

Riku Lokka

MITTAUSJÄRJESTELMÄ PESÄPALLOMAILAN NOPEUDEN MITTAAMISEEN

Insinööri
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2004

Kajaani Polytechnic

Faculty Faculty of Engineering	Degree programme Information Technology
Author(s) Riku Lokka	
Title A Measurement System for the Speed and Kinetic Energy of the Finnish Baseball Stick	
Optional professional studies Measurement technology and machine vision	Instructor(s) / Supervisor(s) Tuomo Rantala
Date April 2004	Total number of pages 31+8
<p>Abstract</p> <p>The purpose of this final year project was to create a measurement system, which will help Finnish baseball players to choose a baseball stick of the right weight. The purpose of this equipment is to measure the speed and relative kinetic energy of the Finnish baseball stick. The equipment includes a microprocessor which is programmed by the C language.</p> <p>The player strikes the Finnish baseball stick through two infrared beams. The equipment measures the time between the two infrared beam. From this time result the equipment calculates the speed [m/s] and kinetic energy [J], which are then shown on an LCD display.</p> <p>One Finnish baseball player tested the equipment in Kajaani Halli. The measurement system worked impeccably and will help Finnish baseball players to choose baseball sticks of the right weight in the future.</p>	
Confidential	
Yes	
No x	
Keywords Measurement system, Finnish baseball	
Deposited at Kajaani Polytechnic	

ALKUSANAT

Tämän insinööriyön tilaajaksi ryhtyi Kajaanin Pallokerhon valmentaja Pentti Määttä, jolle ehdotin tällaisen projektin toteuttamista. Insinööriyössä suunniteltiin ja toteutettiin mittausjärjestelmä pesäpallomailan valintaan. Mittausjärjestelmä rakennettiin mikrokontrolleripohjaiseksi, ja sitä ohjelmoitiin C-kielillä.

Haluan kiittää insinööriyön valvojaa Tuomo Rantalaa ja laboratorioinsinööri Ismo Talusta saamistani neuvoista sekä Eero Soinista kielellisestä ohjauksesta.

Riku Lokka

7. huhtikuuta 2004

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

ALKUSANAT

1 JOHDANTO	6
2 MITTAUSJÄRJESTELMÄN TOTEUTUSVAIHTOEHTOJA.....	7
3 MIKROTIETOKONE JA SEN PÄÄOSAT	8
3.1 Prosessori	8
3.2 Ohjelmamuisti, Program Memory	9
3.3 Datamuisti, Data Memory	10
3.4 Kello-oskillaattori, Clock Oscillator	11
3.5 Nollaus, Reset	11
3.6 I/O-liitännät	11
3.7 Väylät, Buses	11
3.7.1 Osoiteväylä, Address Bus.....	12
3.7.2 Dataväylä, Data Bus	12
3.7.3 Ohjausväylä, Control Bus	12
4 INTEL MSC-51 -MIKROKONTROLLERI	13
4.1 Prosessorin ajastin/laskuri-toiminto	17
4.2 Prosessorin keskeytystoiminnot	18
5 C-KIELI OHJELMOINTIKIELENÄ	20
6 MITTAUSJÄRJESTELMÄ.....	21
6.1 Valokennot ja niiden valinta.....	21
6.2 Nopeuden ja suhteellisen liike-energian laskeminen.....	22
6.3 Järjestelmässä käytetyt laitteet ja komponentit	22
6.4 Työn suoritus ja sen eri vaiheet.....	23
6.5 Mittausjärjestelmän testaus.....	26
7 MITTAUSTULOKSET	28
8 TYÖN TAVOITTEIDEN TOTEUTUMINEN.....	29
9 YHTEENVETO.....	30
LÄHDELUETTELO	31

LIITTEET

1 JOHDANTO

Urheilun ja tietotekniikan yhdistäminen on ollut suhteellisen vähäistä kautta aikojen. Erilaisia mittausjärjestelmiä voitaisiin kehittää lisää useisiin eri urheilulajeihin. Esimerkiksi käytetyimmät ajanottolaitteet ovat usein hyvin vanhentuneita. Yleisessä käytössä ovat edelleen käsikäyttöiset kellot, joilla ajanotto on pikemminkin suuntaa antavaa. Tänä päivänä olisi kohtuullisen edullista rakentaa tietokoneohjattuja mittausjärjestelmiä, joilla esimerkiksi ajanotto saadaan uuteen tarkkuusluokkaan. Toki huippu-urheilussa on jo käytössä tarkkoja mittauslaitteita, mutta niiden saatavuus harrastelijatasolle on vähäistä. Esimerkiksi yksinkertaisen ajanottolaitteen rakentaminen ei materiaalikustannuksiltaan poikkea suuresti hyvästä ajanottokellosta.

Tietokoneohjattuja mittaus-, ajanotto- ja testausjärjestelmiä on markkinoilla hyvin vähän ajatellen, kuinka suurta on urheilun harrastaminen. Suomessa tämäntapaisia järjestelmiä kehittää Kilpa- ja huippu-urheilun Tutkimuskeskus (KIHU), jonka projektit on kuitenkin suunnattu lähinnä palvelemaan huippu-urheilua. Tällaiset tuotteet eivät usein tavoita seuratason urheilijoita. Huippu-urheilussa käytettävät järjestelmät ovat erittäin laadukkaita, mutta samalla hyvin kalliita. Olisi hyvä saada kohtuuhintaisia, mutta samalla laadukkaita mittauslaitteita yleisempään käyttöön. Tällöin myös tavalliset urheilun harrastajat pääsevät nauttimaan tietotekniikan kehityksen tuomasta mukavuudesta.

Insinööriyössäni kehitän mittausjärjestelmän pesäpallon pelaajille. Tarkoituksena on helpottaa pelaajaa valitsemaan itselleen oikeanpainoinen maila, jolla hän aikaansaa maksimaalisen liike-energian pesäpallomailaan. Tällä on urheilijan sisäpelisuoritukseen olennainen merkitys. Mitä suuremman liike-energian pelaaja saa tuotettua mailalla palloon, niin todennäköisesti sitä vaarallisempi on lyöntikin. Työ toteutettiin mikrokontrolleripohjaisena. Sillä voidaan mitata valokennojen avulla pelaajan heilauttaman mailan liikenopeutta ja suhteellista liike-energiaa eripainoisilla mailoilla.

2 MITTAUSJÄRJESTELMÄN TOTEUTUSVAIHTOEHTOJA

Pesäpallomailan liikenopeuden mittaamiselle on useita erilaisia toteutusvaihtoehtoja. Mittauslaitteena voidaan käyttää esimerkiksi lasertutkaa, kiihtyvyyssanturia, valokennoja tai konenäkökameraa.

Lasertutkan käyttäminen mittauslaitteena voidaan toteuttaa sijoittamalla se syöttölautasen taakse, lyöntisuuntaan nähden vastakkaiselle puolelle. Näin lasertutka mittaa mailan nopeuden. Ongelmana tällä tavalla toteutettuna on sen epäluotettavuus. Missä vaiheessa lyöntiliikettä tutka mittaa nopeuden? Oleellista on saada mailan nopeus syöttölautasen kohdalta.

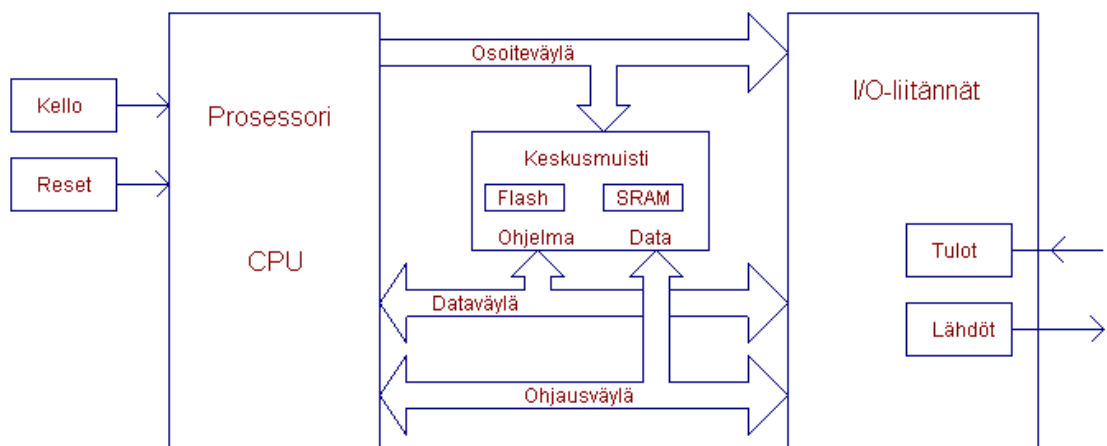
Kiihtyvyyssanturilla toteutettuna mittausjärjestelmä on hyvä. Tällä tavoin voidaan mitata mailan liikenopeutta koko sen heiluriliikkeen matkalta. Näin voidaan analysoida nopeuden lisäksi myös tekniikkaa. Missä vaiheessa lyöntisuoritusta mailan liikenopeus on maksimissaan? Kiihtyvyyssanturi sijoitetaan mailanpäähän, jolloin saadaan vakioitua mittauskohta. Ongelmana tällä tavoin toteutettuna on se, että useimmat kiihtyvyyssanturit mittaavat g-voiman muutokset x- ja y-suunnassa. Näin jää huomioimatta z-suunnan voima, joka kuitenkin kiertoliikkeessä esittää suurta roolia.

Konenäkökameralla toteutus on varmasti paras vaihtoehto, mutta samalla erittäin kallis vaihtoehto. Kameran avulla voidaan tietyillä kuvankäsittelyohjelmilla analysoida lähes kaikki mitattavat kohteet, kuten nopeus, kiihtyvyys, mailan heilahduskulma ja pallon lähtönopeus.

Valokennojen valintaan päädyttiin sen toimintavarmuuden ja edullisuuden vuoksi. Lisäksi koulun tietokonetekniikan harjoitustöiden perusteella saatiin varmuus työn onnistumiseen. Tutkan valinta ei ollut järkevää, koska tavoitteena oli saada toimiva laite. Kiihtyvyyssanturia ei valittu, koska sen ohjelmointi osoitautui liian haastavaksi aikatauluun nähden. Konenäkökameran kallis hinta ei mahdollistanut sen käyttöä.

3 MIKROTIETOKONE JA SEN PÄÄOSAT

Tietokonetta, jonka keskusyksikkönä on mikroprosessori, kutsutaan mikrotietokoneeksi. Prosessorin ollessa jokin muu kuin mikroprosessori, niin konetta kutsutaan pelkäksi tietokoneeksi. Mikrotietokone on elektroninen laite, joka suorittaa sille annettua tehtävää eli ohjelmaa. Niitä voi olla yleiskäyttöisiä, kuten PC, tai johonkin tiettyyn erikoistarkoitukseen suunniteltu kone, jota kutsutaan sulautetuksi järjestelmäksi. Kuvassa 1 havainnollistetaan mikrotietokoneen rakenne. [1, s. 15.]



Kuva 1. Mikrotietokoneen rakennekaavio [1]

Nykyään mikrokontrollerit ovat yleisessä käytössä, joihin on samaan piiriin liitetty kaikki kuvassa 1 olevat mikrotietokoneen osat.

3.1 Prosessori

Mikroprosessori (μP) on tietokoneen keskusyksikkö, joka on integroitu yhdelle piisirulle. Mikroprosessorin rakenne on pysyvä, joten käyttäjä ei voi muuttaa sen sisäistä rakennetta eikä sen ominaisuuksia. Mikroprosessori sisältää miljoonia transistoreita. Kytkettäessä nämä kaikki transistorit sopivasti yhteen saadaan erilaisia porttipiirejä. Näistä koostetaan kiikkuja, jotka liitettynä yhteen muodostavat laskureita ja rekistereitä. [2, s. 12.]

Mikroprosessorista käytetään myös nimityksiä suoritin ja CPU (Central Processing Unit). CPU on mikro-ohjaimen sydän. Se hakee ohjelmamuistista käskyt ja suorittaa ne. CPU huolehtii tietokoneen ohjauksesta ja ajoituksesta, eli se toimii koneen "isäntänä", joka määrää, mitä milläkin hetkellä tapahtuu. [1, s. 16.]

3.2 Ohjelmamuisti, Program Memory

Ohjelmamuisti säilyttää ohjelman käskyjä. Se ei poista tietoja, vaikka sähkötkatkaistaan. Tätä muistia kutsutaan haihtumattomaksi muistiksi eli ROM-muistiksi (Read only Memory). ROM-muistia on useita eri malleja, Maski-ROM, PROM, EPROM, EEPROM ja Flash-ROM. [4.]

Maski-ROM:n ominaisuuksia:

- Ohjelmointi tapahtuu tuotteen valmistuksen yhteydessä.
- Ei voida ohjelmoida uudelleen.

PROM:n (Programmable ROM) ominaisuuksia:

- piiri voidaan ohjelmoida vain kerran erillisellä ohjelmointilaitteella.

EPROM:n (Erasable PROM) ominaisuuksia:

- On erasoituva PROM.
- Piiri voidaan ohjelmoida uudelleen useita kertoja erillisellä ohjelmointilaitteella.

EEPROM:n (Electrically EPROM) ominaisuuksia:

- On sähköisesti erasoituva piiri.
- Piiriä ei tarvitse irrottaa piirilevyiltä uudelleenohjelmointia varten.

Flash-ROM:n ominaisuuksia:

- Erasointi tapahtuu sähköpulssein avulla.
- Piiriä ei tarvitse irrottaa piirilevyltä uudelleenohjelmointia varten.

3.3 Datamuisti, Data Memory

Datamuistista käytetään nimitystä käyttömuisti. Sinne tallennetaan käsiteltävä ja käsitelty tieto, joten sitä on kyettävä sekä lukemaan että kirjoittamaan. Muistia kutsutaan RAM-muistiksi (Random Access Memory), josta tieto häviää, kun käyttöjännitteet katkaistaan (poikkeus NVRAM). Datamuistina käytetään SRAM-, DRAM- ja NVRAM-muistia. [4.]

SRAM:n (Static RAM) ominaisuuksia:

- Staattinen RAM.
- Muistielementtinä on ”kiikku”.
- Ei tarvitse ylimääräistä virkistyslogiikkaa, kuten DRAM.

DRAM:n (Dynamic RAM) ominaisuuksia:

- Dynaaminen RAM.
- Muistielementtinä on ”kondensaattori”.
 - varaus purkautuu
 - varattava aika ajoitin uudelleen

NVRAM:n (Nonvolatile RAM) ominaisuuksia:

- Haihtumaton RAM.
- Siinä on kaksi periaatteellista ratkaisua:
 - paristovarmennettu SRAM
 - samaan piiriin integroitu EEPROM ja SRAM

3.4 Kello-oskillaattori, Clock Oscillator

Kello-oskillaattori tahdittaa kaikki toiminnot mikrotietokoneessa. Jokainen toiminto koneessa tapahtuu yhden kellopulssin aikana. Konekielinen komento voi sen laajuudesta riippuen vaatia aikaa useita kellopulsseja. [1, s. 16.]

3.5 Nollaus, Reset

Resetin avulla voidaan pakottaa ohjelma alkamaan aina samasta osoitteesta tietokoneen käynnistäessä. Tällä tavalla saadaan aikaan niin sanottu hallittu käynnistys. [1, s. 16.]

3.6 I/O-liitännät

I/O-liitännät ovat IC-piirejä (Integrated Circuit) tai irtokomponenteista rakennettuja kytkentöjä. Näiden avulla prosessori liittyy ympäristöönsä. I/O-liitännät voivat joko rinnakkais- tai sarjamuotoisia sekä analogisia tai digitaalisia liittimiä. [1, s. 16.]

3.7 Väylät, Buses

Mikroprosessori, ohjelma- ja datamuistit liittyvät sekä I/O-piireihin että toisiinsa väylien avulla. Väylät mahdollistavat mikrotietokoneen pääosien kommunikoinnin keskenään. Väylät muodostuvat rinnakkaisista johtimista, jotka välittävät siirrettävän datan johtimien loogisten jännitetasojen perusteella. Mikrotietokone sisältää sekä osoite- että dataväyliä. [1, s. 17.]

3.7.1 Osoiteväylä, Address Bus

Sen avulla prosessori osoittaa laitteen osan tai piirin, jonka kanssa se haluaa kommunikoida. Osoiteväylän rinnakkaiset johtimet määrittävät osoitteen. Osoiteväylä on yksisuuntainen. Mitä enemmän on osoitelinjoja, sitä suurempi on osoitettavien muistipaikkojen määrä eli muistiavaruus. [2, s. 8.]

3.7.2 Dataväylä, Data Bus

Dataväylällä siirtyy se tieto, jonka prosessori lukee piiristä tai kirjoittaa siihen. Dataväylä on kaksisuuntainen. Käskyt kulkevat yhteen suuntaan ohjelma-muistista prosessoriin, mutta datan täytyy voida kulkea molempiin suuntiin. Dataväylästä määräytyy, kuinka moni bittinen tietokone on. Väylän ollessa 16 bittiä leveä, niin puhutaan 16-bittisestä tietokoneesta. Toisin sanoen tietokoneen sanaleveys on tällöin 16 bittiä. [2, s. 8.]

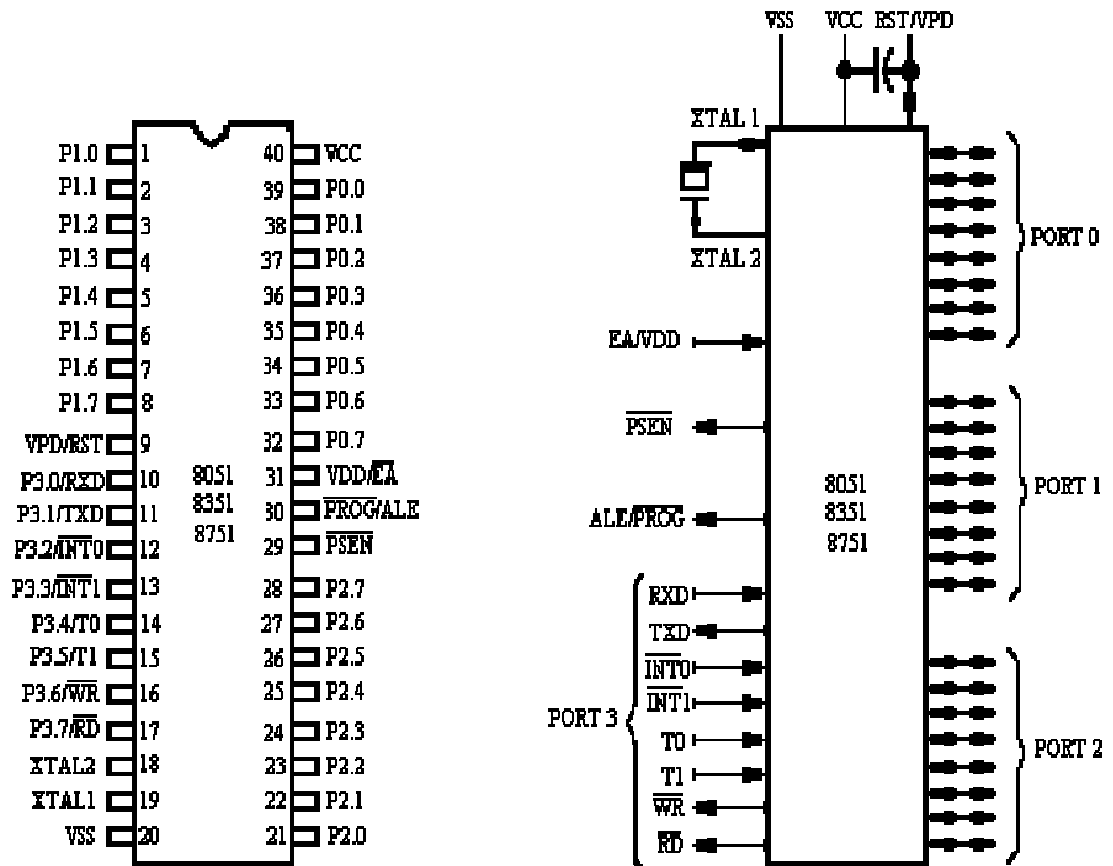
3.7.3 Ohjausväylä, Control Bus

Ohjausväylä ei ole varsinaisesti väylä samassa mielessä kuin edelliset. Voidaan sanoa, että se on ”nippu” signaaleja, joilla kullakin on oma merkityksensä. Prosessori hoitaa ohjausväylän signaalien avulla ajoituksen itsensä ja ympäryspiirien välillä. [2, s. 9.]

8051:n pääominaisuudet:

- 8-bittinen keskusyksikkö
- 64 ktavun muistiavaruus ohjelmamuistille
- 64 ktavun muistiavaruus datamuistille
- 4 ktavua sisäistä ohjelmamuistia (ROM)
- 128 tavua sisäistä datamuistia
- 32 kaksisuuntaista ja yksittäin osoitettavaa I/O-linjaa, jotka on ryhmitelty neljäksi 8-bitin portiksi P0, P1, P2, P3
- kaksi 16-bittistä ajastinta/laskuria
- täysi kaksisuuntainen sarjaportti (full duplex UART)
- vektoroitu keskeytysrakenne, jossa kaksi prioriteettitasoa
- sisäinen kello-oskillaattori
- laaja käskykanta: kuusi osoitusmuotoa ja yksittäisten bittien osoitusmahdollisuus [2.]

Kuvassa 3 havainnollistetaan MSC-51-perheen prosessorin nasta- ja logiikka-kaaviot.



Kuva 3. 8051:n nasta- ja logiikkakaavio [5].

Nastoja on yhteensä 40 ja piiri toimii 5 voltin jännitteellä. Seuraavassa kuvataan nastojen toimintaa.

VCC:

Piirin virtalähde

VSS:

Piirin maapotentiaali

PORTIT 0-3:

Tyypillisessä tilanteessa portit P0 ja P2 toimivat osoite- ja datalinjoina, jolloin portit P1 ja P3 jäävät käyttäjän käyttöön [2].

8051-mikrokontrollerissa portteja voidaan käyttää tavallisen I/O-liikenteen lisäksi erilaisiin erikoistoimintoihin [5].

Portteja 0 (P0.0-P0.7) ja 2 (P2.0-P2.7) käytetään osoittaessa ulkoista muistia. Portissa 0 kulkee osoiteväylän 8 alinta bittiä sekä 8 bittinen data. Portti 2 antaa osoiteväylän 8 ylintä bittiä. [5.]

Portti 3 (P3.0-P3.7) sisältää erilaisia erikoislippuja, joilla ohjataan ulkoisia keskeytyksiä, ajastimia ja laskureita [5]. Taulukossa 1 havainnollistetaan portti 3:n vaihtoehtoiset toiminnot.

Taulukko 1. Portti P3:n toiminnot [2]

Portin P3 linja	Vaihtoehtoinen toiminto
P3.0	RxD sarjaportin tulo
P3.1	TxD sarjaportin lähtö
P3.2	INT0 ulkoinen keskeytys 0
P3.3	INT1 ulkoinen keskeytys 1
P3.4	T0 timer 0:n ulkoinen tulo
P3.5	T1 timer 1:n ulkoinen tulo
P3.6	WR ulkoisen datamuistin kirjoituspulssi
P3.7	RD ulkoisen datamuistin lukupulssi

Lisäksi portteja 0, 1 (P1.0-P1.7) ja 2 voidaan käyttää ohjelmointi- ja tarkastusportteina [5].

RST:

Tällä nastalla tehdään järjestelmän nollaus ohjaamalla sen tila nollasta ykköseen (n. 3 voltin muutos). Tällöin keskusyksikkö asetetaan alkutilaan. [5.]

ALE:

Normaalissa käytössä ALE:lla (address latch enable) lukitaan osoite ulkopuoliseen muistiin. Samaan nastaan tuodaan ohjelmoitaessa ohjelmointipulssi. [5.]

PSEN:

PSEN (program store enable) kytkee ulkoisen ohjelmamuistin väylälle. Tällöin ohjelmamuistialue alkaa sisäisellä ja jatkuu ulkoisella ROM -muistilla. [5.]

EA:

EA (effective address) suorittaa käskyjä sisäisestä ROM- tai EPROM- muistista, jos ohjelmalaskurin (program counter)arvo on alle 4096. Maapotentiaaliin kytkettynä 8051 hakee käskyt ulkoisesta PROM- muistista. [5.]

XTAL1 ja XTAL2:

Näitä nastoja käytetään lähinnä ohjaimen sisäisessä ajastuksessa. XTAL tulee englanninkielisestä sanasta "crystal" eli kide. [5.]

4.1 Prosessorin ajastin/laskuri-toiminto

MCS-51-prosessoreissa on kaksi itsenäisesti toimivaa 16-bittistä ajastin/laskuria, joita kutsutaan timer/counter 0:ksi ja timer/counter 1:ksi. Niissä laskenta tapahtuu vain ylöspäin. Laskurin pyörähtäessä ympäri tulee keskeytyspyyntö. Laskurien ytimenä toimivat rekisterit TH0 ja TL0 sekä TH1 ja TL1. Kyseisten rekisterin sisältö päivittyy automaattisesti laskennan edetessä. [2.]

Käytettäessä ajastin-toimintoa, laskuri askeltaa yhdellä ylöspäin jokaisella konejaksolla. Konejakso koostuu 12 kellojaksosta, joten laskentataajuus on kellotaajuus/12. [2.]

Käytettäessä laskuri-toimintoa, laskuri askeltaa yhdellä ylöspäin, kun vastavassa ulkoisessa T0- tai T1-pinnissä esiintyy laskeva reuna. Laskentataajuus on kellotaajuus/24. [2.]

Ajastin- ja laskuri-toiminnoissa voidaan käyttää neljää erilaista toimintamuotoa, jotka valitaan TMOD-rekisterissä olevilla M0- ja M1-biteillä [2].

4.2 Prosessorin keskeytystoiminnot

MSC-51-proessoreissa on viisi eri keskeytyslähdettä, joista kukin tekee keskeytyspyynnön asettamalla sitä vastaavan keskeytyspyyntölipun tilaan "1". Taulukossa 2 esitetään eri keskeytyslähdeet ja niitä vastaavat keskeytysliput. [2.]

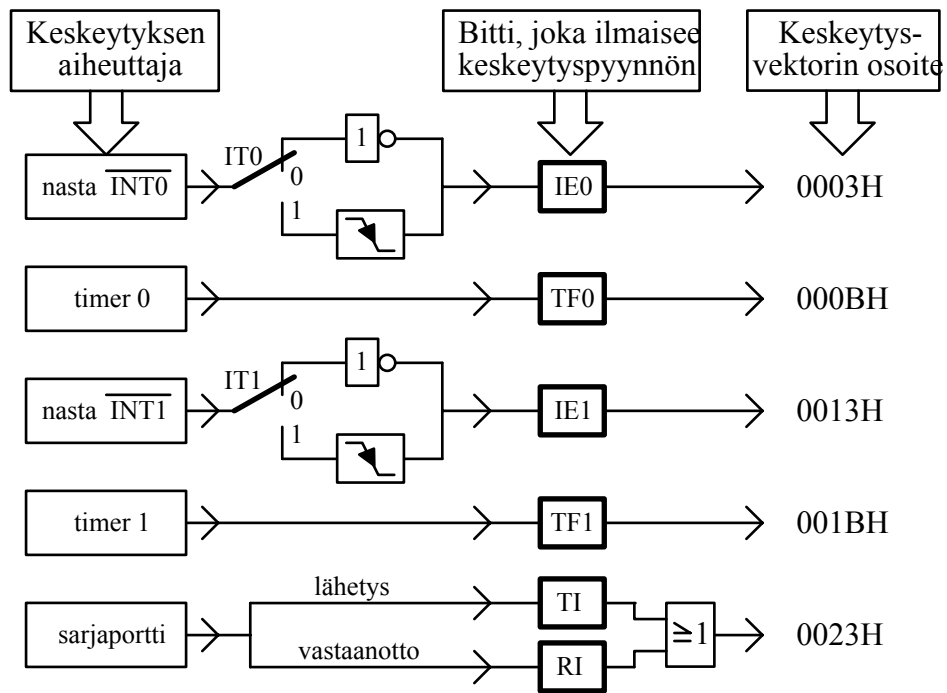
Taulukko 2. Keskeytyslähdeet ja niiden liput

Keskeytyslähde	Lippu
ulkoinen keskeytys 0	IE0
ulkoinen keskeytys 1	IE1
timer/counter 0	TF0
timer/counter 1	TF1
sarjaportti	lähetys TI ja vastaanotto RI

Kaikki viisi keskeytyslähdettä voidaan erikseen estää tai sallia keskeytyksen sallinta -rekisterissä olevilla biteillä. Kaikki keskeytykset voidaan myös estää yhdellä bitillä samassa keskeytyksen sallinta -rekisterissä. [2.]

Jokainen keskeytyspyyntölippu voidaan asettaa tai nollata myös ohjelmallisesti. Näin voidaan ohjelmallisesti aiheuttaa keskeytys tai estää palvelua odottava keskeytyspyyntö. [2.]

Kuvassa 4 havainnollistetaan prosessorin keskeytysjärjestelmä.



Kuva 4. 8051:n keskeytysjärjestelmä [2].

Ulkoiset keskeytykset $\overline{INT0}$ ja $\overline{INT1}$ ovat joko reuna- tai tasoherkkiä. Tähän vaikuttaa se, että millainen arvo biteillä IT0 ja IT1 on rekisterissä TCON. [2.]

5 C-KIELI OHJELMOINTIKIELENÄ

C-kieli on alun perin ammattilaisten käyttöön tarkoitettu. Dennis Ritchie kehitti sen B-kielen pohjalta vuonna 1972 UNIX-käyttöjärjestelmän lähdekoodin kirjoittamista varten. C-kieli ei neuvo käyttäjiänsä eikä korjaa kirjoitusvirheitä vaan se olettaa, että ohjelmoija tietää tarkalleen mitä on tekemässä ja osaa sen myös pilkun tarkkuudella kirjoittaa. Sillä on mahdollista tehdä hyvin tiivistä ja vaikeaselkoista koodia. Tämä ei ole kuitenkaan tarkoituksenmukaista, vaan hyvään C-ohjelmointiin kuuluvat kommenttirivit, välit ja sisennykset, jotka helpottavat koodin lukua. [6.]

C-kieli on lohkorakenteinen ohjelmointikieli. Se koostuu selkeistä käskyistä ja valinta- ja toistorakenteista. C-kielinen ohjelma sisältää lyhyen pääohjelman ja useita eri toimintoja suorittavia funktioita eli aliohjelmaa. Tämän johdosta C-kielinen ohjelma on helppo esittää vuokaaviona, joka taas helpottaa koodin kirjoittamista. [6.]

C-kieli on tarkasti standardoitu, jonka virallinen standardi on ANSI X3.159-1989. Kääntäjien valmistajat ovat myös noudattaneet standardia, niinpä C-kielessä ei ole eri "murteita" vaan eri valmistajien C-kääntäjät käsittelevät täsmälleen samanlaista lähdekoodia. [6.]

C-kieli on käyttöympäristöstä riippumaton. Sama lähdekoodi voidaan kääntää eri käyttöjärjestelmissä toimivaksi ohjelmaksi. Tämä johtuu siitä, että C-kieli ei sisällä käyttöympäristöstä riippuvia syöttö- tulostus- tai muistinhallintatoimintoja. Ne on toteutettu erillisillä funktioilla, jotka ovat samannimisiä, mutta eri lailla toteutettuja eri käyttöjärjestelmille. [6.]

6 MITTAUSJÄRJESTELMÄ

Pesäpallon mailanvalintaan suunniteltu mittausjärjestelmä kehitettiin mikrokontrolleripohjaiseksi. Mikroprosessoriksi valittiin Atmelin AT89S8252-24PC, jossa on Flash-ROM-tyyppinen muisti ohjelmakoodia varten. Laitteeseen liitettiin LCD-näyttö, näppäimistö ja kaksi valokennoa.

6.1 Valokennot ja niiden valinta

Valokennoista oli markkinoilla useita eri malleja. Niiden ominaisuudet vaihtelivat paljon eri valmistajien ja eri mallien välillä. Kennot tuli valita tarkasti käyttötarkoituksen mukaisesti. Mittausjärjestelmän olisi voinut toteuttaa kolmella eri valokennoperiaatteella. *1. kaksi kennoa vastakkain, 2. kenno ja heijastin, 3. kohdeheijastava valokenno.* Lähes kaikilla valmistajilla oli tarjolla valokennoja kyseisillä toimintaperiaatteilla.

Valittiin kenno ja heijastin -periaate. Kahden kennon periaatteella kustannukset olisivat nousseet liian korkeiksi, koska kennoja olisi pitänyt olla neljä kappaletta. Kohdeheijastavien valokennojen ongelmaksi olisi noussut toimintavarmuus, koska IR -säteen takaisin heijastavan kohteen pinta ei saanut poiketa 15°:tta enempää lähettimestä. Lisäksi niiden toimintaetäisyys oli maksimissaan 0,6 m, joka olisi vaikeuttanut mittausjärjestelmän fyysistä rakentamista.

Valokennoiksi valittiin italialaisvalmisteiset DECOUT-kennot. Näiden kennojen ominaisuudet vastasivat parhaiten mittausjärjestelmän oleellisiin vaatimuksiin. Lisäksi hinta oli edullinen, joka on tärkeää protolaitetta suunniteltaessa.

6.2 Nopeuden ja suhteellisen liike-energian laskeminen

Nopeuden mittaaminen toteutettiin kahden valokennon avulla. Kennojen etäisyyden ollessa vakio saatiin mailan nopeus [m/s] selville mittaamalla kulunut aika kennojen välillä kaavan 1 mukaisesti,

$$v = \frac{l}{\Delta t} \quad (1)$$

jossa l [m] on kuljettu matka ja Δt [s] käytetty aika.

Pesäpallomailan tuottama suhteellinen liike-energia [J] saatiin kaavasta 1 las-
ketun nopeuden ja kaavan 2 avulla,

$$E_k = \frac{1}{2}mv^2 \quad (2)$$

jossa m [kg] on mailan massa ja v [m/s] on nopeus.

6.3 Järjestelmässä käytetyt laitteet ja komponentit

Insinööriyössä käytettiin seuraavia laitteita ja komponentteja:

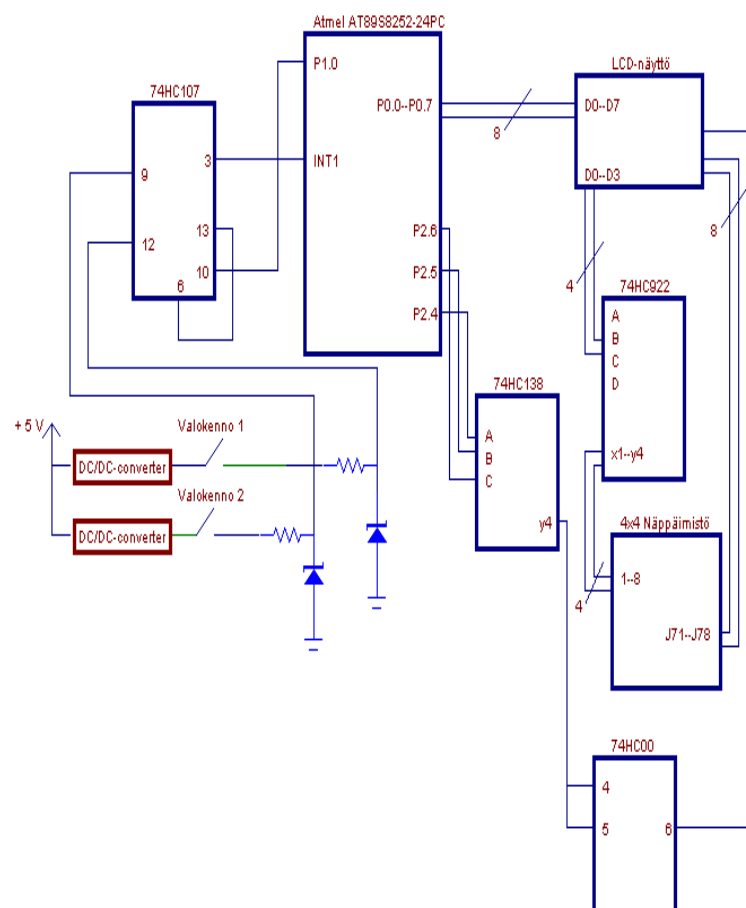
- piirilevy KAJTOL3 11.6.99
- DC/DC-converter NMV0515SA, 2 kpl
- mikropiiri 74HC107
- zenerdiodi 5,6 V
- vastus 1 k Ω , 2 kpl
- vastus 6,8 k Ω , 2 kpl
- tippatantaali 10 μ F, 2 kpl
- mikroprosessori Atmel AT89S8252-24PC
- valokenno DECOUT, 2 kpl
- 16-nappinen matriisinäppäimistö Claviers ECO
- yksirivinen Hyundai LCD -näyttö
- alumiininen kotelo (22x18x6,5) mm

6.4 Työn suoritus ja sen eri vaiheet

Työn suorittamisen ensimmäinen vaihe oli laitteen suunnittelu, koska kyseessä oleva laite oli prototyyppi. Määritettiin mittausjärjestelmään tarvittavat komponentit ja niihin saatavuus.

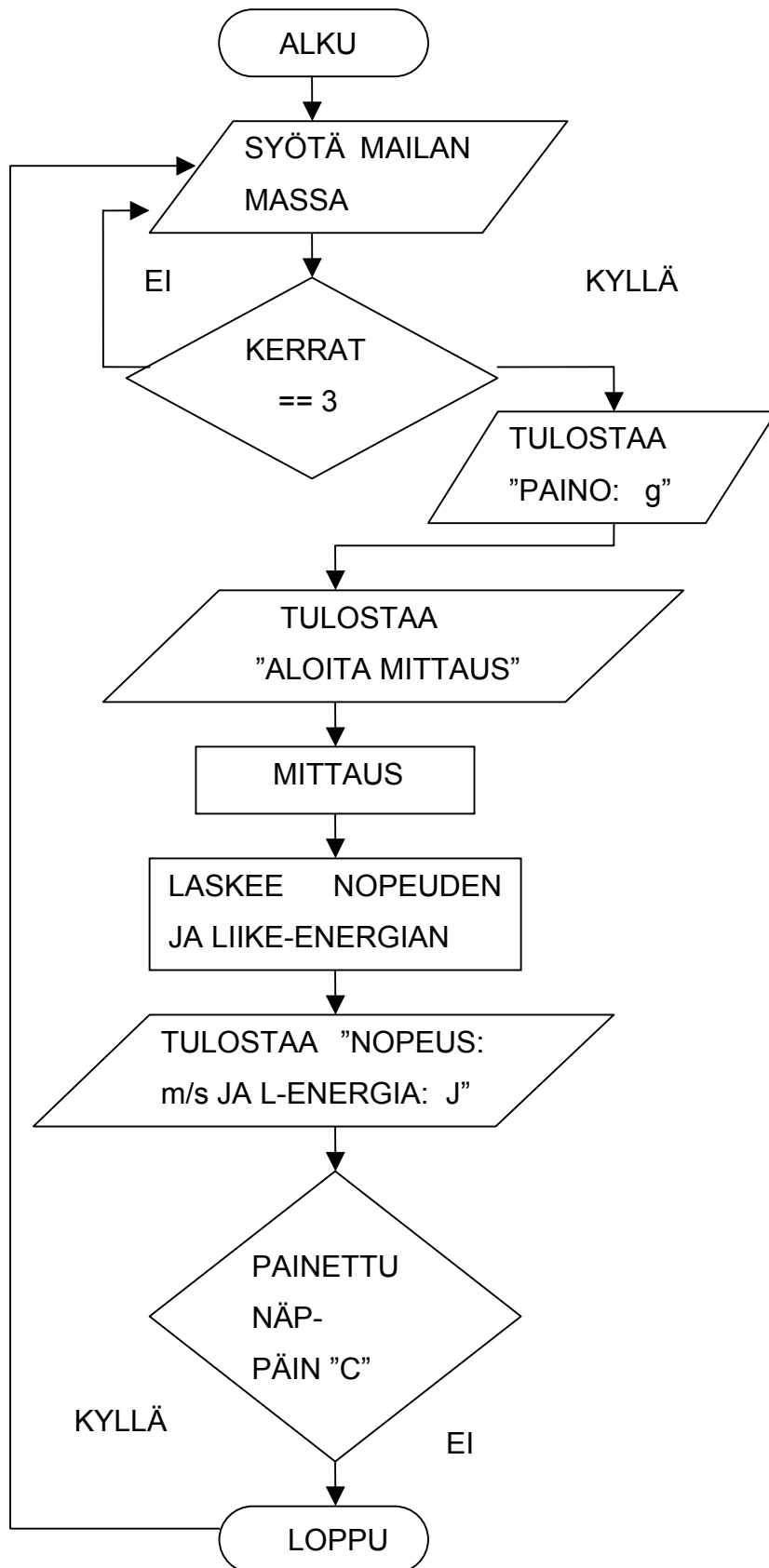
Seuraavaksi rakennettiin piirilevy KAJTOL3 11.6.99. Kyseessä oli kasaamaton piirilevy, johon liitettiin toiminnan kannalta kaikki tarvittavat komponentit.

Tämän jälkeen piirilevylle kytkettiin kuvan 5 mukaisesti mittausjärjestelmän vaativat komponentit, kuten kiikkupiiri 74HC107, DC/DC-converter, valokennot, LCD-näyttö ja näppäimistö. Piirien väliset kytkennät tehtiin kiertoliittymällä.



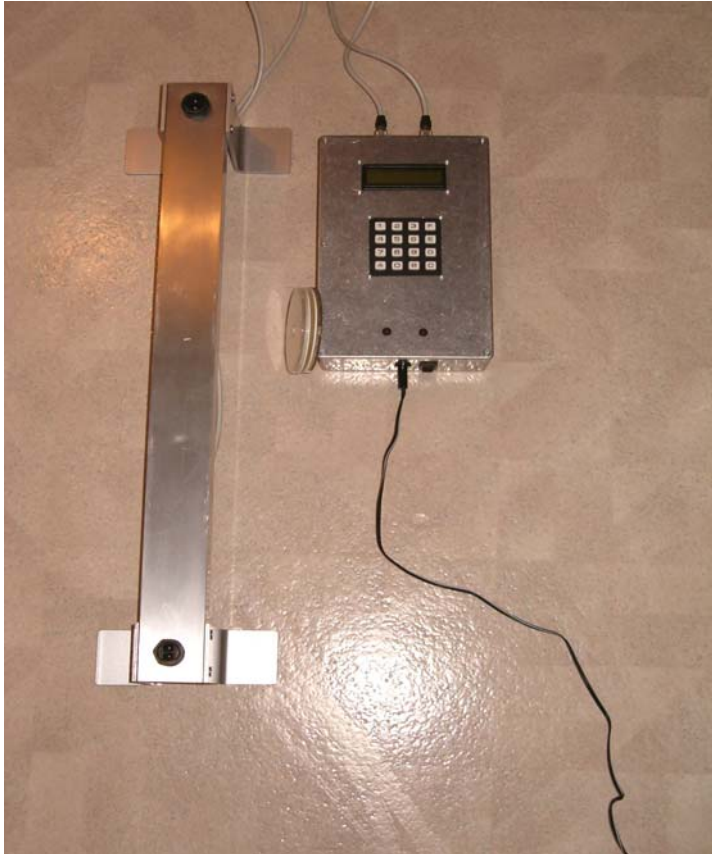
Kuva 5. Mittausjärjestelmän kytkentäkaavio

Ohjelmakoodin vuokaavion (kuva 6) laatiminen oli seuraava projektin vaihe. Vuokaavion rakenteen perusteella aloitettiin mikrokontrollerin ohjelmoinnin suunnittelu ja toteuttaminen. Ohjelmakoodi toteutettiin C-kielellä (liite A).



Kuva 6. Ohjelmakoodin vuokaavio

Projektin viimeisessä vaiheessa mikro-ohjain koteloitiin alumiiniseen koteloon, jonka kanteen leikattiin näppäimistölle ja LCD-näytölle paikat kuvan 7 mukaisesti. Lisäksi valokennojen ja virtajohdon liittimet kiinnitettiin koteloon.



Kuva 7. Mittausjärjestelmä

6.5 Mittausjärjestelmän testaus

Laitteen toimivuutta testattiin koko sen kehitystyön ajan Kajaanin ammattikorkeakoulun tietotekniikan laboratoriossa.

Aluksi valokennojen toimintaa testattiin digitaalisella oskilloskoopilla. Kennot asetettiin toimintavalmiuteen ja niiden reagointia kohteeseen testattiin käden heilautuksella infrapunasäteen läpi. Kokeiltiin myös kädenheilautusta eri nopeuksilla, jotta saatiin selville kennojen reagointi myös erittäin nopeaan liikkeeseen. Havaittiin, että valokennot reagoivan hyvin kohteeseen. Ohjelmakoodia käännettiin IAR Embedded Workbench demo for 8051-käännös-

ohjelmalla. Tämän jälkeen ohjelmakoodi siirrettiin emulaattoriohjelma Emul51:een, jolla voitiin emuloida PC-ympäristössä mikro-ohjaimen toimintaa ilman varsinaista mikroprosessoria. Tämän avulla voitiin havainnoida ohjelmakoodin puutteet ja korjata ne sekä kehittää edelleen mittausjärjestelmää muuttamalla koodia toimivammaksi.

7 MITTAUSTULOKSET

Mittausjärjestelmän testaus pelaajien todellisissa harjoitteluolosuhteissa toteutettiin Kajaani Hallissa 3.4.2004. Koehenkilöinä toimi eräs pesäpallon pelaaja.

Taulukossa 3 esitetään mittauks tulokset kyseisestä testaustapahtumasta.

Taulukko 3. Pelaaja x:n mittauks tulokset

Pelaaja x		
Mailan massa [g]	Nopeus [km/h]	Liike-energia [J]
630	124	373
650	118	249
720	111	345

Pelaajan saavuttamat suurimmat nopeuden ja suhteellisen liike-energian arvot:

Pelaaja x:	mailan massa	630 g
	nopeus	124 km/h
	liike-energia	373 J

8 TYÖN TAVOITTEIDEN TOTEUTUMINEN

Insinööriyössä saavutettiin kaikki sille ennakolta asetetut tavoitteet. Mittausjärjestelmä saatiin toimivaksi laitteeksi, joka tulosti näytölle eripainoisten pesäpallomailan liikenopeuden ja suhteellisen liike-energian. Pelaaja voi halutessaan testata eripainoisten mailojen sopivuutta itselleen.

Mittaustulosten perusteella voidaan todeta, että laitteella on käyttöä tulevaisuudessa. Laitteen toimintavarmuutta kannattaa kuitenkin kehittää vaihtamalla paremmat valokennot, jotka reagoivat vielä nopeampaan liikkeeseen. Testaus-tilanteessa pesäpallomailaan liitettiin kartongista pieni levike, jonka avulla saatiin mailan pinta-alaa isommaksi. Tällä tavoin varmistettiin, että laite mittaa jokaisen suorituksen. Kartonki kuitenkin muuttaa mailan ilmanvastusta, joka vääristää mailan todellista liikenopeutta. Laadukkaampien valokennojen avulla tämä ongelma poistuu.

Mittausjärjestelmän kehittämisideoita tuli paljon lisää työn suorittamisen aikana. Paras kehitysidea olisi ottaa laitteistoon käyttöön kahden valokennon sijasta esimerkiksi neljä valokennoa. Tällä tavoin voidaan analysoida paremmin mailan liikenopeutta ennen ja jälkeen syöttölautasta. Samalla saadaan analysoitua lyöntisuorituksen teknistä osapuolta. Onko esimerkiksi mailan liikenopeus suurempi ennen syöttölautasta, kuin sen kohdalla? Tällöin voidaan todeta lyöntisuorituksen olevan puutteellinen teknisesti.

Hyvä kehitysversio olisi liittää mittausjärjestelmään videokamera tai konenäkökamera, jolla voidaan analysoida visuaalisesti lyöntisuoritusta. Näin itse testattava voi myös analysoida omaa suoritusta. Hyvällä konenäkökameralla voidaan myös tarkastella mailan liikerataa tarkemmin, kuten mailan kulmaa horisontaalitasossa ja vertikaalisuunnassa. Lisäksi voidaan ottaa pesäpallo mittaus-suoritukseen mukaan. Tällä tavoin kyetään analysoimaan pallon vaikutusta suoritukseen. Tekeekö pelaaja täysin samanlaisen liikesuorituksen pallon kanssa, kuin ilman sitä.

9 YHTEENVETO

Insinööriyön tavoitteena oli suunnitella ja toteuttaa mittausjärjestelmä pesäpallomailan valintaan.

Työ aloitettiin kartoittamalla järkevät mittauskohteet pesäpallossa. Valittiin kyseinen aihe työlle, koska pelaajien mailanvalinta ei perustu mihinkään järkevään mittaukseen. Pelaaja ottaa sen painoisen mailan, joka on ollut ennenkin käytössä. Tällaisella valintatavalla ei varmasti ole kaikilla pelaajilla oikean painoinen maila ottaen huomioon hänen fyysiset ominaisuutensa.

Seuraavaksi työssä rakennettiin piirilevy, johon liitettiin juottamalla ja kiertoliittymällä kaikki mittausjärjestelmän vaatimat komponentit. Tässä työvaiheessa ei ollut suuria vaikeuksia, jos malttoi juottaa komponentit huolellisesti kiinni piirilevyyn.

Tämän jälkeen testattiin koulun valokennojen toimintaa. Sen avulla voitiin tilata vastaavilla ominaisuuksilla olevat valokennot mittausjärjestelmälle. Valokennojen testaamista jatkettiin liittämällä siihen digitaalinen oskilloskooppi, jonka avulla voitiin tarkastella kennojen reagointia nopeasti liikkuvaan kohteeseen. Havaittiin, että kennot reagoivat hyvin nopeaankin kädenheilautukseen läpi infrapunasäteen. Tämä oli projektin kulminaatiopiste. Jos kennot eivät olisi ehtineet reagoida nopeaan liikkeeseen, niin se olisi vesittänyt koko työn. Tässä vaiheessa usko projektin kunnialliseen valmistumiseen kasvoi huomattavasti.

Seuraavaksi alkoi työn mittavin vaihe eli ohjelmakoodin suunnittelu ja laadinta. Koodin valmistaminen aloitettiin perehtymällä C-kielen oppimateriaaleihin, joita olivat sekä luentomuistiinpanot että kirjastosta lainatut ohjelmointikirjat. Ohjelmoinnin vaikeimmaksi osaksi osoittautui mittauksetulosten tarkka laskeminen ja niiden tulostaminen ymmärrettävässä muodossa. Ongelma ratkaistiin useiden kokeilu- ja testausyritysten ansiosta.

LÄHDELUETTELO

- 1 Vahtera, P. Mikro-ohjaimen ohjelmointi C-kielillä. Ensimmäinen painos. Porvoo: WS Bookwell Oy, 2003. 339 s. ISBN 951-0-27160-8
- 2 Rantala, P. Tietokonetekniikka osa 2. Mikrotietokonetekniikka. Oulu: Kymdata, 1993. ISBN 951-559-103-1
- 3 Harju, T. C-kieli. Tietotekniikan monistesarja. Kymdata, 1990. ISBN 951-559-019-1
- 4 Airaksinen, R. Tietokonetekniikan luentomuistiinpanot. 2002.
- 5 Saalasti, S. Sulautetut järjestelmät, mikro-ohjaimet ja niiden ohjelmointi. Osa pro gradu-tutkielmaa. 1997. [www-dokumentti] <http://www.mit.jyu.fi/opiskelu/seminaarit/ohjelmistotekniikka/sulautetut/>
- 6 Sihvo, T. OHJELMOINNIN PERUSTEET / C-KIELI. 2002.

```
/*Riku Lokka,  
TTIOS, Päättötyö  
7.4.2004*/
```

```
#include <io51.h>  
#include <string.h>  
#include <stdio.h>  
#define NAPPAIMET (*(char*)0x01B000)  
#define LCD_CONTROL (*(char*)0x01C000)  
#define LCD_DATA (*(char*)0x01C002)  
#define RESET P1.0
```

```
void delay(int);                               /* Muuttujat ja aliohjelmat */  
void LCD_alustus();  
void tulostus(char[]);  
void tulos();  
int luku1=0, luku2=0, luku3=0;  
int nappis=0;  
unsigned int luku=0;  
float voima=0, voima1=0;  
unsigned int luku_ds=0;  
unsigned int voima_ds=0;  
int voima2=0;  
int summa=0;  
int i=0;  
int keskeytys=0;  
int kerrat=0;  
int kiikku=0;  
int ylivuotojen_maara=0;
```



```

int pyorahdys=0;

void main(void)                                     /*Päälause */
{
char merkit[20];

TR1=0;                                             /* Laskurien valinnat, keskeytykset */
EX0=1;
EX1=0;
IT1=1;
PCON=0x00;
TMOD=0x99;
TCON=0x05;
ET1=1;
TH1=0;
TL1=0;
EA=1;

LCD_alustus();
delay(100);
/*****/
sprintf(merkit,"ANNA PAINO:  g");
tulostus(merkit);

delay(500);
LCD_CONTROL=0xC3;

for(;;)
{
while(keskeytyks==0);

nappis=NAPPAIMET;
nappis=nappis & 0x0F;
}
}

```

```

switch(nappis)                                     /* Näppäimistön toiminnot */
{

case 0:
    if(kerrat==1)
        luku1=1;
    if (kerrat==2)
        luku2=1;
    if (kerrat==3)
        luku3=1;
        LCD_DATA='1';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;

case 1:    if (kerrat==1)
        luku1=2;
    if (kerrat==2)
        luku2=2;
    if (kerrat==3)
        luku3=2;
        LCD_DATA='2';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;

case 2:    if (kerrat==1)
        luku1=3;
    if (kerrat==2)
        luku2=3;
    if (kerrat==3)
        luku3=3;

```

```
LCD_DATA='3';  
for (i=0; i<1000; i++);  
keskeytys=0;  
break;
```

case 3:

```
LCD_DATA='F';  
for (i=0; i<1000; i++);  
keskeytys=0;  
break;
```

case 4: if (kerrat==1)

```
luku1=4;
```

```
if (kerrat==2)
```

```
luku2=4;
```

```
if (kerrat==3)
```

```
luku3=4;
```

```
LCD_DATA='4';
```

```
for (i=0; i<1000; i++);
```

```
keskeytys=0;
```

```
break;
```

case 5: if (kerrat==1)

```
luku1=5;
```

```
if (kerrat==2)
```

```
luku2=5;
```

```
if (kerrat==3)
```

```
luku3=5;
```

```
LCD_DATA='5';
```

```
for (i=0; i<1000; i++);
```

```
keskeytys=0;
```

```
break;
```

```
case 6:  if (kerrat==1)
        luku1=6;
        if (kerrat==2)
            luku2=6;
        if (kerrat==3)
            luku3=6;
        LCD_DATA='6';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;
```

```
case 7:
        LCD_DATA='E';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;
```

```
case 8:  if (kerrat==1)
        luku1=7;
        if (kerrat==2)
            luku2=7;
        if (kerrat==3)
            luku3=7;
        LCD_DATA='7';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;
```

```
case 9:  if (kerrat==1)
        luku1=8;
        if (kerrat==2)
            luku2=8;
```

```
if (kerrat==3)
    luku3=8;
    LCD_DATA='8';
    for (i=0; i<1000; i++);
    keskeytys=0;
    break;
```

```
case 10: if (kerrat==1)
    luku1=9;
    if (kerrat==2)
        luku2=9;
    if (kerrat==3)
        luku3=9;
        LCD_DATA='9';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;
```

```
case 11:
    LCD_DATA='D';
    for (i=0; i<1000; i++);
    keskeytys=0;
    break;
```

```
case 12:
    LCD_DATA='A';
    for (i=0; i<1000; i++);
    keskeytys=0;
    break;
```

```
case 13: if (kerrat==1)
    luku1=0;
    if (kerrat==2)
```

```
    luku2=0;
    if (kerrat==3)
        luku3=0;
        LCD_DATA='0';
        for (i=0; i<1000; i++);
        keskeytys=0;
        break;
```

case 14:

```
    LCD_DATA='B';
    for (i=0; i<1000; i++);
    keskeytys=0;
    break;
```

case 15:

```
    LCD_DATA='C';
    for (i=0; i<1000; i++);
    kerrat=0;
    luku1=0;
    luku2=0;
    luku3=0;
    summa=0;
    TH1=0;
    TL1=0;
    luku=0;
    luku_ds=0;
    voima=0;
    voima_ds=0;
    LCD_CONTROL=0x01;
    delay(500);
    sprintf(merkit,"ANNA PAINO:  g");
    tulostus(merkit);
    delay(500);
```

```

        LCD_CONTROL=0xC3;
        delay(100);
        keskeytys=0;
        break;
    }

    if(kerrat==3)
    {
        summa=0;
        luku1=luku1*100;
        luku2=luku2*10;
        luku3=luku3;
        summa=luku1+luku2+luku3;

        sprintf(merkit,"PAINO: %d g",summa);
        LCD_CONTROL=0x01;
        delay(500);
        tulostus(merkit);
        LCD_CONTROL=0xC8;
        delay(2000);                                     /*Odottaa hiukan*/

        tulos();
        delay(500);
        kerrat=0;
        keskeytys=0;
    }
}

void LCD_alustus()
{
    int i;
    char

```

```

alustus[11]={0x30,0x30,0x30,0x38,0x38,0x01,0x07,0x01,0x0F,0x06,0x01};
for(i=0;i<11;i++)
{
LCD_CONTROL=alustus[i];
delay(10);
}
}

```

```

/*****/

```

```

void tulostus (char merkit[])
{
int i;

for (i=0;i<strlen(merkit);i++)
{
if(i==8)
{
LCD_CONTROL=0xC0;
delay(10);
}
if (merkit[i]=='ä')
{
LCD_DATA=0xE1;
delay(1);
}
else
{
LCD_DATA=merkit[i];
delay(1);
}
}
}

```



```

/*****/
interrupt void EX0_int()                /* Näppäimistön keskeytys */
{
    keskeytys=1;
    kerrat++;
}
/*****/

interrupt void T1_int()                /* Timer 1 Overflow */
{
    ylivuotojen_maara++;
}
/*****/

interrupt void EX1_int()                /* Kiihun keskeytys */
{

    kiikku=1;

}
/*****/

void delay(int pituus)                /* Viive aliohjelma */
{
    int i;
    for(i=0;i<pituus;i++)
    {
        TL0=0x18;
        TH0=0xFC;                    /* Laskuriin 18FCH */
        TR0=1;                        /*Laskurin käynnistys*/
        while(TF0==0);                /*kunnes tulee ylivuoto*/
        TR0=0;                        /*pysäytetään laskuri*/
        TF0=0;                        /*nollataan ylivuoto*/
    }
}

```

```

/*****/

void tulos()                                     /* Mittaus-aliohjelma */
{
  unsigned int LSB=0,MSB=0;
  char merkit[20];
  char merkit2[20];
  sprintf(merkit," ALOITA MITTAUS ");
  LCD_CONTROL=0x01;
  delay(200);
  tulostus(merkit);
  delay(1000);
  LCD_CONTROL=0x01;
  P1.0=0;
  for(i=0;i<1500;i++);
  P1.0=1;
  EX1=1;
  TR1=1;
  TH1=0;
  TL1=0;

  while(kiikku==0);

  EA=0;
  TR1=0;
  LSB=TL1;
  MSB=TH1;
  TL1=0;
  TH1=0;

  luku=MSB;
  luku=luku<<8;                                /* bittien vyörytys */
}

```

```

luku=luku+LSB;

pyorahdys=ylivuotojen_maara*65.536;
ylivuotojen_maara=0;
luku=(luku/1000.0)+pyorahdys;          /* luku = ms */
luku=1000.0/luku;                      /* m/s */
voima1=summa/1000.0;                   /* Mailan massa kg */
voima=luku*luku;                       /* voima = 1/2mv2 */
voima2=((float)voima1/2.0)*(float)voima;
voima2=voima2/2.0;
voima2=voima2/2.0;
luku=luku/2.0;
luku=luku*3.6;
sprintf(merkit, " %d km/h",luku);       /* tulostus nopeus */
sprintf(merkit2, "L-energia: %d J",voima2); /* tulostus L-energia */

for(i=0;i<2;i++)
{
    LCD_CONTROL=0x01;
    delay(100);

    tulostus(merkit);
    delay(2000);
    LCD_CONTROL=0x01;
    delay(100);
    tulostus(merkit2);
    delay(2000);
}
kiikku=0;
TH1=0;
TL1=0;
EA=1;
}

```