

# Sisukord

Käsklused ja nende kasutamine	23
Ülevaade	23
BASIC	23
Laiendatud BASIC	23
PICAXE	23
Assembler	23
Matemaatilised operaatorid	23
Käsklused	24
Konstandid	24
Muutujad	25
Käsklused (A – W)	26
backward	27
branch	27
debug	28
end	28
for...next	29
forward	29
gosub	30
goto	30
halt	31
high	31
if...then	32
if...and...then	32
if...or...then	32
infrain	33
input	34
let	34
low	35
nap	35
output	35
pause	36
play	36
pulsin	37
pulsout	37
pwm	37
random	38
readadc	38
readadc10	38
return	39
reverse	39
sleep	40
sound	40
switch on	40
switch off	41
symbol	41
toggle	41
tune	42
wait	42

PICAXE toodete edasimüüja Eestis:

**Step Systems Eesti Oü**

Laki 14  
PK 16  
10621 Tallinn

tel. 651 7701

fax. 651 7702

[myk@stepsystems.ee](mailto:myk@stepsystems.ee),

<http://www.stepsystems.ee>

# Käsklused ja nende kasutamine

## ÜLEVAADE

Järgnevalt on välja toodud Programm Editori peamised BASIC – keele käskluste kirjeldused. Kuna Programm Editori programmi najal töötavad mitmesugused seadmed, siis käsitleme vaid neid käsklusi, mida olulisteks pidasime. Neli erinevat vormi on:

### BASIC

See on algupärane PBASIC™ formaat BASIC Stamp™ seadmetele, samuti Stamp Mikrokontrolleritele ja Stamp Kontrolleritele.

### Laiendatud BASIC

Laiendatud BASIC keel sarnaneb algupärasele, kuid sisaldab lisakäsklusi nagu näiteks ("if input0 is on then"...programm käivitatakse "if pin0 = 1 then"...käsklusi)

### PICAXE

Need on käsklused, mis toetavad PICAXE'i mikrokontrollerit.

### Assembler

Need käsklused teisendavad käsklused automaatselt assembleri keelde ning BASIC keelde BASIC > Assembleri omaduste kaudu. Assembleri kohta võid leida lisainformatsiooni assembleri kodulehelt.

## 1. Osa

See peatükk on jaotatud

- Käskluste selgitused ja näited
- Konstandid
- Sümbolid
- Muutujad
- Kommentaarid

### Matemaatilised Operaatorid

Matemaatiliste operaatorite üksikasjad on lahti seletatud "let" käskluse juures.

## Käskluste selgitused ja näited

Selgitused on eraldatud käsklustest ülakomaga (^) või semikooloniga (;). Avasõna REM- võib kasutada ka kommentaaride kirjutamisel.

Näiteks:

```
high 0          ' Muudab pin0-i esimeseks
high 0          ; Muudab pin0-i esimeseks
REM muudab pin0-i esimeseks
```

## KONSTANDID

Konstandid võivad esineda neljas erinevas vormis, detsimaal, heksta, binaar ja ASCII vormis

Detsimaali konstante ei ole vaja eraldada erisümbolitega. Heksta numbrite ette lisatakse dollari märk (\$). Binaar numbritele lisatakse ette protsendi sümbol (%) ja ASCII konstandid on jutumärkide vahel.

Näiteks:

```
100             ' 100 detsimaalina
$64             ' 64 heksta
%01100100      ' 01100100 binaar
"A"            ' "A" ASCII-na (65)
"Hello"        ' "Hello" – sama kui "H","e","l","l","o"
B1 = B0 ^ $AA  ' XOR muutuja B0 heksta AA-ga
```

## SÜMBOLID

Sümbolit võib nimetada erinevateks konstantideks, muutujateks või programmiaadressideks. Konstantide, muutujate ja sümbolite väärtused eraldatakse võrdusmärgiga (=). Võrdusmärgile järgneb konstandi arv, vms, millisenä soovitakse antud sümbolit defineerida.

Sümboliks võib määrata ükskõik millise sõna, kuid mis ei sarnane programmi enda käsklustega. Sümbolid võivad ka sisaldada numbreid (flash1, flash2, jne), kuid sümboli esimene tähe märk ei saa olla number (näiteks 1flash).

Programmi päise lisamisel kasutatakse päise eraldamist programmist eristusmärgina koolonit (:)

Näiteks:

```
symbol PUN_LED = 7          ' konstant 7 sümboliks määratakse PUN_LED
symbol LOENDUR = B0        ' B0 sümboliks määratakse LOENDUR
let LLOENDUR = 200        ' arvu 200 sümboliks määratakse let LOENDUR
KUUSE TULED:              ' programmi päis (võib olla ka programmi nimetus)
high PUN_LED              ' 7. harukahvel lülitatakse mikrokontrolleris sisse
pause LOENDUR             ' viivitus 0.2 sekundit
low PUN_LED               ' 7. harukahvel lülitatakse mikrokontrolleris välja
pause LOENDUR             ' viivitus 0.2 sekundit
goto KUUSETULED          ' suunatakse tagasi programmi päisesse
```

# MUUTUJAD

## Kasutamine:

Järgnevalt on toodud näited muutujate kasutamisest erinevates olukordades:

- PICAXE´i kasutatakse PICAXE´ mikrokontrolleri programmeerimisel
- BASIC-ut ja EXTENDED-it kasutatakse Stamp programmeerimisel
- Assemblerit kasutatakse kui Assembler keele kodeerijat.

Kestvusmuutujad (0-255) on nimetatud b0 - b13

Sõnamuutujad (0-65535) on nimetatud w0 - w6

Iga Sõnamuutuja koosneb kahest tavamuutujast:

w0	w1	w2	w3	w4	w5	w6
b0-b1	b2-b3	B4-b5	b6-b7	B8-b9	b10-b11	b12-13

## PICAXE

PICAXE – mikrokontrolleerit programmeerimist toetavad järgnevad muutujad:

Sõnamuutujad: W0, W1, W2, W3, W4, W5, W6

Silbid: PINS, DIRS (vain PICAXE-08), INFRA, KEYVALUE,  
B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13

Bitid: PIN0, PIN1, PIN2, PIN3, PIN4, PIN5, PIN6, PIN7 (kõik koos = PINS)  
BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7 (kõik koos = tavu B0)  
BIT8, BIT9, BIT10, BIT11, BIT12, BIT13, BIT14, BIT15 (kõik koos = tavu B1)

Lisaks valenimed

INPUT0, INPUT1 jne. Võib kasutada PIN0, PIN1 jne. kasutamisel  
OUTPUT0, OUTPUT1 jne. Võib kasutada konstantide 0, 1, 2 jne. kasutamisel

## BASIC Stamp

BASIC –u kasutamine toetab järgmisi muutujaid:

Sõnad: PORT

W0, W1, W2, W3, W4, W5, W6

Silbid: DIRS, PINS

B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13

Bitid: DIR0, DIR1, DIR2, DIR3, DIR4, DIR5, DIR6, DIR7 (kõik koos = DIRS)  
BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7 (kõik koos = B0)  
BIT8, BIT9, BIT10, BIT11, BIT12, BIT13, BIT14, BIT15 (kõik koos = B1)

B12:a ja B13:a (W6) kasutatakse GOSUB – käskluste puhul pini mäluna (vaid Stamp 1).

## Extended ja Assembler

Neid kasutatakse BASIC muuttujate toetamisel.

Lisaks:

INPUT0, INPUT1 jne. võib kasutada pin0, pin1 jne. puhul  
OUTPUT0, OUTPUT1 jne. Võib kasutada konstant numbrite 0, 1, 2 jne. puhul

B12:a ja B13:a (W6) kasutatakse GOSUB – käskluste puhul pini mäluna (vaid Stamp 1).

## KÄSKLUSED (A-W)

Allolevas tabelis on välja toodud erinevad käsklused ning samuti märgitud ära milliste mikrokontrollerite puhul on neid võimalik kasutada.:

- BASIC-ut ja EXTENDED-it kasutatakse Stamp programmeerimisel
- PIC16F84A-d kasutatakse kui programmi laetakse otse PIC16F84A/PIC16F627 mikrokontrolleritele.
- PICAXE-t kasutatakse PICAXE´ i mikrokontrollerite programmeerimisel.

Mitmed käsklused on kasutusel vaid PICAXE tüüpi mikrokontrollerite A, M või X mikroskeemi paigutuse puhul. Tabelisse on need märgitud sulgudega.

Käsklus	BASIC	Extended	PIC16F84A	PICAXE08	PICAXE18	PICAXE28/40
Backward		X	X		X	X
Branch	X	X	X	X	X	X
Debug	X			X	X	X
End	X	X	X	X	X	X
For... Next	X	X	X	X	X	X
Forward		X	X		X	X
Gosub	X	X	X	X	X	X
Goto	X	X	X	X	X	X
Halt		X	X		X	X
High	X	X	X	X	X	X
If... then	X	X	X	X	X	X
Infrain				X (M)	X (A)	X
Input	X	X	X	X		
Let	X	X	X	X	X	X
Low	X	X	X	X	X	X
Nap	X	X	X	X	X	X
Output	X	X	X	X		
Pause	X	X	X	X	X	X
Play				X (M)		
Pulsin	X	X	X	X	X	X
Pulsout	X	X	X	X	X	X
Pwm	X	X	X	X		
Random	X	X	X	X	X	X
Readadc				X	X	X
Readadc10				X (M)	X (X)	X (X)
Return	X	X	X	X	X	X
Reverse	X	X	X	X		
Sleep	X	X	X	X	X	X
Sound	X	X	X	X	X	X
Switch on		X	X	X	X	X
Switch off		X	X	X	X	X
Symbol	X	X	X	X	X	X
Toggle	X	X	X	X	X	X
Tune				X (M)		
Wait		X	X	X	X	X

## backward

### BACKWARD mootor

Saab panna mootori tööle tagurpidi suunas.

- Mootor = mootori nimi, A või B.

**NB:** Sellise nimetusega käsklus kirjutatakse ka high/low – käsklusena 4,5 (A) ja 6,7 (B).

Näiteks:

<b>algus:</b>	<b>forward A</b>	‘ mootor A liigub edasi
	<b>wait 5</b>	‘ viivitus 5 sekundit
	<b>backward A</b>	‘ mootor A hakkab tööle tagurpidi
	<b>wait 5</b>	‘ viivitus 5 sekundit
	<b>halt A</b>	‘ mootor A seiskub
	<b>wait 5</b>	‘ viivitus 5 sekundit
	<b>goto algus</b>	‘ tagasi programmi päisesse

## branch

### BRANCH offset,(address0,address1...addressN)

Käsklus *offset* suunatakse eelnevalt defineeritud aadressile (päisele).

- Offset on muutuja/konstant, mis määratakse millist aadress# - numbrit (0-N) kasutatakse.

- Aadressid on tunnused, kuhu programm suunatakse.

Näiteks:

<b>reset:</b>	<b>let infra = 0</b>	‘ nimetatakse ümber muutuja infra
<b>silmukka:</b>	<b>if infra &gt; 0 then testi</b>	
		‘ suunatakse päisesse testi, kui võetakse vastu uus signaal
	<b>goto silmukka</b>	‘ suunatakse päisesse silmukka
<b>testi:</b>		
	<b>branch infra,(btn0,btn1, btn2, btn3)</b>	
		‘ suunatakse sinna, kus defineeritakse btn
	<b>goto reset</b>	‘ suunatakse tagasi reset päisesse

**btn0:** jne.

**btn1:** jne.

**btn2:** jne.

**btn3:** jne.

## debug

### DEBUG muutuja

Näitab programmi koostamisel muutujate arvvaartusi, avatud aknas uuesti.

- Muutuja hetkelist vaartust saab vaadata lihtsalt, kui kirjutame muutuja nime.

Näiteks:

#### algus:

<b>debug b1</b>	' näitab b1- arvvaartust
<b>let b1 = b1 + 1</b>	' suurendatakse b1 ühe ühiku võrra
<b>pause 500</b>	' viivitus 0.5 sekundit
<b>goto algus</b>	' tagasi päisesse algus

## end

### END

Lülitab välja mikrokontrolleri, kuni järgmise sisselülitumiseni või kui arvutist laetakse uus programm.

NB: Käsklus END kirjutatakse programmi viimaseks käskluseks.

Näiteksi:

#### Algus:

<b>let b2 = 15</b>	' b2 defineeritakse arvvaartuseks 15
<b>pause 2000</b>	' viivitus 2 sekundit
<b>gosub flsh</b>	' käivitatakse alamprogramm
<b>let b2 = 5</b>	' b2 defineeritakse arvvaartuseks 5
<b>pause 2000</b>	' viivitus 2 sekundit
<b>gosub flsh</b>	' käivitatakse alamprogramm
<b>end</b>	' lõpetus, kui alamprogrammiks nõutav tingimus ei ole täidetud

#### flsh:

<b>for b0 = 1 to b2</b>	' muudetakse arvvaartust b2 võrra
<b>high 1</b>	' lülitatakse sisse harukahvel 1
<b>pause 500</b>	' viivitus 0,5 sekundit
<b>low 1</b>	' Lülitatakse harukahvel 1 välja
<b>pause 500</b>	' viivitus 0,5 sekundit
<b>next b0</b>	' lõpetatakse programm
<b>return</b>	' tagasi alamprogrammist põhiprogrammi

## for...next

**FOR** muutuja = algus TO lõpp

**NEXT** {muutuja}

Ühendab käsklused *FOR-NEXT*.

- Muutujat kasutatakse loendurina.

- Alguses on muutuja arvväärus, mida võetakse loendamise alnumbriks programmi kestel.

- Lõpus on kirjeldatud number, millest alates loendamine lõpeb.

Näiteks:

**algus:**

<b>for b0 = 1 to 20</b>	‘ määratakse loendus 20 korrale
<b>high 1</b>	‘ lülitatakse sisse harukahvel 1
<b>pause 500</b>	‘ viivitus 0,5 sekundit
<b>low 1</b>	‘ harukahvel 1 lülitatakse välja
<b>pause 500</b>	‘ viivitus 0,5 sekundit
<b>next b0 peale</b>	‘ Lisatakse b0 uus väärtus ja loendus algab algusest
<b>pause 2000</b>	‘ kui on kahekümnes kord, siis ‘ viivitus 2 sekundit
<b>goto loop</b>	‘ tagasi algusesse

## forward

**FORWARD** mootor

Selle käsklusega saab mootorit käivitada õiges suunas (edasi).

- Mootor = mootori nimi, A tai B.

**NB:** Sellise nimetusega käsklus kirjutatakse ka high/low – käsklusena 4,5 (A) ja 6,7 (B). Tähega defineeritakse ümber mootori nimetus porogrammijadas.

Näiteks:

**algus:**

<b>forward A</b>	‘ mootor A töötab päripäeva
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>backward A</b>	‘ mootor A töötab vastupäeva
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>halt A</b>	‘ seiska mootor A
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>goto alku</b>	‘ tagasi päisesse algus



## gosub

### GOSUB address (päis)

Programm suunatakse kasutaja poolt etteantud aadressile, ehk programmi alamprogrammi. Ühes programmis võib kasutada korraga 16 erinevat alamprogrammi neljal korral. Kui PICAXE mikroskeemidel toimub teatud kohas katkestus, on üks gosub katkestatud, kuid ülejäänud 15 on töövalmis. PICAXE 18X (firmware >=8.2) või 28X (firmware >=7.4) – versioonides on võimalik kasutada 256 alamprogrammi.

- Aadress on tunnus, kuhu programm suunatakse.

Näiteks:

#### algus:

<b>let b2 = 15</b>	' b2 määratakse arvvärtuseks 15
<b>pause 2000</b>	' viivitus 2 sekundit
<b>gosub flsh</b>	' kutsutakse esile alamprogramm
<b>let b2 = 5</b>	' b2 määratakse arvuks 5
<b>pause 2000</b>	' viivitus 2 sekundit
<b>gosub flsh</b>	' alustab alamprogramm
<b>end</b>	' lõpetus

flsh:

<b>for b0 = 1 to b2</b>	' suurendatakse b2 ühe arvvärtuse võrra
<b>high 1</b>	' lülitatakse sisse harukahvel 1
<b>pause 500</b>	' viivitus 0,5 sekundit
<b>low 1</b>	' lülitatakse harukahvel 1 välja
<b>pause 500</b>	' viivitus 0,5 sekundit
<b>next b0</b>	' lõpetatakse loendus
<b>return</b>	' tagasi põhiprogrammi

## goto

### GOTO aadress (päisele)

Suunatakse etteantud päisele.

- Aadress on tunnus, kuhu programm suunatakse.

Näiteks:

#### algus:

<b>high 1</b>	' lülitatakse sisse harukahvel 1
<b>pause 5000</b>	' viivitus 5 sekundit
<b>low 1</b>	' lülitatakse välja harukahvel 1
<b>pause 5000</b>	' viivitus 5 sekundit
<b>goto algus</b>	' tagasi algusesse

## halt

### HALT mootor

*Selle käskluseag peatatakse mootor.*

- Mootor = mootori nimi, A tai B.

**NB:** See pole sobiv variant. Selle käskluse asemel kasutame käsklust high/low – käsklusena 4,5 (A) ja 6,7 (B). Tähega defineeritakse ümber mootori nimetus porogrammijadas.

Näiteks:

#### algus:

<b>forward A</b>	‘ mootor A pannakse tööle päripäeva
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>backward A</b>	‘ mootor A pannake tööle vastupäeva
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>halt A</b>	‘ mootor A seiskumine
<b>wait 5</b>	‘ viivitus 5 sekundit
<b>goto algus</b>	‘ tagasi alguse päisesse

## high

### HIGH pin-i number

*Pin-i number, ehk harukahvli väljundi number muutub 1-ks.*

- Pin-i number on muutuja/konstant (0-7), mis määrab millist harukahvli sisendit/väljundit kasutatakse.

Näiteks:

<b>algus:</b>	<b>high 1</b>	‘ lülitatakse sisse harukahvel 1
	<b>pause 5000</b>	‘ viivitus 5 sekundit
	<b>low 1</b>	‘ lülitatakse välja harukahvel 1
	<b>pause 5000</b>	‘ viivitus 5 sekundit
	<b>goto algus</b>	‘ tagasi alguse päisesse

## if...then, if...and...then, if...or...then

**IF** muutuja ?? arv {AND/OR muutuja ?? arv ...} **THEN** aadress

?? võib olla =, <>, >=, <=, >, <

*Annab võimaluse võrrelda ja suunata programmi erinevaid operatsioone.*

- muutuja/muutujad võrreldakse arvuga/arvudega.

- Arv on muutuja/konstant.

- Aadress on tunnus (päise nimi), mis määrab, kuhu suunatakse programmi protsessid, kui tingimus on täidetud.

Näiteks:

**algus:**

**if pin0 = 1 then vahele**

‘ kui pin0 = 1, siis mine päisesse vahele

**goto alku**

‘ kui tingimus ei ole täidetud, mine algusesse tagasi

**vahele:**

**high 1**

‘ lülita sisse harukahvel 1

**pause 5000**

‘ viivitus 1 sekund

**low 1**

‘ lülita harukahvel 1 välja

**goto algus**

‘ tagasi algusesse

Sõnu is (=), on (1) ja off (0) võib samuti kasutada (nt. if pin0 is on then vahele).

## infrain

### INFRAIN

*Oota kuni uus infrapuna signaal vastu võetakse.*

Seda käsklust kasutatakse infrapuna signaalide vastuvõtmisel (sarnane telekapuldi toimimisele). Kõik protsessid peatatakse, kuni vastuvõtja ei saa uut signaali. Vastuvõetud käskluse arv asendatakse ´infra´ muutujaga. Infrapuna tootekoodid on (AXE040), (LED020) vastuvõtuks kasuta vaid LEDO20.

Infrapuna tuli on sisend 0, väljaarvatud PICAXE-08M mikrokontrolleri puhul kus on see harukahvli sisend 3. Muutuja ´infra´ on muutuja kõigi mikrokontrolleerite puhul, välja arvatud PICAXE-08M, see on (B13).

Näiteks:

#### algus:

```
infrain                                'oota uut signaali
    if infra = 1 then swon1             'lülita harukahvel 1 sisse
    if infra = 2 then swon2             'lülita harukahvel 2 sisse
    if infra = 3 then swon3             'lülita harukahvel 3 sisse
    if infra = 4 then swoff1            'lülita harukahvel 1 välja
    if infra = 5 then swoff2            'lülita harukahvel 2 välja
    if infra = 6 then swoff3            'lülita harukahvel 3 välja
    goto algus
swon1:  high 1
        goto algus
swon2:  high 2
        goto algus
swon3:  high 3
        goto algus
swoff1: low 1
        goto algus
swoff2: low 2
        goto algus
swoff3: low 3
        goto algus
```

## input

### INPUT pin-i number (harukahvli number)

Teeb pin-ist (harukahvlist) sisendi

- Pin-i numbri muutuja/konstant (0-7), mis määrab millist sisendi/väljundi harukahvli kasutatakse.

Näiteks:

<b>algus:</b>	<b>input 1</b>	<b>' teeb pin 1-st sisendi</b>
	<b>reverse 1</b>	<b>' teeb pin 1-st väljundi</b>
	<b>reverse 1</b>	<b>' teeb pin-ist sisendi</b>
	<b>output 1</b>	<b>' teeb pin-ist väljundi</b>

## let

### {LET} muutuja = {-}arv ?? arv ...

+	;kasvav loendur
-	;kahanev loendur
*	;kordade loendur (taastab sõna alumise osa tulemuse)
**	;kordade loendur (taastab sõna ülemise osa tulemuse)
/	;jagajaloendur (taastab osaliselt)
// (tai %)	;jagajaloendur (taastab jääkväärtuse)
MAX	;make less than or equal to the maximum (vähem või võrdne maksimumiga)
MIN	;make greater than or equal to the minimum (suurem või võrdne miinimumiga)
&	;JA
	;VÕI (klaviatuuril Alt Gr + < vajutada üheaegselt klaviatuurinuppu)
^	;väljalülitav VÕI (klaviatuuril SHIFT + ^ vajutada üheaegselt klaviatuurinuppu)
&/	;EI-JA (nand)
/	;EI-VÕI (nor)
^/	;väljalülitav EI-VÕI (xnor)

Muudab muutujate kasutust (sõnast sõnani). Loenduri puhul vasakult paremale.

- Muutujate kasutamist vaatame hiljem.

- Arv(ud) on muutujad/konstandid, mis vahetavad muutujaid.

Sõnu AND ja OR kasutatakse ainult if...then –lausete keskel, kuid nad ei võrdu matemaatiliste sümbolitega & (ja) ja | (või), mida kasutatakse let –käskude puhul.

Näiteks:

<b>algus:</b>	<b>let b0 = b0 + 1</b>	<b>' lisame b0-le arvu 1</b>
	<b>sound 7,(b0,50)</b>	<b>' mängi heli</b>
	<b>if b0 &gt; 50 then reset</b>	<b>' kui heli on jõudnud 50-ni mine päisesse reset</b>
	<b>goto algus</b>	<b>' tagasi algusesse</b>
<b>reset:</b>	<b>let b0 = b0 max 10</b>	<b>' nulli b0 arvuni 10,</b>
	<b>goto algus</b>	<b>' millest tuleb b0-i maksimum väärtus</b>
		<b>' tagasi algusesse</b>

## low

### LOW pin-i number

*Pin (harukahvel) lülitatakse välja.*

- Pin-i numbri muutujad/konstandid (0-7), mis määratakse millist sisend/väljund harukahvlit kasutatakse.

Näiteks:

<b>algus:</b>	<b>high 1</b>	‘ lülita harukahvel 1 sisse
	<b>pause 5000</b>	‘ viivitus 5 sekundit
	<b>low 1</b>	‘ lülita harukahvel 1 välja
	<b>pause 5000</b>	‘ viivitus 5 sekundit
	<b>goto algus</b>	‘ tagasi algusesse

## nap

### NAP aeg

*Katkestab hetkeks programmi. Selle käskluse puhul kasutatakse väga väikesi väärtusi.*

- Aeg on muutuja/konstant, mis määrab katkestusperioodi kestuse. Katkestuse kestvus kirjutatakse  $2^{\text{aeg}} * \sim 18\text{ms}$ . Aja väärtuseks võin olla vahemikus 0 - 7.

Näiteks:

<b>algus:</b>	<b>high 1</b>	‘ lülita harukahvel 1 sisse
	<b>nap 4</b>	‘ peata programm $2^4$ x 18ms pikkuseks ajaks
	<b>low 1</b>	‘ lülita harukahvel 1 välja
	<b>nap 7</b>	‘ peata programm $2^7$ x 18ms pikkuseks ajaks
	<b>goto algus</b>	‘ tagasi algusesse

## output

### OUTPUT pin-i number

*Teeb pin-ist väljundi*

- Pin-i numbri muutujad/konstandid (0-7), mis määratakse millist sisend/väljund harukahvlit kasutatakse.

Näiteks:

<b>algus:</b>	<b>input 1</b>	‘ muudab pin-i sisendiks
	<b>reverse 1</b>	‘ muudab pin-i väljundiks
	<b>reverse 1</b>	‘ muudab pin-ist sisendi
	<b>output 1</b>	‘ muudab pin-i väljundiks

## pause

### PAUSE millisekundit

Viivitus. Säilitab käskluse kestvusaega. Viivituse aeg sõltub resonaatorist.

- Millisekundid on muutujad/konstandid (0-65535), mis määravad viivituse ajalise pikkuse.

Näiteks:

<b>algus:</b>	<b>high 1</b>	' lülitab harukahvli 1 sisse
	<b>pause 5000</b>	' viivitus 5 sekundit
	<b>low 1</b>	' lülitab harukahvli 1 välja
	<b>pause 5000</b>	' viivitus 5 sekundit
	<b>goto algus</b>	' tagasi algusesse

## play

### PLAY meloodia,LED

Mängib PICAXE-08M mikroskeemidel (pin 2).

Enda loodud meloodiate puhul kasutatakse käsklust 'tune'.

Vaata, kuidas ühendada PICAXE-08M mikroskeemile Piezo summerit.

- Meloodia on muutuja/konstant (0 - 3), mis määrab millist meloodiat mängida.

0 – Palju õnne sünnipäevaks

1 – Tiliseb, tiliseb

2 – Püha öö

3 – Peeter Punanina

- LED on muutuja/konstant (0 -3), mis määrab, kas meloodiaga samaaegselt ka tuled vilguvad.

0 – Ei põle

1 – Väljund 0 vilgub

2 – Väljund 4 vilgub

3 – Väljundid 0 ja 4 vilguvad kordamööda

Näiteks:

<b>play 3,1</b>	' Peeter Punanina meloodia mängib ja väljund 0 vilgub
-----------------	---

## pulsin

### PULSIN pin-i number, kestvus, muutuja

*Mõõda sisendpulssi.*

- Pin-i number on muutuja/konstant (0-7), mis määrab millist sisendit/väljundit kasutatakse.
- Kestvus on muutuja/konstant (0 or 1), mis määrab kas väärtust pulseeritakse harukahvlil 1-1 või 0-1.
- Muutuja vastab väärtustele (1-65536). Kui mõõtmine jõuab lõppu (65536s), on tulemus 0.

Näiteks:

**pulsin 3,1,b1**      ' **salvesta pulsi pikkus pin 3-st muutujaks b1**

## pulsout

### PULSOUT pin-i number, aeg

*Lülita pin hetkeks välja.*

- Pin-i number on muutuja/konstant (0-7), mis määrab sisendi/väljund pin-i, mida kasutatakse.
- Aeg on muutuja/konstant, mis määrab ajaperioodi (0-65535).

Näiteks:

#### algus:

**pulsout 4,150**      ' **lülita 1.50ms pulss harukahvlist 4**  
**pause 20**      ' **viivitus 20 ms**  
**goto algus**      ' **tagasi algusesse**

## pwm

### PWM pin-i number, pinge , periood

*Käsklust pwm kasutatakse ainult PICAXE-08 ja Stamp mikrokontrollerite puhul. Seda käsklust kasutatakse, kui harukahvlist vähendatakse pinget järk-järgult. Seda saab määrata numbriliselt (0-5V),olenevalt missugust väljund harukahvlist kasutatakse.*

- Pin-i number on muutuja/konstant (0-7), mis määrab missugust sisendit/väljundit kasutatakse.
- pinge on muutuja/konstant (0-255), mis määrab analoogpinge taseme.
- periood on muutuja/konstant (0-255), mis määrab perioodi kestvuse. Iga periood kestab 5ms.

Näiteks:

#### algus:

**pwm 4,150,20**      ' **lülitab sisse pwm käskluse harukahvlist 4**  
**pause 20**      ' **viivitus 20 ms**  
**goto algus**      ' **tagasi algusesse**



## random

### RANDOM muutuja

*Loob muutujale juhuslikkuse väärtuse.*

- Muutuja on nii algus kui ka lõppväärtus. Kui luua juhuslikkus, kutsutakse esile muutuja iseeneselikku muutumist teatud väärtuste hulgas.

Näiteks:

```
algus:      random b1          ' salvesta b1 juhuslikuks muutujaks
           let pins = b1      ' salvesta juhuslikkuse esinemine väljund
             harukahvlile
           pause 100          ' viivitus 0,1 sekundit
           goto algus         ' tagasi algusesse
```

## readadc, readadc10

### READADC kanal, muutuja

### READADC10 kanal, sõnamuutuja

*Readadc - loetakse ADC kanaliks (8 bitine eraldusvõime) sisaldab muutujaid.*

*Readadc10 - loetakse ADC kanaliks (10 bitine eraldusvõime) sisaldab muutujaid.*

- Kanal on muutuja/konstant, mida määratakse silpidest koosnevate päistega (0-3).  
- Muutuja loetakse andmesilpidega.

Näiteks:

```
algus:      readadc 1,b1       ' arv loetakse b1-ks
           if b1 > 50 then ohi ' mine päisesse ohi, kui b1 > 50
           goto algus         ' vastasel korral mine algusesse tagasi
ohi:        high 1             ' lülita harukahvel 1 sisse
           pause 5000          ' viivitus 5 sekundit
           low 1               ' lülita harukahvel 1 välja
           goto algus         ' mine tagasi algusesse
```

## return

### RETURN

*Suunab alamprogrammist tagasi põhiprogrammi.*

Näiteks:

```
algus:      let b2 = 15          ' b2 asendatakse arvuga
            pause 2000       ' viivitus 2 sekundit
            gosub ohi        ' kutsutakse esile alamprogramm
            let b2 = 5       ' b2 asendatakse arvuga
            pause 2000       ' viivitus 2 sekundit
            gosub ohi        ' kutsutakse esile alamprogramm
            end              ' lõpeta, et valikutes minna alamprogrammi

edasi:      for b0 = 1 to b2   ' korda käsklust b2 korda
            high 1           ' lülita harukahvel 1 sisse
            pause 500        ' viivitus 0,5 sekundit
            low 1            ' lülita harukahvel 1 välja
            pause 500        ' viivitus 0,5 sekundit
            next b0         ' lisa b0-le üks ja alusta programmi uuesti
            return          ' välju alamprogrammist
```

## reverse

### REVERSE pin-i number

*Muuda pin-i tähendust (sisend väljundiks ja vastupidi)*

- Pin-i number on muutuja/konstant (0-7), mis määrab, missugust sisendit/väljundit kasutatakse.

Näiteks:

```
algus:      input 1          ' muudetakse pin-st sisend
            reverse 1       ' muudetakse pin-ist väljund
            reverse 1       ' muudetakse pin-st sisend
            output 1        ' muudetakse pin-ist väljund
```

## sleep

### SLEEP sekundit

Maga teatud ajahulk sekundeid (eristamisvõime on ~2.3s, täpsus on ~99.9%).

- Sekundid on muutujad/konstandid, mis määrab magamise aja perioodiks 2.3 sekundit ning režiimi kordmise ajaks (0-65535).

Näiteks:

<b>algus:</b>	<b>high 1</b>	' lülita sisse harukahvel 1
	<b>sleep 10</b>	' maga 23 sekundit
	<b>low 1</b>	' lülita harukahvel 1 välja
	<b>sleep 100</b>	' maga 230 sekundit
	<b>goto algus</b>	' mine tagasi algusesse

## sound

### SOUND pin-i number,(meloodia,kestvus,meloodia,kestvus...)

Mängi meloodiat soovitud ajahulk. Pin peab olema ühendatud Piezo heliväljundiga (summeriga) või valjuhääldiga ja 10 uF elektrolüütkondensaatoriga.

- Pin-i number on muutuja/konstant (0-7), mis määrab, missugust sisendit/väljundit kasutatakse.

- Meloodiad on muutujad/konstandid (0-255), mis määravad heli kõrgusi ja tugevusi. Meloodia 0 on hääletu, meloodiad 1-127 on helitugevused ja meloodiad 128-255 kõrgsagedushelid.

- Kestvuse numbrid on muutujad/konstandid (0-255), mis määravad meloodia pikkuse (12ms mitmekordselt).

Näiteks:

<b>algus:</b>	<b>let b0 = b0 + 1</b>	' lisatakse b0-le üks
	<b>sound 7,(b0,50)</b>	' mängi heli
	<b>goto algus</b>	' tagasi algusesse

## switch on

### SWITCH ON pin-i number / SWITCHON pin-i number

Lülitatakse sisse Pin harukahvel.

- Pin-i number on muutuja/konstant (0-7), mis määrab, missugust sisend/väljundpin-i kasutatakse.

**NB:** See on varjatud käsklus, mille tegelik nimetus on käsklus ' high ' .

Näiteks:

<b>algus:</b>	<b>switch on 7</b>	' lülita haukahvel 7 sisse
	<b>wait 5</b>	' viivitus 5 sekundit
	<b>switch off 7</b>	' lülita harukahvel 7 välja
	<b>wait 5</b>	' viivitus 5 sekundit
	<b>goto algus</b>	' tagasi algusesse

## switch off

### SWITCH OFF pin-i number / SWITCHOFF pin-i number

Lülitatakse välja Pin harukahvel.

- Pin-i number on muutuja/konstant (0-7), mis määrab, missugust sisend/väljundpin-i kasutatakse.

**NB:** See on varjatud käsklus, mille tegelik nimetus on käsklus ´ low ´ .

Näiteks:

<b>algus:</b>	<b>switch on 7</b>	´ lülita harukahvel 7 sisse
	<b>wait 5</b>	´ viivitus 5 sekundit
	<b>switch off 7</b>	´ lülita harukahvel 7 välja
	<b>wait 5</b>	´ viivitus 5 sekundit
	<b>goto algus</b>	´ tagasi algusesse

## symbol

### SYMBOL sümboli nimi = arv

Määratakse arvu vasteks sümbol.

- Sümboli nimi on tekst, mis võib olla kirjutatud tähtedes ning kus võib kasutada alakriipsu. Peale esimest tähemärki võib kasutada ka numbreid ('0'-'9' ).

- Arv on muutuja või konstant, millele võib anda sümboli nime.

Näiteks:

	<b>symbol PUN_LED = 7</b>	´ 7 asendatakse sümboliga PUN_LED
	<b>symbol LOENDUR = B0</b>	´ B0 asendatakse sümboliga
	<b>let LOENDUR = 200</b>	´ 200 on LOENDUR
<b>teine:</b>	<b>high RED_LED</b>	´ lülita harukahvel 7 sisse
	<b>pause LOENDUR</b>	´ viivitus 0,2 sekundit
	<b>low RED_LED</b>	´ lülita harukahvel 7 välja
	<b>pause LOENDUR</b>	´ viivitus 0,2 sekundit
	<b>goto teine</b>	´ mine tagasi päisesse teine

## toggle

### TOGGLE pin-i number

Tee pin-ist väljund ja muuda selle funktsiooni.

- Pin-i number on muutuja/konstant (0-7), mis määrab, missugust sisend/väljundpin-i kasutatakse.

Näiteks:

<b>algus:</b>	<b>toggle 7</b>	´ vaheta harukahvel 7 funktsiooni
	<b>pause 1000</b>	´ viivitus 1 sekund
	<b>goto algus</b>	´ tagasi algusesse

## tune

### TUNE LED, kiirus (meloodia, meloodia,meloodia...)

Mängib meloodiaid PICAXE-08M mikrokontrolleritel (i/o pin2).

PICAXE-08M –mikrokiibil on 4 erinevat sisemist meloodiat, mis ei kasuta palju mälu, kui teeb seda ´tune´ käsklus. Käskluse kasutamine eeldab, et mikroskeemile on ühendatud väljundsummer .

- LED on muutuja/konstant (0 -3), mis määrab kas meloodiaga samaaegselt tuled vilguvad või mitte.

0 – Ei põle

1 – Väljund 0 vilgub

2 – Väljund 4 vilgub

3 – Väljundid 0 ja 4 vilguvad kordamööda

- kiirus on muutuja/konstant, mis määrab meloodia tempo ja noodipikkused.

- meloodiad on kodeeritud. Koodidega meloodiate kirjutamiseks kasuta Tune Wizardit. Lisainformatsiooniks vaata Internetist ´PICAXE-08M –mikrokontrolleri muusikapalade lehekülge.

Näiteks :

#### ´Vilgu vilgu täheke

**tune 3,4,(\$99,\$99,\$A4,\$A4,\$A6,\$A6,\$E4,\$A2,\$A2,\$A1,\$A1,\$9B,  
\$9B,\$D9,\$A4,\$A4,\$A2,\$A2,\$A1,\$A1,\$DB,\$A4,\$A4,\$A2,\$A2,\$A1,\$A1,  
\$DB,\$99,\$99,\$A4,\$A4,\$A6,\$A6,\$E4,\$A2,\$A2,\$A1,\$A1,\$9B,\$9B,\$D9)**

´(vaata telefonihelinate või PICAXE´i lehekülgi kui soovid lisainformatsiooni)

## wait

### WAIT sekundit

Oota mõnda aega.

- Sekundid on muutujad/konstandid (0-65), mis määrab viivituse aja.

**NB:** See on varjatud käsklus, mille tegelik nimetus on käsklus ´pause´ .

Näiteks:

#### algus:

<b>switch on 7</b>	´ Lülita sisse harukahvel 7
<b>wait 5</b>	´ viivitus5 sekundit
<b>switch off 7</b>	´ lülita välja harukahvel 7
<b>wait 5</b>	´ viivitus5 sekundit
<b>goto algus</b>	´ tagasi algusesse