

M2 Programmeerimisvahendid



Õppematerjali koostas ITT Group OÜ 2014/05

Koostajad: Raivo Sell, Rain Ellermaa, Heiko Pikner

Litsents: CC BY-SA 3.0 EE



Mikrokontrollerite ja miniarvutitel programmeerimisel on vaja valida sobiv programmeerimiskeel ja keskkond. Õppematerjal annab ülevaate erinevatest programmeerimiskeeltest ja keskkondadest ja aitab valida sobivat lahendust sõltuvalt püstitatud probleemile.

C-keel mikrokontrolleritel

Intelligentse seadme toimimiseks on vaja arendajal koostada seadmele konkreetne toimeleotika, mis arvestab konkreetse süsteemi riistvaraga. Levinud praktika süsteemi käitumise eeskirja kirjapanekuks on algoritmi koostamine. Algoritmi võib teha graafiliselt kasutades näiteks plokkdiagrammi. See annab hea ülevaate, mida süsteem tegema peaks ja kuidas erinevates olukordades käituma. Graafilist plokk skeemi, aga otse mikrokontrolleri masinkoodiks üldjuhul ei genereerita (va erilähendused nn Lego Mindstorm, Labview, vms) vaid selleks tuleb luua algoritmist lähtuv programmi kood. Kuigi mikrokontrolleritele on olemas enamus kõrgkeelte kompilaatorid on kauaaegne populaarne PC arvutite programmeerimiskeel C laialt levinud ka mikrokontrollerite programmeerimisel. C on lakooniline riistvaralähedane keel, mis võimaldab kirjutada riistvara võimalusi efektiivselt kasutatavat programmikoodi. C-keele süntaksis on võtmesõnu minimaalselt, kuid sisaldab palju erineva tähendusega märke, mis teevad keele õppimise raskeks. Järgnev peatükk sisaldab lühiülevaadet algoritmide ja C-keele süntaksi osadest, mis on kasutusel ka praktiliste näidete juures. Algajal on C-keele kohta soovitatav lugeda täiendavaid materjale.

Programmi ülesehitus

Põhimõtteliselt võib C-keele programmi kirjutada ükskõik mis kujul, kas või üherealisena, sest kompilaator eeldab vaid süntaksireeglite järgimist. Samas on selguse ja lihtsuse huvides ikkagi oluline tähelepanu pöörata ka programmikoodi stiilile. Tüüpiline C-keele programmi ülesehitus:

```
/* Pääsefailide kaasamine */
#include <avr/io.h>
#include <stdio.h>

/* Makro-deklaratsioonid */
#define PI 3.141

/* Andmetüübid */
typedef struct
{
    int a, b;
}
element;

/* Globaalsed muutujad */
element e;

/* Funktsioonid */
int main(void)
{
    // Lokaalsed muutujad
    int x;

    // Programm
    printf("Tere maailm!\n");
}
}
```

Programmi kompileerimine

Selleks, et C keeles kirjutatud koodist oleks võimalik saada mikrokontrollerile arusaadav masinkood kasutatakse kompilaatorit. Kompilaatori ülesandeks on C failidest ja kasutatud tarkvarateekidest kokku panna tervik kood, see optimeerida, et tagada võimalikult kiire ja/või andmemahult väike kood ning teisendada kood mikrokontrollerile arusaadavaks andmejadaks, milleks tavaliselt on .hex fail. Üks populaarseim vabavaralised kompilaatorid on näiteks GCC (GNU C Compiler), mida kasutavad paljud AVR mikrokontrolleri programmeerimiseks mõeldud lahendused (nt WinAVR, Atmel Studio).

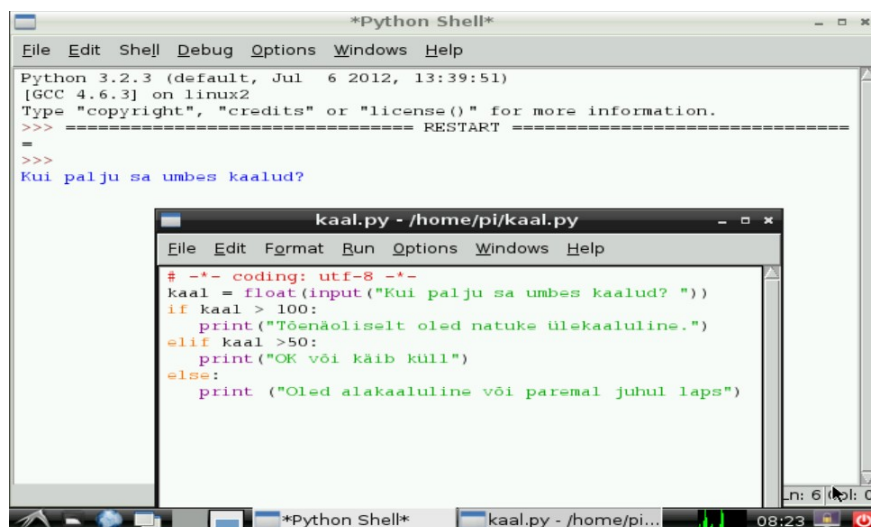
```
:100010000C9434000C9434000C9434000C94340090
:100020000C9434000C9434000C9434000C94340080
:100030000C9434000C9434000C9434000C94340070
:100040000C9434000C9434000C9434000C94340060
:100050000C94340011241FBECFE5D4E0DEBFCDBF29
:100060000E9436000C944C000C940000DF93CF9358
:100070000CDB7DEB787E390E0FC01108284E390E027
:100080002FEFFC01208385E390E026E330E0F901C7
:0C0090002081FC012083F7CFF894FFCF03
:00000001FF
```

Miniarvutite arendusvahendid

Miniarvutite programmeerimiseks saab kasutada mitmeid programmeerimiskeeli, nagu näiteks C, C++, Scratch ja Python. Keerulisemate süsteemsete rakenduste loomiseks on neist sobilikud C-pere keeled. Samas on need oma keerulise süntaksi tõttu raskemini õpitavad. Scratch on visuaalne programmeerimiskeel, mis on mõeldud eelkõige programmeerimise õppimiseks ja prototüüpide tegemiseks. Järgnevalt on pikemalt kirjeldatud Pythoni keelt, arendusvahendeid ja Qt keskkonda kui C-pere keelte arendusvahendit miniarvutitel.

Python miniarvutitel

Python on üks lihtsamini kasutatavaid keeli miniarvutite programmeerimisel. Raspberry Pi miniarvutile on Pythoni tugi vaikimisi lisatud Wheezy operatsioonisüsteemile ja see on ametlikult soovitatud programmeerimiskeel.



```
*Python Shell*
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Jul 6 2012, 13:39:51)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>>
Kui palju sa umbes kaalud?

kaal.py - /home/pi/kaal.py
File Edit Format Run Options Windows Help
# -*- coding: utf-8 -*-
kaal = float(input("Kui palju sa umbes kaalud? "))
if kaal > 100:
    print("Tõenäoliselt oled natuke ülekaaluline.")
elif kaal > 50:
    print("OK või käib küll")
else:
    print("Oled alakaaluline või paremal juhul laps")
n: 6 | pl: 0
```

Programmide sisestamiseks, redigeerimiseks ja täitmiseks kasutatakse interaktiivset kasutajaliidest IDLE (Interactive DeveLopment Environment), mille haldur kuulub Pythoni põhikomplekti. IDLE käivitamiseks võib kasutada töölaual olevat ikooni IDLE 3. Peale seda kuvatakse IDLE käsurea ehk Shelli aken. Sinna väljastatakse tulemused ja teated. Lisaks on võimalik sisestada valemeid, avaldise ja programmikoodi.

Pikemate programmide sisestamiseks kasutatakse redaktorit. Selleks valida menüüst File->New Window. Programmi saab käivitada redaktorist klahviga F5. Programmikoodi sisaldav fail peab olema enne täitmist salvestatud laiendiga .py.

Python on interpreteeritav programmeerimiskeel ehk sellel puudub kompileerimise etapp. Sellevõrra on arendus kiirem ja valminud programmikoodi saab kohe käivitada. Kui programmis on süntaksivigu, kuvab interpretaator vastava teate.

Programme saab käivitada vajadusel Linuxi käsurealt. Näiteks on see vajalik, kui töölaual keskkond ei ole kasutusel. Selleks tuleb käsoreal minna programmi kataloogi sisestada: *python programm.py*, kus „programm“ on käivitatava faili nimi. Kui programm nõuab administraatori õiguseid, tuleb lisada *sudo*. Sellisel juhul tuleb kirjutada: *sudo python programm.py*

Programmi ülesehitus

Python on väga lihtsa süntaksiga keel. Selles puuduvad spetsiifilised eraldajad lausete struktuuri määramiseks võtmesõnade või looksulgude näol. Lausete struktuur on selge ja kompaktne. Lausete ja võtmesõnade hulk keeles on üsna väike. Suur osa tegevusi määratakse funktsioonide ja objektide meetodite abil. Näitena on toodud Pythoni kood, mis muudab GPIO viigu külge ühendatud LED-i olekut vastavalt sisestatud väärtusele.

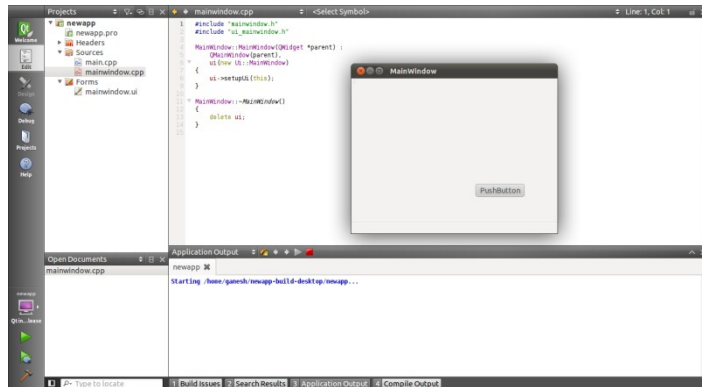
```
import RPi.GPIO as GPIO
ledPin = 11
GPIO.setmode(GPIO.BOARD)
GPIO.setup(ledPin, GPIO.OUT)
while True:
    input_var = input("Enter LED state: ")
    if input_var == 1:
        GPIO.output(ledPin, True)
    elif input_var == 0:
        GPIO.output(ledPin, False)
    else:
        print "Unknown command!"
```

Qt

Qt raamistiku kuuluv Qt Creator on sobilik kiireks ja mugavaks rakenduste loomiseks C ja C++ keeltes. Keskkond on porditud operatsioonisüsteemidele Linux, Mac OS X ja Microsoft Windows. See on installeeritav ka Raspberry Pi miniarvutile, võimaldades ühte ja sama programmikoodi lihtsalt ümber tõsta erinevate operatsioonisüsteemide vahel. Qt Creatori pakett sisaldab koodiredaktorit, Qt Designer (UI arendamiseks),

projektihalduse tööriistu, silurit (debugger) ja mobiilse kasutajaliidese emulaatorit. Qt on vabalt saadaval GNU LGPL litsentsi alusel.

Raspberry Pi miniarvutil on väikese jõudluse tõttu suuremate programmide kompileerimine ajamahukas. Keskkonna seadistamine algajale küllaltki keeruline. Siiski on Qt raamistik oma universaalsuse tõttu hea valik süsteemsete ja keerulisemate rakenduste arendamiseks.

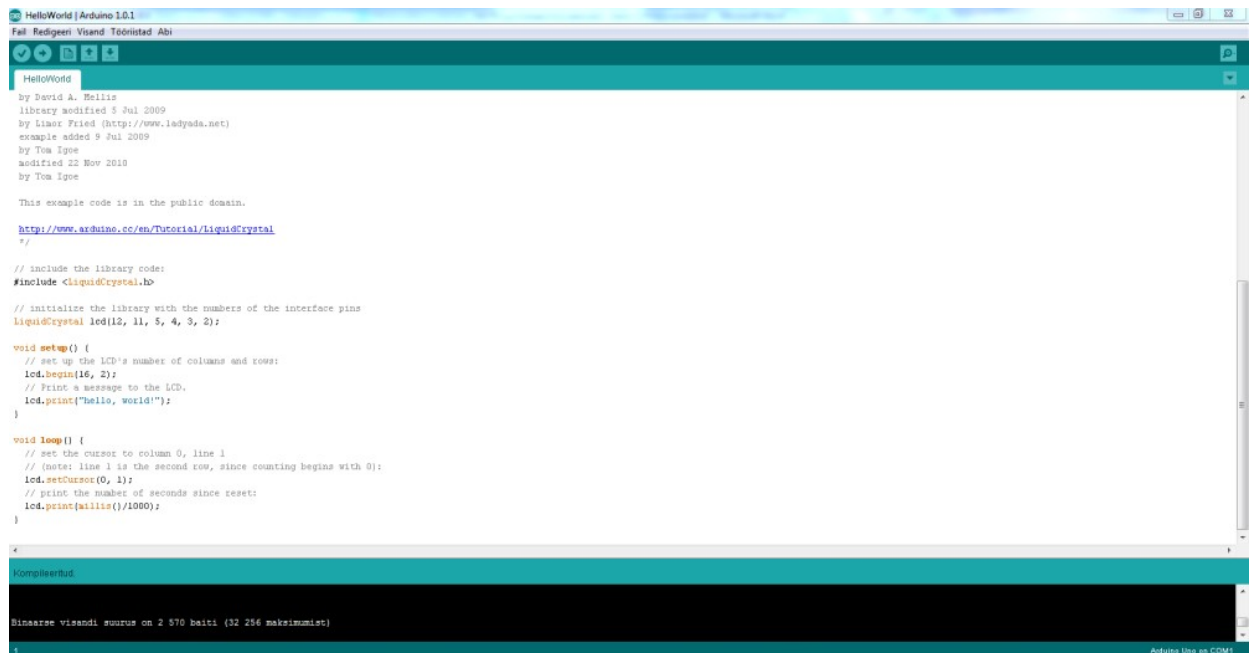


Muud võimalused

Mikrokontrolleritele tarkvara arendamiseks on saadaval mitmeid erinevaid arendusvahendeid. Mõned neist on vabavaralised, mõned jällegi nõuavad litsentsi ostmist. Valik oleneb sellest mida täpsemalt teha tahetakse. Õppimiseks on sobilik valida midagi lihtsat ja vabavaralist. Keeruliste lahenduste loomiseks on vaja teisi vahendeid. Määravaks võib osutuda tehnilise toe olemasolu, versioonihaldus, koodi automaatgenerimine jms.

Üks tuntumaid algajale sobilikke platvorme on Arduino. See on avatud lähtekoodiga ja vabavaraline platvorm millel on mugavamaks kasutamiseks välja toodud erinevad sisend- ja väljundviigud. Tarkvara koosneb standardsest programmeerimiskeskkonnast ja mikrokontrolleris olevast alglaadimise haldurist.

Arduino integreeritud programmeerimiskeskond (IDE) on tarkvaraplatformist sõltumatu Javas kirjutatud rakendus, mis on disainitud selliselt, et seda saaksid kasutada tarkvaraarenduses vähekogenud inimesed. Kommertsrakendustest on Arduino kasutajaliidese baasil arendatud näiteks Libeliumi tarkade anduri võrkude kontseptsioon.



```
>HelloWorld | Arduino 1.0.1
Fail Redigeeri Viisand Teõnõistad Abi

>HelloWorld
by David A. Mellis
library modified 5 Jul 2009
by Limor Fried (http://www.ladyada.net)
example added 9 Jul 2009
by Tom Igoe
modified 22 Nov 2010
by Tom Igoe

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/LiquidCrystal
*/

// include the library code:
#include <liquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

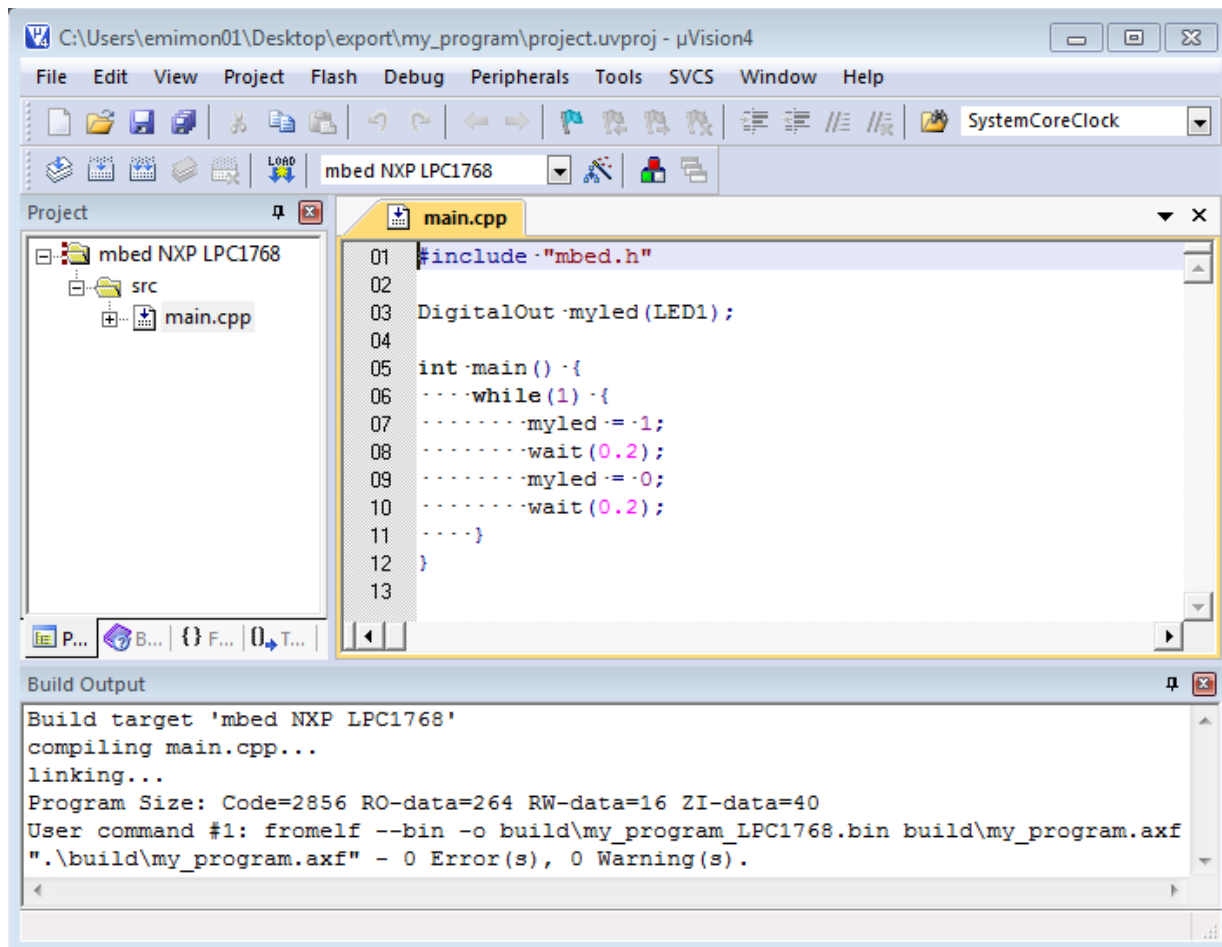
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}

Kompileeritud
Binaarse viisandi suurus on 2 570 baiti (32 256 maksimumist)
Arduino Uno on COM1
```

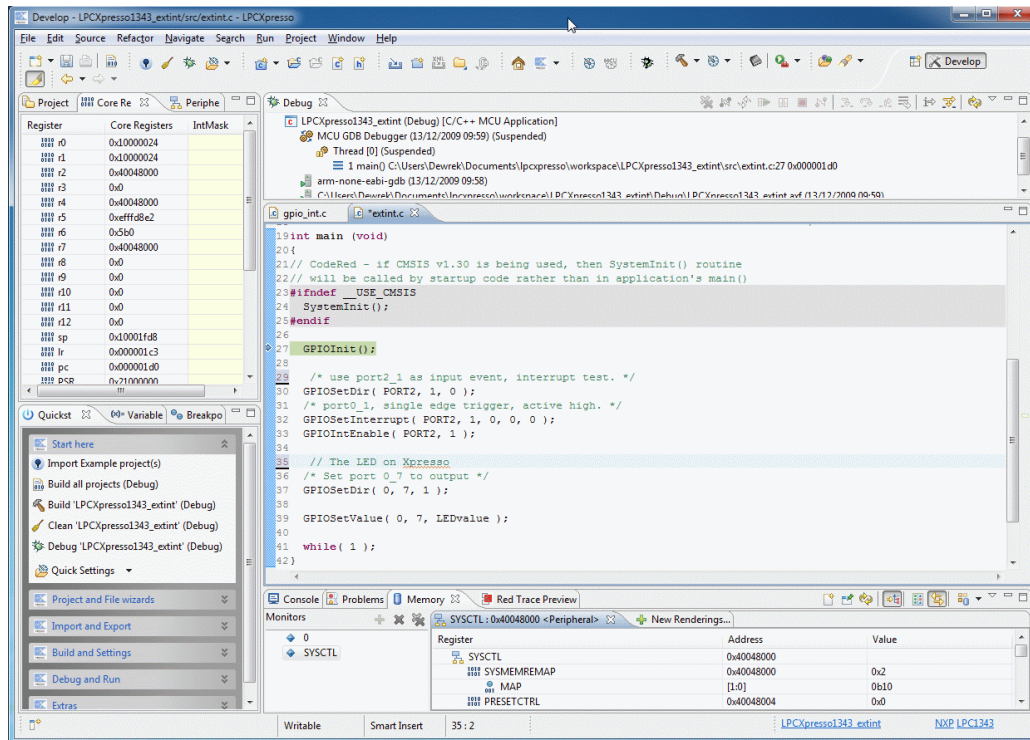
Keil µVision IDE

Keil µVision IDE on võimas programmeerimiskeskond edasijõudnud arendajatele mikrokontrollerite tarkvara loomiseks C ja C++ keeltes. µVision IDE sisaldab komponente koodi silumiseks ja mikrokontrollerite simuleerimiseks. Piiratud võimalustega tarkvara on vabalt allalaaditav, täisversioon on tasuline. Toetatud on väga mitmete tootjate mikrokontrollerid, nagu STM32, NXP ja Stellaris.



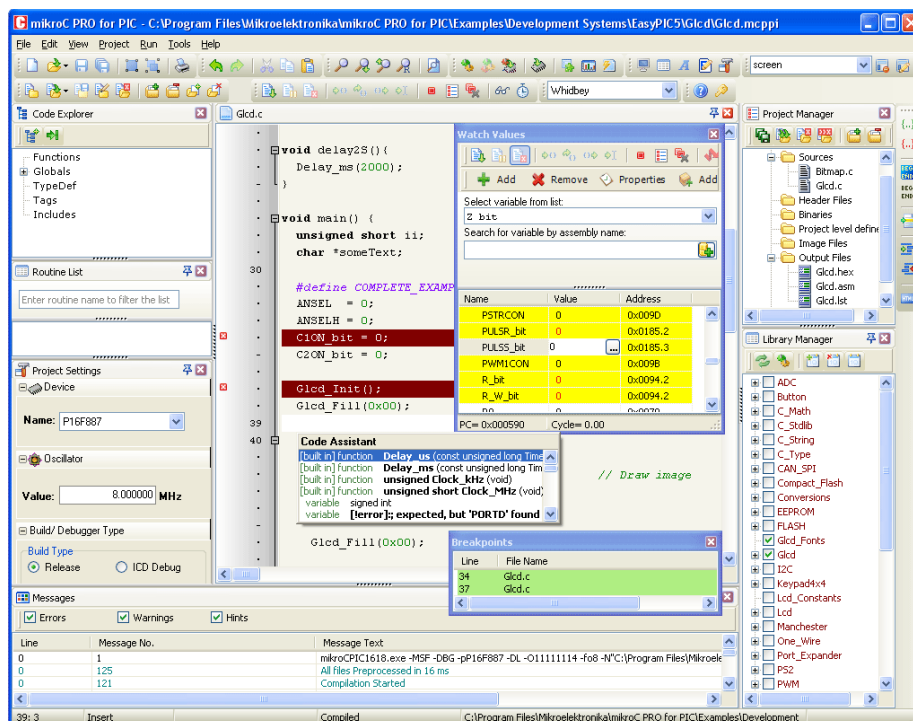
LPCXpresso

LPCXpresso on NXP arendatav arenduskeskkond nende enda ARM-i põhiste mikrokontrollerite programmeerimiseks. Keskkond sisaldab kompilaatorit, silurit ja paljusid teke mugavaks tööks. Paljudele programmeerijatele on LPCXpresso tuttav ja kergesti kasutatav, kuna on ehitatud Eclipse põhjale. Lisaks tasuta programmeerimiskeskonnale pakub NXP veel veebi põhist liidest programmide koostamiseks, ja koodi haldamiseks.



mikroC

mikroC on täisfunktsionaalne arendusvahend peamiselt PIC ja mõnede ARM mikrokontrollerite tarkvara arendamiseks. Sisaldab ANSI C kompilaatorit, optimeerimisvõimalusi ja suurt valikut erinevaid teeke. Lisaks on olemas valmis näitekoodid ja kättesaadav rohke abimaterjal. mikroC litsents sisaldab uuendusi ja tehnilist tuge. Saadaval on piiratud prooviversioon toote testimiseks.



Kokkuvõte

Antud mooduli eesmärgiks oli anda ülevaade erinevatest programmeerimiskeskondadest, keeltest ja töövahenditest, mida läheb vaja miniarvutite ja mikrokontrollerite tarkvara arendamiseks. Pikemalt on peatunud C keelel ja Pythonil, mis on enamlevinumad programmeerimiskeeled robotika ja teiste tarkade rakenduste loomisel. Python on soovitatud programmeerimiskeel Raspberry Pi jaoks, kuid on ära tood ka mitmed alternatiivsed programmeerimiskeskonnad nii Raspberry Pi, kui ka teiste levinumate miniarvutite ja mikrokontrollerite programmeerimiseks.

Lisalugemist ja viited

1. C- keele kiirkursus - <http://home.roboticlab.eu/et/programming/c/crashcourse>
2. Python programmeerimiskeele juhend - <http://rlpa.ttu.ee/python/Python.pdf>
3. Qt raamistik (Seminaritöö) - www.cs.tlu.ee/teemaderegister/get_file.php?id=175
4. Arduino veebileht - <http://www.arduino.cc>
5. Keil µVision - <http://www.keil.com/uvision/>
6. LPCXpresso - <http://www.lpcware.com/lpcxpresso/home>
7. Mikroc erinevad versioonid - <http://www.mikroe.com/mikroc>