μ**CC**

Micro Controlled Car

Työryhmä: Mysse[1]

Käyttöohje v1.4

1

Tiina Kumento	40512E
Ilari Lehti	39098T
Joni Suominen	40954L
Anu Vilvala	38304A

Sisältö

1 Yleiskuvaus

Tampereen TKK:n Ohjelmistotekniikan laitos ja TKK:n Tietojenkäsittelyopin laboratorio ovat yhteistyössä kehittämässä sulautettujen järjestelmien opetustaan. TTKK:n osuus on suunnitella ja toteuttaa harjoitustöissä käytettävä laitteisto: Kauko-ohjattavasta autosta poistetaan radiovarustus ja tilalle laitetaan pieni mikrokontrolleri, jolla auton nopeutta ja etupyöriä voidaan säätää. Tämä on siis mikro-ohjattava auto (micro-Controlled Car, μCC). Autossa on anturit, joilla voidaan havaita lattiaan kiinnitetty heijastava teippi. Ohjauksen tehtävänä on ajaa mahdollisimman nopeasti teipillä tasaiseen lattiaan merkittyä rataa pitkin.

Tässä kuvattu ohjelmisto toteuttaa kyseiselle laitteistolle simulointijärjestelmän. Simulaattorin on tarkoitus auttaa autolle tehdyn ohjausohjelman testauksessa siten, että ohjelma voidaan ajaa läpi kerralla ja seurata auton liikettä radalla tai suorittaa ohjelma haluttuun pisteeseen ja seurata rekisterien arvoja ja muistin sisältöä, jotta saadaan selvitettyä ohjausohjelman algoritmiset tai koodausvirheet. Ohjelma simuloi todellisen fyysisen auton toimintaa, eikä jotakin teoreettista esimerkiksi välittömästi ohjaussignaaleihin reagoivaa autoa. Simulaattorissa on samat osajärjestelmät kuin fyysisessä, Tampereella tehtävässä laitteessa. Nämä osat ovat mikrokontrolleri, uBoard-piirikortti, anturit, toimilaitteet, johdotus, auto ja rata. Lisäksi on käyttöliittymä mikrokontrollerin ohjaamista ja auton liikkeen seuraamista varten. Käyttöliittymässä on useita ikkunoita, joissa näkyvät auton paikka radalla, auton nopeus ym. tiedot, toimilaitteiden ja antureiden tilat ja prosessorin tila.

Auton fysiikkamallissa on aiheen kompleksisuuden vuoksi tehtävä kompromisseja. Ohjelmiston nopeus tullee käytännössä riippumaan nopeaa sykliä pyörivien mikrokontrolleriosan ja toimilaiteosan nopeudesta.

2 Käyttäjän ohje

2.1 Ohjelman käynnistys

Ohjelma käynnistetään sanomalla komentorivillä:
uCC &

Käynnistettäessä ohjelma voi sille antaa myös komentoriviargumentteja. Lista ohjelman ymmärtämistä argumenteista:

Hakasulkeiden sisällä ovat optionaalisia argumentteja.

OPTIO	SELITYS	OLETUSARVO
[**-b** *N*]	pysähtymispiste N:n käskyn jälkeen	0
[**-c** *koodi*]	kooditiedoston nimi	code.s19

[**-r** *ehto*]	pysähtymispiste rekisteriehdoilla	
[**-t** *rata*]	ratatiedoston nimi	track.uCC

Esimerkki:
uCC -r 'PC < 321a' -t rata.uCC -c koodi.s19 &
Nyt ohjelman käynnistyessä oletusarvo rekisterien pysähtymispiste-ehdolle korvataan (rekisteriehto: PC < 321a) ja ohjelmaan ladataan koodi tiedostosta koodi.s19. Rata luetaan tiedostosta rata.uCC. Mikäli kyseistä tiedostoa ei ole (tai se on väärässä formaatissa), käytetään sisäänrakennettua rataa. Ratatiedoston oikea formaatti on kuvattu kappaleessa 5. S19-koodin luomisesta on kerrottu kappaleessa 4.1.

2.2 Asetukset

Ohjelman asetuksia voidaan muuttaa, kun simulaation suoritus ei ole käynnissä. Asetuksilla on ohjelman käynnistyttäessä oletusarvot, oletusarvoja voi antaa myös komentoriviargumenteina *(ks. Kappale 2.1)*.

2.2.1 Pysähtymispisteet (Breakpointit)

Pysähtymispisteet ovat simulaation suorituksen lopetusehtoja ja ne asetetaan päälle painamalla tekstin *breakpoints* vieressä olevaa nappia. Pysähtymispisteitä voidaan antaa seuraavilla tavoilla:

- Suoritettujen käskyjen määrä

Suoritus pysähtyy, kun on suoritettu käyttäjän haluama määrä mikrokontrollerin käskyjä.

- Data-alue

Jos koodia suoritetaan käyttäjän määrittelemältä data-alueelta, pysäytetään simulaatio. Näin käyttäjällä on mahdollisuus 'kieltää' mikrokontrolleria suorittamasta dataa.

- Rekisteriehdot

Suoritus pysähtyy kun ehtolauseke on tosi. Ehtolauseke on muotoa:
EHTO1 [*looginen operaattori* EHTO2 ...], missä EHTON on
REG *vertailuoperattori* ARVO. Loogisia operaattoreita ovat *and* ja *or*, vertailuoperaattoreita ovat *<, >, =, # (≠)*.

Ehdoissa käytettävät sallitut rekistereiden ja lippujen tunnukset ovat seuraavat:

Rekisterit:
- A – *akku A (ei vielä käytössä)*
- B – *akku B (ei vielä käytössä)*
- D – *kaksoisakku D*
- IX – *indeksirekisteri IX*
- IY – *indeksirekisteri IY*
- SP – *pino-osoitin*
- PC – *ohjelmalaskuri*

Liput:
- S – *stop disable*
- X – *x interrupt mask*
- H – *half-carry*
- I – *i interrupt mask*
- N – *negative*
- Z – *zero*
- V – *overflow*
- C – *carry*

And operaattorilla katsotaan olevan korkeampi presedenssi kuin *or* operaattorilla.

Esimerkki:
A = 45 or PC > 34ff.

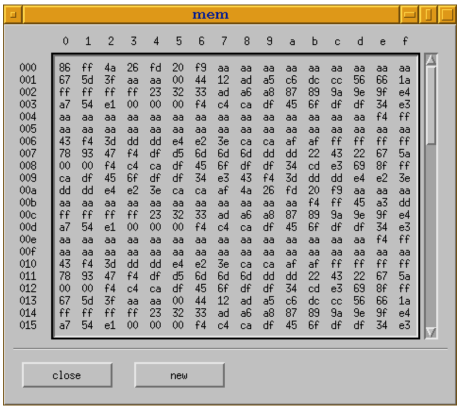
2.2.2 Loki

Lokiin kirjoitetaan simuloitavan prosessorin rekisterit ja muisti. Muistialueen koon voi rajata haluamakseen ja rekisterien tulostuksen voi asettaa pois päältä. Mikäli ei halua mitään tietoja lokiin, voi lokiin tulostuksen ottaa pois päältä sille varatusta painonapista. Lokia kannattaa pitää päällä vain kun todellakin tarvitsee lokitietoa, koska lokin tekeminen hidastaa jonkin verran simulaatiota.

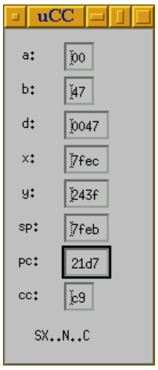
2.2.3 Tilatiedon esittäminen

Pääkonsolin visualisointikohdasta voidaan vaikuttaa siihen, mitkä tiedoista näytetään ruudulla *(ks. kuva 4)*. Näitä tietoja voi myös muutella simulaation ollessa käynnissä.

Muisti esitetään taulukossa heksadesimaalilukuina *(ks. kuva 1)*. Näitä ikkunoita voi avata tarpeen vaatiessa lisää napista NEW ja siten voi seurata niitä muistiosoitteita, joista on kiinnostunut. Muistiin voi kirjoittaa painamalla hiiren vasenta nappia halutun muistiosoitteen kohdalla, jolloin muisti-ikkunan alareunaan aukeaa kohta syöttötietoa varten.



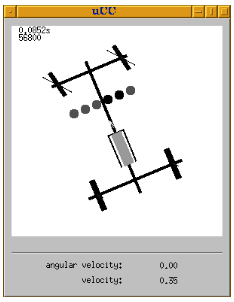
Kuva 1: Muistin esittäminen



Kuva 2: Rekisterit

Rekisteri-ikkunasta voi katsoa ja muuttaa rekisterien arvoja *(ks. kuva 2)*. Arvojen muutos tapahtuu aktivoimalla halutun rekisterin arvo-osa ja kirjoittamalla siihen uusi arvo. Arvoja voi muuttaa ainoastaan silloin kun simulaatio ei ole käynnissä. Ehtorekisteri on ikkunan alalaidassa nähtävissä myös aukikirjoitettuna (bittitasolla).

Auton tila esitetään kahdessa eri ikkunassa. Toisessa auton sijainti radalla ja toisessa auton parametrit (voimat, suunta, nopeus jne.) Kuvassa 3 on hahmoteltu parametrien esitystapa. Renkaista lähtevät voimat ovat autoon vaikuttavia kitkavoimia. Rengaskulma esitetään eturenkaiden kääntymisellä, samoin kuin ohjauksen kohdekulma (ohuella viivalla piirretyt eturenkaat). Moottorin teho visualisoidaan auton rungossa olevalla palkilla, josta näkee kuinka monta prosenttia moottorin tehosta yritetään saada. Anturit ovat värillisiä ympyröitä, joiden sävyt vaihtelevat antureiden kirkkauden mukaan. Nopeusvektori lähtee auton rungon keskipisteestä. Auto myös pyörii keskipisteensä suhteen, riippuen missä suunnassa auto on. Muut arvot, eli kulmanopeus, suoritettujen käskyjen lukumäärä ja aika ovat skalaareja ja nämä näytetään numeroarvoina.



Kuva 3: Auton tila

Rata näytetään kaksiuloitteisena ja auto liikkuu tätä rataa pitkin. Rataan piirtyy jälki auton kulkemasta reitistä.

2.3 Suoritus

2.3.1 Suorituksen käynnistys

Suoritus käynnistyy painamalla nappia EXECUTE, tätä ennen on radalla oltava jokin aloituskohta. Mikäli aloituskohtaa ei ole annettu, ilmoitetaan siitä ikkunan tilarivillä *(ks. kappale 2.4)*. Aloituskohta asetetaan klikkaamalla hiiren vasenta nappia rataikkunassa halutussa kohdassa. Ennen tätä voi auton suuntaa vaihtaa painelemalla hiiren oikeaa nappia (auto kääntyy hieman myötäpäivään, joka painalluksella) tai hiiren keskimmäistä nappia (auto kääntyy vastapäivään).

2.3.2 Suorituksen pysäytys

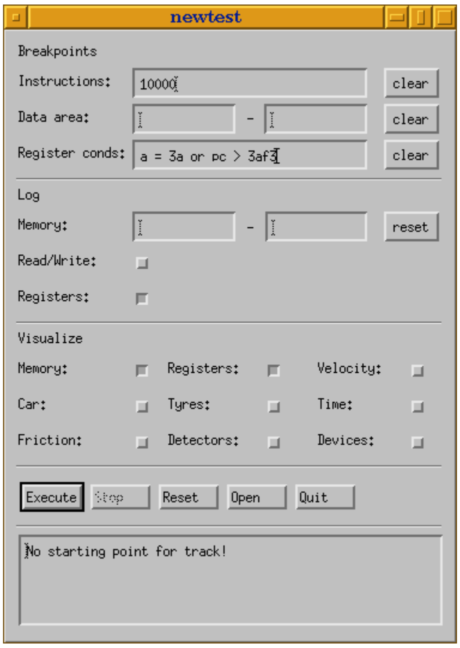
Suoritus pysähtyy seuraavissa vaihtoehdoissa:

- Käyttäjä pysäyttää suorituksen

Käyttäjä voi halutessaan pysäyttää simulaation. Tämä tapahtuu painamalla nappia STOP. Suoritus pysähtyy pienellä viiveellä ja voidaan käynnistää jälleen painamalla nappia EXECUTE (tällä välillä käyttäjä voi esimerkiksi muutella asetuksia ym.).

- Havaitaan pysähtymispiste

Ohjelman suoritus pysähtyy kun havaitaan käyttäjän asettama pysähtymispiste (ks. Kappale 2.2.1.).



Kuva 4: Pääkonsoli

- Prosessorisimulaattori havaitsee virheen

Prosessori pysähtyy, kun yritetään suorittaa simulaattorille tuntematon käsky, kun yritetään kirjoittaa muistin ROM-alueelle tai jos suoritetaan STOP-käsky.

- Auto ei ole enää radalla

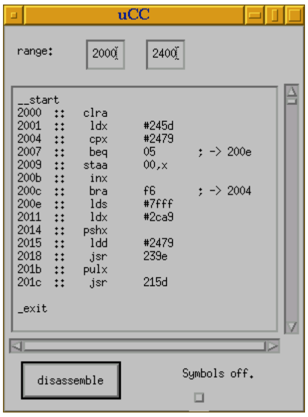
Käyttöliittymä saa fysiikkamallilta tiedon auton sijainnista. Jos auto ei ole radalla pysäytetään simulaatio (tämä on ohjausohjelman kannalta virhetilanne, auto ei saa poistua radalta). Koska teipissä voi olla lyhyitä katkoksia, esimerkiksi tietona tulevasta mutkasta, pysäytetään simulaatio vasta kun auto on ollut 2 s pois radan päältä.

2.4 Virheiden havaitseminen

2.4.1 Disassembler

Ohjausohjelman virheiden havaitsemiseksi on käytössä disassembler *(ks. kuva 5)*.

Disassemblerille pitää antaa koodialue mikä halutaan disassembloida. Aina kun suoritus pysähtyy, näytetään ikkunasta korostettuna juuri suoritettu



Kuva 5: Disassembler

käsky (mikäli ohjelmalaskuri on disassembloidulla alueella). Mikäli symbolitaulu on käytössä (ks. Kappale 2.4.2), voi alalaidassa olevasta napista ottaa symbolit käyttöön. Suhteelliset hypyt kommentoidaan käskyn jälkeen.

2.4.2 Symbolitaulu

Jos suoritettavasta ohjelmasta on olemassa symbolitaulu, avautuu tätä varten oma ikkunansa mistä näkyy symbolin nimi, osoite ja osoitteen osoittaman muistipaikan sisältö. Muistipaikan sisällön tyypin voi muuttaa valitsemalla uusi tyyppi pull-down menusta.

Symbolitaulutiedoston nimi pitää olla sama kuin kooditiedoston + .map. Esimerkiksi jos kooditiedosto on *code.s19*, niin symbolitaulutiedoston nimi on *code.s19.map*. Symbolitaulussa määritellään symbolien nimet sekä osoitteet ja lisäksi onko symboli koodia vai dataa. *Text* eroittimen jälkeen olevat symbolit ovat koodia ja vastaavasti *data* eroittimen jälkeiset ovat dataa.

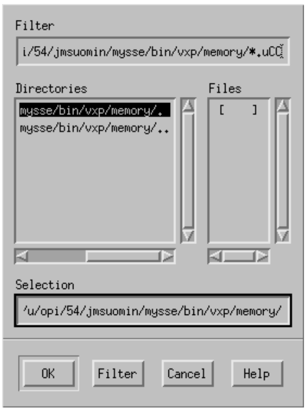
Esimerkki symbolitaulusta:

```
text
       2000   __start
       201F   _exit
       2021   _servoint
       2045   _rtint
       215D   _main
       219C   _steer
       21A7   _drive

data
       23F3   _drvval
       2433   _steerval
       2449   _ch2
```

```
244B  _ch4
244D  _realtime
244F  _mmask
2451  _kmask
2453  _attmask
2455  _speedpoint
2457  _speed
2459  _disable
245B  _laststeer
```

2.5 Uuden koodin lataus

Pääkonsolin painonapista OPEN *(ks. kuva 4)* voi ladata uuden koodin suoritettavaksi. Tällöin avautuu valinta-dialogi *(ks. kuva 6)*, josta voi valita halutun koodin. Vaikka valinta-dialogin suodatin onkin oletusarvoisesti *.s19, ei kooditiedoston prefixin tarvitse olla s19. Ohjelmalaskurin (PC) alkuarvo luetaan reset-vektorista (osoitteet 0xFFFE, 0xFFFF).



Kuva 6: Uuden koodin lataus

2.6 Tilarivi ja virhetilanteet

Konsoli-ikkunan alalaidassa sijaitsee tilarivi, jonne mahdolliset virheilmoitukset tai muut huomautukset kirjoitetaan. Kaikki tilarivin ilmoitukset eivät ole virheilmoituksia!

TILARIVIN TEKSTI	SELITYS
Executing...	Simulaatio on käynnissä.
No starting point for track!	Radalle ei ole annettu aloituspistettä.
Breakpoint:	Suoritus pysähtyi pysähtymispisteeseen,

	myös syy tulostetaan.
Unknown instruction!	Simulaattori havaitsi tuntemattoman käskyn.
Invalid settings!	Asetuksissa on vikaa.
Code file: *filename* **non-existent!**	Kooditiedostoa ei löydy.
Error in Code file: *filename***!**	Kooditiedostossa on jokin virhe.

3 Auton tekniikka ja fysiikka

Auton liikkeen mallinnus on tehty renkaiden ja lattian välisiin kitkavoimiin perustuen. Näin saatu dynamiikka on tasaisella alustalla suhteellisen tarkka. Mallissa on kuitenkin lukuisia epävarmuustekijöitä kuten esimerkiksi kitkakertoimien tai moottorin antaman tehon määrittely. Tästä syystä simulaattorin perusteella ei pidä määrittää esimerkiksi kiihdytystä aivan pidon rajalle, koska todellisen auton käyttäytyminen ei välttämättä vastaa simuloitua tapausta.

Sen sijaan mallista tulee huomioida seikat, joista voidaan päätellä ohjausohjelmiston loogiset virheet. Esimerkiksi tapauksessa, jossa seurattava teippi uhkaa karata auton alustan ulkopuolelle, kannattaa tarkkailla onko renkaiden käännöskulma juuri sellainen mitä on tarkoitettu. Tällöinkin on ohjausohjelman tietenkin huomoitava, että se kykenee antamaan käskyjä nopeammin, mitä todellinen auto tai simulaattori kykenee vastaamaan.

Jos ohjausohjelma tarkkailee autossa olevaa käynnistyskytkimen painamista, auton liikkeellelähtö viivästyy. Ohjelmassa nimittäin simuloidaan kyseistä kytkintä siten, ett se painetaan pohjaan simuloinnin alkaessa ja vapautetaan noin puolen sekunnin kuluttua. Tätä joutuu simuloinnin aluksi hetkisen odottamaan, koska simulointi ei pyöri reaaliajassa.

4 Prosessorisimulaattori

4.1 Suoritettava koodi

Prosessorisimulaattori lukee suoritettavan koodin s19-formaatissa. Kooditiedoston voi tehdä esimerkiksi C-ristiinkääntäjällä. Kooditiedosto ladataan simulaattoriin käyttöliittymästä. Simulaattori osaa suorittaa kaikki Motorola 68HC11A1:n dokumentoidut käskyt.

4.2 Suorituksen pysäytys

Koodin suoritus on mahdollista pysäyttää asettamalla käyttöliittymästä pysähtymispisteitä (breakpointteja). Pysähtymispisteitä voivat olla suoritettujen käskyjen määrä, ohjelmalaskurin (program counterin) joutuminen dataalueelle, jonka käyttäjä voi määritellä, ja rekisterien arvot. Pysähtymispisteen jälkeen koodin suorittamista voi jatkaa siitä, mihin on jääty.

Prosessori pysähtyy myös silloin, kun yritetään suorittaa tuntematon käsky. Silloin koodin suoritusta ei enää voi jatkaa, mutta prosessorin tilaa voi tutkia normaaliin tapaan.

4.3 Muisti, rekisterit ja loki

Rekisterien ja muistin sisältö on mahdollista saada näytölle, aina kun prosessori on pysähtynyt.

Lokitulostukset menevät tekstitiedostoon. Lokiin voi tulostaa kaikki rekisterit, käyttäjän määrittelemän muistialueen ja muistiosoitteet, joista on luettu tai joihin on kirjoitettu, ja mitä on luettu/kirjoitettu.

5 Rata

5.1 Radan kuvaus

Ratatiedosto sisältää vektorikuvauksen radasta (alkupiste ja loppupiste pareja). Lisäksi tiedostossa on oikea mittakaava, eli kuinka monta metriä on yksikkö. Tiedostossa on myös tieto siitä, kuinka paljon teipinleveys on metreinä.

Oikeassa formaatissa olevan tiedoston ensimmäisen numeron on oltava radan mittakaava ja tämän jälkeen teipinleveys. Näiden jälkeen luetellaan ratavektorit. Risuaidan jälkeiset merkit ovat kommentteja seuraavaan rivinvaihtoon saakka. Tyhjillä riveillä, tabulatuureilla tai välilyönneillä ei ole merkitystä.

Esimerkki ratatiedostosta:

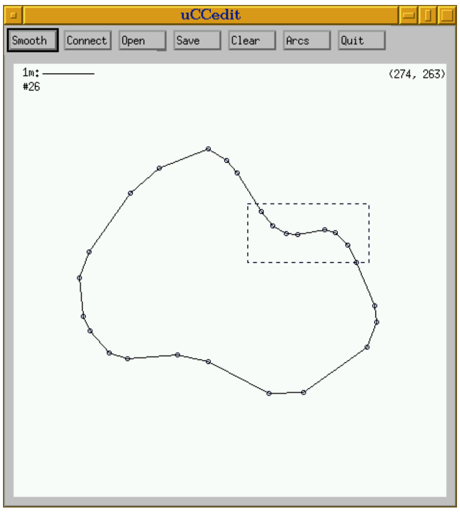
```
#
# Esimerkkirata.
#

# Mittakaava:
1.0

# Teipinleveys:
0.015

# Ratavektorit:
0.88  5.0   1.3   5.1
1.3   5.1   1.9   5.25
1.9   5.25  3.2   5.15
3.2   5.15  4.15  5.0
4.15  5.0   4.9   4.4
```

4.9	4.4	5.25	3.6
5.25	3.6	5.27	3.35
5.27	3.35	5.24	2.9
5.24	2.9	4.9	2.03
4.9	2.03	4.02	1.7
4.02	1.7	3.0	1.67
3.0	1.67	1.8	1.78
1.8	1.78	0.88	2.3
0.88	2.3	0.6	3.0
0.6	3.0	0.48	3.6
0.48	3.6	0.53	3.8
0.53	3.8	0.65	4.5
0.65	4.5	0.88	5.0

Jos ratatiedoston parsiminen epäonnistuu (esimerkiksi tiedosto on virheellinen tai sitä ei löydy), ohjelma käyttää sisäänrakennettua rataa.

5.2 Rataeditori

Ratojen tekemisen helpottamiseksi on tehty editori *(ks. kuva 7.)*



Kuva 7: Rataeditori

Ohjelma käynnistyy komentoriviltä käskyllä:
uCCedit &

Ohjelmalle voi antaa komentoriviargumentteina teipinleveyden (-t) ja mittakaavan (-s). Ikkunan ylälaidassa näkyy metrin mittainen jana, ratavektorien lukumäärä sekä kursorin sen hetkinen asema.

Ratavektoreita piirretään vasemmaisimmalla hiiren napilla ja piirtäminen lopetetaan painamalla keskimmäistä tai oikeaa nappia. Keskimmäisellä napilla voi valita alueen käsiteltäväksi.

Toiminnot:

- **Smooth**

  Pyöristää valitulla alueella olevia kulmia.

- **Connect**

  Yhdistää valitulla alueella olevat pisteet. Tätä voi käyttää myös kaikkien vektorien poistamiseen valitsemalla kaikki vektorit ja yhdistämällä ne.

- **Open**

  Tiedoston avaus.

- **Save**

  Tiedoston tallennus.

- **Clear**

  Poistaa valitulla alueella olevia vektoreita yksi kerrallaan siinä järjestyksessä kun ne on piirretty.

- **Arcs**

  Poistaa vektorien päätepisteissä olevat ympyrät näkyvistä. Painamalla nappia uudestaan tulevat päätepisteet taas näkyviin.

- **Quit**

  Lopettaa ohjelman.