas näites võib valida suvaliselt).



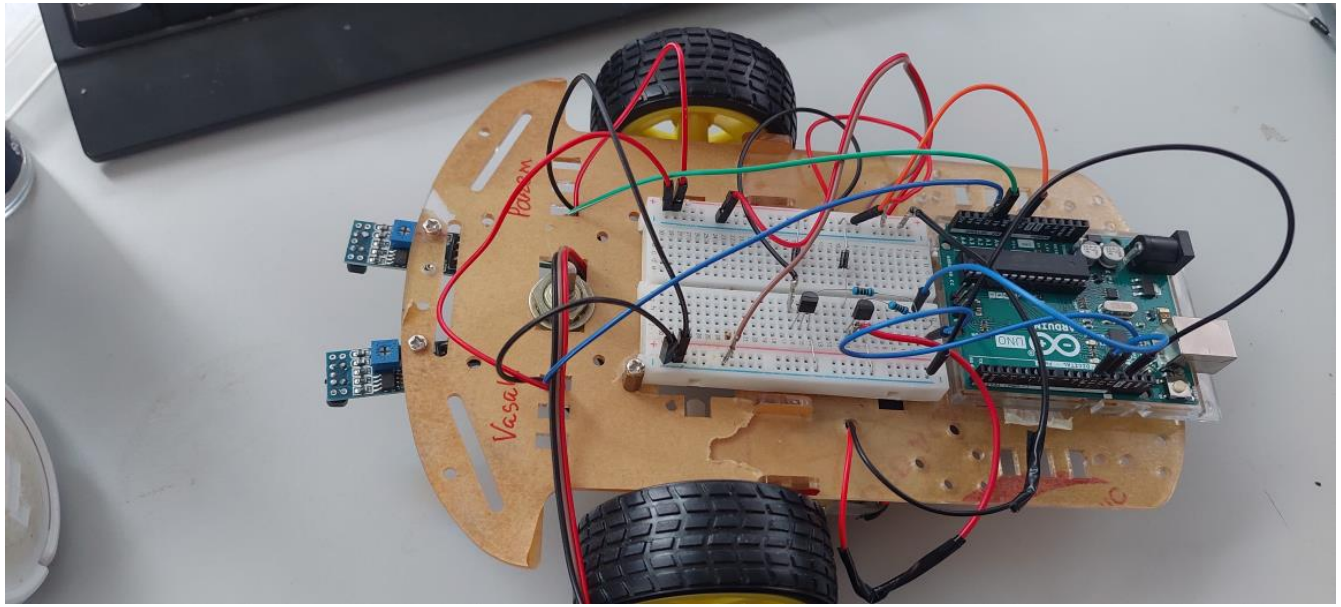


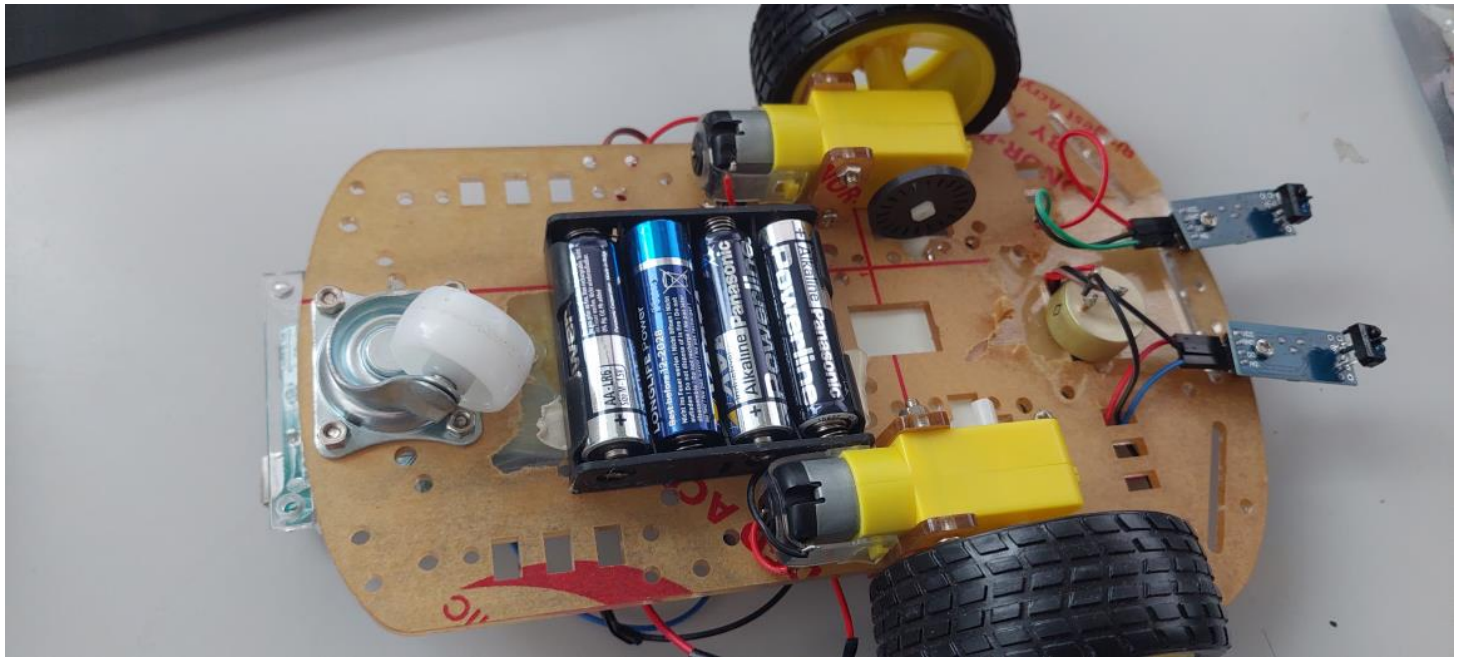
A1 – parem andur
A2 – vasak andur
12 – vasak mootor
11 – parem mootor

Võib kasutada ka Hiina mooduleid

NB ! Siis võivad olla klemmid ette määratud.

Koostame skeemi





Definitsioonid

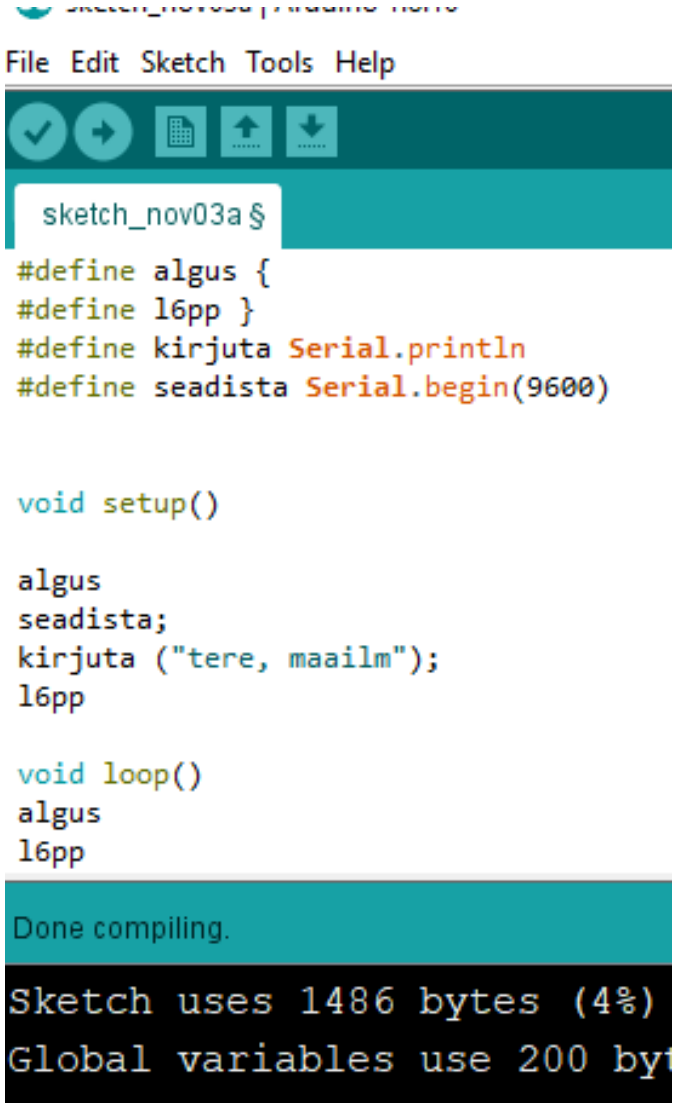
- Saab lihtsustada programmeerija tööd, saab teha koodi loetavamaks, muudetavamaks, erineva riistvara jaoks.

#define kartuleid 5 // igale poole kus on tekst “kartuleid” pannakse 5

#define porgandeid 12

#define doktorikraadiga_programmeerija Martinj

- byte tulemus=kartuleid+porgandeid;
- byte tulemus=5+12; //sama mis eelmine rida
- Mugav, kui on vaja kasutada programmi mitmes kohas , lähteandmeid muutes.
- Definitsioon ei ole muutuja !
- Ehk kartuleid=kartuleid+3 ;
- on vigane lause (omistamistehe !)



Saab ka sedasi – Üks võimalikke viise, kuidas koodist arusaamine võõrastel (ja ka endal) äärmiselt ebamugavaks teha.

Arduino “keele mugavused” on ka suures osas definitsioonid.

Defineerime Arduino keskkonnas sisendid, väljundid

- Riistvara definitsioonid on üliväga mõistlik teha (veidi pikema koodi puhul läheb kergesti slime ees kirjuks) , kuigi saab ka ilma. Anname klemmidele “nimed”
- Pannakse programmikoodi kõige algusesse
- Kui on vaja teha riistvaras muudatusi, saab muuta vaid definitsioone

```
#define Vasak_Andur A2
#define Parem_Andur A1
#define Vasak_Mootor 12
#define Parem_Mootor 11
#define valge LOW
#define must HIGH
```

```
void setup() {
//---teeme anduriklemmid sisendiks------
pinMode (Vasak_Andur, INPUT);
pinMode (Parem_Andur, INPUT);
//---teeme täituriklemmid väljundiks-----
pinMode (Vasak_Mootor, OUTPUT);
pinMode (Parem_Mootor, OUTPUT);
}
```

Koostame algoritmi jaoks sobivad funktsioonid

- Kasulik hoida põhialgoritmist eraldi – võimalik, et on vaja riistvara vahetada, saab teha ÜHES kohas vaid muudatuse.

```
//---ALGORITMIKÄSUD
void Liigu_OTSE ()
{
  digitalWrite (Parem_Mootor, HIGH);
  digitalWrite (Vasak_Mootor, HIGH);
  }
void Liigu_PAREMALE ()
{
  digitalWrite (Parem_Mootor, LOW);
  digitalWrite (Vasak_Mootor, HIGH);
  }
void Liigu_VASAKULE ()
{
  digitalWrite (Parem_Mootor, HIGH);
  digitalWrite (Vasak_Mootor, LOW);
  }
void Liigu_STOP ()
{
  digitalWrite (Parem_Mootor, LOW);
  digitalWrite (Vasak_Mootor, LOW);
  }
//-------------------------------
```

Keegi ei keela teha ka sedasi,aga kui on mingil põhjusel vaja muuta mootori klemmi, on muutused vaja teha igal pool !

```
void Liigu_VASAKULE ()
{
  digitalWrite (11, HIGH);
  digitalWrite (12, LOW);
  }
```

joonej_rgija2022 | Arduino 1.8.10

File Edit Sketch Tools Help

joonej_rgija2022 §

```
/* Sissejuhatus Mehhatroonikasse 2022
 *  Joonejärgija
 *  Martin Jaanus
 */
#define Vasak_Andur A2
#define Parem_Andur A1
#define Vasak_Mootor 12
#define Parem_Mootor 11
#define valge LOW
#define must HIGH
//---ALGORITMIKÄSUD
void Liigu_OTSE ()
{
  digitalWrite (Parem_Mootor, HIGH);
  digitalWrite (Vasak_Mootor, HIGH);
  }
void Liigu_PAREMALE ()
{
  digitalWrite (Parem_Mootor, LOW);
  digitalWrite (Vasak_Mootor, HIGH);
  }
void Liigu_VASAKULE ()
{
  digitalWrite (Parem_Mootor, HIGH);
  digitalWrite (Vasak_Mootor, LOW);
  }
void Liigu_STOP ()
{
  digitalWrite (Parem_Mootor, LOW);
  digitalWrite (Vasak_Mootor, LOW);
  }
//-------------------------------
```

Valmisprogramm

```
void setup() {
//---teeme anduriklemmid sisendiks------
pinMode (Vasak_Andur, INPUT);
pinMode (Parem_Andur, INPUT);
//---teeme täituriklemmid väljundiks-----
pinMode (Vasak_Mootor, OUTPUT);
pinMode (Parem_Mootor, OUTPUT);
}
```

```
void loop() {
  // loeme andurite signaalid
  bool vasak_varv=digitalRead(Vasak_Andur);
  bool parem_varv=digitalRead(Parem_Andur);
// algoritmi algus
 if (vasak_varv==valge)
 {
  if (parem_varv==valge)
  {
    Liigu_OTSE();
    }
    else
    {
      // parem värv on must
      Liigu_PAREMALE();
      }
  }
  else
  {
    // vasak värv on must
   if (parem_varv==valge)
  {
    Liigu_VASAKULE();
    }
    else
    {
      // parem värv on must, võimalik, et ristumiskoht
      Liigu_OTSE();
      }
    }
// algoritmi lõpp
}
```

Algorimi muutmine

- Kui on loodud riistvarahaldusfunktsioonid, on algoritmi muutmine imelihtne ! (kui on olemas toetav riistvara)

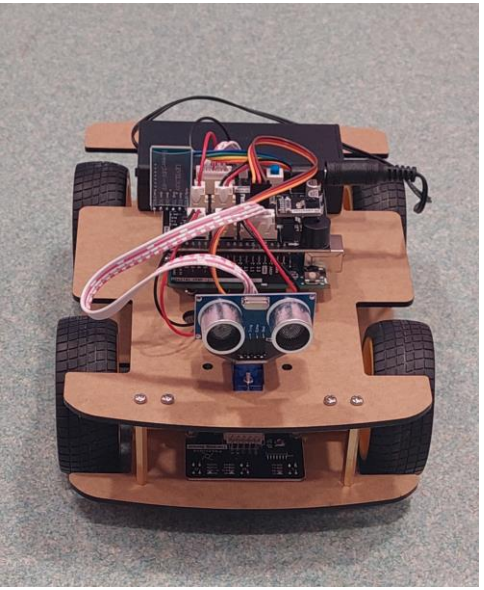
```
{
  Liigu_OTSE();
  delay(2000);//liigub 2 sekundit otse
  Liigu_PAREMALE();
  delay(3000);//liigub 3 sekundit paremale
  Liigu_STOP ();
  //jääme seisma

  }
```

```
{
  pinMode (11, OUTPUT);
  pinMode (12, OUTPUT);
  digitalWrite (11, HIGH);
  digitalWrite (12, HIGH);
  delay(2000);
  digitalWrite (11, LOW);
  digitalWrite (12, HIGH);
  delay(3000);
  digitalWrite (11, LOW);
  digitalWrite (12, LOW);


  }
```

Halb näide programmikoodist, kuigi toimiv – võõras sellest aru ei saa (NB! –meeskonnatöö), ega tõenäoliselt mõne aja möödudes ka autor. Sama asi, mis vasakul.

Liigutame Freenove autoplatvormi

- Erinevus eelmiste slaididega – mootorid saavad liikuda mõlemat pidi (kuidas seda tehakse , mõne loengu pärast)
- On olemas kummagi mootori jaoks kaks signaali – “LIIGU” ja “SUUND”
- Saab muuta ka kiirust



Autokiti kasutusjuhendist -->

Relationship between extension board and the control board is as below.

Pins of Arduino	Ports of extension	Pin multiplexing	Description
0	UART-Bluetooth		Bluetooth
1			
2	Servo		
3	Direction-Right		Direction of right motor
4	Direction-Left		Direction of left motor
5	Motor-Right		Speed of right motor -PWM
6	Motor-Left		Speed of right motor -PWM
7	Trig		Ultrasonic module
8	Echo		
9	SPI-NRF24L01	IR-remote	CE/IR-remote
10			CS
11	SPI-NRF24L01		MOSI
12			MISO
13			SCK
A0	Battery voltage	Buzzer	The ADC uses an internal 5V reference and the acquisition voltage is the battery voltage *1/4. Output to the Buzzer through the comparator. When higher than the 4.5V, output HIGH.
A1	Line-tracking sensor		Left
A2			Middle
A3			Right
A4	I2C		SDA
A5			SCL

Liigutame Freenove autoplatvormi

- Definitsioonid

```
/* Sissejuhatus Mehhatroonikasse 2023
 *  Algoritmidemo
 *  Martin Jaanus
 */

#define Vasak_Mootor_Liigu 6
#define Vasak_Mootor_Suund 4
#define Parem_Mootor_Suund 3
#define Parem_Mootor_Liigu 5
#define Piiksuja A0
```

- Otseliikumine

NB !

```
//---ALGORITMIKÄSUD
void Liigu_OTSE ()
{
  digitalWrite (Parem_Mootor_Suund, LOW); //parem mootor on tagurpidi !!!
  digitalWrite (Vasak_Mootor_Suund, HIGH);
  digitalWrite (Parem_Mootor_Liigu, HIGH);
  digitalWrite (Vasak_Mootor_Liigu, HIGH);
  }
```

Kasulik on läbi testida kõik elementaarliikumiskäsud.

Liigutame Freenove autoplatvormi

```
//---ALGORITMIKÄSUD
void Liigu_OTSE ()
{
  digitalWrite (Parem_Mootor_Suund, LOW); //parem mootor on tagurpidi !!!
  digitalWrite (Vasak_Mootor_Suund, HIGH);
  digitalWrite (Parem_Mootor_Liigu, HIGH);
  digitalWrite (Vasak_Mootor_Liigu, HIGH);
  }
void Liigu_PAREMALE ()
{
  digitalWrite (Parem_Mootor_Suund, LOW);
  digitalWrite (Vasak_Mootor_Suund, HIGH);

  digitalWrite (Parem_Mootor_Liigu, LOW);
  digitalWrite (Vasak_Mootor_Liigu, HIGH);
  }
void Liigu_VASAKULE ()
{
  digitalWrite (Parem_Mootor_Suund, LOW);
  digitalWrite (Vasak_Mootor_Suund, HIGH);

  digitalWrite (Parem_Mootor_Liigu, HIGH);
  digitalWrite (Vasak_Mootor_Liigu, LOW);
  }
```

```
void Liigu_STOP ()
{
  digitalWrite (Parem_Mootor_Liigu, LOW);
  digitalWrite (Vasak_Mootor_Liigu, LOW);
  }
void Liigu_TAGASI ()
{
   digitalWrite (Parem_Mootor_Suund, HIGH);
   digitalWrite (Vasak_Mootor_Suund, LOW);
  digitalWrite (Parem_Mootor_Liigu, HIGH);
  digitalWrite (Vasak_Mootor_Liigu, HIGH);
  }
void Piiks ()
{
digitalWrite (Piiksuja, HIGH);
delay(100);
digitalWrite (Piiksuja, LOW);
  }
```

Liigutame Freenove autoplatvormi

- Klemmide seadistus ja algoritm.

```
//--------------------------------
void setup() {

//---teeme täituriklemmid väljundiks-----
pinMode (Vasak_Mootor_Liigu, OUTPUT);
pinMode (Parem_Mootor_Liigu, OUTPUT);
pinMode (Vasak_Mootor_Suund, OUTPUT);
pinMode (Parem_Mootor_Suund, OUTPUT);
pinMode (Piiksuja, OUTPUT);
//--------------------------------
```

```
//algoritm ise

Liigu_OTSE ();
delay(1000);
Liigu_STOP ();
delay(3000);
Piiks ();
Liigu_TAGASI ();
delay(1000);
Liigu_PAREMALE ();
delay(1000);
Liigu_STOP();
Piiks ();
delay(2000);
Liigu_VASAKULE ();
delay(1000);
Liigu_OTSE ();
delay(1000);
Liigu_STOP();
}
void loop() {
//siia pole vaja,kuna ühekordne
}
```