

Jarmo Salo

SULAUTETTUIJEN JÄRJESTELMIEN HYÖDYNTÄMINEN
TOISEN ASTEEN AMMATILISESSA OPETUKSESSA

Tietotekniikan Pro gradu

Ohjelmistotekniikka

20.12.2017

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Kokkolan yliopistokeskus Chydenius

Tekijä: Jarmo Salo

Yhteystiedot: jarmo.salo@kpedu.fi

Työn nimi: Sulautettujen järjestelmien hyödyntäminen toisen asteen ammatillisessa opetuksessa

Title in English: How to utilize embedded systems within second grade vocational education

Työ: Pro gradu -tutkielma

Sivumäärä: 68 (+ liitteet)

Linja: Opettajalinja

Ohjaaja: Professori Ismo Hakala

Avainsanat: Sulautettu järjestelmä, mikrokontrolleri, mikroprosessori, ohjelmointi

Keywords: Embedded System, microcontroller, microprocessor, programming

Tiivistelmä: Tässä Pro gradu -tutkielmassa paneudutaan sulautettujen järjestelmien hyödyntämiseen toisen asteen opetuksessa Tieto- ja tietoliikennetekniikan opetussuunnitelmassa. Tutkielmassa tutkitaan LEGO Mindstorms -, Arduino UNO - ja MatrixTSL:n ECIO-järjestelmien soveltuvuutta eri tietotekniikka-alueiden ammatilliseen opetukseen.

Abstract: This Master's Thesis focuses on the utilization of embedded systems in secondary education in the curriculum of Information and Communication Technology. The thesis examines the suitability of the LEGO Mindstorms, Arduino UNO and MatrixTSL ECIO systems for vocational education in various IT fields.

Termiluettelo

TCP/IP	Transmission Control Protocol / Internet Protocol
PIC	Microcontroller Family, Microchip Technology Inc
dsPIC	Digital Signal Controller Family, Microchip Technology Inc
AVR	Microcontroller Family, Atmel Corporation
ARM	Microcontroller Family, Advanced RISC Machines
Arduino	Open-source electronics platform
JTAG	Joint Test Action Group
FLASH	Flash memory
RAM	Random access memory
USB	Universal Serial Bus
RS232	Recommended Standard 232
Bluetooth	Wireless Technology
CAN	Controller Area Network
C	High-level programming language
C++	Object-oriented programming (OOP) language
FPGA	Field Programmable Gate Array
DSP	Digital Signal Processor
RFID	Radio Frequency Identification
ZigBee	Protocol stack for wireless networks
GPS	Global Positioning System
LIN	Local Interconnect Network
IrDA	Infrared Data Association

RF(ISM)	Radio Frequency (Industrial, Scientific and Medical)
RC5	Rivest Cipher 5, encryption algorithm
RS485	Recommended Standard 485

Sisältö

1	JOHDANTO	1
2	TIETO- JA TIETOLIIKENNETEKNIIKAN OPETUSSUUNNITELMA	3
2.1	ELEKTRONIIKKA- JA ICT-ASENTAJAN AMMATILLISET AINEET	3
2.2	ELEKTRONIIKAN JA ICT:N PERUSTEHTÄVÄT	5
2.3	TIETOKONE- JA TIETOLIIKENNEASENNUKSET	5
2.4	AMMATTIELEKTRONIIKKA	6
2.5	SULAUTETUT SOVELLUKSET JA PROJEKTITYÖT	7
2.6	SULAUTETTUIEN JÄRJESTELMIEN OPETUSALUSTOJEN ROOLI OPS:SSA	8
3	SULAUTETTUIEN JÄRJESTELMIEN TEKNIikka	9
3.1	SULAUTETTUIEN JÄRJESTELMIEN PERUSKOMONENTIT	9
3.2	SULAUTETUN JÄRJESTELMÄN TOIMINTAPERIAATE.....	12
3.3	LEGO MINDSTORMS -ROBOTTI	13
3.4	MATRIXTSL	18
3.5	ARDUINO	23
4	SULAUTETUT JÄRJESTELMÄT OPETUSKÄYTÖSSÄ	32
4.1	LEGO MINDSTORMS -ROBOTTI	32
4.1.1	Esimerkkejä opetuskäytöstä.....	33
4.2	MATRIXTSL-JÄRJESTELMÄT	36
4.2.1	Esimerkkejä opetuskäytöstä.....	37
4.3	ARDUINO-JÄRJESTELMÄT.....	40
4.3.1	Esimerkkejä opetuskäytöstä.....	41
4.4	JÄRJESTELMIEN VÄLISET EROT.....	44
4.5	SULAUTETTUIEN JÄRJESTELMIEN MAHDOLLISUUDET OPS:SSA	45
5	CASE ”KPEDU - SULAUTETUT JÄRJESTELMÄT OPETUKSESSA”	48
5.1	JÄRJESTELMIEN KÄYTTÖTILANNE KPEDU:SSA	48
5.2	TUTKIMUKSEN TAVOITE JA KUVAUS	49
5.3	TUTKIMUKSEN TOTEUTUS	50
5.4	TULOKSET KYSELYISTÄ	53
5.5	JOHTOPÄÄTÖKSET.....	59
6	YHTEENVETO	62
7	LÄHTEET	64
8	LIITTEET	69

1 Johdanto

Tässä Pro gradu -tutkielmassa selvitetään sulautettujen järjestelmien opetusalojen soveltuvuutta ja käyttöä tietotekniikan opetuksessa toisen asteen ammatillisessa koulutuksessa Tieto- ja tietoliikennetekniikan perustutkinnon mukaisessa opetussuunnitelmassa (OPS). Uusittu Tieto- ja tietoliikennetekniikan opetussuunnitelma otettiin ammattiopistoissa käyttöön 1.8.2015. Tieto- ja tietoliikennetekniikan perustutkinnosta valmistuu sekä elektroniikka- että ICT-asentajia. Tässä tutkimuksessa keskitytään valtakunnallisessa Tieto- ja tietoliikennetekniikan perustutkinnossa oleviin elektroniikka- ja ICT-asentajan opetussuunnitelman mukaisiin ammatillisiin tietotekniikan alueen tutkinnon osiin.

Tutkimuksessa tarkastellaan miten sulautettujen järjestelmien opetukseen kehitettyjä opetusaloja voidaan integroida toisen asteen ammatillisen koulutuksen tietotekniikan perusteiden opetukseen. Työssä selvitetään myös sitä, tuleeko tietotekniikan ja ohjelmoinnin opettaminen opetusaloilla opiskelijoille konkreettisemmaksi ja mielenkiintoisemmaksi. Yksi tämän tutkimuksen tavoitteista on saada selville se, onko sulautettujen järjestelmien opetusalojen käytöllä ohjelmoinnin opetuksessa opiskelijoiden opiskelumotivaatiota parantavaa vaikutusta. Lisäksi selvitetään, onko jokin tähän tutkimukseen valituista sulautettujen järjestelmien opetusaloista opiskelijoiden mielestä ominaisuuksiltaan ja käytettävyydeltään muita parempi.

Toisessa luvussa luodaan ensin katsaus siihen, miten valtakunnallinen Tieto- ja tietoliikennetekniikan opetussuunnitelma rakentuu ja mitä ammatillisia tutkinnon osia siihen kuuluu. Luvussa esitellään elektroniikka- ja ICT-asentajan tutkintoon kuuluvat ammatilliset tutkinnon osat. Lopuksi tehdään pieni katsaus sulautettujen järjestelmien opetusalojen käytömahdollisuuksiin OPS:n puitteissa.

Luvun 3 myötä selvitetään opetukseen tarkoitettujen sulautettujen järjestelmien opetusalojen taustaa, historiaa ja rakennetta. Sulautettujen järjestelmien rakennetta ja toimintaperiaatteita kuvataan yleisellä tasolla. Tähän tutkimukseen on valittu mukaan LEGO:n, Arduinon ja MatrixTSL:n tarjoamat alustat. Näillä kaikilla järjestelmillä on mahdollista opettaa ja demonstroida monia toisen asteen tietotekniikan opetuksessa esille tulevia asioita ja tekniikoita. Tähän mennessä KPEDUssa on ollut käytössä LEGO ja MatrixTSL. Arduino on

otettu käyttöön tämän tutkimuksen yhteydessä vaihtoehtoisena opetuslustana muiden rinnalle.

Luvussa 4 tarkastellaan, millaisia tietotekniikan asioita on mahdollista opettaa noilla kolmella sulautettujen järjestelmien opetuslustalla ja miten valitut opetuslustat soveltuvat ominaisuuksiensa perusteella opetettaviin asioihin. Niiden käytöstä opetuksessa annetaan esimerkkejä monista oppilaitoksista. Seuraavaksi esitellään noiden kolmen järjestelmän periaatteelliset erot. Lopuksi katsotaan OPS:n pakollisten ammatillisten tutkinnon osien kohdalla, miten esiteltyjä opetuslustoja voidaan hyödyntää opetuksessa. Opetuslustoja käytetään sekä elektroniikka- että ICT-asetajien opetuksessa lähinnä toisella ja kolmannella luokalla, vaikka laitteet soveltuvat myös osin ykkösluokkien opetukseen sekä muiden sähköalan luokkien yleiseen tietotekniikkaopetukseen.

Viidennessä luvussa tarkastellaan, miten KPEDUssa on ICT-asetajalinjalla hyödynnetty opetuslustoja tietotekniikan ja ohjelmoinnin opetuksessa. ICT-asetajaryhmälle tehdään vertailututkimus kahdella opetuslustalla. Tutkimuksen toteutus esitellään ja sen jälkeen kerrotaan tulokset tutkimuksesta. Vertailututkimukseen on valittu Arduinon UNO ja MatrixTSL:n ECIO sulautettujen järjestelmien opetuslustoina. Opiskelijat tekevät molemmilla opetuslustoilla harjoitustehtävät ja antavat palautetta kyselykaavakkeilla. Johtopäätöksistä nähdään kuinka hyvin valitut opetuslustat soveltuvat tietoteknisten asioiden opettamiseen ja onko niiden käytöllä omalta osaltaan opiskelijoiden opiskelumotivaatiota parantavaa vaikutusta. Lisäksi nähdään auttavatko opetuslustat oman opetuksen kehittämisessä ja kannattaako niiden käyttöä vielä laajentaa.

Luvussa 6 annetaan yhteenveto tutkimuksesta ja kerrotaan miten tutkimustulokset vastaavat tutkimukselle asetettuja tavoitteita. Katsotaan myös, kuinka hyvin harjoitustehtävien käytännön järjestelyt sujuivat ja esiintyikö niiden aikana esiin ongelmia. Lopuksi esitellään, miten näitä tutkimuksia voitaisiin jatkaa tulevaisuudessa.

2 Tieto- ja tietoliikennetekniikan opetussuunnitelma

Valtakunnallinen Tieto- ja tietoliikennetekniikan perustutkinnon opetussuunnitelma (OPS) [1] käsittää sekä elektroniikka-asetajan että ICT-asetajan (Information, Communication, Technology) perustutkinnot. Tutkinto koostuu pakollisista tutkinnon osista, valinnaisista tutkinnon osista, yhteisistä tutkinnon osista sekä vapaasti valittavista tutkinnon osista, ja niiden yhteenlaskettu kesto on 180 osp (osaamispistettä). Jokainen oppilaitos tekee oman ratkaisunsa siitä, että kumpaa perustutkintoa oppilaitoksessa opetetaan, vai opetetaanko molempia. Valtakunnallisesta opetussuunnitelmasta jokainen ammattiopisto muokkaa oman opetussuunnitelmansa. Kaikissa toisen asteen tutkinnoissa opiskelijat suorittavat ammatillisissa tutkinnon osissa näyttötehtävän, jolla arvioidaan osaamista. Näytöt sisältyvät opetettaviin tutkinnon osiin ja ne tehdään pääsääntöisesti työssäoppimispaikoilla. Mikäli siellä ei voida järjestää vaaditun laajuuden mukaista näyttötehtävää, niin silloin näyttöä voidaan täydentää oppilaitoksessa tai toisessa työssäoppimispaikassa.

2.1 Elektroniikka- ja ICT-asetajan ammatilliset aineet

Tieto- ja tietoliikennetekniikan perustutkinnon muodostuminen on esitetty liitteen 1 taulukossa 2. Tutkinto muodostuu yhteensä 180 osp:stä, josta varsinaiset ammatilliset tutkinnon osat ovat 135 osp. Sen lisäksi tutkintoon sisältyy yhteisiä tutkinnon osia 35 osp:een edestä ja vapaasti valittavia tutkinnon osia 10 osp.

Elektroniikka- ja ICT-asetajan ammatillisten tutkinnon osien rakenne on liitteen 1 taulukoissa 2.1 ja 2.2 valtakunnallisen OPS:n mukaisesti. Ammattiaineet jakaantuvat kahteen osaan: *Pakolliset tutkinnon osat* (Taulukko 2.1) ja *Valinnaiset tutkinnon osat* (Taulukko 2.2). Pakollisiin tutkinnon osiin kuuluu ensimmäisellä luokalla suoritettava *Elektroniikan ja ICT:n perustehtävät* 45 osp. Tämä osa on sama sekä Elektroniikka- että ICT-asetajilla. Toisena vuonna ICT-asetajilla ovat pakollisina opintoina *Tietokone- ja tietoliikenneasennukset* 30 osp ja Elektroniikka-asetajilla *Ammattielektroniikka* 30 osp. Näiden lisäksi on

ICT-asentajan valittava valinnaisista ammatillisista tutkinnon osista joko *Palvelinjärjestelmät ja projektityöt* 30 osp tai *Tietoliikennelaitesennukset ja kaapelointi* 30 osp. Elektroniikka-asentajan on valittava joko *Sulautetut sovellukset ja projektityöt* 30 osp tai *Elektroniikkatuotanto* 30 osp. Elektroniikka-asentajan ja ICT-asentajan erikoistuminen tapahtuu näillä kuudella tutkinnon osalla. Näiden lisäksi opiskelijat valitsevat erilaisia ammatillisia tutkinnon osia liitteen 1 mukaan (Taulukko 2.2, 2.2.5 – 2.2.11) kuitenkin niin, että vaadittava minimi osp-määrä täyttyy ja myös sen mukaan mitä tutkinnon osia koululla järjestetään.

Yhteiset tutkinnon osat jakaantuvat 4 osaan (Liite 1, Taulukko 2.3). Ensimmäisessä osassa on *Viestintä- ja vuorovaikutusosaaminen*, johon kuuluu pakollisia osia 8 osp ja valinnaisia osia 3 osp. Tästä osasta hyvänä esimerkkinä ovat *Äidinkieli* ja *Ruotsi*. Toinen osa, *Matemattis-luonnontieteellinen osaaminen*, sisältää pakollisia osia 6 osp ja valinnaisia osia 3 osp. Hyvinä esimerkkeinä tästä osasta ovat *Matematiikka* ja *Fysiikka*. Kolmannessa osassa on *Yhteiskunnassa ja työelämässä tarvittava osaaminen*, joka koostuu 5 osp:n pakollisista opinnoista ja 3 osp:n valinnaisista opinnoista. Esimerkkeinä tästä osasta voisi mainita *Yhteiskuntataidot* ja *Työelämätaidot*. Viimeisessä, neljännessä osassa on *Sosiaalinen ja kulttuurinen osaaminen*, johon kuuluu ainoastaan valinnaisia opintoja 7 osp. Tässä osassa opiskeltavia aineita ovat esimerkiksi *Kulttuurien tuntemus* ja *Psykologia*.

Vapaasti valittavat tutkinnon osat ovat sellaisia, että ne tukevat opiskelijan ammatillista kehittymistä ja lisäävät tutkintoon vaadittavia osaamistavoitteita (Liite 1, Taulukko 2.4). Vapaasti valittaviin tutkinnon osiin voidaan ottaa mukaan osia ammatillisista tutkinnon osista, ne täydentävät silloin ammatillista osaamista. Tämä on mahdollista niin Elektroniikka- kuin ICT-asentajan opinnoissakin. Tutkinnon osat voivat olla ammattitaitoa syventäviä tai laajentavia ammatillisia tutkinnon osia. Voidaan myös tehdä paikallisiin ammattitaitovaatimuksiin tai osaamistavoitteisiin perustuvia tutkinnon osia. Vapaasti valittaviin tutkinnon osiin voidaan myös lukea mukaan YTO-aineita (*Yhteiset tutkinnon osat*) sekä ammattilukiosta tulevat YTO-aineet. Jatko-opiskelua ajatellen voidaan mukaan ottaa jatko-

opintovalmiuksia tai ammatillista kehittymistä tukevia opintoja. Lisäksi mukaan voidaan ottaa tutkintoa yksilöllisesti laajentavat tutkinnon osat.

2.2 Elektroniiikan ja ICT:n perustehtävät

Elektroniiikan ja ICT:n perustehtävät -tutkinnon osa (45 osp) on ensimmäisenä vuonna toteutettava tutkinnon osa (Liite 1, Taulukko 2.1), joka kuuluu pakollisena osana sekä Elektroniiikka- että ICT-asentajan opetussuunnitelmaan. Tässä tutkinnon osassa on tarkoitus oppia elektroniiikan-, sähkötekniikan- ja sähköasennustekniikan perusteita. Harjoitustehtäviin kuuluu erilaisten sähkökytkentöjen tekeminen sekä mittaaminen. Lisäksi opittavina asioina ovat erilaiset sähkö- ja elektroniiikka-alan komponenttivalikoimat sekä niiden perustoiminnot. Opiskeltaviin asioihin kuuluvat myös käsi- ja sähkötyökalujen käyttö sekä erilaiset elektroniiikka-alueen kytkentä-, mittaus- ja laskuharjoitukset. Tutkinnon osassa opiskellaan myös sulautetun ohjelmankehitysympäristön käyttö yhdessä ohjelmointialustojen kanssa. Lisäksi sekä Internetin että IP-osoitejärjestelmän ja tietotekniikan perusteiden oppiminen on olennainen asiakokonaisuus.

2.3 Tietokone- ja tietoliikenneasennukset

Tietokone- ja tietoliikenneasennukset -tutkinnon osa (30 osp) on ICT-asentajille kuuluva pakollinen ammatillinen tutkinnon osa (Liite 1, Taulukko 2.1). Osa tästä tutkinnon osasta täydentyy työssäoppimisjakson (TOP) aikana. Tämä tutkinnon osa jakaantuu kolmeen osa-alueeseen:

1. Tietokonelaite- ja järjestelmätyöt
2. Sulautettujen järjestelmien toteutus
3. Tietoliikennetekniikan työt

Tietokonelaite- ja järjestelmätöiden osiossa on tarkoitus oppia tietokoneen rakenteen perusteet, lisälaitteet, käyttöjärjestelmien ja ohjelmien asennukset sekä laitteiden liittämiset tietoverkkoon. Vianetsintä niin ohjelmisto- kuin laitepuolella sekä tietoverkkopuolella kuuluvat opiskeltaviin asioihin. HTML-kielen (HyperTextMarkupLanguage) perusteiden opiskelu ja WWW-palvelimen (WorldWideWeb) asentaminen sekä omien WWW-sivujen luominen palvelimeen tulevat tutuiksi opiskelun aikana. Tietoturva-asioista on tavoitteena

oppia olennaiset perusasiat, koska tietoturva on tietotekniikassa tärkeässä roolissa. Tämän tutkinnon osan harjoituksiin kuuluu myös langattoman tukiaseman käyttöönotto erilaisilla määrittelytiedoilla.

Sulautettujen järjestelmien toteutusosiossa keskitytään sulautetun järjestelmän käytön ja ohjelmoinnin periaatteisiin. Lisäksi perehdytään laitetekniikan perusteisiin niin komponenttitasolla kuin sähköisten liitännätarvikkeiden osalta sekä teoriassa että harjoitusten avulla. Tavoitteena on oppia erityyppisten sulautettujen järjestelmien perusohjelmointia ja niiden testausta. Ohjelmointi suoritetaan laiteläheisellä C-ohjelmointikielellä ja tavoitteena on oppia tulkitsemaan lausekieltä sekä ohjelmoimaan hyvää ohjelmointitapaa noudattaen. Samalla opitaan ohjelmistokehitysympäristön käyttö ja ohjelmistokehityksen periaatteet. C-ohjelmointikielellä tehdään harjoituksia, jossa luetaan tietoa sisään mikrokontrollerin portteihin ja lähetetään tietoa porttien kautta erilaisille antureille. Ohjelmiin luodaan erilaisia ehto- ja valintarakenteita sekä aliohjelmiä. Harjoituksissa tavoitteena on oikein jäsentynyt C-kielinen ohjelma, jota ohjelmoija osaa tulkita sekä korjata.

Tietoliikennetekniikan töiden osiossa keskitytään analogisen- ja digitaalisen tiedonsiirtoverkon periaatteisiin, tietoliikenteen peruskäsitteisiin ja siirtoteiden ominaisuuksiin. Laajakaistatekniikoiden ja mobiililiikenteen opiskelu sekä laitteiden ja tietoverkon tietoturvasasiat kuuluvat opiskeltaviin asioihin. Tietokonelaitteiden väliset toimintaperiaatteet ja tietoliikennemenetelmät ovat keskeisinä opiskeltavina asioina. Tavoitteena on myös oppia perusteet TCP/IP-protokollaperheen eri yhteyksiä hoitavista protokollista. Pienen yrityksen tietoverkon rakentaminen ja määrittely harjoitustehtävänä kokoaa yhteen edellä käsitellyjä opiskeltavia asioita.

2.4 Ammattielektronikka

Ammattielektronikka (Liite 1, Taulukko 2.1) on elektroniikka-asentajille kuuluva pakollinen ammatillinen tutkinnon osa (30 osp), josta osa suoritetaan työssäoppimisjakson (TOP) aikana. Tämä tutkinnon osa jakaantuu neljään osa-alueeseen:

1. Elektroniikan virtapiirit ja rakentaminen

2. Sulautettujen järjestelmien toteutus
3. Tietokonelaite ja -järjestelmäsennukset
4. Tietoliikennetekniikan työt

Ensimmäisessä osiossa, Elektroniikan virtapiirit ja rakentaminen, ovat olennaisia opiskeltavia asioita elektroniikan yleisimmät komponentit, perusvirtapiirit, elektroniikkakytkentöjen rakentaminen sekä niiden mittaukset. Piirilevyjä suunnitellaan CAD-ohjelmalla ja piirilevyt valmistetaan joko syövytys- tai jyrsinmenetelmällä. Lisäksi tutustutaan analogia- ja digitaalipiireistä muodostuviin peruskytkentöihin. Käydään myös hieman läpi elektroniikkatuotantoa sivuavia asioita. Seuraavissa osioissa, 2, 3 ja 4 on tavoitteena oppia pääpiirteissään samat asiat, jotka on esitelty kohdassa 2.3 *Tietokone- ja tietoliikenneasennukset*.

2.5 Sulautetut sovellukset ja projektityöt

Sulautetut sovellukset ja projektityöt (Liite 1, Taulukko 2.2) on valinnainen ammatillinen tutkinnon osa (30 osp) sekä elektroniikka- että ICT-asentajille. Tässä tutkinnon osassa on tarkoitus oppia suunnittelemaan ja rakentamaan sulautetun järjestelmän piirikortti. Kortille valittavat komponentit ja niiden huolellinen kiinnittäminen juottamalla on tärkeä asia piirikortin rakentamista ajatellen. Lisäksi opitaan C-ohjelmoinnin eri perusrakenteita ja oikean tyyppinen ohjelman rakenne. Sulautettujen järjestelmien ohjelmointiin kuuluu olennaisesti aliohjelmien, keskeytysten ja ajastimien käyttö. Kokonaisuuteen kuuluu sulautetun järjestelmän ohjelman lisäominaisuuksien tekeminen ja lisälaitteiden lisääminen sekä niiden ohjelmallinen päivittäminen järjestelmään. Järjestelmä testataan mittauksin tai muulla asianmukaisella tavalla. Komponenttituntemus on tärkeää sekä niiden toimintaperiaatteiden tunteminen.

Muista valinnaisista ammatillisista tutkinnon osista (Liite 1, Taulukko 2.2) voidaan valita sellaisia osia, joita on valittu omassa oppilaitoksessa suoritettaviksi tutkinnon osiksi. Esimerkiksi seuraavat valinnaiset ammatilliset tutkinnon osat tukevat ammatillista osaamista sekä auttavat opiskelijaa työllistymään: *Huoltopalvelut* 15 osp, *RF-työt* 15 osp, *Elektroniikkatuotanto* 30 osp, *Kodin elektroniikka ja asennukset* 15 osp, *Valvonta- ja ilmoitusjärjestelmäsennukset* 15 osp, *Sähköasennukset* 15 osp ja *Paikallisiin ammattitaitovaatimuk-*

siin perustuvia tutkinnon osia 5-15 osp (vaihtelee tutkinnoittain ja alueittain). Vapaasti valittavista tutkinnon osista (Liite 1, Taulukko 2.4) voidaan myös valita tutkinnon osia.

2.6 Sulautettujen järjestelmien opetusalojen rooli OPS:ssa

Eri tutkinnon osien valinnassa on eri ammattiopistoilla ja paikkakunnilla itsenäinen rooli. Eri koulut painottavat opetuksessa erityyppisiä tutkinnon osia. Kolmen eri vuosiluokan välillä on myös painotuseroja ja tutkinnon osien valinnalla on ratkaiseva merkitys siihen, kuinka paljon sulautettujen järjestelmien opetusaloja voidaan käyttää hyväksi opetuksessa. Opetusalojen käyttö keskittyy kummallakin linjalla toiseen vuosikurssiin. Ensimmäisenä vuonna on enemmän yhteisiä aineita (YTO) ja muita ammatillisia perusasioita. Kolmantena vuonna taas tulee enemmän työssäoppimista (TOP) sekä erityisiä syvällisempiä ammattiaineita liittyen omaan opintoalaan. Vanhan OPS:n aikana vapaasti valittaviin kursseihin voitiin sisällyttää lähes kaikenlaisia kursseja. Uuden OPS:n (2015) myötä muuttuivat suositukset niin, että opiskelijat valitsisivat vapaasti valittavien 10 osp:n piiriin ammattiaineita. Oppilaitoksissa onkin räätälöity isommista ammatillisista tutkinnon osista pienempiä 5 osp:een kokonaisuuksia vapaasti valittaviin tutkinnon osiin.

Ohjelmoinnin ja sulautettujen järjestelmien opetukseen kehitettyjen opetusalojen kuten LEGO, Arduino ja MatrixTSL hyödyntäminen soveltuu hyvin OPS:ssa olevien monien eri tutkinnon osien opettamiseen. Ensimmäisenä vuonna voidaan monia elektroniikan asioita konkreettisesti esitellä sulautettujen järjestelmien opetusalojen avulla. Toisena ja kolmantena vuonna voidaan sulautetuilla opetusaloilla eri tutkinnon osien yhteydessä tuoda teoria-asioita käytännön tasolle. Näitä konkreettisesti esitettäviä teoria-asioita löytyy esimerkiksi tietokone-, tietoliikenne- ja ohjelmointitehtävistä. Kun teoria-asioita voidaan esittää käytännön esimerkeillä, on sillä iso merkitys opiskelumotivaatioon ja sitä kautta myös opiskelijan oppimiseen.

3 Sulautettujen järjestelmien tekniikka

Sulautettuja järjestelmiä käytetään tänä päivänä hyvin monenlaisissa enemmän tai vähemmän älyä vaativissa elektronisissa laitteissa, esimerkiksi kodinkoneissa, roboteissa, ruohonleikkuuroboteissa, viihde-elektroniikan laitteissa, langattomissa järjestelmissä, pankkiautomaateissa, lääketieteellisissä kehon mittausvälineissä, kuntoilulaitteissa tai muissa vastaavissa itsenäisesti toimivissa laitteissa. Sulautettu järjestelmä on yleensä itsenäisesti toimiva laite, joka sisältää elektroniikkaa ja siihen kiinteästi liitetystä muistipiiristä olevan ohjelman. Tätä yhdistelmää kutsutaan nimellä firmware (rauta + ohjelma). Laite suunnitellaan kuluttamaan vähän tehoa, jolloin se yleensä toimii pienellä paristolla tai akulla päiviä ehkä jopa vuosia. Esimerkkeinä voidaan mainita sisä- / ulkolämpömittari tai elektroninen herätyskello, joka hakee tarkan ajan itseksensä radioteitse. Käyttäjän ei tarvitse määrittää kellossa muita asetuksia kuin ainoastaan aktivoida herätys sekä muuttaa herätysaikaa. Mitä enemmän ohjelmallisia toimintoja sulautettuun järjestelmään otetaan mukaan, niin sitä enemmän tehoa yleensä kuluu ja laitteen toiminta-aika lyhenee. Tästä esimerkkinä ovat nykyajan tehokkaat ja monipuoliset matkapuhelimet eli ”älykännykät”. Niissä on niin paljon toimintoja, että tehokkaankin akun toiminta-aika jää muutamaan päivään. Toinen esimerkki on robotti, joka liikkuu ja kuluttaa paljon tehoa liikkessaan.

Tässä työssä keskitytään kolmen valmistajan sulautettujen järjestelmien opetukseen kehitettyyn alustaan LEGO-roboteihin sekä Matrixin - ja Arduinon mikrokontrollerikortteihin. Syy miksi nämä laitteet valittiin on se, että ne ovat laajasti käytössä opetusmaailmassa eri tasoisissa oppilaitoksissa aina peruskouluista yliopistoihin.

3.1 Sulautettujen järjestelmien peruskomponentit

Sulautettu järjestelmä koostuu useista elektroniikan komponenteista ja niiden avulla toimivista käyttöjärjestelmistä, sovellusohjelmista sekä tietoliikenneyhteyksistä.

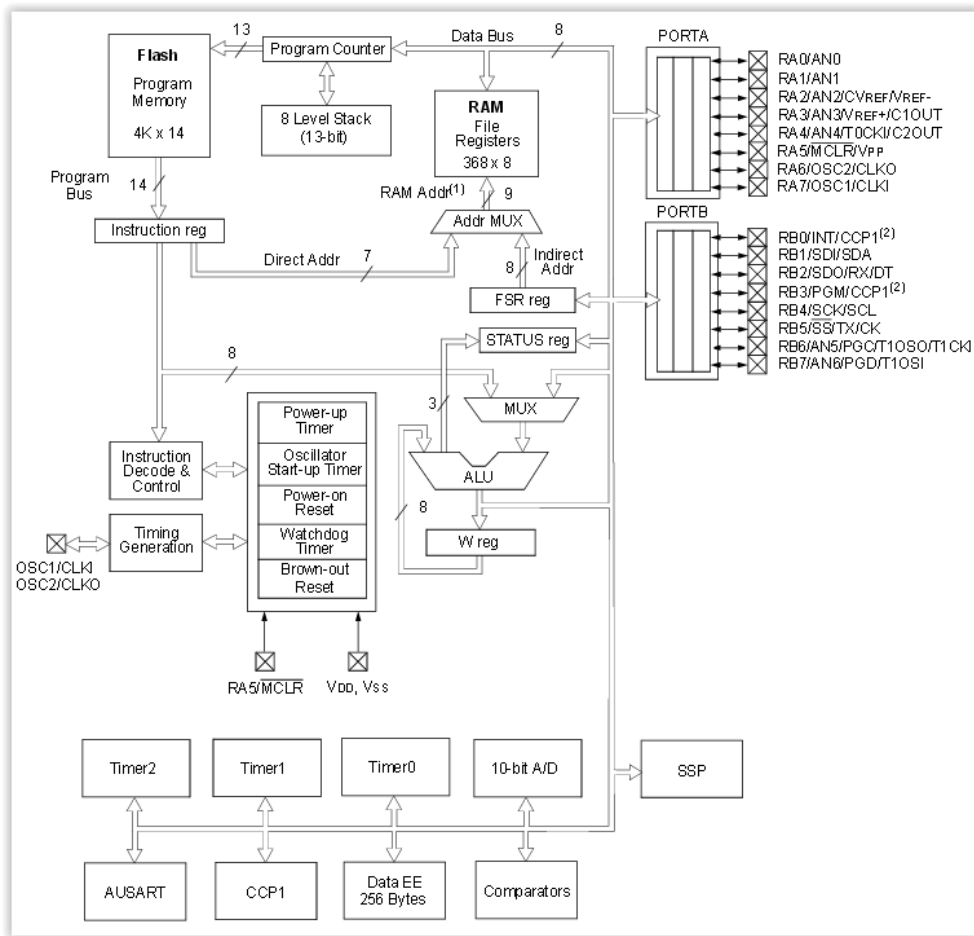
Peruskomponentteja ovat muun muassa:

- mikrokontrolleri

- muistit, RAM, EEPROM, Flash
- väylät, data- muisti-, ohjausväylä
- virtalähde, patteri, akku
- anturit, ADC-, lämpö-, liike-, valo-, ultraääni-, kaasuanturi
- tietoliikenneohjelma, joko kaapeli- tai langaton yhteys
- käyttöjärjestelmä
- sovellusohjelma

Ohjelmamuisteina käytetään usein Flash-muistia tai EEPROM-muistia (Electrically Erasable Programmable Read Only Memory), jotka ovat sähköisesti ohjelmoitavia muistipiirejä ja niissä ohjelma säilyy virtakatkon ajan. Työmuistina on yleensä käytössä RAM-muistia (Random Access Memory), jonne ladataan esimerkiksi Flash-muistista suoritettava ohjelma laitteen käynnistyessä. Sitä voidaan käyttää myös tilapäisesti tietovarastona, esimerkiksi luettaessa lämpötila-anturilta lämpötila-arvoja sovituin väliajoin. Luetut tiedot täytyy tarvittaessa tallettaa RAM-muistista muualle, koska tiedot häviävät virtakatkossa. Ulkoisena anturina voi toimia esimerkiksi lämpötila-, paine-, ääni-, ultraääni-, kosketus- tai valoanturi. Niistä saatava sähköinen signaali johdetaan mikrokontrollerille sopivaan sisäänmeno-noon. Anturilta saatava analoginen tai digitaalinen signaali voidaan muuntaa mikrokontrollerin sisällä esimerkiksi AD/DA-muuntimen kautta sopivaan muotoon jatkokäsittelyä varten. Firmware tarkoittaa muistipiirin ja siihen talletetun ohjelman toiminnallista kokonaisuutta. Firmware-ohjelmisto talletetaan sulautetussa järjestelmässä yleensä EEPROM- tai Flash-muistiin. Tuo ohjelmisto on yleensä laitevalmistajan tekemä, joka mahdollistaa laitteen eri toimintojen käytön yhdessä laitteelle tehdyn sovellusohjelman kanssa.

Mikrokontrolleri on sulautetun järjestelmän ydin, joka suorittaa kaikki laskutoimitukset ja loogiset operaatiot. Se sisältää ALU:n (Arithmetic Logical Unit), sisäiset rekisterit, AD/DA-muuntimet, ajastimet, laskurit, RAM-muistin, I/O-portit (Input/Output) ja väylät eri osien välillä. Alla olevassa kuvassa 3.1 on esimerkkinä Matrixin multiprogrammerissa [4] tehdaskokoonpanossa olevan PIC16F88 mikrokontrollerin periaatekuva.

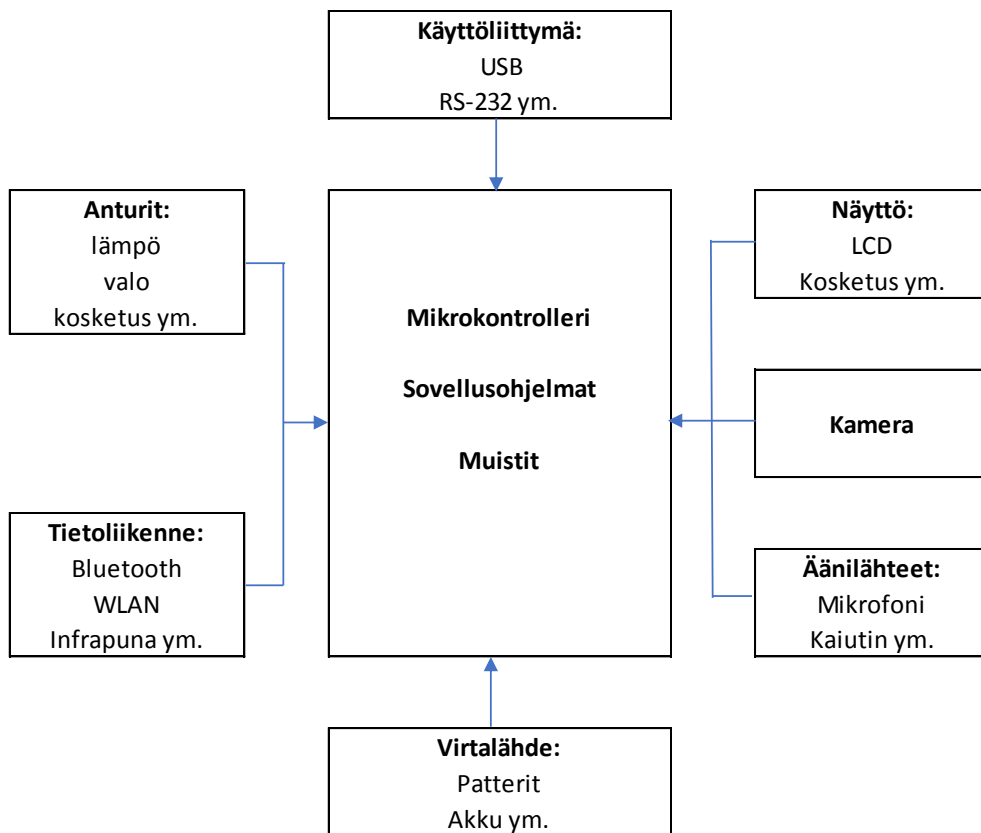


Kuva 3.1. PIC16F88 mikrokontrollerin periaatekuva [4]

Mikrokontrolleri eroaa mikroprosessorista muun muassa siinä, että siihen on integroituna sarjaliikenneportteja, ajastimia, FLASH-muistia, I/O-portteja ja RAM-muistia. Mikroprosessoria käytetään yleensä tietokoneissa ja se kuluttaa paljon enemmän tehoa sekä on tehokkaampi laskentakyvyiltään ja tiedonsiirto-ominaisuuksiltaan kuin mikrokontrolleri. Mikrokontrollerin ohjelmointi tapahtuu yleensä RS232-sarjaliikenneväylän tai nykyään useimmiten sarjaliitännäisen USB-väylän kautta. Kellotaajuudet mikrokontrollereissa ovat huomattavasti matalampia kuin mikroprosessoreissa ja niiden suunnittelussa paneudutaan siihen, että ne saadaan kuluttamaan huomattavasti vähemmän tehoa kuin mikroprosessorit. Tämä on tärkeää, koska mikrokontrolleripohjaiset laitteet toimivat yleensä pattereilla tai akuilla. Sulautetussa järjestelmässä ei yleensä ole ulkoista muistia ohjelmille, vaan ohjelmat talletetaan mikrokontrollerin sisällä olevaan muistiin.

3.2 Sulautetun järjestelmän toimintaperiaate

Sulautettu järjestelmä koostuu mekaanisista osista, elektroniikkakomponenteista ja tietynlaista toimintaa varten suunnitellusta ohjelmistosta sekä yleensä virtalähteenä ovat patterit tai akku. Laite voi olla pieni, vaikka elektroninen kuumemittari tai iso kuten hissi. Kun sulautettuun järjestelmään kytketään virta, niin siinä olevassa muistipiirissä oleva käynnistys-ohjelma (firmware) käynnistyy ja lataa varsinaisen sovellusohjelman käyttömuistiin, jolloin laite alkaa toimia ns. ikuisessa silmukassa. Laite suorittaa ohjelmiston ohjaamana haluttuja toimenpiteitä, esimerkiksi lukee johonkin porttiin kytketyn lämpötila-anturin tietoja määrätyn väliajoin. Lämpötilat voidaan tallettaa käyttömuistiin ja myöhemmin esimerkiksi Flash-muistiin talteen myöhempää käyttöä varten. Sulautetussa järjestelmässä on yleensä joko ledejä tai LCD-näyttö (Liquid Crystal Display), joista nähdään järjestelmän toiminnan tilat. Pieneltä näppäimistöltä tai kytkimiltä voidaan vastaavasti antaa komentoja ohjelman vaatimusten mukaisesti.



Kuva 3.2. Sulautetun järjestelmän periaatekuva

Keskeinen yksikkö sulautetussa järjestelmässä (Kuva 3.2) on itse mikrokontrolleriyksikkö, jossa sijaitsee sovellusohjelma ja tarvittavat muistit. Keskusyksikkö kommunikoi ulkoisten anturien tai näyttölaitteiden kanssa ohjelman mukaisesti. Tietoliikennettä varten on älykkäämmissä laitteissa yleensä jokin radioyhteys kuten Bluetooth tai WLAN (Wireless Local Area Network). Itse laite on yleensä liikuteltava, joten siinä on yleensä patterien tai akun käyttömahdollisuus.

Esimerkkeinä sulautetuista järjestelmistä esitellään seuraavana kolme opetuskäyttöön tarkoitettua järjestelmää, LEGO Mindstorms, Arduino ja MatrixTSL. Noiden valmistajien järjestelmillä voidaan opettaa monenlaisia sulautettujen järjestelmien toimintoja.

3.3 LEGO Mindstorms -robotti

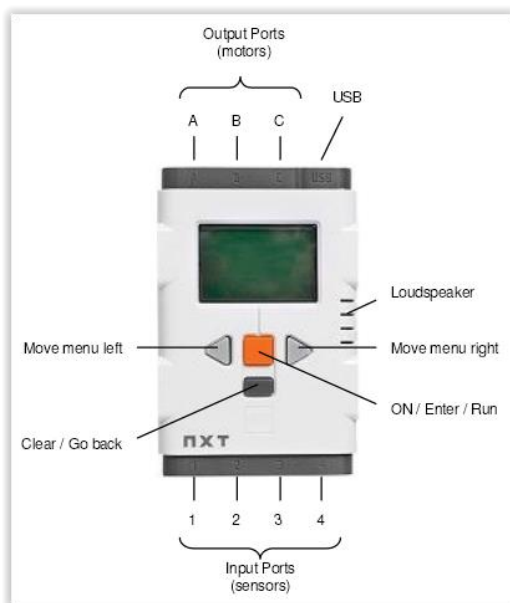
Tietokoneohjattavien LEGO-robottien historia alkaa vuodesta 1986. Ensimmäisen laajemmin myyntiin tulleen robotin keskusyksikköä kutsuttiin nimellä RCX, joka julkistettiin vuonna 1998. Seuraavasta LEGO-robotista käytettiin nimeä NXT, joka julkistettiin vuonna 2009. LEGO-robotin viimeisin versio on EV3 (tilanne 1.6.2017), joka julkaistiin syyskuussa vuonna 2013.

Ensimmäinen versio LEGO-robotista oli teknisiltä ominaisuuksiltaan aika rajoittunut ja siinä oli paljon puutteita ohjelmointipuolella. Tehdasasetuksena siihen tallennettiin ohjelmat RIS-kielellä (Robotics Invention System) [5]. Se oli graafinen ohjelmointikieli, joka sopi lapsille. Ohjelmoitavuutta parannettiin siten, että Dave Baum modifioi C-kielestä version NQC (Not Quite C) [15], joka paransi tilannetta huomattavasti. ADA-ohjelmointikieltä [8] voitiin myös käyttää, mutta siitäkään ei saatu kunnolla toimivaa versiota. LEGO:n alkuperäisen ohjelmointikielen mukainen versio tehtiin myös LabVIEW:sta, joka sai nimekseen RoboLab [13]. Siinä oli edistyksellisiä piirteitä alkuperäiseen verrattuna. Myöhemmin suuri parannus oli korkean tason ohjelmointikielen ominaisuuksia omaava legOS [12], jolla korvattiin alkuperäinen RIS-kieli. Tuo legOS-ohjelmointikieli oli Open Source -projektin puitteissa luotu ilmaiseksi jaettava ohjelmisto [25], samoin kuin NQC. Sitten saataville tuli uusi Open Source -ohjelmisto nimeltään leJOS [14], jossa oli osa Java

Virtual Machine -koodista [5]. Näillä toimenpiteillä parannettiin robotin ohjelmistoa monelta osin, mutta silti siihen jäi paljon sujuvaa käyttöä haittaavia vikoja [24].

NXT-version käyttö on huomattavasti edistyksellisempää ja joustavampaa. Sillä pystytään toteuttamaan opetusmielessä monipuolisia ohjelmointiharjoituksia. Siinä on helposti vaihdettava firmware ja tekniset ominaisuudet mahdollistavat melko laajojen sovellusohjelmien tekemisen. Moottorien ohjelmointimahdollisuudet ovat monipuolisemmat. Ohjelmointiliittymänä käytetään USB-liitäntää, joka on helppokäyttöinen. Lisäksi robottia voidaan käyttää Bluetooth- ja IR-yhteyksien avulla. Robotin näytön saa myös näkymään tietokoneen näytöllä (Kuva 3.5), jolloin robotin käyttö onnistuu tietokoneen avulla [24].

Teknisiltä ominaisuuksiltaan LEGO Mindstorms NXT-malli on jo melko tehokas. Robotin keskusyksikössä on mikrokontrollerina 32-bittinen ARM7. Flash-muistia siinä on 256 KB ja 64 KB RAM-muistia. Keskusyksikössä on myös toinen 8-bitin mikrokontrolleri, joka on AVR-mikrokontrolleriperheestä. Siinä on 4 KB Flash-muistia ja 512 tavua RAM-muistia. Ulkoiset yhteydet keskusyksikköön tapahtuvat Bluetoothin tai USB:n kautta. Robottiin ei saa ulkoisia muisteja, joten kaikki ohjelmat tulee sopia laitteen sisäiseen muistiin. Ohjelmointi tapahtuu USB-kaapelin kautta [16, 18].

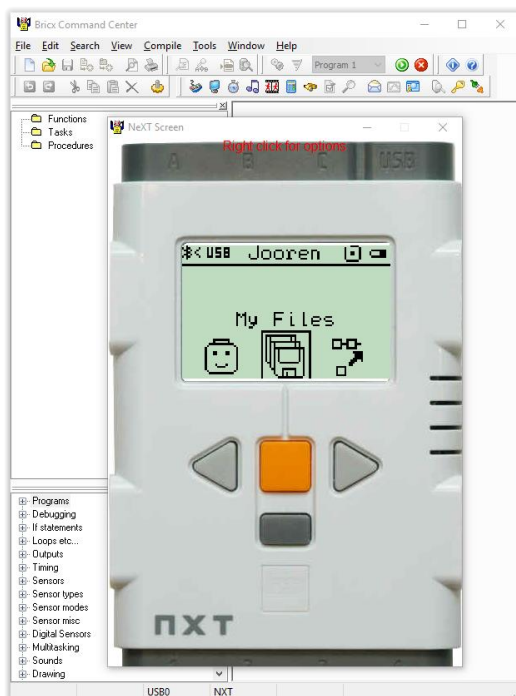


Kuva 3.3. LEGO-robotin (NXT) liitännät ja käyttöpainikkeet [18]

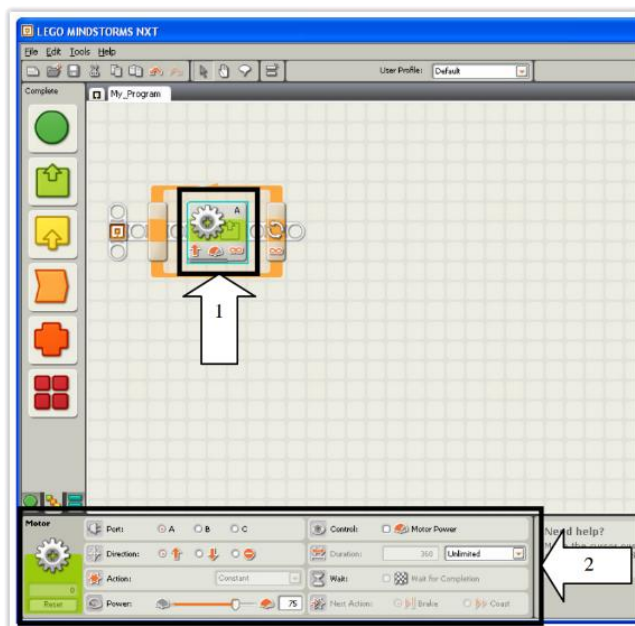


Kuva 3.4. LEGO-robotin periaatekytkentä [18]

LEGO:n keskusyksikössä on 3 ulostuloa moottoreiden ohjaamiseen ja 4 sisäänmenoa anturien tai sensorien lukemiseen, jotka näkyvät kuvassa 3.3. Normaalisti LEGO-robotti kytkeään kuvan 3.4 mukaisesti. Sovelluksesta riippuen sen sisäänmenoihin voidaan tarvittaessa kytkeä kaikki 4 anturia sekä ulostuloihin 3 moottoria [18]. Keskusyksikössä on 100 x 64 kuvapisteen (Pixel) LCD-näyttö, kaiutin sekä käsiohjausta varten hallintanäppäimet. Sisäänmenoportteihin voidaan kuhunkin liittää jokin ultraääni-, ääni-, kosketus- tai valosensoreista [18].



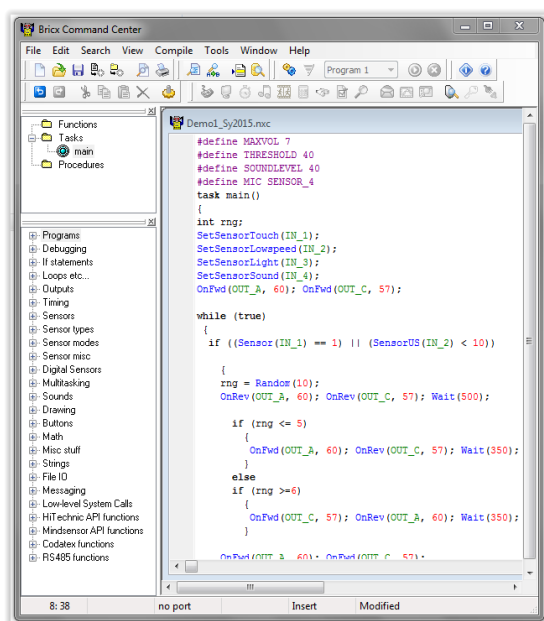
Kuva 3.5. LEGO-robotin ohjausikkuna



Kuva 3.6. LEGO-robotin graafinen ohjelmointikäyttöliittymä

LEGO-robotin ohjaus voidaan suorittaa myös tietokoneelta käsin hiirtä käyttäen samalla tavalla kuin suoraan NXT-yksikön ohjauspainikkeiltakin kuvan 3.5 mukaisesti. Käyttöliittymäikkuna saadaan avattua *Tools*-valikon valinnalla *NXT Screen*. Sitten valitaan LEGO:n ruudun päällä hiiren oikealla painikkeella *Polling*. Tästä on apua robotin etäohjauksessa. Kun haluttu ohjelma on valittu näyttöön nuolinäppäimillä, niin se käynnistetään oranssia käynnistä-nappia painamalla. Ohjelmat tallennetaan USB-väylän kautta robotin mikrokontrolleriin.

LEGO-robottiin on mahdollista vaihtaa sen sisäinen firmware-ohjelmisto. Siinä tehdasasennuksena oleva firmware mahdollistaa LEGO-robotin ohjelmoinnin Labview-pohjaisella graafisella ohjelmointikäyttöliittymällä kuvan 3.6 mukaisesti. Tuohon kuvaan on käyttöliittymässä valittu graafinen moottori-elementti, jonka ominaisuuksia voidaan muuttaa. Mikäli robotin firmware vaihdetaan, niin voidaan käyttää joko NXC-ohjelmointikieltä, tai jotain muuta ohjelmointikieltä minkä vaihdettu firmware mahdollistaa [18].



Kuva 3.7. LEGO-robotin NXC-ohjelmointikäyttöliittymä

Mikäli käytetään NXC-ohjelmointia, niin vaihdetaan ensin robotin firmware. Sen jälkeen asennetaan tietokoneeseen BricxCC-ohjelmointikäyttöliittymä, jollainen on esitetty kuvassa 3.7. Nyt on mahdollista käyttää NXC-ohjelmointikieltä (Not eXactly C), joka on C-ohjelmointikielen näköinen. Siihen on tehty lisäfunktioita, jotta kieli sopii paremmin robotien ohjelmointiin [19].

Viimeisimmästä LEGO-robotin EV3-rakennussarjasta voidaan rakentaa 17 erilaista robotihahmoa. Sarjan mukana toimitettavasta ohjelmointiympäristöstä löytyy myös robotin ohjelmointi- ja ohjausohjelmistot älypuhelimien ja tablet-tietokoneeseen. EV3-robotti on teknisten ominaisuuksiensa puolesta NXT:tä kehittyneempi, monipuolisempi ja nopeampi [24].

LEGO Mindstorms -robottien opetusmateriaali

LEGO-yhtiöllä on koulutusta varten sivustot, joilta löytyy paljon koulutusmateriaalia eri kouluasteille. Sivustolta löytyy yleistä ohjeistusta robotin ohjelmoimisesta, opiskelijoille tietoa sekä opettajalle ohjeistusta erilaisten ohjelmointitehtävien tekemiseen. Opetus kes-

kittyy uusimman robotin EV3:n ympärille ja siinä käytettävään ohjelmistoon. Tehtäviä ja ohjeita voidaan ladata pdf-muodossa omaan käyttöön ilmaiseksi [38].

Yhteenveto LEGO-robotista

LEGO-robotti on edullinen ja nopea ottaa käyttöön opetuksessa. Sillä voi harjoitella monipuolisesti liikkuvan laitteen ohjelmointia. Robotissa on useita moottoreita ja antureita, joiden toimintaa voidaan harjoitella ohjelmoimalla. Se soveltuu hyvin ohjelmoinnin aloittamiseen, koska se on hyvin konkreettinen laite. Se esimerkiksi kääntyy väärään suuntaan, jos ohjelmoinnissa on vikaa, jolloin ohjelmointivirheen jäljille on helpompi päästä. Tämä osaltaan alentaa ohjelmointikynnystä ja helpottaa opiskelijoiden ohjelmoinnin motivoinnissa.

3.4 MatrixTSL

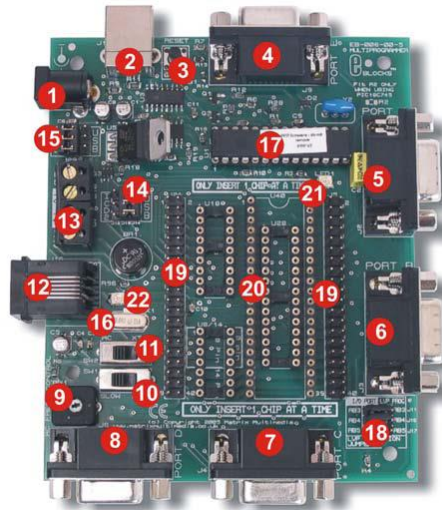
Aiemmin MatrixMultimedia, nyky nimeltään MatrixTSL on kehittänyt laajan sekä edullisen mikrokontrolleri- ja sensorikortisarjan opetuskäyttöön [3]. Ohjelmointialustoja löytyy esimerkiksi PIC- ja ARM- sekä AVR-mikrokontrollereille. Sensori- ja koekytkentäkortteja sekä muita erilaisten laitteiden ohjauskortteja löytyy jo yli 50 kpl. Uusia eri käyttökohteille tarkoitettuja kortteja tulee markkinoille koko ajan. Sarja on hyvin oppilaitoskäyttöön soveltuva, edullinen ja monipuolinen. Flowcode-vuokaavio-ohjelmointiympäristö on mielenkiintoinen ja siinä on ohjelmointikynnys saatu matalalle tasolle. Opiskelijoiden ohjelmoinnin motivaatio kasvaa helpokäyttöisen Flowcode-vuokaavio-ohjelmoinnin ansiosta, koska ohjelmoinnin aloittaminen yksinkertaistuu [3].

Järjestelmä sisältää kehityskortit PIC-, AVR- ja ARM-mikrokontrollereille. Lisäksi on saatavilla valmiit kehitysalustat seuraaville laajemmille sovellusalueille: GSM-, USB-, Bluetooth-, CAN-, Zigbee-, Internet (TCP/IP) -, Audio DSP-, RFID- ja FPGA-sovellukset. Lisälaittevalikoimaan kuuluu sensorikortteja eri sovellusalueille kaikkiaan yli 50 kpl. Tuettuja tietoliikenneyhteyksiä ovat muun muassa SPI, RS232, I2C, TCP/IP, USB, LIN, CAN, Bluetooth, GSM, GPS, ZigBee, RFID, MIDI, IrDA, RF(ISM), RC5 ja RS485. Flowcode-ohjelmointiympäristöt löytyvät PIC-, AVR- ja ARM-mikrokontrollereille [3].

Multiprogrammerin rakenne

MatrixTSL valmistaa PICmicro[®]-nimellä PIC-mikrokontrollereiden ohjelmointi- ja kehityskorttia, jota kutsutaan nimellä Multiprogrammer (EB006). Siihen voidaan liittää yhteensä 5 erilaista E-blocks[®]-lisäkorttia 9 pinnisen RS232-liittimen avulla. Kuvassa 3.8 on esitetty Multiprogrammer ja sen tärkeimmät osat [40].

1. Power connector - either polarity
2. USB connector
3. Reset switch
4. Port E I/O
5. Port A I/O
6. Port B I/O
7. Port C I/O
8. Port D I/O
9. RC clock speed potentiometer
10. RC clock speed switch
11. Clock crystal / RC switch
12. ICD2 socket
13. Power screw terminals
14. USB/ICD2 programming selector
15. USB/ICD2 power selector
16. Removable crystal
17. USB control chip – do not remove
18. Low Voltage Program pin selector link block
19. Expansion connector – two off
20. Turned pin DIL sockets for 8, 14, 18, 28, 40 pin PICmicro devices
21. 'Ready to go' programming LED
22. Power LED



Kuva 3.8. Multiprogrammer-ohjelmointikortti [40]

Korttiin on saatavilla useita lisäkortteja. Multiprogrammeria sekä lisäkortteja kutsutaan yhteisellä nimellä E-blocks[®]. MatrixTSL toimittaa ohjelmointikorttiin ohjelmointiympäristön assembly-, C- ja Flowcode-ohjelmointia varten. Kortilla voidaan ohjelmoida 8, 14, 18, 28 ja 40 pinnisiä DIP-tyyppisiä (Dual Inline Package) mikrokontrollereita. Ohjelmat tallennetaan USB-väylän (Universal Serial Bus) kautta PPP-ohjelmalla (Point to Point Protocol) mikrokontrolleriin. Flowcode-ohjelmointiympäristön kääntäjä tuottaa HEX-koodina (Hexadecimal) ohjelman, jonka PPP-ohjelma tallentaa mikrokontrolleriin.

Taulukossa 3.1 on esimerkkinä suurin osa PIC-mikrokontrollereista, joita Flowcode ja Matrixin PIC-mikrokontrollerikortti tukevat. AVR- ja ARM-mikrokontrollerikortteille löytyy vastaavat luettelot.

Taulukko 3.1. Matrixin PIC-ohjelmointikorttiin sopivat mikrokontrollerit [40]

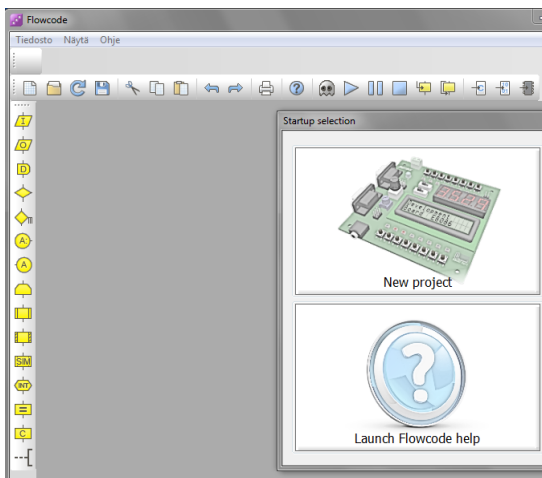
8-bittiset mikrokontrollerit	16-bittiset mikrokontrollerit	32-bittiset mikrokontrollerit
PIC12F629, PIC12F675, PIC12F635, PIC12F683	PIC16F627A, PIC16F627, PIC16F628A, PIC16F628, PIC16F630, PIC16F648A, PIC16F676, PIC16F684, PIC16F688, PIC16F636, PIC16F716, PIC16F72, PIC16F737, PIC16F73, PIC16F747, PIC16F74, PIC16F767, PIC16F76, PIC16F777, PIC16F77, PIC16F818, PIC16F819, PIC16F83, PIC16F84A, PIC16F84, PIC16F870, PIC16F871, PIC16F872, PIC16F873A, PIC16F873, PIC16F874A, PIC16F874, PIC16F876A, PIC16F876, PIC16F877A, PIC16F877, PIC16F87, PIC16F88	PIC18F242, PIC18F248, PIC18F252, PIC18F258, PIC18F442, PIC18F448, PIC18F452, PIC18F458, PIC18F1220, PIC18F1320, PIC18F2220, PIC18F2320, PIC18F2331, PIC18F2410, PIC18F2420, PIC18F2431, PIC18F2439, PIC18F2455, PIC18F2510, PIC18F2515, PIC18F2520, PIC18F2525, PIC18F2539, PIC18F2550, PIC18F2585, PIC18F2586, PIC18F2610, PIC18F2620, PIC18F2680, PIC18F2681, PIC18F4220, PIC18F4320, PIC18F4331, PIC18F4410, PIC18F4420, PIC18F4431, PIC18F4439, PIC18F4455, PIC18F4510, PIC18F4515, PIC18F4520, PIC18F4525, PIC18F4539, PIC18F4550, PIC18F4585, PIC18F4586, PIC18F4610, PIC18F4620, PIC18F4680, PIC18F4681

Ohjelmointikortilla tulee oletuksena PIC16F88 mikrokontrolleri. Kortin kellooskillaattoriksi voidaan valita RC-piiri tai kideoskillaattori. Kortissa on 5 I/O-porttia (Input/Output). Vastaavatyypiset ohjelmointikortit löytyvät myös AVR- ja ARM-mikrokontrollereille.

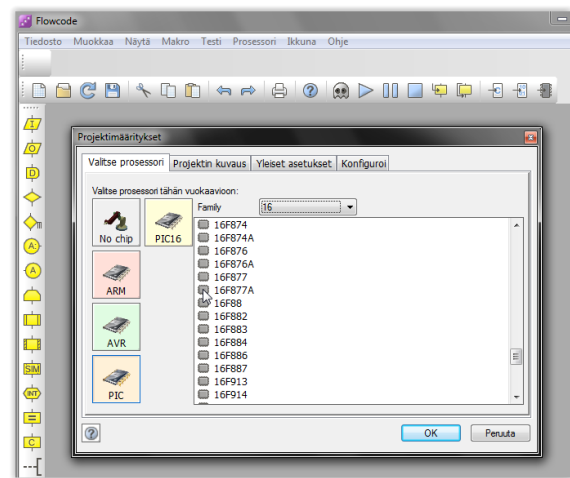
Ohjelma voidaan siirtää mikrokontrollerin EEPROM-muistiin myös JTAG-liitännän [20] (Joint Test Action Group) kautta. Ohjelmasta voidaan etsiä myös virheitä JTAG-liitännän kautta. Silloin käytetään esimerkiksi Microchip-yrityksen MBLAB[®]-kehitysohjelmistoa [21] ICD (In-Circuit Debugger) [22] oman lisäohjelmointikortin avulla (ICD2). Tietokoneen minimivaatimukset ohjelmankehitystä varten ovat Pentium 100 MHz, USB-portti, 2 Mt kiintolevytilaa, 16 Mt RAM. Käyttöjärjestelmistä yhteensopivat ovat Windows 98/ME/2000/XP/Vista/Win7 sekä uudemmat.

Flowcode-ohjelmointiympäristö

Ohjelmointiympäristöä kutsutaan nimellä IDE (Integrated Development Environment). Flowcode-ohjelmankehitysympäristöön on tehty paljon eri oheiskortteille sopivia komponenttimakroja. Niiden avulla eri korttien ohjelmointi on huomattavasti helpompaa. Alla olevissa kuvissa esitetään Flowcode 6 version käyttöönottoa. Kun Flowcode käynnistetään, niin ensin tehdään ohjelmointityötä varten oma projekti kuvan 3.9 mukaisesti. Tässä vaiheessa täytyy tarkistaa, mikä mikrokontrolleri on fyysisesti kiinnitettynä Multiprogrammerissa, jotta projektiin liitetään oikea mikrokontrolleri kuvan 3.10 mukaisesti.

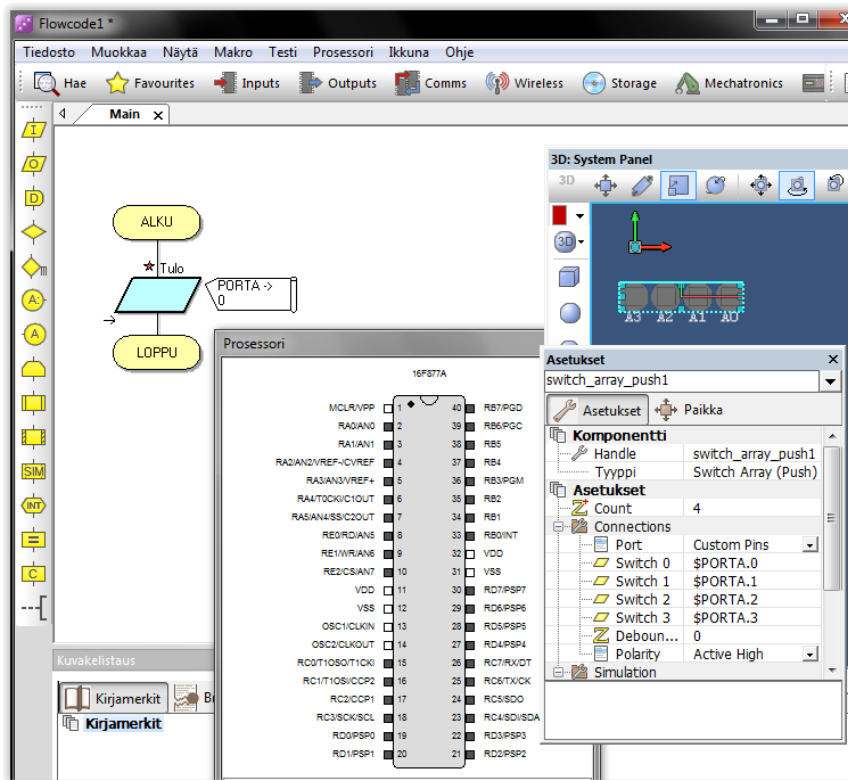


Kuva 3.9. Flowcode 6 projektin luonti



Kuva 3.10. Mikrokontrollerin valinta

Kuvasta 3.10 näkyy, että valittavissa on laaja valikoima PIC-mikrokontrollereita. Lisäksi voidaan käyttää AVR- [3] ja ARM-mikrokontrollereita [3]. Hyvin yleinen PIC-mikrokontrolleri on 16F877A. Sitä käyttämällä voidaan ottaa käyttöön laaja valikoima erilaisia oheiskortteja, enemmän sisäistä muistia sekä monia lisäominaisuuksia. Käyttöliittymästä löytyy myös valinnat esimerkiksi ECIO:n ohjelmointiin. Sitten kun projektin luonti on valmis, päästään heti tekemään haluttua ohjelmaa Flowcode vuokaavio-ohjelmoinnilla.



Kuva 3.11. Ohjelmoinnin aloitus Flowcode-ympäristössä

Yllä olevasta kuvasta 3.11 nähdään Flowcoden vuokaavio-ohjelmointikäyttöliittymän periaate. Kuvassa on näkyvissä projektin luontivaiheessa valittu mikrokontrolleri ja sen kantakytkenä. Mikrokontrollerin valinnan jälkeen on hyvä valita seuraavaksi System Paneliin ne Input- ja Output-laitteet tai muut käytettävät laitteet, joita on tarkoitus ottaa mukaan ohjelmointiin. System Paneliin on tässä tuotu Input-valikon kautta komponentiksi kytkinelementti. Kytkimien asetuksia voidaan muuttaa Asetukset-valikossa. Käyttöliittymään vasemmasta reunasta voidaan hiirellä vetää vuokaavioelementtejä työpöydälle ja siten rakentaa ohjelman runko. Ohjelmaan on ensimmäiseksi tuotu graafinen Input-elementti. Elementtien sisään määritellään tarvittavat muuttujat ja muut parametrit. Käyttöliittymään voidaan ottaa näkyviin erilaisia ominaisuusikkunoita kuten mikrokontrollerin kantakytkenä ja lisäkorttien tärkeimpiä ominaisuuksia esimerkiksi kytkimet ja ledit. Tehtyä ohjelmaa voidaan ensin simuloida ohjelmointiympäristössä, jolloin mahdolliset virheet voidaan poistaa. Ikkunoissa näkyy simuloinnin ajan eri muuttujien vaihtuvia tietoja. Mikrokontrollerin pinnien tilojen muutokset näkyvät eri väreillä. Mikäli LED- tai LCD-elementti on otettu

mukaan ohjelmaan, niin niissä näkyy joko haluttuja tai virheellisiä viestejä. Simuloinnin jälkeen voidaan ohjelmaa korjata tai tallentaa eli ”polttaa” se mikrokontrollerin muistiin.

MatrixTSL -järjestelmien opetusmateriaali

Yritys tarjoaa Internet-sivuillaan laajaa tukea heidän laitteidensa opiskeluun sekä Flowcode-ohjelmoinnin harjoitteluun. Jokainen voi opiskella omatoimisesti heidän Learnig-Center-sivuillaan olevilla harjoitteluohjeilla. Flowcode-ohjelmointiin löytyy aloittelijoille harjoituksia, videoita, esimerkkejä ja testi-ohjelmia. Sivuilta löytyy myös laitteiden tekniset tiedot. Opettaja saa koulutuspaketin silloin kun ostaa tuotteen, joka käsittää jonkin kokonaisjärjestelmän, esimerkiksi CAN-väylä, TCP/IP, Bluetooth ja niin edelleen [39].

Yhteenveto MatrixTSL:n laitteista

Matrixin laitevalikoima on jo melko laaja, ja jos on hankittuna monia erilaisia anturikortteja, niin järjestelmää voidaan hyödyntää tosi monenlaisten ohjelmien suunnitteluun ja erilaisten ilmiöiden tutkimiseen. Vuokaavio-ohjelmoinnilla helpotetaan ohjelmoinnin opiskelua sekä ohjelman loogisuuden ymmärtämistä. Eri anturikortteille tehtyjen komponenttimakrojen avulla säästytään pikkutarkalta ohjelmointikikkailulta, joka saattaa olla este joillekin opiskelijoille ohjelmointitöissä. Komponenttimakrolle annetaan vain tietynlaiset arvot ja haluttu tehtävä suoritetaan. Tällöin voidaan keskittyä enemmän ohjelman loogiseen toimintaan eikä pikkutarkkaan ohjelmointikielen syntaksiin. Uusin versio Flowcode-ohjelmointikielestä on versio 7 mikä julkaistiin 2016.

3.5 Arduino

Arduino-järjestelmiä on käytetty tuhansissa pienissä opetusikäisten sovelluksissa, niin kuin myös ammattilaisten sovellusprojekteissa. Mikrokontrollerialustat ovat edullisia. Ne ovat aloittelijallekin helppoja käyttää ja sovellusohjelmisto toimii useissa käyttöjärjestelmissä. Arduino perustuu avoimeen lähdekoodiin. Arduino-järjestelmien ja ohjelmoinnin taustalla on laaja kehittäjäyhteisö [36].

Arduino-mikrokontrollerialustoja saa usealle eri mikrokontrolleriperheelle, kuten taulukosta 3.2 nähdään. Yleiskäytössä ovat Uno- ja 101-versiot. Mikäli halutaan pienikokoinen ja vähän virtaa kuluttava laite eikä tarvita paljon I/O-yhteyksiä, niin silloin voidaan käyttää

Nano-versiota [36]. Alla olevassa taulukossa 3.2 on lueteltu eri tuotekategorioiden liittyviä Arduino-tuotteita. Tummennetut tuotteet ovat elektroniikkakortteja. Shield tarkoittaa tuon kortin päälle kiinnitettävää lisäkorttia, joka antaa jonkin lisäominaisuuden peruskortille. Kit on tuotepaketti tiettyjen harjoitusten tekemiseen [36].

Taulukko 3.2. Arduino-tuotteiden käyttökategoriat

Entry level	UNO	Leonardo	101	ROBOT	ESPLORA	MICRO	NANO	MINI	
	MKR2UNO Adapter Shield	Starter Kit	Basic Kit	LCD Screen					
Enhanced Features	MEGA	ZERO	DUE	MEGA ADK	PRO	MO	MO PRO	MKRZERO	PRO MINI
	Motor Shield	USB HOST Shield	Proto Shield	MKR Proto Shield	4 Relays Shield				
	Mega Proto Shield	MKR Relay Proto Shield	ISP	USB2Serial Micro					
	USB2Serial Converter								
Internet of Things	YÚN	Ethernet	TIAN	Industrial 101	Leonardo Eth	MKRFOX 1200	MKR1000		
	YÚN Mini	WiFi Shield	WiFi 101 Shield	YÚN Shield	Wireless SD Shield				
	Wireless Proto Shield	Ethernet Shield V2	GSM Shield V2	MKR1000 Bundle Kit					
Education	CTC 101 Kit								
Wearable	GEMMA	LILYPAD ARDUINO USB	LILYPAD ARDUINO Main Board	LILYPAD ARDUINO Simple					
	LILYPAD ARDUINO Simple Snap								
3D Printing	Materia 101 Kit								

Edellä olevasta taulukosta on poimittu yksi protokortti jokaisesta tuotekategoriasta ja esitelty niiden yleisimmät tekniset tiedot taulukossa 3.3. Taulukosta nähdään erilaisuuksia niiden rakenteessa, ja siten voidaan valita oikea mikrokontrolleri haluttuun sovellukseen.

Taulukko 3.3. Arduinon mikrokontrollerimallit ja muut tekniset tiedot

Card name	Micro-controller	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital I/O/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6 / 0	14 / 6	1	2	32	Regular	1
MEGA	ATmega2560	5 V / 7-12 V	16 MHz	16 / 0	54 / 15	4	8	256	Regular	-
Ethernet	ATmega328	5 V / 7-12 V	16 MHz	6 / 0	9 / 4	1	2	32	-	1
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1 / 0	3 / 2	0.512	0.512	8	Micro	-

Aloitustason (Entry-level) sovellusalustoista Uno-malli on pienempiin projekteihin suunniteltu yleiskäyttömalli. Kehittyneemmistä sovellusalustoista (Enhanced Features) MEGA-malli on isompiin projekteihin soveltuva. Siinä on runsaasti analogia- ja digitaali-tuloja sekä -lähtöjä. Lisäksi siinä on huomattavasti enemmän muistikapasiteettia isompia ohjelmia varten. Esineiden Internet (Internet of Things) sarjan piirikorteista Ethernet-malli on suunniteltu tietoverkkoon liitettäväksi. Puettavan sovelluksen (Wearable) suunnittelualustoista Gemma-malli on suunniteltu vaatteissa mukana kuljetettavaksi. Siinä on vähemmän I/O-portteja ja muistia. Gemman piirikorttiin voidaan liittää 2-3 kpl I/O-antureita ja 3.7 V Li-Ion akku. Liikuttaessa pitempiä matkoja laite on mahdollista kytkeä päälle virtakytkimestä ja pois päältä.

Arduinot myydään USA:ssa tuotenimellä Arduino UNO ja USA:n ulkopuolella nimellä Genuino UNO. Monista saatavilla olevista malleista UNO on parhaiten dokumentoitu ja se sopii hyvin ensimmäiseksi mikrokontrollerialustaksi kestävyytensä ja helpon ohjelmoitavuutensa takia. Kuvassa 3.12 on molempien mikrokontrollerikorttien kuvat [36].



Kuva 3.12. Arduino UNO - ja Genuino UNO -mikrokontrollerikortit [36]

Arduino UNO ja samoin Genuino sisältävät seuraavia teknisiä ominaisuuksia. Mikrokontrollerina on ATmega328P, jossa on runsaasti digitaalisia ja analogisia I/O-liitäntöjä. Niistä osa voi toimia PWM-ulostulona (Pulse Width Modulation). UNO toimii 16 MHz:n taajuudella ja siitä löytyvät lisäksi virtaliitin ulkoiselle virtalähteelle, reset-kytkin ja USB-liitäntä [36]. Taulukosta 3.4 selviää tarkemmat Arduino UNO:n tekniset tiedot.

Taulukko 3.4. Arduino UNO:n mikrokontrollerin tekniset tiedot

Microcontroller (Atmel)	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limit values)	6-20 V
Analog Input Pins	6
PWM Digital I/O Pins	6
Digital I/O Pins	15 (6 provide PWM output)
DC Current for 3.3 V Pin	50 mA
DC Current per I/O Pin	20 mA
SRAM	2 KB
Flash Memory	32 KB, bootloader uses 0.5 KB
EEPROM	1 KB
LED (built in)	13
Width	53,4 mm
Length	68,6 mm
Weight	24 g

Mikrokontrollerialustan ohjelmointi tapahtuu IDE-ympäristössä, joka toimii kaikissa tärkeimmissä käyttöjärjestelmissä. Ohjelmointiin käytetään USB-liitäntää, joten sovellusten testausvaiheessa UNO toimii USB-liitännän antamalla jännitteellä. Ohjelmointikieli on

lähellä C++ -kieltä. Arduinon ohjelmointikieli jakaantuu kolmeen pääalueeseen: rakenne (structure), arvot (variables and constants) ja funktiot (functions) [37]. Ohjelmia kutsutaan luonnoksiksi eli sketseiksi (sketches). Ohjelmat kirjoitetaan IDE:ssä tekstieditorilla ja ne tallennetaan loppuliitteellä *.ino. Arduinon IDE-ohjelmiston uusin versio on 1.8.3 [35], joka toimii kaikilla Arduinon mikrokontrollerialustoilla Windowsissa, Mac OS X:ssa ja Linuxissa.

Arduinon voi hankkia myös aloituspakkauksena (Starter Kit tai Basic Kit), jolloin Arduinon mukana tulee valittu joukko komponentteja. Niiden avulla voi tehdä 15 erilaista harjoitustyötä, jotka on myös ohjeistettu ja harjoituksista (10 kpl) löytyvät lisäksi opetusvideot Youtube:sta. Arduino-korteilla voidaan lukea usealtakin sensorilta tai kytkimeltä tietoja, joilla sitten vastaavasti ohjataan esimerkiksi merkkivaloja, relettä tai moottoria. Arduino-piirilevyn päälle voidaan liittää useita erilaisia tytärkortteja (shield) kuten Ethernet-, Wi-Fi- tai GSM-käyttöä varten. Tytärkortit on valmistettu niin, että ne voidaan kiinnittää Arduinokortilla oleviin pinnirimoihin painamalla. Alla olevassa kuvassa 3.13 on Ethernet-kortti (Ethernet Shield). Arduinolla voidaan siten muodostaa esimerkiksi kodin hälytysjärjestelmä. Siinä tapahtuvan liike- tai muun sellaisen hälytyksen takia lähetetään omistajalle tietoverkon kautta hälytysviesti esimerkiksi omaan matkapuhelimeen [36].

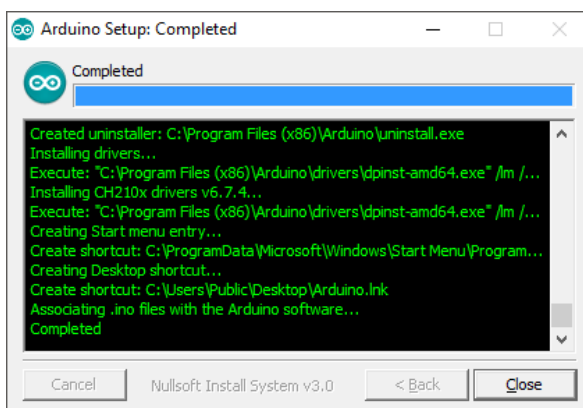


Kuva 3.13. Arduinon liitettävä Ethernet-kortti [36]

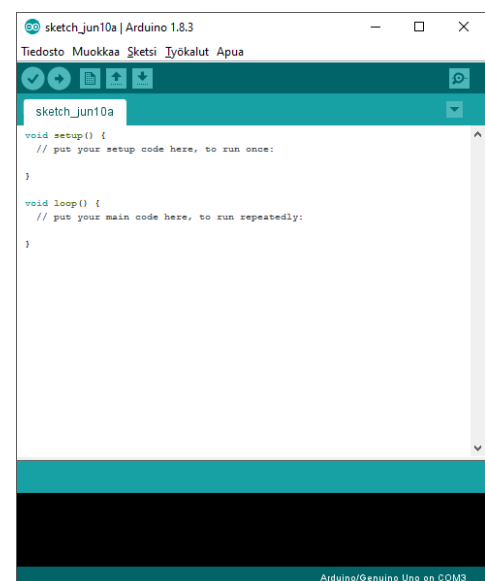
Sen jälkeen, kun Ethernet-kortti on kiinnitetty Arduinon päälle, niin Arduino liitetään USB-kaapelilla tietokoneeseen ja tehty sketsi (ohjelma) ladataan mikrokontrolleriin. Ohjelmankehitysympäristöön (IDE) voidaan hakea erilaisia ohjelmointikirjastoja, riippuen siitä mitä ominaisuuksia tarvitaan. Kirjastoja löytyy todella paljon. Ethernet-korttia ohjelmoitaessa tarvitaan Ethernet-kirjasto (Ethernet Library). Sitten on mahdollista ohjelmoida esimerkiksi yksinkertainen WebServer-palvelu, vaikka lämpötilojen esittämiseen netin kautta. Ethernet-kortilla olevaa SD-korttiliitäntää varten tarvitaan SD-kirjasto (SD Library). Kirjaston avulla on mahdollista käsitellä SD-kortin muistia ja luoda kortille hakemistoja tai tiedostoja sekä kirjoittaa tiedostoihin tietoa [36].

Arduino Software (IDE)

Arduino Software toimii kaikilla Arduino-versioilla ja se on saatavilla Windowsille, Mac OS X:lle ja Linuxille. Windows-versio 1.8.3 vaatii noin 400 MB levytilaa. Kun Arduino Softwaren Windows-versio on asennettu onnistuneesti, tulee lopuksi kuvan 3.14 mukainen ikkuna.



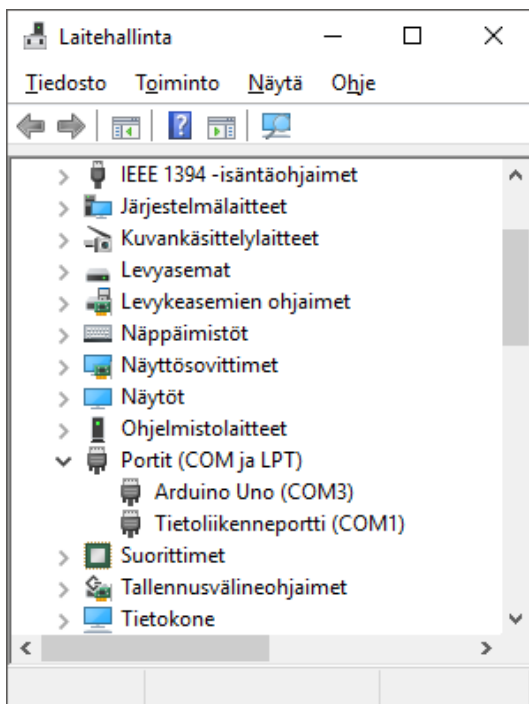
Kuva 3.14. Arduino Software v. 1.8.3 asennettuna



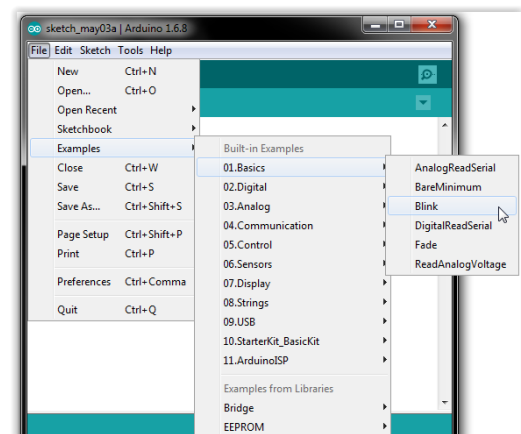
Kuva 3.15. Arduinon tekstieditori

Kun Arduino Software (IDE) käynnistetään, niin silloin tulee ensiksi näkyviin kuvan 3.15 mukainen sketch-ohjelmien kirjoittamiseen tarkoitettu tekstieditori, ja käytettävä COM-portti. COM-portti täytyy kuvan 3.16 mukaisesti tarkistaa Windowsin Laitehallinta-

ohjelmalla (Device Manager). Tässä tulee hieman opiskeltua myös tietokonetekniikkaa sekä Laitehallinta-ohjelman käyttöä, jota tarvitaan melko usein vianmäärityksessä. Arduino-mikrokontrollerikortti täytyy olla kytkettynä USB-porttiin, jotta nähdään, minkä COM-portin Windows on sille määritellyt. Kun Arduinolle valittu portti on selvillä, liitetään IDE:n Tools-valikosta Arduino kyseiseen porttiin. Arduinon laiteajurit Windowsia varten tulee olla myös asennettuna ennen kuin portti löytyy.



Kuva 3.16. Arduinolle määritelty COM-portti



Kuva 3.17. File-valikon rakenne

File-valikon alta löytyvästä Examples-valikosta löytyy paljon erilaisia esimerkkiohjelmiä kuvan 3.17 mukaisesti. Valikossa on niin digitaalisen kuin analogisen puolen sekä tietoliikenteeseen ja sensoreihin liittyviä esimerkkiohjelmiä. LED:in vilkutusohjelma (Blink) on yksi usein esitetty ohjelman malliesimerkki. Siinä tulee hyvin esille sketsi-ohjelman rakenne ja miten I/O-portteja ohjelmassa käsitellään.

```

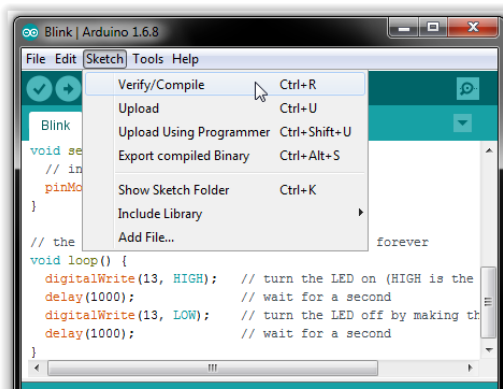
/* This example code is in the public domain. */

int pushButton = 2;    //digital pin 2 has a pushbutton attached to it. Give it a name.
void setup()           //the setup routine runs once when you press reset.
{
  Serial.begin(9600); //initialize serial communication at 9600 bits per second.
  pinMode(pushButton, INPUT); // make the pushbutton's pin an input
}
void loop()            // the loop routine runs over and over again forever
{
  // read the input pin and
  int buttonState = digitalRead(pushButton); // print out the state of the button
  Serial.println(buttonState);               // print to the serial monitor
  delay(1);                                  // delay in between reads for stability
}

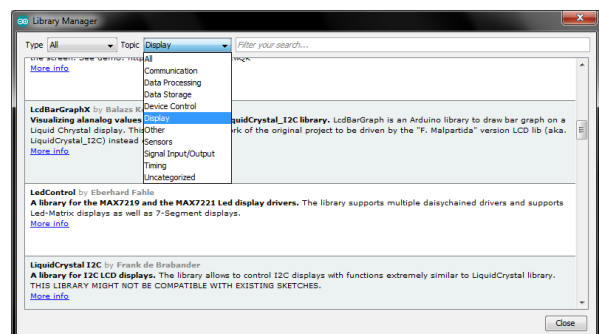
```

Kuva 3.18. Arduinon esimerkkiohjjelma

Arduinon sketch-ohjelmien mallirakenne on esitetty kuvassa 3.18. Ennen setup-osaa merkitään yleensä ohjelman muuttujat. Setup-osassa määritellään tapahtumia, jotka suoritetaan vain ohjelman käynnistysvaiheessa. Sitten loop-osassa ollaan ikuisessa silmukassa koko ohjelman suorituksen ajan ja siellä tehdään kaikki muut toimintaan kuuluvat tehtävät. Ohjelman kääntäminen ja ohjelman tallennus mikrokontrolleriin tapahtuvat kuvan 3.19 mukaisesti Sketch-valikon kautta. Kääntäminen tapahtuu *Verify/Compile* -valikon kautta ja tallennus mikrokontrolleriin *Upload*-valinnalla. Ajurikirjaston lisäys tapahtuu kuvassa 3.20 *Sketch/Include Library* -valinnan kautta, jolloin seuraavaan ikkunaan ilmestyvät hakukriteerien mukaisesti eri komponenttien kirjastot.



Kuva 3.19. Sketch-valikko



Kuva 3.20. Ajurikirjaston hakuvalikko

Ajurikirjastoja voidaan etsiä eri aihealueiden perusteella. Silloin kun aloitetaan uudenlaista ohjelmointiprojektia, niin usein joudutaan lataamaan uusia ajureita kirjastoon. Ajureita löytyy todella laajalle komponenttivalikoimalle ja avoimen lähdekoodin yhteisö tuhansine koodaajineen tuottaa jatkuvasti lisää ajurikirjastoja.

Arduino-järjestelmien koulutusmateriaali

Arduino toimii avoimen lähdekoodin periaatteella ja tarjoaa verkkosivuillaan hyvin tukimateriaalia laitteidensa opiskeluun. Arduinon avoimen lähdekoodin yhteisöön kuuluu tuhansia harrastajia, opiskelijoita ja asiantuntijoita, jotka ovat kehittäneet laajan tukimateriaalin Arduinolla ohjelmoitavien tehtävien suunnitteluun ja opiskeluun. Arduino-foorumilta voi hakea vastauksia kysymyksiin, esittää kysymyksiä tai opiskella uusia asioita ohjelmoinnista tai erityyppisistä elektroniikkakorttien sovellusratkaisuista [35].

Yhteenveto Arduinon laitteista

Arduino on helppo ottaa käyttöön ja sen ohjelmointikäyttöliittymä on miellyttävän helppokäyttöinen. Tätä laitesarjaa voi hyvin suositella niin elektroniikan - kuin elektroniikan komponenttienkin toiminnan tutkimiseen sekä myös vaativampien sovellusten tekemiseen. Proto-kortille voi tehdä virtapiirejä ja Arduinon voi melko nopeasti tehdä pienen ohjelman, jonka avulla tutustutaan eri komponenttien ominaisuuksiin. Mikrokontrolleripohjaisen ohjelmistonkehityslaitteiston käyttöön ja tutustumiseen pääsee tällä laitteistolla pienellä vaivalla. Verkkosivuilta löytyy runsaasti lisäinformaatiota ja käyttäjillä on aktiivinen toimiva foorumi, josta voi lukea tietoja erilaisista ohjelmointi- ja kytkentäratkaisuista sekä kysellä lisätietoja tarvittaessa. Ajurikirjastot ovat laajat eri komponenteille ja tietoliikenneyhteyksille.

4 Sulautetut järjestelmät opetuskäytössä

Tässä luvussa esitellään, miten sulautettujen järjestelmien opetusaluustoja on yleisesti käytetty opetuksen tukena eri kouluasteilla. Lukujen 4.1, 4.2 ja 4.3 alussa esitellään yleisesti LEGO-robottien, Matrix-järjestelmien ja Arduino-järjestelmien käyttömahdollisuuksia opetuksessa. Jokaisessa luvussa on lisäksi mukana esimerkkien kautta noilla järjestelmillä opetuksessa tehtyjä tutkimuksia eri oppilaitoksissa. Edellä kuvattujen järjestelmien välisiä eroja esitellään luvussa 4.4. Luvussa 4.5 tarkastellaan mahdollisuuksia soveltaa LEGO-robotteja sekä Arduinon ja Matrixin sulautettujen järjestelmien opetusaluustoja opetuksessa valtakunnallisen opetussuunnitelman mukaisissa tutkinnon osissa.

4.1 LEGO Mindstorms -robotti

LEGO-robotit ovat edullisia hankkia ja opiskelijat voivat vähilläkin ohjelmointitaidoilla saada robotit liikkumaan, puhumaan tai väistämään esteitä. Robotit ovat oivallinen väline ohjelmoinnin opettamisessa, koska ne konkretisoivat ohjelmoinnin näkyväksi liikkeeksi. LEGO-robottien ohjelmoinnilla voidaan oppia moottorien monipuolisia ohjausmenetelmiä. Robotin moottorit voidaan ohjelmoida kulkemaan eteen- ja taaksepäin eri nopeuksilla, pyörittää paikallaan tai liikuttaa jonkin kuvion mukaisesti. Robotissa olevien ääni-, valo-, ultraääni- ja kosketussensoreiden avulla päästään havainnoimaan laitteen ulkopuolelta tulevia viestejä ja niiden avulla voidaan ohjelmoida robotin toimintaa halutulla tavalla. Eri sensorien lisääminen ohjelmaan kohde kerrallaan auttaa ohjelman rakenteen hahmottamisessa. Eri firmware-versioiden vaihtaminen on robotissa helppoa ja näin päästään ohjelmoimaan erilaisilla ohjelmointikielillä. Lisäksi on mahdollista harjoitella langattoman Bluetooth- ja IR-yhteyden käyttöä ohjelmointitehtävissä. LEGO-robotteja on käytetty jo pitkään opetuksessa eri oppilaitoksissa. Esimerkiksi yliopistoissa on tehty lukuisia tutkimuksia [5,8,26,27,28], joissa on tutkittu LEGO-robottien hyödyntämistä opetuksessa. Ohjelmointiympäristö toimii erilaisissa käyttöjärjestelmissä ja robotti on helppo ohjelmoida USB-väylän kautta.

4.1.1 Esimerkkejä opetuskäytöstä

Vuonna 2001 tekivät Frank Klassner ja Scott D. Anderson tutkimuksen [5] siitä, miten ACM:n (Association for Computing Machinery) opetussuunnitelmassa (ACM:n tietokone-tekniikan opetussuunnitelma 2001) [6] olevien tietotekniikkakurssien yhteydessä voitaisiin hyödyntää LEGO-robotteja. Tutkimuksen tekijät olivat huomanneet, että opiskelijoiden motivaatio oppia tietojenkäsittelyn periaatteita nousi, kun he saivat soveltaa oppimaansa robottien rakentamiseen ja suunnitella niiden ongelmien ratkaisuun ohjelmakoodia [7,8,9,10,11]. Tutkijat halusivat parantaa LEGO-robottia ja tekivät itse siihen lukuisia parannuksia. He paransivat muun muassa robotin firmware-ominaisuuksia, joiden avulla he saivat poistettua muutamia havaittuja puutteita. He valitsivat muutamasta silloin saatavilla olevista robotiikkaperustaisista laitteista juuri LEGO Mindstormsin, koska sen hinta oli edullinen. Sen lisäksi se oli joustava ja monipuolinen moniin opetustarkoituksiin. Se myös kiinnosti ja motivoi opiskelijoita. Lisäksi heillä oli oma ammatillinen kiinnostus tähän robottimalliin ja halusivat nähdä mitä kaikkea sillä voi tehdä [5].

He tutkivat omaa opetussuunnitelmaansa ja totesivat, että lähes 14 eri kurssissa voitaisiin käyttää robotiikkaa opetuksessa. Tärkeimpään 7 kurssin ryhmään he valitsivat, ohjelmoinnin perusteet, algoritmit ja kompleksisuus, ohjelmointikielet, arkkitehtuurit, käyttöjärjestelmät, älykkäät järjestelmät ja verkko-keskeinen tietojenkäsittely. Heidän tutkimuksensa perusteella Mindstormsin perussarja ei täytä kaikilta osin niitä vaatimuksia, joita tarvitaan noiden kurssien toteuttamisessa. He huomasivat silloin, että robotissa ei ollut moottoreissa pyörimisen tunnistavaa sensoria. Lisäksi sitä ei voinut ohjelmoida esimerkiksi seuraavilla ohjelmointikielillä, Java, C++, ja Lisp. Ohjelmamuistin määrä oli myös rajoitettu ja IR-lähetysprotokolla oli esteenä esimerkiksi protokollatekniikoita opetettaessa. Vastapainoksi sarja oli halpa ja siihen saadaan kytkettyä tarpeeksi erilaisia antureita. Puutteista huolimatta voitaisiin robottia käyttää opetuksessa, kunhan käytettäisiin joitakin lisäohjelmistoja [5].

Frank Klassner ja Scott D. Anderson paransivat havaittuja ohjelmointikielten puutteita ohjelmisto-työkalupakilla, jota he kutsuivat nimellä MTM (More Than Mindstorms). Sen avulla korjattiin firmwaren alkuperäistä versiota, jolloin eri ohjelmointikielten käytettävyys parani ja saatiin korjattua IR-protokolla-ohjelmistoa. Sen jälkeen IR-protokollalla voitiin ohjata useampaa kuin yhtä robottia. Näiden muutosten jälkeen tutkijat totesivat, että RCX

on hyvä alusta hyödynnettäväksi käyttöjärjestelmien, tietokone-arkkitehtuurien ja ohjelmointikielten opetuksessa [5].

Kreikkalaiset V. Dagdilelis et.al. [26] tekivät vuonna 2005 tutkimuksen LEGO-robottien käyttämisestä toisen asteen opiskelijoilla. Heikin olivat aikaisemmin huomanneet, että opiskelijoilla on ongelmia löytää ohjelmoinnissa aivan perusohjelmointivirheitä. He halusivat helpottaa sekä opettamista että oppimista ja loivat ensin ohjelmointiympäristön, jossa opiskelijat liikuttelivat esineitä ruudulla. Tämä helpotti aloittelevienkin opiskelijoiden kykyä hahmottaa ohjelmointia. Kun ruudulla liikkuva esine törmää seinään, niin se konkretisoituu ja on helppo ottaa ohjelmoinnissa huomioon eri tavalla. Seuraavassa vaiheessa he ottivat avuksi LEGO-robotit ja valitsivat ohjelmointikieleksi RoboLab-kielen [26].

Opettaessaan ohjelmointia V. Dagdilelis et.al. olivat myös havainneet, että opiskelijoille tuottaa suuria vaikeuksia ymmärtää ohjelmoinnissa mitä tarkoittavat parametrit, toistot ja ehtolauseet. He laativat harjoitustehtäviä, jotka suoritettiin kahdessa luokassa ja mukana oli yhteensä 15 opiskelijaa. Opiskelijat työskentelivät pareittain soveltaen yhteistyötaitoja ja keskustelivat ratkaisuista. Ensimmäinen harjoitus käsitti tutustumisen LEGO-robottiin, infrapunalähettimeen, ohjelman siirtämiseen robottiin sekä termin ”ohjelma” selvittämiseen. Lopuksi muokattiin hieman harjoitusohjelmia. Toisen oppitunnin teemana olivat ohjelmoinnissa Input ja Output-lauseet sekä sensoreihin tutustuminen. Kolmannen harjoituksen tarkoituksena oli harjoitella toistorakenteita. Käytiin myös läpi toistojen hyötyjä ja lopuksi harjoiteltiin erilaisia toistokomentoja. Lopuksi opiskelijoille esitettiin kysymykset, joihin heidän piti vastata ja ratkoa joitakin ongelmia. Johtopäätöksinä opettajat huomasivat, että jo harjoitusten aikana oikea toimiva laite aiheutti huomattavaa motivaation kasvua. Paljon määrittely- ja käyttöongelmia aiheuttivat niin Input- kuin Output-porttienkin ohjelmointi, mutta erityisesti ohjelmointiongelmia tuli Input-porttien kanssa. Vaikka osalla opiskelijoista ei ollut vaikeuksia ymmärtää ohjelman loogista etenemistä, niin hekin unohtelivat perusohjelmointiohjeita ja -määrittelyitä, jolloin heidän piti kysyä opettajilta usein neuvoja. Opettajat painottivat kuitenkin, että pieni hyvin motivoitunut opiskelijajoukko selviytyi ongelmien selvittelyistä hyvin [26].

Korealaisessa yliopistossa teettivät korealaiset Seung Han Kim ja Jae Wook Jeon opiskelijoilla LEGO-harjoitukset vuonna 2009 [27]. Kaikki 80 opiskelijaa olivat huippuopiskelijoita matematiikassa ja he olivat suorittaneet ANSI-C -ohjelmointikurssin (perus-C-ohjelmointi). Heillä ei monellakaan ollut aikaisempaa kokemusta sulautetuista järjestelmistä. Heidät jaettiin viiden opiskelijan ryhmiin ja heille annettiin 3 kotitehtävää ja 3 projektia, sekä teorialuennot. Kotitehtävässä piti robotin liikkua eri kuvioita tehden ja päästään ääniä sekä piirtäen kynällä jälkiä liikkueensa. Ensimmäisessä projektissa opiskelijoiden piti rakentaa robotti ja ohjelmoida se niin, että se osaa poistua labyrintista. Toisessa projektissa piti rakentaa skannerirobotti, jonka tuli lukea pisteitä ja tulostaa paperille pisteistä luotu kuvio. Kolmas projekti oli opiskelijoiden mielestä mielenkiintoisin. Siinä piti rakentaa ja ohjelmoida sumorobotti, jonka tehtävänä oli työntää toinen robotti pois kehästä. Kaikista projekteista ja kotitehtävistä kyseltiin opiskelijoilta mielipiteitä niiden vaikeudesta ja mielenkiintoisuudesta. Vaikka LEGO-robotti aluksi vaikutti enemmän lelulta, osoittautui se lopulta tehokkaaksi välineeksi opiskella sulautettua järjestelmää ja sen ohjelmointia. Opiskelijoiden mielestä kurssi oli raskas, mutta mielenkiintoinen ja se antoi hyvän kuvan siitä, miten sulautettu järjestelmä toimii [27].

Baskimaan yliopistossa, Automation and Engineering Systems Department (DISA) -osastossa työskentelevät E. Irigoyen et.al. [28] ovat tehneet vuonna 2013 selvityksen siitä, miten halvoilla suunnittelualustoilla opiskelijat voivat suunnitella monipuolisia sovelluksia nopealla aikataululla. Tutkijoiden mukaan on kaksi syytä, miksi näitä halpoja alustoja otetaan mielellään mukaan opiskelijaprojekteihin. Ensiksi ne ovat monipuolisia ja edullisia hankkia, sekä opiskelijoiden yleinen tietämys tällaisista laitteista on kasvanut. Toiseksi ne mahdollistavat, että opiskelijat voivat suunnitella nopeasti jonkin järjestelmän prototyyppi-vaiheeseen ja testata sen toiminnallisuuden. Automaatio-osaston opiskelijoilla oli mahdollista tehdä projekteja sekä vanhemmalla LEGO-robotilla NXT 2.0 että uudemmalla EV3:lla. LEGO-robotit oli valittu sen takia, että ne olivat edullisia ja niitä voidaan ohjelmoida monilla ohjelmointikielillä. Lisäksi niissä voidaan käyttää laajaa valikoimaa erilaisia sensoreita ja moottoreita. Opiskelijat tekivät kaksi sovellusharjoitusta, joista ensimmäinen käsitteli teollisuuden havainnointijärjestelmiä. Opiskelijat opiskelivat joko teollisuusjärjestelmien tai teollisuusautomaation parissa. Jokaiselle ryhmälle annettiin LEGO-robotti NXT

2.0, jonka he saivat itse rakentaa omanlaisekseen. Robotin tuli lukea valkoista viivaa alustalla ja liikkua sen mukaan. Lisäksi sen piti lukea ultraäänisensorilla etäisyyksiä ja väistää esteitä. Opiskelijoiden tuli ohjelmoida robotti niin, että nämä eri ominaisuudet otettiin huomioon. Lopputuloksena oli, että tutkimukseen osallistuneet opiskelijat saivat parempia arvosanoja. He hyödynsivät näissä projekteissa laajasti muissakin aineissa saatuja tietoja ja hyötyivät näistä ohjelmointi- ja suunnittelutaidoista taas muissa kursseissa [28].

Edellä olevan perusteella voidaan nähdä, että LEGO-roboteilla on jo pitkät perinteet opetuksessa erilaisten kurssien yhteydessä. Tutkimuksissa [26,27,28] todettiin, että LEGOjen käyttö ohjelmoinnissa antoi opiskelijoille motivaatiota ja innostusta. Vuonna 2009 lanseerattiin RCX-keskussyksiköstä uusi versio NXT [16], jonka myötä ovat parantuneet sekä ohjelmointimahdollisuudet että laitteen muut ominaisuudet. Nykyään esimerkiksi moottoreista löytyy pyörimisen tunnistavat sensorit ja ohjelmointi onnistuu USB 2.0 -väylän kautta. Firmwaren helppo vaihtaminen mahdollistaa eri ohjelmointikielten käytön.

4.2 MatrixTSL-järjestelmät

Aiemmalta nimeltään tunnettu Matrix Multimedia toimii nykyään nimellä MatrixTSL (Matrix Technology Solutions Ltd). Matrix on valmistanut ja kehittänyt tuotteitaan vuodesta 1993. Yrityksen tuotevalikoima on laaja ja kaikki tuotteet on suunniteltu opetuskäyttöä varten. Kokonaisia laitepaketteja opetuspaketteineen löytyy eri sovellusten ohjelmointiin kuten CAN-, USB-, AudioDSP-, Bluetooth-, Internet-, Zigbee-, GSM- ja RFID-sovelluksiin. Sovellusalustoja on saatavilla erilaisilla mikrokontrollereilla varustettuna kuten PIC-, AVR-, Arduino-, dsPIC-, tai ARM-mikrokontrollerilla. Myös opettajalle tulee laitepakettitilauksen mukana valmiit noin 50 sivuiset opetuspaketit noihin sovelluksiin. Edellisistä nähdään, että Matrixin tuotteet yhdessä vuokaaviotyypin Flowcode®-kehitysympäristön kanssa soveltuvat hyvin kokonaisten järjestelmien opettamiseen ja opiskeluun. Matrixin laitteiden käyttö opetuksessa on hyödyllistä myös silloin, kun keskitytään enemmän mikrokontrollerin sisäiseen ohjelmointiin. Valikoimaa täydentää E-blocks®-sarja, joka sisältää noin 50 kpl elektroniikkakortteja. Flowcode-kehitysympäristössä on kaikkiin lisäkortteihin jo valmiiksi tehtynä komponenttimakroja, jolloin ei jokaista ohjel-

mointitehtävää tarvitse pikkutarkasti ohjelmoida. Tämä helpottaa opiskelijoiden pääsyä ohjelmointiin sisälle. [3].

Vuonna 2008 tuli mukaan Locktronics-laitteet, joilla opetetaan sähköön ja elektroniikkaan liittyviä asioita. Automaatiojärjestelmiä voidaan opiskella ja rakentaa MIAC-laitteilla (Matrix Industrial Automotive Controller). FormulaAllCode-ohjelmointikieltä käytetään robottien ohjelmoimiseen ja yhteydet robotteihin toimivat Bluetooth-yhteydellä. FormulaAllCode on Windows, Android-, OS X -, ja Raspberry Pi -yhteensopiva. Electronic Workstation -laitteistossa on käytettävissä kaikki tärkeimmät elektroniikan opettamiseen liittyvät laitteet. Siirrettävänä työasemana se sopii elektroniikan prototyyppien testaamiseen. Vuonna 2012 tuli yrityksen valikoimiin Automatics-tuotteet, jotka on suunniteltu pneumaattikajärjestelmien opettamiseen [3].

4.2.1 Esimerkkejä opetuskäytöstä

A. Dabroom et. al. [43] esittelivät vuonna 2013 tutkimuksen, jossa he suunnittelivat mikrokontrollerin ohjelmointikurssin oheislaitteineen. Heidän kurseilleen tuli paljon opiskelijoita, joilla oli hyvin eritasoiset tiedot ja taidot mikrokontrollereista, liityntärajapinnoista ja niiden ohjelmoinnista. Heidän mielestään myös kurssin laitteiston tulisi olla helppokäyttöinen ja joustava sekä opiskelijoille mahdollisimman mielenkiintoinen. He suunnittelivat dsPIC-mikrokontrolleripohjaisen sovellusalustan, jossa oli useita erityyppisiä I/O-liityntäportteja. Alustaan he suunnittelivat itse 6 kpl E-blocks -elektroniikkakortteja erilaisien harjoitusten tekemistä varten. Lisäkorteissa on erityyppiset liityntärajapinnat ja ne on helppo liittää mikrokontrollerikortin eri portteihin. Koska osalla opiskelijoista on ennakkoluuloja ohjelmointia kohtaan, halusivat kurssin suunnittelijat rohkaista ja auttaa heitä. Tätä varten he valitsivat dsPIC-mikrokontrollerin, jossa voitiin käyttää Flowcode-vuokaavio-ohjelmointia. Suunnittelijoiden mielestä Flowcode on helppo oppia sekä käyttää ja sillä saa nopeasti ohjelmoitua monimukaisempiakin sovelluksia. Aloituskurssin tehtävät opiskelijat tekevät Flowcodella, jotta he saavat varmuutta ohjelmointiin. Kurssin ohjelmointitehtäviksi on valittu mm. ledin vilkutus, releen toiminta, kytkimen käyttö, 7-segmentti-näytön ohjaus, LCD-näytölle tulostus, DC-moottorin käyttö ja sen pyörimisnopeuden mittaaminen. Jatkokurssilla otetaan käyttöön korkeamman tason C-kieli tai vastaava. Aloituskurssin tarkoituksena on opettaa opiskelijoille perusteet mikrokontrollerin tekniikasta ja ohjelmoinnista sekä

syventää heidän osaamistaan kokonaisen järjestelmän rakentamisesta ja ohjelmoimisesta näillä itse suunnitelluilla laitteilla [43].

Haasteena näillä kursseilla oli se, kuinka saada opiskelijat motivoituneiksi ja olemaan aktiivisia tunneilla. Tätä yritettiin parantaa siten, että opiskelijat tekivät tiimeissä useita harjoittelutehtäviä, joita opettajat ohjasivat tehtävä kerrallaan. Lisäksi opiskelijat ohjeistettiin valitsemaan tiimeihinsä eri osa-alueen osaajia, jotta tietoa voidaan jakaa. Opiskelijoille karttui tällä tavalla luottamusta omiin taitoihinsa. Lopuksi opiskelijat tekivät pienen sovel-lusprojektin ryhmissä sekä siitä raportin. Opettajien tavoitteena oli myös rohkaista opiske-lijoita jakamaan tietoa tiimeissä sellaisista asioista, joita olivat jo oppineet ja olemaan ak-tiivisia kysymään epäselvistä asioista joko tiimin jäseniltä tai opettajalta. Suunnittelijat halusivat suunnitella ohjelmointikortista sellaisen, että sitä voidaan päivittää helposti ilman että E-blocks -kortteja tarvitsee muunnella. E-blocks -korteilla voidaan saada säästöjä, kun eri kursseilla voidaan käyttää samoja kortteja ja opettaja voi valita vain ne kortit mitä ky-seisellä kurssilla tarvitaan. Järjestelmä soveltuu hyvin ”opiskele tekemällä” -malliin ja opiskelijoiden motivaatio paranee. Opiskelijoilla on myös mahdollisuus jatkokursseilla itse suunnitella lisää kortteja eri käyttötarkoituksiin ja näin järjestelmää on mahdollista käyttää monenlaisiin elektroniikka- ja tietokonealan kursseihin [43].

Suunnittelijoiden mukaan näillä järjestelyillä päästään vanhasta aikansa eläneestä mikro-kontrollerialustasta eroon, jossa käytettiin assembly- (matalan tason ohjelmointikieli) ja C-kieltä. Uusilla järjestelmillä ja uudella modernilla ohjelmointikielellä saatiin opiskelijoiden opiskelumotivaatiota nostettua. Uudella alustalla ja ohjelmointikielellä tuli mikrokontrole-rikurssista opettamisen ja oppimisen kannalta tehokkaampi ja saatiin parempia oppimistu-loksia. Uusi laitealusta on myös helppo päivittää, jotta pysytään uusien tekniikoiden muka-na [43].

Romanialaiset D. Mihai ja C. Caramida tekivät vuonna 2010 yliopistollaan opetusjärjes-telmän [44], jolla voidaan sumean logiikan avulla lukea säiliön sisälämpötilaa. Ohjauslo-giikan laitteistoksi he valitsivat suosituilla PIC16F877-mikrokontrollerilla varustetun sovel-lusalustan [40] ja siihen liityntäkorteiksi valmiit E-blocks -kortit. Tuosta korttisarjasta oli käytössä LCD-, Sensori-, DAC-, Graafinen näyttö- ja LED-kortti. Tämä laitesarja on edul-

linen sekä suosittu. Ohjelmointiympäristöksi valittiin graafinen ohjelmointikieli Flowcode. Tällä järjestelyllä suunnittelijoiden mielestä saatiin joustava ja nopeasti käyttöön otettava ohjausjärjestelmä lämpötilan mittaukseen, joka on sekä helppokäyttöinen että edullinen. Suunnittelijoiden mielestä näin saatiin hyvä opetusala sulautettujen järjestelmien opettamiseen ja heidän mielestään näillä laitteilla onnistui säiliön lämpötilan mittaaminen sumealla logiikalla oikein hyvin [44].

Toisessa artikkelissaan D. Mihai [45] esitteli vuonna 2016 ilmanpaineen mittaamista säiliöstä valmiin kaupallisen PIC16F877-mikrokontrollerilla varustetun ohjausjärjestelmän ja Flowcode-ohjelmiston avulla. Järjestelmään kuului teollisuudessa käytettävä ilmanpaine-tunnistin, digitaalinen ohjausyksikkö, pienikokoinen auton ilmanpaine kompressori sekä painesäiliö. Sekä teollisuudessa käytettävä digitaalinen ohjausyksikkö, että mikrokontrollerin ohjausjärjestelmä ohjaavat kompressoria, joka pitää säiliön ilmanpaineen tasaisena. Tämänlaisen toimintaan perustuvat auton renkaan ilmanpaineen valvontajärjestelmät. Tässä tutkimuksessa käytettiin ilmanpaineen mittaamiseen algoritmeja, jotka noudattivat sumeaa ohjauslogiikkaa ja ohjelmisto oli toteutettu Flowcodella. Järjestely onnistui hyvin ja lopputuloksissa laskennallisten - ja reaalityönten ero oli niinkin pieni kuin 2 %. Osa tuosta virheestä johtuu vain kahdeksalla bitillä tehtyjen A/D-mittausten tarkkuudesta sekä muista mittausteknisistä asioista. Tällä järjestelyllään suunnittelija todisti, että Flowcode on tehokas työkalu monimutkaistenkin järjestelmien toteuttamisessa. Ohjelmiston simuloituympäristö antoi myös hyvin realistiset lopputulokset verrattuna oikeassa ympäristössä saatuihin tuloksiin. Tätä järjestelmää voidaan opetuksessa käyttää moniin eri harjoitustilanteisiin kuten mikrokontrollerin ohjelmointiin, teollisuuden ohjainyksikön säätöihin sekä laitteistojen ja ohjelmistojen rajapintojen suunnitteluun [45].

Näiden muutamien esimerkkien kautta nähdään hyvin, että MatrixTSL:n mikrokontrollerialustat, E-blocks lisäkortit ja vuokaavio-ohjelmointityökalu Flowcode soveltuvat hyvin monimutkaisempienkin sovellusten tekemiseen. Niillä sovelluksen suunnittelu aika nopeutuu ja yksinkertaistuu. Valmiin prototyypin käyttöönotto tapahtuu nopealla aikataululla ja näin ollen suunnittelun tehokkuus kasvaa. Opetuskäytössä näillä tuotteilla on mahdollista tehdä lähes loputon määrä erilaisia harjoituksia.

4.3 Arduino-järjestelmät

Arduinon kehitys perustuu vuonna 2003 Italiassa Ivrea Interaction Design Institute -yliopistossa aloitettuun Wiring-projektiin [33]. Kehittäjinä olivat Hernando Barragán, Brett Hagman ja Alexander Brevig. Toisaalta Wiring-projekti perustuu taas Processing-ohjelmointikieleen [34], joka on alusta alkaen suunniteltu ensimmäiseksi ohjelmointikieleksi taide-painotteiseen ohjelmointiin. Ohjelmointitehtävä aloitetaan Processing-ohjelmointiympäristössä piirtämällä, josta sitten saadaan nopeasti visuaalinen tai toiminnallinen ohjelma. Ne ohjelmoijat, joilla ei ole aikaisempaa ohjelmointikokemusta ovat olleet tyytyväisiä tähän ohjelmointikieleen ja -tapaan. Processing-kielen suunnittelun aloittivat Ben Fry ja Casey Reas vuonna 2001 ja se toimii Windows- OS X - sekä GNU/Linux-alustoilla. Arduinoja on käytetty eritasoisissa oppilaitoksissa opetuskäytössä laajalti jo vuosia.

Arduino on erityyppisiin mikrokontrollereihin perustuva mikrokontrollerikorttiperhe, jonka ohjelmistot ovat avoimen lähdekoodin ohjelmia [36]. Arduinot ohjelmoidaan yleensä USB-väylän kautta, mutta löytyy myös Bluetoothin kautta ohjelmoitavia sovellusalustoja. Arduinon yleisyyttä on lisännyt niiden taustalla toimiva laaja yhteisö, joka kehittää ja julkaisee nettiin uusia sovelluksia. Käyttökohteita eri Arduino-alustoille on runsaasti (Taulukko 3.2). Sovellusalustasta riippuen on Arduinolla mahdollista tehdä vaatiiviakin sovellusprojekteja nopealla aikataululla. Järjestelmään on saatavilla paljon opiskelumateriaalia ilmaiseksi. Tutkimuksessaan A. A. Galadima toteaa [29], että Arduino valitaan usein opetusvälineeksi sen halpuuden ja helppokäyttöisyyden takia. Sitä on helppo ohjelmoida ja se sopii yleisimpiin käyttöjärjestelmiin. Tutkimuksen [29] mukaan Arduinosta on tullut erittäin suosittu opiskelijoiden keskuudessa. Jopa sellaisetkin opiskelijat, joilla ei ollut paljon kokemusta ohjelmoinnista tai elektroniikasta, innostuivat Arduinosta. Käyttöliittymän toimivuus ja koko ohjelmointikonaisuuden selkeä rakenne puoltavat laitteiston käyttöä opetuksessa. Kirjastotuki eri komponenteille on erinomaisen hyvä ja kirjaston ajureiden lisääminen sekä käyttö onnistuvat helposti. Verkkosivuilla oleva hyvä tukimateriaali on suurena apuna hyvien esimerkkien kautta ja virheiden etsinnässä. Helpon ohjelmoitavuutensa ja monipuolisen sovellusalustavalikoiman takia voidaan sanoa, että vain mielikuvitus on rajana sille, kuinka moninaisia projekteja Arduinolla voidaan tehdä.

4.3.1 Esimerkkejä opetuskäytöstä

Miamin Yliopistossa opetustehtävissä toimiva Peter Jamieson on ottanut sulautettujen järjestelmien kurseille käyttöön erilaisia mikrokontrollerialustoja. Hän esitteli tutkimukseen [30] Arduinon käyttöä opetuksessa. Tutkimuksen kysymyksenä oli, että onko Arduino-alusta sopiva opetettaessa tietotekniikkainsinöörejä ja tietokonetutkijoita sulautettujen järjestelmien kurssilla. Jamiesonin mielestä, vaikka Arduinossa on puutteitakin, on siinä monia hyviä ominaisuuksia opetuskäyttöön. Laaja Arduino-yhteisö, joka toimii avoimen lähdekoodin periaatteella netissä, antaa opiskelijoille erittäin hyvän lähestymistavan Arduino-maailmaan. Hyvä puoli on myös se, että he voivat opiskella netissä yhteisön sivuilla olevia opetusmateriaaleja ja tutustua lukuisiin tehtyihin projekteihin. Tutkijan mielestä haittana on myös se, että opiskelijat voivat hakea vapaasti tietoa ja malliprojekteja avoimen yhteisön nettisivuilta. Tällöin kouluttajan on vaikea arvioida mitä opiskelija todellisuudessa on oppinut ja kuinka paljon hän ymmärtää laitteista ja ohjelmista. Jamiesonin järjestämällä sulautettujen järjestelmien kurssilla oli käytävissä oleva aika jaettu 3 tunnin teoriajaksoon ja 2 tunnin laboratoriotyöskentelyyn. He suunnittelivat 3 prototyyppiä, toteuttivat ne (ohjaaja hyväksyi) ja esittelivät lopuksi kaikki työnsä. Aiempina vuosina opiskelijat käyttivät muun muassa PIC-mikrokontrollerialustaa, mutta niiden kanssa oli enemmän rajoituksia. Jamiesonin mielestä Arduinon myötä opiskelijat ovat voineet tehdä mielenkiintoisempia projekteja. Vaikka heillä on ollut valittavanaan muitakin mikrokontrollerialustoja, niin opiskelijat ovat ylistäneet Arduinoa ja avoimen lähdekoodin nettiyhteisön tarjoamaa tukea, ja siten he ovat valinneet useammin sen projekteihinsa.

P. Jamiesonin ja J. Herdtnerin mielestä [31] Arduino laitteena on opiskelijoille helppo oppia. Heidän mielestään Arduino UNO on parhaimmillaan esimerkiksi bittitason ohjelmoinnissa, silloin kun ohjataan biteillä Input/Output-portteja. Lisäksi opiskelijat voivat hyödyntää itsenäisesti netissä olevaa laajaa Arduino-yhteisön tekemää käyttö- ja ohjelmointiharjoitusmateriaalia. Yhteisön avuliaisuus ja ystävällisyys asettavat myös sitä kautta haasteita ohjaajille ja opettajille, jotka tuottavat Arduinolle tehtäviä. Miamin yliopistossa on Arduino UNOa käytetty muun muassa seuraavien kurssien yhteydessä: *Sähkö- ja tietotekniikan aloituskurssi, Virtapiirit kurssi 2, Sulautettujen järjestelmien suunnittelukurssi, Tietokoneen rakennekurssi* (valinnainen). Kurssien lisäksi opiskelijat käyttävät Arduinoja omien

projektiensa parissa. Oppilaitos tarjoaa opiskelijoille Arduino-laitteet kahteen ensiin mainittuun kurssiin. Jamieson kertoo, että näille laitteille tarvittava käytön opetus on minimaalista. Esimerkkinä hän kertoo, että tuossa aloituskurssissa käydään läpi kaksi 55 minuutin laitteiston käyttöönottoharjoitusta. Harjoituksessa ohjeistetaan, kuinka Arduinoon kytetään LED ja ylösvetovastus. Ohjelmoimalla näytetään kuinka luodaan perusohjelma, joka muodostaa PWM-ohjauksen LED:ille, jolla LED saadaan himmenemään ja kirkastumaan. Tämän jälkeen oletetaan, että opiskelija osaa tehdä itsenäisesti loput harjoitukset. Opiskelijoiden oletetaan hyödyntävän netissä olevia opastustehtäviä ja projekteja ja käyttävän niiden tietoja omissa harjoituksissaan.

Opettajien mielestä kursseilla on tullut esiin muutamia haasteita. Vaikeimpana haasteena he kokevat avoimen lähdekoodin projektien arvioimisen. Niistä on vaikea arvioida mitä opiskelija on itse tehnyt ja oppinut. Valmiita projekteja on netissä paljon ja ne kehittyvät koko ajan. Yksi kurssilla käytetty metodi oli sellainen, että opiskelijan tuli tehdä matalan tason liityntä (interface) mikrokontrollerikortin ja lisälaitteen välille sekä luoda niiden välille tiedonsiirto-ohjelma. Näin opiskelija näki konkreettisesti sen, miten laitteet keskustelivat. Toisessa mallissa opiskelijat saivat käyttää kaikkia mahdollisia valmiita laitteita ja ohjelmia sekä sekoittaa niitä saadakseen halutun projektin aikaiseksi. Heidän tuli käyttää aikaa selvittääkseen eri laitteiden ja ohjelmien toiminnat. Mutta laitteiden tullessa monimutkaisemmiksi opettajat huomasivat, että opiskelijoiden projektityöt alkoivat näyttää kaupallisesti myytäviltä projekteilta ja luopuivat tästä tavasta. Kolmas tyyli oli sellainen, että opiskelijat saivat käyttää mitä tahansa koodia ja tytärkortteja, mutta heidän tuli dokumentoida kaikki omat muutoksensa ohjelmiin ja laitteisiin ja lopuksi esitellä työnsä opettajalle. Toinen haaste koskee projektin läpivientiä. Opiskelijoiden on vaikea ymmärtää sitä, kuinka iso tehtävä koko projektin loppuunsaattaminen on. Opettajat käyttivät kahta metodia. Ensimmäisessä opiskelijat esittelevät suunnittelemansa projektin sekä kirjallisesti että suullisesti. Ohjaaja voi sitten kommentoida sitä ja tehdä siihen muutoksia. Toisessa metodissa kaikki opiskelijat tekevät luokassa samanlaista suunniteltua projektia. Kuitenkin molemmissa metodeissa projektien onnistuminen on ohjaajan kokemuksen varassa.

Tehdyn tutkimuksen tarkoituksena oli tuoda kouluttajille esille hyötyjä ja haasteita silloin, kun käytetään opetuskäytössä mikrokontrolleripohjaisia elektroniikkakortteja harjoitusten

tekoon. Tutkimuksen tekijöiden mielestä, vaikka he tiesivät ja esittelivät järjestelyjen ongelmakohdat, niin silti ei ole vielä löytynyt laitteiden oikeaa käyttötapaa. Opiskelijat mielellään käyttävät näitä laitteita, koska niitä on helppo käyttää ja niillä voi suunnitella hämmästyttävän laajoja projekteja. He voivat myös itse hankkia laitteita niiden halpuuden takia. Ohjaajien mielestä opiskelijat kuitenkin kehittyvät järjestelmien suunnittelussa. Heidän luovuutensa motivoi heitä kohti oikeita suunnittelujärjestelmiä. Ohjaajien mielestä juuri tuo on niin iso asia, että näitä laitteita voi käyttää opetussuunnitelman kursseihin, vaikka edellä kuvattuja riskejäkin on.

MSU-yliopistossa (McNeese State University) työskentelevät W. Albrecht et.al. [32] tekivät tutkimuksen, jossa he esittelevät kuinka yliopisto aikoo käyttää tietokonetekniikan opetussuunnitelmassa mikrokontrolleripohjaisia suunnittelualustoja. Tutkimuksen tarkoituksena oli yhdistää opiskelijoiden monialaiset opiskelut säännöllisesti reaaliaikaisen harjoitukseen ja tämän projektin jälkeen he osaavat laajemmin soveltaa oppimaansa teoriaa tietokonetekniikan käytännön esimerkkeihin. MSU:n tietokonetekniikan opetussuunnitelmaan sisältyy 16 kolmen pisteen kurssia. Tällä hetkellä yhdelläkään kurssilla ei opiskella ohjelmointia suoraan mikrokontrollerikorteilla tai muilla sulautetuilla järjestelmillä. Heidän visionaan oli ottaa yksi laitealusta käyttöön koko tietokonetekniikan opetussuunnitelmassa, joka olisi Arduino. Tavoitteena oli, että opiskelijoiden ohjelmointikokemus laiteistorajapintojen ohjelmoinnissa kasvaa. Tarkoitus oli, että opiskelijat ohjelmoivat mikrokontrollereita ja sulautettuja järjestelmiä sekä oppivat miten ne liittyvät muihin reaaliaikaisen järjestelmiin rajapintojen (interface) kautta. Projektissa oli tarkoitus tehdä yhteistyötä alueen polttoaineisiin keskittyneen kemianteollisuuden sekä maatalous- että ympäristöteknologia-alojen kanssa. Yliopiston opiskelijoiden suunnitelmissa oli kerätä Arduinon avulla anturitietoja teollisuusalueelta ja käsitellä kerättyjä tietoja. Näin opiskelijat oppivat elektroniikasta interface-suunnittelua sekä ohjelmointitaitoja, kun anturitietoja siirretään interfacen kautta mikrokontrollerikortille käsiteltäviksi. Alemman tason tietokonetekniikan kurssilla opiskelijat käyttävät C-ohjelmointikieltä ja heillä tulisi näin ensikosketus Arduinon ohjelmointiharjoituksissa. Harjoituksissa tehtäisiin ohjelmia, jotka liittyvät digitaalisiin I/O-toimintoihin. Niitä voisi olla esimerkiksi ledien sytyttäminen tai sammuttaminen sekä erityyppisistä digitaalisista laitteista tietojen lukua Arduinon. Lisäksi he ohjelmoisivat sen-

soreita ja toimilaitteita, joissa käsitellään analogisia I/O-signaaleja. Tavoitteena olisi esimerkiksi tehdä Arduinon ohjelma, joka kerää siihen liitetystä sensorista reaaliaikaista tietoa, jota käsitellään jollakin algoritmilla tai numeerisella menetelmällä [32].

Edellä olevista esimerkeistä nähdään, että Arduino on hyvin suosittu oppilaitoksissa. Opiskelijoiden opiskelumotivaatio kasvaa, kun oppilaitoksesta löytyy samanlaisia laitteita kuin mitä moni käyttää vapaa-ajallaankin. Opiskelijat selkeästi haluavat opiskella ohjelmointia yhdessä laitteiden kanssa, koska silloin ohjelman tekemät asiat ovat paljon konkreettisempia. Arduinot ovat todella edullisia hankkia sekä monipuolisia. Niitä löytyy valmiina mikrokontrollerialustoina eri sovelluskohteita varten ja lisäkorttivalikoimalla voi vielä laajentaa sovelluskohteita. Ohjelmointiympäristö on nopea oppia ja aktiivinen taustayhteisö maailmalla jakaa omia sovellusratkaisujaan netissä sekä antaa neuvontaa ja tukea.

4.4 Järjestelmien väliset erot

Alla on lyhyesti lueteltu periaatteellisia eroja näiden kolmen järjestelmän rakenteesta ja niiden pääkäyttöalueista. LEGO-robotti on selkeästi suunniteltu liikkuvan laitteen ohjelmointiin. Robotti soveltuu esineiden siirtelyyn ja nostamiseen tiettyä rataa pitkin (Luku 4.1). Edellisten asioiden lisäksi voidaan opetuksessa keskittyä siihen, kuinka robotti osaa väistellä esteitä ja tunnistaa esimerkiksi värejä tavaroita liikutellessaan. Matrixin laaja E-blocks elektroniikkakorttien valikoima [3] on suunniteltu ohjelmointialustan ympärille rakennettavien valmiiden sekä uusien sovellusten ohjelmointiin (Luku 4.2). Opetuksessa on näiden lisäkorttien avulla mahdollista saada nopeasti aikaan jokin mallisovellus tai opiskella valmiilla sovelluspaketilla. Hyvät teoriamateriaalit löytyvät valmiiden järjestelmien opiskeluun. Arduino-järjestelmät taas soveltuvat hyvin niin analogisen kuin digitaalisen tiedon keräämiseen ja prosessointiin, koska niihin saa tarvittavan määrän I/O-portteja vaihtamalla sovellusalustaa (Luku 4.3). Järjestelmä soveltuu hyvin myös elektroniikan ja elektroniikan komponenttien opiskeluun. Arduinon käyttöä opetukseen tukee se, että netissä toimiva avoimen lähdekoodin yhteisö jakaa tietoa tekemistään sovellusprojekteista ja antaa neuvoja. Kirjastotuki eri komponenteille on netissä helposti saatavilla ja tarjonta on laajaa. Arduino-järjestelmän saa myös helposti liitettyä tietoverkkoon, kun lisää järjestelmään esimerkiksi ethernet-tytärkortin. Sen jälkeen järjestelmä toimii etähallittavana tie-

donkeruulaitteena. Kaikki järjestelmät ovat edullisia hankkia ja helppoja ottaa käyttöön, Arduinon ollessa kuitenkin edullisin.

4.5 Sulautettujen järjestelmien mahdollisuudet OPS:ssa

Mitä ammattiaineita voitaisiin sulautetuilla järjestelmillä kansallisen Tieto- ja tietoliikennetekniikan opetussuunnitelman [1] mukaisesti opettaa? Alla on yhteenvetona elektroniikka- ja ICT-asantajan tutkinnon osiin liittyviä ammatillisia asioita, joissa sulautetut järjestelmät voisivat toimia mainiosti niin oppimista hyödyntäen kuin opiskelumotivaatiota nostaan. Monipuolisilla käytännön harjoituksilla voidaan edesauttaa vaikeidenkin asioiden selkeyttämistä ja estää opiskelijoiden turhautumista teoria-asioissa. Joillekin on taas vaikeaa saada teoriasta irti olennaista tärkeää teknistä asiaa, mutta käytännön harjoitusten kautta se onnistuu.

Elektroniikan ja ICT:n perustehtävät -tutkinnon osa

Sulautettuja järjestelmiä ajatellen olisi ensimmäisen vuoden pakollisessa *Elektroniikan ja ICT:n perustehtävät* -tutkinnon osassa toteuttamiskelpoisimpina aihealueina mikrokontrollerin kehitysympäristön -, elektroniikan komponenttien -, Internetin toimintaperiaatteen - ja IP-osoitejärjestelmän opiskelu. Samalla tutustuttaisiin yleisellä tasolla sulautetun järjestelmän rakenteeseen. Johdantona toisena vuonna tuleviin ohjelmointitöihin olisi hyvä käydä läpi muutama C-kielen perusharjoitus. Tähän sopisi hyvin esimerkiksi ilmainen Dev-C++ -ohjelmointiympäristö [23]. Sillä olisi hyvä harjoitella C-ohjelmoinnin perusteita ilman sulautettuja järjestelmiä. LEGO-robotilla tehtäisiin muutama perusharjoitus joko National Instrumentsin valmistamalla graafisella LabVIEW-pohjaisella ohjelmointiympäristöllä [24] LEGO Mindstorms NXT, joka on oletuksena asennettuna LEGO-roboteissa, tai voitaisiin käyttää roboteille suunniteltua C-ohjelmointikieltä NXC (Not eXactly C). Arduinoihin voitaisiin tutustua käyttämällä sen omaa Sketch-ohjelmointiympäristöä. Sen avulla voisi tehdä harjoituksia esimerkiksi I/O-portteja ohjelmoimalla ja tekemällä elektroniikan tutkimiseen painottuvia ohjelmia, johon Arduino sopii hyvin. MatrixTSL:n järjestelmien ohjelmointia harjoiteltaisiin vuokaaviotyypisellä ohjelmointiympäristöllä Flowcodella

tekemällä muutama pieni harjoitus. Matrix-järjestelmään on saatavilla valmis sovellus WEB-serverin luomiseen ja siihen on hyvä ohjeistus valmiina. Näin tässä tulisi samalla harjoiteltua IP-osoitejärjestelmän perusmäärittelyt, sekä Internetin rakennetta. Näin näiden järjestelmien käyttö ja ohjelmointi olisi toisena vuonna tuttua ja sujuvampaa. Tässä tutkinnon osassa voisi harjoitusten painopiste olla Arduinossa, koska se on helppo käyttää ja sopii hyvin yhteen elektroniikan opiskeluun. Muiden järjestelmien kohdalla riittää, että niihin tutustutaan ja osataan tehdä niille ohjelmia sekä tallentaa ne mikrokontrolleriin.

Ensimmäisenä vuonna haasteena on se, että lähtötasojen erot opiskelijoilla voivat olla suuret. Siksi ajan löytäminen kaikelle tälle voi olla haasteellista. Silti edellisestä nähdään, että sulautettuja järjestelmiä voitaisiin hyödyntää ensimmäisenä vuonna laajalti OPS:n puitteissa. Sitten kun edetään toiselle ja kolmannelle luokalle, niin tulee edellisten asioiden lisäksi sekä syvällisempää että uutta tietoa.

Tietokone- ja tietoliikennetekniikan perusteet -, ja ammattielektroniikka -tutkinnon osat

Seuraavat asiat koskevat lähinnä toisen ja kolmannen luokan opetettavia asioita. Yllämainitut tutkinnon osat jakaantuvat kahdelle vuodelle. Tietokoneen ja tietoliikenteen perusrakenteita sekä sulautetun järjestelmän komponenttituntemusta voidaan harjoitella esimerkiksi seuraavissa tutkinnon osien asiayhteyksissä: mikrokontrollerit, tietoliikenne, sulautetut järjestelmät ja niiden ohjelmointi, elektroniikka, tietokoneen tietoliikenneväylät, langattomat järjestelmät, Internet-tietoverkko. Nuo aihealueet jakaantuvat toisen ja kolmannen vuoden osalle. Sulautettujen sovellusten kokonaisjärjestelmien opettamiseen sopisi hyvin Matrixin järjestelmät, koska siihen löytyy monia valmiita sovelluksia ohjeineen. Flowcodella voi nopeasti ohjelmoida sovellusalustoja, joissa on PIC-, AVR- tai ARM-mikrokontrolleri. Internetin perusteiden ja tietoliikenneväylien tutkimiseen sopii Matrix-järjestelmät myös hyvin. LEGO-robotti sopii hyvin liikkuvan sulautetun järjestelmän opiskeluun. Arduinon ja koekytkentälevyn avulla voitaisiin harjoitella monenlaisia elektroniikkaan liittyviä asioita. Arduinossa on paljon I/O-portteja ja AD/DA-muuntimia, joten se soveltuisi hyvin esimerkiksi mikrokontrollerin toiminnan opiskeluun. Tietoliikenneasioita ja väylätekniikkaa tulee käsiteltyä kaikkien järjestelmien kohdalla.

Tutkimuksessa olevista kolmesta järjestelmästä olisi näiden tutkinnon osien kohdalla Matrix-järjestelmät varteenotettavin, koska sillä voidaan tehdä melko laajasti erilaisia harjoituksia. Erityisen hyvin Matrix sopii TCP/IP-tietoliikenteen harjoitteluun. Myös väylä- ja tietoliikennetekniikan kuten CAN-, Bluetooth-, Zigbee-, WiFi- ja GSM-harjoitukset onnistuvat hyvin.

Edellisistä esimerkeistä nähdään selkeästi, että näitä kaikkia kolmea sulautettua järjestelmää voitaisiin hyödyntää todella monipuolisesti ICT-asentajien kaikkien kolmen vuoden aikana. Näin tulisi käytäntö ja teoria mukavasti yhdistettyä ja opiskelijoiden mielenkiinto ja motivaatio paranevat.

5 Case ”KPEDU - sulautetut järjestelmät opetuksessa”

Keski-Pohjanmaan koulutusyhtymän (KPEDU) alaisuudessa toimivassa Keski-Pohjanmaan ammattiopistossa (KAO) on valittu Tieto- ja tietoliikennetekniikan linjavaihtoehtoksi ICT-asentajan perustutkinto. Liitteen 1 taulukossa 2.1 ovat ne tutkinnon osat merkittynä harmaalla pohjalla, jotka on valittu KAO:n ICT-asentajalinjan ammatillisiksi tutkinnon osiksi. KAO:lla valitut pakolliset - ja valinnaiset tutkinnon osat ovat rakenteeltaan samoja kuin valtakunnallisessa OPS:ssa [1]. KAO:n opetussuunnitelmaan ei ole lisätty teollisuuden erityispiirteisiin liittyviä paikallisiin ammattitaitovaatimuksiin [1, s. 2] kuuluvia erikseen määriteltyjä tutkinnon osia. *Vapaasti valittaviin* tutkinnon osiin on opiskelijoille KAO:lla suositeltu otettavaksi räätälöidyt opetuspaketit (5 + 5 osp) tutkinnon osista *Kappaletavara-automaatio* ja *Valvonta- ja ilmoitusjärjestelmät*. Ne tukevat ja laajentavat hyvin ICT-asentajan työnkuvaa ja mahdollistavat paremman työllistymisen teollisuuden pariin tai valvonta- ja hälytysalan työtehtäviin. Resursseista johtuen vapaasti valittavista tutkinnon osista toteutetaan ne, joihin ilmoittautuu eniten opiskelijoita. Tässä tutkimuksessa tutkimuksen aiheena on opiskelijoiden käyttämien sulautettujen järjestelmien opetusalustojen hyödyllisyys opetuksessa ja kuinka näillä järjestelmillä voisi kehittää ja parantaa omaa opetusta.

5.1 Järjestelmien käyttötilanne KPEDU:ssa

ICT-asentajien opinnoissa käytetään LEGO-robotteja ohjelmoinnin opiskelussa, jossa harjoitellaan sen eri moottoreiden avulla liikeratoja sekä hyödynnetään siinä olevia ääni- ja valosensoreita eri reittien valinnoissa. Robotin C-kieliset ohjelmat käytännön opetusharjoituksiin tehdään NXC-ohjelmointikielelle suunnitellussa BricxCC-ohjelmointiympäristössä (IDE). Näitä harjoituksia tehdään ICT-asentajien *Tietokone- ja tietoliikenneasennukset* -tutkinnon osassa. Kaikkien opiskelijoiden kanssa pyritään siihen, että perusasiat robotista ja sensoreista käydään läpi teoriassa ja ohjelmointiharjoitusten kautta. Nopeammin eteneville annetaan haastavampia ohjelmointitehtäviä.

KPEDUn ICT-asentajien opetuksessa käytetään nykyisin Matrix-ohjelmointialustan kanssa monipuolisesti siihen kuuluvia eri sensorikortteja. Ohjelmointiharjoituksissa aloitetaan ensin yksinkertaisimmasta elektroniikkakortista (E-blocks -sarja), sen jälkeen otetaan käyttöön useampia kortteja ja harjoitellaan niiden yhteiskäyttöä. Kaikki opiskelijat käyvät läpi perusasiat käytettävistä sulautetuista järjestelmistä ja lisälaittekorkeista sekä teoriassa että ohjelmointiharjoitusten kautta. Nopeammin etenevät saavat haastavampia ohjelmointitehtäviä. Tällä hetkellä käytetään eniten PIC-mikrokontrollerikorttia ja siihen liitettäviä lisäkortteja kuten kytkin-, LED-, LCD-, 7-segment-, moottorinohjaus- ja koekytkentäkorttia. Edellisten lisäksi ICT-asentajien *Tietokone- ja tietoliikenneasennukset* -tutkinnon osan harjoituksissa käytetään Flowcodella ohjelmoitavaa Formula-autoa [3] sekä Matrixin ECIO-mikrokontrollerialustaa [3]. ECIO liitetään koekytkentäalustaan, johon tehdään virtapiirikytkennät ja kytketään tarvittavat komponentit, joiden toimintaa Flowcode-ohjelmoinnilla harjoitellaan. Formula-autoa ohjelmoidaan Flowcodella kulkemaan erilaisia reittejä eri nopeuksilla käyttäen hyväksi auton sensoreita.

LEGO-robotit ja Matrixin ohjelmoitavat mikrokontrollerikortit ja sensorikortit ovat opetuksessa mukana ICT-asentajien *Tietokone- ja tietoliikenneasennukset* -tutkinnon osassa niin paljon kuin tuntimäärissä on varaa. Mikäli kurssilta jää aikaa niin käytettävissä on myös muutama vanhempi AVR-mikrokontrollerilla [2] varustettu ohjelmointikortti joilla voidaan tehdä muutama ohjelmointiharjoitus C-kielellä. Niillä ohjelmointi tapahtuu AVR Studio -ohjelmointiympäristössä (IDE).

5.2 Tutkimuksen tavoite ja kuvaus

ICT-asentajien *Tietokone- ja tietoliikenneasennukset* -tutkinnon osassa opiskelijat käyttävät erilaisia sulautettujen järjestelmien opetusalustoja. Tutkimuksen tavoitteena on vertailla kahden sulautettujen järjestelmien opetukseen tarkoitettun alustan soveltuvuutta KPEDUn ICT-asentajien opetukseen sekä arvioida voiko näillä järjestelmillä kehittää ja parantaa omia opetusmenetelmiä. Laitteistoiksi tutkimukseen valittiin Arduinon ja MatrixTSL:n sulautettujen järjestelmien opetusalustat. Kortit valittiin sillä perusteella, että ne ovat edullisia ja niitä käytetään melko yleisesti monissa eri oppilaitoksissa aina peruskouluista yli-

opistoihin. Lisäksi ne tarjoavat vertailuun kaksi erityyppistä ohjelmointitapaa ja kaksi erilaista ohjelmointiympäristöä.

Arduino ja Matrix ovat samankaltaisia järjestelmiä opetuksen kannalta, joten on luonnollista verrata näitä keskenään. Alustojen vertailevalla käytöllä arvioidaan niiden toimivuus niin elektroniikan kytkentäalustana kuin ohjelmoinnin sovellusalustana. Molempien järjestelmien toimittajien tarjoamien tukisivustojen ja opetusmateriaalien hyödyllisyyttä pyritään myös arvioimaan. Arviointia varten laadittiin aluksi harjoitustehtävät Arduinolle. Arduino-laitteilla tehtyjen harjoitusten jälkeen opiskelijoille tehtiin kirjallinen kysely, jossa kartoitettiin opiskelijoiden näkemyksiä alustan ja tukimateriaalin käytöstä. Tämän jälkeen laadittiin samantyylliset harjoitustehtävät Matrixille. Myös näiden harjoitusten jälkeen pyydettiin käyttökokemuksista vastaukset kirjallisesti. Molempien järjestelmien laitteisto- ja ohjelmointiharjoitusten jälkeen tehtiin vielä yksi kysely, jossa pyydettiin opiskelijoita vertailemaan alustoja keskenään.

Tutkimus toteutettiin pilottitutkimuksena yhdelle KPEDU:n ICT-luokan 10 opiskelijan ryhmälle. Tutkimus oli koululla ensimmäinen laatuaan ja sen aikana tehtiin kyselytutkimus ryhmälle kahdesta käytetystä harjoittelulaitteistosta.

5.3 Tutkimuksen toteutus

Tutkimus toteutettiin *Tietokone- ja tietoliikenneasennukset* -tutkinnon osassa vuonna 2016–17 harjoitustöiden ja kyselytutkimuksen yhdistelmänä yhdelle KPEDU:n ICT-opiskelijaryhmälle, jossa oli 10 opiskelijaa. Harjoitustyöt Arduinolla tehtiin syksyllä 2016 ja Matrixille keväällä 2017. Opiskelijoille annettiin harjoitustyöt niin Arduinolle kuin Matrixillekin, joille kummallekin tehtiin hieman erilaiset harjoitustehtävät ja niille ohjelmat. Harjoitustehtävistä haluttiin tehdä hieman erilaiset, jotta opiskelijoiden mielenkiinto säilyisi yllä paremmin. Arduinon harjoitusten jälkeen pyydettiin opiskelijoilta vastaukset kyselylomakkeelle. Samoin Matrixin kohdalla ja lopuksi vielä annettiin opiskelijoille järjestelmien vertailukysymykset. Seuraavaksi esitellään opiskelijoiden kanssa tehtyjen töiden valmistelutoimet ja harjoitustyöt.

Arduino

Valmistelutöinä opiskelijoille jaettiin ensin Arduino Uno -ohjelmointikortit. Seuraavaksi haettiin ja asennettiin mikrokontrollerikortin ohjelmointiympäristö (IDE), jonka mukana tulivat Arduinon vaatimat ohjelmistoajurit Windows-käyttöjärjestelmää varten. Ohjelmointiympäristön asentamisen jälkeen käytiin tunnilla läpi ohjelmointikortin käyttöönottoon liittyvät yleiset asiat. Seuraavana opiskeltiin ne asiat, joita tulee ottaa huomioon, kun aloitetaan Arduino-ohjelmointikortin käyttö. Tässä tulee samalla myös opiskeltua tietokonetekniikkaa, eli miten laitteisto, käyttöjärjestelmä ja ohjelmisto toimivat yhdessä. Ensimmäkin varmistetaan, että IDE löytää Arduinon samasta COM-portista kuin käyttöjärjestelmä. Laitehallinnan mukainen oikea COM-portti pitää valita IDE:ssä, ja sen jälkeen voi Arduino-yhteyden testata IDE-valikon valinnalla *Lue Arduino-tiedot*. Mikäli yhteys toimii, niin työpöydälle ilmestyy pieni ikkuna, jossa näkyy muutamia tietoja Arduino-kortilta. Tämän jälkeen IDE ja Arduino toimivat yhteen ja voidaan aloittaa harjoitustöiden tekeminen.

Tehtävien kuvaukset

Ensimmäisenä Arduino-harjoituksena opiskeltiin kolme pientä harjoitteluohjelmaa (Liite 4), jotta opiskelijat tottuvat ohjelmointiympäristön ja koekytkentälevyn käyttöön. Harjoituksessa käytiin läpi peruskomentoja ja niistä muodostettuja pieniä ohjelmia. Komentoja, joita käytettiin, olivat esimerkiksi `digitalRead`, `digitalWrite`, `analogRead` ja `pinMode`. Ensimmäin käytettiin `pinMode`- ja `digitalWrite`-komentoja ohjaamaan LED:iä. Seuraavana kytkentään ja ohjelmaan lisättiin kytkin, jota luettiin `digitalRead`-komennolla. LED:iä sytytettiin ja sammutettiin kytkimen mukaan. Kolmas harjoitteluohjelma oli PWM-ohjelma (Pulse Width Modulation). Siinä luettiin potentiometrin tilaa `analogRead`-komennolla analogiaportin kautta ja luetulla signaalilla ohjattiin LED:in valon voimakkuutta. Samalla havainnollistettiin teoriassa sitä, kuinka monta jännitetasoa on käytössä analogia- ja digitaalisignaaleissa ja miten se pitää huomioida ohjelmassa. Ohjelmassa otettiin mukaan ohjelmointiympäristössä oleva sarjamonitori-ikkuna, johon ohjelmallisesti tulostettiin muuttujien arvoja. Kytkennät tehtiin koekytkentälevylle.

Seuraavana käytiin läpi alustus toiseen harjoitusohjelmaan (Liite 5), joka oli toimiston ovivälilyhteen ohjelma. Siinä käytettiin kytkimiä ja ledejä, jotka opiskelijoiden piti kytkeä kytkentäalustalle ja johdottaa oikein. Kytkentää ja ohjelmaa rakennettiin toiminto kerrallaan. Kytkennässä oli ulkopuolella ovikytkin (push-button) sisäänpääsypyynnön varten, sekä

ovivaloina väriledit, punainen varattu- ja vihreä vapaana-valolle. Ovikytkin sytytti toimistossa vihreän LED:in. Muita ovivaloja ei otettu mukaan, koska yritettiin pitää ohjelma suhteellisen yksinkertaisena. Toimistoon sijoitettiin kaksi kytkintä ja yksi vihreä LED. Kytkimien merkitykset olivat varattu-kytkin (on-off) ja vapaana-kytkin (push-button). Varattu-kytkin ohjaa ovella punaista LED:iä ja vapaana-kytkin ohjaa ovella vastaavasti vihreää LED:iä sekä soittaa lisäksi summeria sisääntuloluvan merkiksi.

Kolmantena harjoitusohjelmana oli lämpömittariohjelma (Liite 6). Siinä oli käytössä yksi DS18B20-tyypin lämpöanturi [41]. Teoriassa käytiin läpi lämpötila-anturin kytkentävaihtoehtoja. Ohjelman tuli tulostaa sarjamonitori-ikkunaan lämpötila-anturin lämpötila tietyin väliajoin. Ohjelman ja kytkennän toimivuus testattiin lämmittämällä anturia sormilla, jolloin lämpötilalukemat muuttuivat. Tässäkin tehtiin komponenteista elektroniikkakytkentä kytkentälevylle. Arduino-harjoituksissa oli lisätehtävänä myös piirtää jokainen kytkentä Frizing-ohjelmalla [42]. Tässä tuli hieman opettelua myös ilmaisen CAD-ohjelman (Computer Aided Design) käyttöön. Opettajan toimesta kaikki tehdyt ohjelmat testattiin tekijöiden kanssa ja kirjattiin ylös.

Matrix

Oppilaille jaettiin Matrixin ECIO-mikrokontrollerikortit ja niillä tehtiin kolme ohjelmaa. Tarvittava ohjelmointiympäristö oli jo aiemmin asennettu tietokoneisiin. Ensin käytiin läpi IDE:n peruskäyttö ja testattiin, miten ohjelmointikäyttöliittymästä saadaan yhteys mikrokontrollerikorttiin. Samalla kokeiltiin yhteyttä mikrokontrolleriin, jolloin nähtiin, että toimivatko Windows-ajurit ohjelmointikäyttöliittymän kautta. Koska aikaisemmin oli jo kehitetty Matrixin Flowcode-ohjelmistoa, oli tämä opiskelijoille jo melko tuttua asiaa. Pientä kertausta piti tehdä, niin asiat palailivat opiskelijoille pikkuhiljaa mieliin.

Tehtävien kuvaukset

Ensimmäinen harjoitus oli rallivalojen ohjelmointi (Liite 7). Siinä käytettiin Matrixin tuotesarjaan kuuluvia ledivalo- ja kytkinkortteja, jotka ovat valmiita elektroniikkakortteja. Muita komponentteja ei käytetty. Matrix kytkettiin kiinni sille suunniteltuun apukorttiin, jossa oli valmiit liittimet elektroniikkakortteille. ”Rallin startti” tapahtui tiettyä kytkintä

painamalla ja ledivalot syttyivät tietyn aikataulun mukaan. Sitten kun varsinainen startti tapahtui, niin kaikki kolme lähetysvaloa paloivat yhtä aikaa kolme sekuntia.

Toinen harjoitus oli kytkinpainalluksia binäärisesti laskeva ohjelma (Liite 8). Tässäkin käytettiin apuna ledivalo- ja kytkinkortteja. Matrixin nollauskytkimen painalluksen jälkeen painettiin sovittua kytkintä 15 kertaa ja kytkinpainallusten määrä näkyi ledikortilla binäärisesti. Sitten kun painettiin kytkimeltä 16. kerta, niin kaikki ledit sammuiivat, ja laskenta alkoi alusta. Tässä käytettiin samaa kytkentää, kuin ensimmäisessä Matrix-harjoituksessa.

Kolmannessa harjoituksessa tehtiin ohjelma hämärekytkimen toiminnan simuloimiseksi (Liite 9). Hämärekytkin ohjaa esimerkiksi huoneen valojen syttymistä ja sammumista. Tämän harjoituksen elektroniikkakytkentä tehtiin koekytkentälevylle, jossa käytettiin apuna LDR-vastusta (Light Dependent Resistor) ja muutamaa lisäkomponenttia. Potentiometrillä säädettiin perus-hämäretilä, jolla LED juuri ja juuri sammui. Sen jälkeen, kun LDR-vastusta peitettiin, syttyi LED huonevalojen syttymisen merkiksi. Kaikkien opiskelijoiden ohjelmat tarkastettiin, ja niistä tehtiin merkinnät.

5.4 Tulokset kyselyistä

Arduino- ja ECIO -järjestelmien yleiseen käyttöön liittyvistä 6 kysymyksestä olivat 3 ensimmäistä laitteiden käyttöön ja ohjelmointiin liittyviä (Liite 2). Kysymykset olivat molempien järjestelmien kohdalla samat ja jokaiseen kysymykseen oli 5 erilaista vaihtoehtoa (1=helppo, 2=melko helppo, 3=sopiva, 4=vaikeahko, 5=vaikea). Kysymykset annettiin opiskelijoille aina kummankin järjestelmän kohdalla kolmen harjoitustehtävän jälkeen. Opiskelijaryhmäksi valikoitui kohtalaisen pieni ryhmä, jossa oli 10 opiskelijaa. Tehtävillä ja kysymyksillä pyrittiin selvittämään sitä, kumpi käytetyistä järjestelmistä on enemmän opiskelijoiden mieleen sekä kumman käyttö on helpompaa. Vastauksia saatiin Arduinon osalta 8 opiskelijalta ja Matrixin osalta ainoastaan 4 opiskelijalta. Kysymyksissä 1-3 haluttiin selvittää opiskelijoiden näkemyksiä alustojen käytöstä. Kysymyksistä 1-3 muodostettiin uusi summamuuttuja ”alustan ohjelmointikäyttö”, joka skaalattiin summauksen jälkeen takaisin asteikolle 1-5. Alkuperäiset vastausten arvot ja summamuuttuja on esitetty taulukoissa 5.1 ja 5.2.

Taulukko 5.1. Summamuuttuja ”Arduino-alustan ohjelmointikäyttö”

Vastaaja	Kysymys 1	Kysymys 2	Kysymys 3	Summa	Skaalattu arvo
Opiskelija 1	2	2	4	8	3
Opiskelija 2	2	2	2	6	2
Opiskelija 3	3	3	2	8	3
Opiskelija 4	2	3	3	8	3
Opiskelija 5	2	1	2	5	2
Opiskelija 6	3	2	3	8	3
Opiskelija 7	2	3	3	8	3
Opiskelija 8	2	3	2	7	3
Keskiarvo	2.25	2.38	2.63	7.25	2.75

Taulukko 5.2. Summamuuttuja ”Matrix-alustan ohjelmointikäyttö”

Vastaaja	Kysymys 1	Kysymys 2	Kysymys 3	Summa	Skaalattu arvo
Opiskelija 1	3	3	4	10	4
Opiskelija 2	1	2	4	7	3
Opiskelija 3	1	2	2	5	2
Opiskelija 4	3	3	4	10	4
Keskiarvo	2.00	2.50	3.50	8.00	3.25

Näiden kolmen kysymyksen vastausten perusteella ei tullut suurta eroa esille eri järjestelmien ja ohjelmointiympäristön välille. Ehkä vaaka kallistuu hieman Arduinon puolelle. Taulukon 5.1 perusteella se koettiin hieman helpommaksi käyttää. Lisäksi kolme neljästä opiskelijasta, jotka vastasivat molempiin kyselyihin, kokivat Arduino-alustan helpommaksi.

Liitteen 2 kysymyksissä 4 ja 5 haluttiin selvittää opiskelijoiden näkemyksiä alustojen tukimateriaaleista kuten IDE:n help-toiminnosta ja nettisivuista. Kysymykset olivat molempien järjestelmien kohdalla samat ja kumpaankin kysymykseen oli 5 erilaista vaihtoehtoa (1=en käyttänyt, 2= käytin, ei apua, 3= auttoi vähän, 4=auttoi melko paljon, 5=korvaamaton). Kysymyksistä 4 ja 5 muodostettiin uusi summamuuttuja ” alustojen tukimateriaalin hyödyllisyys”, joka skaalattiin summauksen jälkeen takaisin asteikolle 1-5. Alkuperäiset vastausten arvot ja summamuuttuja on esitetty taulukoissa 5.3 ja 5.4.

Taulukko 5.3: Summamuuttuja ”Arduino-alustan tukimateriaalien hyödyllisyys”

Vastaaja	Kysymys 4	Kysymys 5	Summa	Skaalattu arvo
Opiskelija 1	3	1	4	2
Opiskelija 2	3	1	4	2
Opiskelija 3	4	4	8	3
Opiskelija 4	4	1	5	2
Opiskelija 5	4	3	7	3
Opiskelija 6	3	4	7	3
Opiskelija 7	2	1	3	1
Opiskelija 8	3	3	6	2
Keskiarvo	3,25	2,25	5,50	2,25

Taulukko 5.3: Summamuuttuja ”Matrix-alustan tukimateriaalien hyödyllisyys”

Vastaaja	Kysymys 4	Kysymys 5	Summa	Skaalattu arvo
Opiskelija 1	1	1	2	1
Opiskelija 2	3	3	6	2
Opiskelija 3	1	1	2	1
Opiskelija 4	4	3	7	3
Keskiarvo	2,25	2,00	4,25	1,75

Kysymyksissä 4 ja 5 yhteenvedona voidaan sanoa, että kummankaan järjestelmän valmistajien tarjoamista IDE:n Help-ohjeista eikä nettisivuista ollut juurikaan apua. Taulukoissa 5.2 ja 5.3 olevista vastauksista nähdään, että Arduinon osalta yksi ja Matrixin osalta kaksi opiskelijaa ei tarvinnut tukimateriaalia lainkaan. Matrixin kohdalla syy voi olla se, että Matrixin järjestelmä oli jo tuttu aiemmista harjoituksista. Taulukon 5.2 taulukon summamuuttujien arvoista verrattuna taulukon 5.3 arvoihin nähdään, että Arduinon tukimateriaali koettiin hieman hyödyllisemmäksi. Syitä tähän voi olla useita. Ehkä se vaikutti asiaan, että Arduino oli opiskelijoille uusi alusta, siksi apuvälineitä käytettiin ahkerammin. Joillakin opiskelijoilla voi olla vaikeuksia löytää tarvitsemaansa tietoa. Tiedon etsiminen vaatii aikaa ja vaivaa, siksi joidenkin mielestä on helpompaa saada tieto vieruskaverilta tai kuunnella opettajan antamia vinkkejä.

Kuvassa 5.3 on esitetty molemmista alustoista vastaukset kysymykseen 6 (Antoiko Arduino / ECIO sinulle ohjelmointimotivaatiota, niin että haluat itse opetella lisää?). Samat ky-

symykset esitettiin molempien järjestelmien kohdalla, ja kummassakin oli 5 erilaista vaihtoehtoa (1=ei vaikutusta, 2=motivaatio heräsi, 3=motivoi kohtalaisesti, 4=motivoi paljon, 5=jatkan varmasti). Kysymyksen tarkoitus oli selvittää sitä, että oliko käytännön laitteista apua ja hyötyä ohjelmoinnissa, ja löytyikö sitä kautta motivaatiota jatkaa ohjelmoinnin harjoittelua jopa omaehtoisesti. Taulukossa 5.4. on tulokset molempien alustojen kyselystä ”Arduino, ohjelmointimotivaation lisäys” ja ”Matrix, ohjelmointimotivaation lisäys” sekä niiden keskiarvot.

Taulukko 5.4: Tulokset kysymyksiin ”Antoiko Arduino ohjelmointimotivaatiota” ja ”Antoiko Matrix ohjelmointimotivaatiota” sekä niiden keskiarvot

Vastaaja	Kysymys 6 Arduino	Kysymys 6 Matrix
Opiskelija 1	1	2
Opiskelija 2	2	1
Opiskelija 3	1	2
Opiskelija 4	2	3
Opiskelija 5	3	-
Opiskelija 6	2	-
Opiskelija 7	1	-
Opiskelija 8	2	-
Keskiarvo	1,75	2,00

Vastauksista nähdään, että molempien järjestelmien kohdalla näillä järjestelyillä oli hyvin vähän vaikutusta motivaatioon. Yksi opiskelija totesi molempien järjestelmien kohdalla, että järjestelyt motivoivat kohtalaisesti. Ketään ei motivoinut paljon eikä kukaan vastannut jatkavansa varmasti ohjelmointia näiden harjoitusten perusteella. Arduinon kohdalla kolmen opiskelijan mielestä ei tapahtunut mitään ohjelmointimotivaation kasvua ja Matrixin kohdalla yksi opiskelija oli samaa mieltä.

Harjoitustöiden vaikeusasteet

Käytännön harjoitustöitä oli yhteensä 6 kpl, kolme harjoitusta Arduinolle ja kolme Matrixille. Noiden harjoitusten jälkeen tehtiin kysely harjoitustöiden vaikeusasteesta (Liite 3). Liitteen 3 kysymyksissä 1 - 6 haluttiin selvittää opiskelijoiden näkemyksiä molemmille alustoille tehtyjen harjoitustehtävien vaikeusasteesta. Kysymykset jokaisen harjoitustyön

kohdalla oli samat, ja niihin oli 5 vaihtoehtoa (1=helppo, 2=helpohko, 3=sopiva, 4=vaikeahko, 5=vaikea). Kysymyksistä 1-3 muodostettiin Arduinolle oma taulukko ja siihen luotiin uudet summamuuttujat ”Arduino-harjoitustöiden vaikeus” ja toiseen taulukoon vastaavasti ”Matrix-harjoitustöiden vaikeus” jotka skaalattiin summauksen jälkeen takaisin asteikolle 1-5. Alkuperäiset vastausten arvot ja summamuuttujat on esitetty taulukoissa 5.4 ja 5.5.

Taulukko 5.4. Summamuuttuja ”Arduino-harjoitustöiden vaikeus”

Vastaaja	Kysymys 1	Kysymys 2	Kysymys 3	Summa	Skaalattu arvo
Opiskelija 1	4	4	4	12	4
Opiskelija 2	3	5	3	11	4
Opiskelija 3	3	3	1	7	2
Opiskelija 4	3	5	2	10	3
Keskiarvo	3,25	4,25	2,50	10,00	3,25

Taulukko 5.5. Summamuuttuja ”Matrix-harjoitustöiden vaikeus”

Vastaaja	Kysymys 4	Kysymys 5	Kysymys 6	Summa	Skaalattu arvo
Opiskelija 1	1	3	1	5	2
Opiskelija 2	3	3	4	10	3
Opiskelija 3	1	3	3	7	2
Opiskelija 4	2	3	3	8	3
Keskiarvo	1,75	3,00	2,75	7,50	2,50

Harjoitustöiden vaikeusasteet kokonaisuudessaan näyttävät kuvan 5.4 perusteella olevan Arduinon osalta hieman vaikeampia. Matrixin harjoituksissa ehkä auttoi se, että alusta oli jo tuttu aiemmista tutustumisharjoituksista johtuen. Arduinon kohdalla harjoitus 2 (Kysymys 2) näytti olevan muihin tehtäviin verrattuna huomattavasti vaikeampi. Arduinon harjoituksesta 2 voisi sanoa, että se osoittautui ennakoarvailuja vaikeammaksi. Kahden opiskelijan mielestä se oli vaikea. Yhden opiskelijan mielestä kaikki Arduinon harjoitukset olivat vaikeahkoja. Matrixin tehtävien kohdalla taas harjoitus 1 (Kysymys 4) näytti vastaavasti olleen huomattavasti helpompi. Saattaa johtua osittain siitä, että siinä ei tarvinnut tehdä komponenttikytkeitä. Taulukkojen 5.4 ja 5.5 perusteella voidaan sanoa, että Arduinon harjoitus 1 ja Matrixin harjoitus 2 olivat lähes kaikkien vastaajien mielestä sopivan

vaikeita. Arduinon harjoitus 3 sekä Matrixin harjoitus 3 jakoivat mielipiteitä laidasta laitaan valintojen ollessa 1 ja 4 välillä. Kolmen tehtävän kohdalla vastauksiksi tuli myös helppo-valintoja.

Lopuksi voidaan sanoa, että tehtävissä oli vaikeusasteita laidasta laitaan. On varmasti hyväkin, että välillä opiskelija tuntee onnistuvansa, ettei kaikki tehtävät ole liian vaikeita. Haasteita pitää kuitenkin olla, eli joidenkin tehtävien tulee olla vaativampia. Opetuksellisesti tehtävissä on parantamisen varaa ja niihin tehdään parannuksia seuraavia kursseja ajatellen.

Arduinon ja Matrixin välinen vertailu

Lopuksi opiskelijoille annettiin kysymykset molemmista järjestelmistä sekä niiden ohjelmointiympäristöistä, joilla selvitettiin näiden järjestelmien keskinäistä käytön vertailua. Tässä kyselyssä (Liite 3) oli kysymykset 7 - 13. Niissä kaikissa oli 2 vaihtoehtoa (Arduino on 1 ja Matrix on 2). Alkuperäiset vastausten arvot ja niiden keskiarvot on esitetty taulukossa 5.6.

Taulukko 5.6. Vastaukset kysymyksiin ”Arduino/ECIO-opetusalustan soveltuvuus harjoitustöihin”

Vastaaja	Kysymys 7	Kysymys 8	Kysymys 9	Kysymys 10	Kysymys 11	Kysymys 12	Kysymys 13
Opiskelija 1	2	2	2	2	2	2	2
Opiskelija 2	1	2	1	2	2	1	2
Opiskelija 3	2	2	2	2	1	2	2
Opiskelija 4	1	1	1	1	1	1	1
Keskiarvo	1,50	1,75	1,50	1,75	1,50	1,50	1,75

Taulukon 5.6 perusteella opiskelijat pitivät Matrixin ohjelmointitavasta (kysymys 8) ja se koettiin selkeämmäksi käyttää (kysymys 10). Tulokseen voi vaikuttaa, että Matrixiin ja Flowcodeen on jo hieman tutustuttu aiemmalla kurssilla. Myös kokonaisuutena opiskelijat käyttäisivät mieluiten (kysymys 13) Matrix-alustaa. Opiskelijoista oli vain yksi selkeästi Arduinon kannalla ja kolme muuta opiskelijaa olivat Matrixin kannalla. Muilta osin (kysymykset 7,9, 11 ja 12) vastaukset jakaantuivat tasan eri opetusalustojen kesken. Se minkä takia tulokset olivat tällaiset voi johtua siitä, että kurssi jakaantui aikapulan takia kahdelle

jaksolle ja toisella jaksolla eivät kaikki opiskelijat olleet paikalla. Osa ei myöskään saanut aivan kaikkia tehtäviä tehtyä ja he eivät vastanneet kyselyihin. Jälkikäteen voisi sanoa, että harjoitusten tulisi olla lyhyempiä, yksinkertaisempia ja selkeämpiä, koska aikaa on niin vähän käytettävissä. Opiskelijoiden itseohjautuvuus aiheutti myös yllätyksen, he vaativat kuitenkin melko paljon opettajan ohjausta niin kytkennöissä kuin ohjelmoinnissakin ja siihen ei tullut varauduttua tarpeeksi.

5.5 Johtopäätökset

Arduinon osalta voidaan kyselyjen perusteella sanoa, että sen käyttöönotto oli melko helppoa. Ohjelmointiympäristö oli sopivan vaikea oppia, eli sitä ei koettu liian helpoksi. Arduinon ohjelmoiminen koettiin melko helpoksi tai sopivan helpoksi. Help-toiminnon käyttö auttoi joidenkin mielestä vähän ja toisten mielestä taas melko paljon. Arduinon omia nettisivuja ei käyttänyt moni ollenkaan, mutta niistä jotka käyttivät, kokivat asian niin, että nettisivustot kuitenkin auttoivat vähän tai melko paljon. Tässä taustalla voi olla sekin asia, että osa koki asioiden selvittelyn nettisivujen kautta hankalaksi tai aikaa vieväksi, ja odottivat asioihin apua kaverilta tai opettajalta. Viimeisessä kysymyksessä haluttiin tietää, että herättikö Arduinon käyttö ohjelmointimotivaatiota. Kyselyn perusteella pääsääntöisesti sen käyttö ei auttanut eikä herättänyt juurikaan ohjelmointimotivaatiota. Tästä voi päätellä sen, että pienestä innostuksen poikasesta huolimatta valtaosalle ei löydy ylipäättänsä innostusta ohjelmoimiseen tällä laitteistolla.

Matrixin osalta johtopäätöksinä voidaan todeta, että sen käyttö ohjelmointiin oli joko helppoa tai sopivan vaikeaa. Mielenpitoet jakaantuivat suunnilleen puoliksi. Matrixin ohjelmointiympäristön käyttö oli joko melko helppoa tai sopivan haastavaa, vaikeaksi ei kukaan sitä sanonut. Itse mikrokontrollerikortin ohjelmoiminen oli valtaosalle sopivan haastavaa. Kukaan ei todennut sitä helpoksi tai vaikeaksi. Help-toimintoa ei puolet käyttänyt ollenkaan, ja toisen puolen mielestä sen käyttö auttoi vähän tai melko paljon. Nettisivustoa Matrixin sivuilta käytti vain puolet. Niiden mielestä, jotka käyttivät, oli sivustosta hieman apua. Ohjelmointimotivaatiokysymykseen noin puolet vastasi niin, että motivaatio hieman heräsi. Osalle ei tullut mitään vaikutusta ohjelmointimotivaatioon ja vain yksi totesi, että Matrix

motivoi kohtalaisesti. Tästä voidaan päätellä, ettei myöskään Matrix saanut sen kummemmin opiskelijoiden ohjelmointimotivaatiota herätettyä tai parannettua.

Näiden kahden laitteiston vertailukyselyn tuloksena voidaan sanoa, että laitteistojen valinta jakaantui kahtia. Molemmille laitteistoille löytyi suunnilleen yhtä suuret tukijoukot. Ohjelmointitavoista voiton vei Matrix. Harjoituskytkentöjen teon helppous jakaantui myös puoliksi, mutta Matrix koettiin selvästi selkeämmäksi käyttää. Matrixin avulla koettiin myös oppiminen helpommaksi. Harjoitusten vaikeusasteet olivat helpohkoja tai sopivan vaikeita. Poikkeus tuli Arduinon ovisummeriharjoituksen kohdalla, joka osoittautui valtaosin vaikeahkoksi ja yhdelle jopa vaikeaksi. Myös Matrixin starttivaloharjoituksen kohdalla tuli pieni poikkeus, se osoittautui melko helpoksi.

Tuntityöskentelyn aikana tehdyistä havainnoista voi sanoa, että opiskelijat paria-kolmea lukuun ottamatta eivät olleet järin kiinnostuneita kytkentöjen tekemisestä, eikä ohjelmoinnistakaan. Enemmän kiinnosti netissä olevat muut asiat ja pelaaminen. Silloin kun tehtävistä annettiin ohjeistusta, eivät läheskään kaikki opiskelijat seuranneet ohjeiden antamista. Jonkin ajan päästä kuitenkin kysyttiin monen tehtävän kohdalla, että miten jokin asia tehdään. Kysymyksiä ei tullut paljon tuntien aikana, mutta eniten niitä tuli siinä vaiheessa, kun kytkentä ei toiminut. Ovisummeriharjoitus osoittautui näille opiskelijoille ehkä liian mutkikkaaksi, joten opiskelijat jäivät odottamaan opettajan ohjelmointiohjeita. Muutenkin aikataulun jouduttamiseksi piti opiskelijoille antaa ohjelmointivinkkejä, koska omatoimista ongelmien selvittämistä ei juurikaan esiintynyt.

Tämä tutkimus voisi antaa erilaiset tulokset toisella opiskelijaryhmällä, koska lähes joka vuosi opiskelijaryhmä toimii ja opiskelee eri tavalla. Kurssin tuntimäärä pitäisi myös olla suurempi, näillä tuntimäärillä eivät opiskelijat ehdi oppia ja omaksua asioita. Vaikka osa opiskelijoista totesi kyselyn perusteella osan tehtävistä helpohkoiksi, niin ilman opettajan antamia lisäohjeita ei ohjelmia olisi saatu tehtyä ja kytkentöjen kohdallakin oli osalla melko paljon vaikeuksia. Tunnilla kuitenkin huomasi sen, että kyllä tällaiset käytäntötyyppiset harjoitukset ovat mielenkiintoisempia kuin esimerkiksi pelkät C-ohjelmointitehtävät ilman laitteistoa. Selvästi opiskelija oli tyytyväinen suoritukseensa, kun sekä kytkentä että ohjel-

ma toimivat oikein. Vaikka he saivat apuja opettajalta, niin kuitenkin yritin sopivin välein kannustaa omatoimiseen tutkimiseen, koska asioiden selviäminen sillä tavalla nostaa kuitenkin tyytyväisyyttä ja opiskelumotivaatiota.

6 Yhteenveto

Yhteenvetona tuloksista voidaan todeta, että kummallekin järjestelmälle on kannattajansa. Tehtävät eivät olleet Arduinon ja Matrixin välillä samanlaisia, joten erilaisista vaikeusasteista johtuen voi lopputuloksissa olla jonkun verran poikkeamaa. Poikkeamiin voi vaikuttaa myös se, että Matrixin järjestelmistä ja Flowcodesta oli opiskelijoilla jo aikaisempaa kokemusta jonkin verran. Arduino oli uusi järjestelmä, joka otettiin käyttöön vasta tämän tutkimuksen yhteydessä. Tuloksiin vaikuttaa myös tämän opiskelijaryhmän yleinen aktiivisuus, koska on ollut havaittavissa myös yleinen opiskelumotivaation puute. Sulautettujen järjestelmien ohjelmointi ei ole ollut ryhmän tärkeimpien tehtävien joukossa. Opiskelijaryhmä oli myös melko pieni, eikä kaikilta saatu vastauksia kyselyihin.

Ongelmia tuli myös siitä, että kaikkia tehtäviä ei ehditty tehdä ensimmäisellä jaksolla ja toisella jaksolla tehtiin loput työt muiden töiden lomassa. Osa opiskelijoista ei saanut Matrixin kaikkia tehtäviä tehtyä ja he eivät palauttaneet kyselykaavakkeita. Pari opiskelijaa myös siirtyi toisella jaksolla eri opintoihin ja loput harjoitustehtävät tehdään myöhemmin. Arduinon kysymyksiin vastattiin vielä melko aktiivisesti, mutta Matrixin kohdalla näytti vastaamismotivaatio hävinneen. Aktiivisimmat opiskelijat vastasivat kyselyihin. Mikäli ryhmä olisi ollut isompi, niin tulokset olisivat voineet olla hieman toisenlaiset. Näistäkin tuloksista kyllä nähdään, että mikä on yleinen käsitys käytännön harjoituksista ja näistä käytetyistä järjestelmistä. Aikataulu oli myös tiukka, joten aikaa opetteluun ja oppimiseen oli rajallisesti. Opettaja joutui antamaan melko paljon opastusta sekä kytkennöissä että varsinkin ohjelmoinnissa.

Kyselytutkimuksen perusteella ja tehtävien aikana tuli kyllä selväksi, että tällaiset käytäntöön liittyvät työt ovat kiinnostavampia, kuin pelkät C-kielellä tehtävät harjoitustehtävät. Tuntityöskentelyn aikana huomasin sen, että opiskelijat kuitenkin haluavat konkreettisia välineitä ja välillä olivat jopa innoissaan tehtävistä. Tällaisissa tehtävissä tulee yhdistettyä teoria fyysisiin laitteisiin ja saadaan konkreettinen yhteys oikean käytännön tehtävän ratkaisemiseen. Lopputuloksena voidaan sanoa, että molemmat järjestelmät toimivat käytännön harjoitustehtävissä hyvin ja niitä molempia kannattaa käyttää.

Tutkielmasta on erityisesti hyötyä opetuksessa työkseen sulautettuja järjestelmiä käyttävillä opettajille ja alan harrastajille, unohtamatta myöskään työssään sulautettuja järjestelmiä suunnittelevaa asiantuntijaa. Tutkielmassa käytiin läpi LEGO:n, Arduinon ja Matrixin laitteet oheistoimintoinen. Lisäksi kartoitettiin niiden ohjelmointiominaisuuksia ja soveltuvuutta ammattiopiston elektroniikka- ja ICT-asetajan eri tietotekniikka- ja sulautettujen järjestelmien kurssien toteuttamismahdollisuuksiin. Ongelmaksi voi muodostua lähiopetuksen lyhytkestoiset tutkinnon osien aikataulut, jolloin ei ehdi opettaa kuin pääasiat kustakin aihealueesta. Lisäksi ongelmia voi aiheuttaa se, että oppilaat ovat hyvin heterogeenisiä. Toiset ovat asiassa mukana mielellään ja haluavat oppia uusia asioita, toisilla taas ei ole kovinkaan suurta motivaatiota opiskella asioita kovin syvällisesti. Toisaalta taas, jos mietitään sitä, mitä ICT-asetajan työtehtäviin kuuluu, niin ei heistä ohjelmoijia ole tarkoitus kouluttaa. Näillä kursseilla avataan asioita, jotta ne, jotka kiinnostuvat sulautettujen järjestelmien rakentamisesta ja ohjelmoinnista enemmän, pääsisivät nopeasti haastavampiin tehtäviin tai innostuisivat jatkokouluttamaan itseään. Saatujen tulosten perusteella voidaan todeta, että tutkimus vastasi sille asetettuja tavoitteita. Johtopäätöksenä on, että laitteita voidaan hyödyntää useissa eri tietoteknisissä kursseissa ja niistä on melko paljon apua ohjelmointimotivaation nostamisessa sekä ohjelmoinnin oppimisessa.

Tämä oli ensimmäinen dokumentoitu ICT-asetajille järjestetty harjoituskokonaisuus kyselyineen. Tämän pohjalta on hyvä jatkaa myöhemmin vastaavien harjoitussessioiden tekemistä. Tutkimuksen perusteella voi tulevaisuudessa paremmin valita opetukseen oikeantyyppisiä tehtäviä ja harjoitusmenetelmiä. Tulevaisuudessa on KPEDU:ssa ICT-asetajien opetuksessa tavoitteena laajentaa opetusalueiden käyttöalueita niissä tutkinnon osissa, joissa niitä jo käytetään, sekä laajentaa käyttöä myös uusiin tutkinnon osiin. LEGO-robotit, Arduinon ja Matrixin opetuslaitteet ovat opetuksessa mukana *Tietokone- ja tietoliikenneasennukset* -tutkinnon osassa. Sopiva uusi tutkinnon osa, jossa voitaisiin soveltuvin osin käyttää laajemmin sulautettujen järjestelmien opetusalueita, olisi ensimmäisenä vuonna järjestettävä *Elektroniikan ja ICT:n perustehtävät* -tutkinnon osa.

7 Lähteet

- [1] OPH, Tieto- ja tietoliikennetekniikan perustutkinto, saatavilla PDF-muodossa osoitteessa <URL: http://www.oph.fi/download/162521_tieto_ja_tietoliikennetekniikan_pt_01082015.pdf>, haettu 15.11.2015.
- [2] J. Koskinen, Mikrotietokonetekniikka, Sulautetut järjestelmät, Otava, Keuruu, 1999.
- [3] MatrixTSL:n kotisivut, saatavilla HTML-muodossa osoitteessa <URL: <http://www.matrixtsl.com/index.php>>, haettu 22.11.2015.
- [4] Microchip, PIC16F88:n tekniset tiedot, saatavilla PDF-muodossa osoitteessa <URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/30487D.pdf>>, haettu 20.11.2015.
- [5] F. Klassner and S. D. Anderson, "LEGO Mindstorms: Not just for K-12 anymore", IEEE Robotics and Automation Magazine 10, 2003.
- [6] ACM Computing Curriculum 2001, saatavilla HTML-muodossa osoitteessa <URL: http://portal.acm.org/ft_gateway.cfm?id=375414&type=pdf&coll=GUIDE&dl=GUIDE&CFID=67847880&CFTOKEN=73875971>, haettu 18.11.2015.
- [7] B. Fagin, L. Merkle, and T. Eggers, "Teaching basic computer science concepts with robotics", In Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, 2001.
- [8] F. Klassner, "A case study of LEGO Mindstorms™ suitability for artificial intelligence and robotics courses at the college level." ACM SIGCSE Bulletin. Vol. 34. No. 1., 2002, pp. 8-12.
- [9] D. Kumar and L. Meeden, "A robot laboratory for teaching artificial intelligence." ACM SIGCSE Bulletin. Vol. 30. No. 1., 1998, pp. 341-344.

- [10] N. Mazur and M. Kuhrt, "Teaching programming concepts using a robot simulation", *Journal of Computing in Small Colleges*, Vol. 12 (5) 1997, pp. 4-11.
- [11] L. Stein, "Interactive programming: revolutionizing introductory computer science", *ACM Computing Surveys*, Vol. 28 (4es) 1996, pp. 103.
- [12] M. Noga, LegOS, saatavilla HTML-muodossa osoitteessa <URL: <http://legos.sourceforge.net/HOWTO/>>, haettu 19.12.2015.
- [13] B. Erwin, M. Cyr and C. Rogers, "Lego engineer and RoboLab: Teaching engineering with LabVIEW from kindergarten to graduate school." *International Journal of Engineering Education*, Vol. 16 (3) 2000, pp. 181-192.
- [14] J. Solorzano, LeJOS, saatavilla HTML-muodossa osoitteessa <URL: <http://www.lejos.org/>>, haettu 20.12.2015.
- [15] D. Baum, "Not Quite C (NQC)", saatavilla HTML-muodossa osoitteessa <URL: <http://www.enteract.com/~dbaum/nqc/index.html>>, haettu 14.11.2015.
- [16] LEGO NXT, johdanto teknisiin tietoihin ja ohjelmointiin, saatavilla PDF-muodossa osoitteessa <URL: <https://www.generationrobots.com/media/Lego-Mindstorms-NXT-Education-Kit.pdf>>, haettu 22.10.2017.
- [17] LEGO, EV3 on robotin uusi malli, tuotetiedot saatavilla osoitteessa <URL: <http://www.lego.com/fi-fi/mindstorms>>, haettu 21.11.2015.
- [18] LEGO NXT, syvällisempiä teknisiä ominaisuuksia ja ohjelmointiohjeita, saatavilla PDF-muodossa osoitteessa <URL: <http://www.legoengineering.com/wp-content/uploads/2013/06/download-tutorial-pdf-2.4MB.pdf>>, haettu 14.11.2015.
- [19] LEGO, LEGO NXT -robotin uusi BricxCC-ohjelmointiympäristö Windowsille, ladattavissa osoitteessa <URL: <http://sourceforge.net/projects/bricxcc/files/bricxcc/>>, haettu 20.11.2015.

- [20] JTAG, liitännän tiedot, saatavilla HTML-muodossa osoitteessa <URL: <https://www.xjtag.com/about-jtag/>>, haettu 19.12.2015.
- [21] Microchip, MPLAB-ohjelmankehitysympäristöt, saatavilla HTML-muodossa osoitteessa <URL: <http://www.microchip.com/mplab/mplab-x-ide>>, haettu 3.12.2015.
- [22] Microchip, ICD-ohjelmointi, saatavilla PDF-muodossa osoitteessa <URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/51264B.pdf>>, haettu 3.12.2015.
- [23] Sourceforge, Dev-C++ ohjelmointiympäristö, ladattavissa osoitteessa <URL: <https://sourceforge.net/projects/orwellddevcpp/files/?source=navbar>>, haettu 4.12.2015.
- [24] LEGO, Robottien historia, saatavilla HTML-muodossa osoitteessa <URL: <http://www.lego.com/fi-fi/mindstorms/history>>, haettu 20.12.2015.
- [25] Open Source, vapaan lähdekoodin ohjelma, saatavilla HTML-muodossa osoitteessa <URL: <https://opensource.com/resources/what-open-source>>, haettu 20.12.2015.
- [26] V. Dagdilelis, M. Sartatzemi and K. Kagani, "Teaching (with) Robots in Secondary Schools: some new and not-so-new Pedagogical problems", In Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), 2005.
- [27] S. H. Kim, J. W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms", IEEE Transactions on Education (Volume: 52, Issue: 1, Feb. 2009), 2009.
- [28] E. Irigoyen, E. Larzabal, R. Priego, "Low-cost platforms used in Control Education: An educational case study", IFAC Proceedings Volumes, 2013 – Elsevier, 2013.
- [29] A. A. Galadima, "Arduino as a learning tool", Electronics, Computer and Computation (ICECCO), 2014 11th International Conference, 29 Sept.-1 Oct. 2014.
- [30] P. Jamieson, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?" in Proc. FECS, 2010, pp. 289–294.

- [31] P. Jamieson, J. Herdtner, “More Missing the Boat - Arduino, Raspberry Pi, and Small Prototyping Boards and Engineering Education Needs Them”, *Frontiers in Education Conference (FIE)*, 2015 IEEE, 2015.
- [32] W. Albrecht, P. Bender, and K. Kussmann, “Integrating microcontrollers in undergraduate curriculum”, *Journal of Computing Sciences in Colleges*, vol. 27, no. 4, pp. 45–52, 2012.
- [33] Wiring-projekti, saatavilla HTML-muodossa osoitteessa <URL: <http://wiring.org.co/about.html>>, haettu 5.4.2016.
- [34] Processing-projekti, saatavilla HTML-muodossa osoitteessa <URL: <https://processing.org/overview/>>, haettu 5.4.2016.
- [35] Arduino-foorumi, saatavilla HTML-muodossa osoitteessa <URL: <http://forum.arduino.cc>>, haettu 5.4.2017.
- [36] Arduino-järjestelmät, saatavilla HTML-muodossa osoitteessa <URL: <https://www.arduino.cc/>>, haettu 10.5.2017.
- [37] Yleistä Arduino-ohjelmoinnista, saatavilla HTML-muodossa osoitteessa <URL: <http://www.arduino.cc/en/Reference/HomePage>>, haettu 10.6.2017.
- [38] LEGO-robottien opetussivusto, saatavilla HTML-muodossa osoitteessa <URL: <https://education.lego.com/en-us/support>>, haettu 20.10.2017.
- [39] MatrixTSL:n opetussivusto, saatavilla HTML-muodossa osoitteessa <URL: http://www.matrixttl.com/lc_index.php>, haettu 10.6.2017.
- [40] MatrixTSL:n PIC-ohjelmointialusta, saatavilla PDF-muodossa osoitteessa <http://www.matrixttl.com/datasheets/EB006-30-7.pdf>>, haettu 20.10.2017.
- [41] Lämpötila-anturin tekniset tiedot, saatavilla HTML-muodossa osoitteessa <URL: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>>, haettu 21.10.2017.

- [42] Fritzing-ohjelman kotisivut, saatavilla HTML-muodossa osoitteessa <URL: <http://fritzing.org/home/>>, haettu 21.10.2017.
- [43] A. Dabroom, W. M. Refie, R. Matmti, “Microcontroller-based Learning Kit: Course Design using Constructive Alignment Principles”, 21st Mediterranean Conference on Control & Automation (MED), Plataniias-Chania, Crete, Greece, June 25-28, 2013.
- [44] D. Mihai, C. Caramida, “Experimental Modular Platform for the Fuzzy Control of the Temperature Inside an Enclosure”, Industrial Informatics (INDIN), 8th IEEE International Conference, Osaka, Japan, July 13-16, 2010.
- [45] D. Mihai, “Designing an Experimental Platform for the Air Pressure in a Small Tank by Digital Control”, Applied and Theoretical Electricity (ICATE), 2016 International Conference, Craiova, Romania, October 6-8, 2016.

8 Liitteet

Liite 1

Taulukko 2. Tieto- ja tietoliikennetekniikan perustutkinnon muodostuminen

TIETO- JA TIETOLIIKENNETEKNIIKAN PERUSTUTKINTO
Tutkinnon muodostuminen ammatillisessa peruskoulutuksessa, 180 osp
2 Ammatilliset tutkinnon osat, 135 osp
3 Yhteiset tutkinnon osat, 35 osp
4 Vapaasti valittavat tutkinnon osat, 10 osp

Taulukko 2.1. Elektroniikka- ja ICT-asentajan pakolliset ammatilliset tutkinnon osat

TIETO- JA TIETOLIIKENNETEKNIIKAN PERUSTUTKINTO
2 Ammatilliset tutkinnon osat, 135 osp
2.1 Pakolliset tutkinnon osat, 75 osp
Tieto- ja tietoliikennetekniikan perustutkinto, elektroniikka-asentaja , pakolliset tutkinnon osat ovat 2.1.1 ja 2.1.2.
Tieto- ja tietoliikennetekniikan perustutkinto, ICT-asentaja , pakolliset tutkinnon osat ovat 2.1.1 ja 2.1.3.
2.1.1 Elektroniikan ja ICT:n perustehtävät, 45 osp
2.1.2 Ammattielektroniikka, 30 osp
2.1.3 Tietokone- ja tietoliikenneasennukset, 30 osp
(KPEDUn tarjoamat tutkinnon osat on merkitty harmaalla.)

Taulukko 2.2. Elektroniikka- ja ICT-asentajan valinnaiset ammatilliset tutkinnon osat

<p>2.2 Valinnaiset tutkinnon osat, 60 osp</p> <p>Tieto- ja tietoliikennetekniikan perustutkinto, elektroniikka-asentaja Opiskelijan on valittava tutkinnon osa 2.2.1 tai 2.2.2.</p> <p>Tieto- ja tietoliikennetekniikan perustutkinto, ICT-asentaja Opiskelijan on valittava tutkinnon osa 2.2.3 tai 2.2.4.</p> <p>Lisäksi opiskelijan on valittava 30 osaamispistettä tutkinnon osista 2.2.5–2.2.19 tai suorittamattomista tutkinnon osista 2.2.1–2.2.4.</p> <p>Tutkinnon osat 2.2.12 ja 2.2.13 vastaavat 30 osaamispistettä ammatillisessa peruskoulutuksessa.</p> <p>Tutkinnon osa 2.2.14 vastaa 15 tai 30 osaamispistettä sen mukaan, mitä on määritetty luvun 2 kohdassa 2.2.14.</p>
<p>2.2.1 Sulautetut sovellukset ja projektityöt, 30 osp</p> <p>2.2.2 Elektroniikkatuotanto, 30 osp</p> <p>2.2.3 Palvelinjärjestelmät ja projektityöt, 30 osp</p> <p>2.2.4 Tietoliikennelaiteasennukset ja kaapelointi, 30 osp</p> <p>2.2.5 Huoltopalvelut, 15 osp</p> <p>2.2.6 Valvonta- ja ilmoitusjärjestelmäasennukset, 15 osp</p> <p>2.2.7 Kodin elektroniikka ja asennukset, 15 osp</p> <p>2.2.8 RF-työt, 15 osp</p> <p>2.2.9 Sähköasennukset, 15 osp</p> <p>2.2.10 Yritystoiminnan suunnittelu, 15 osp</p> <p>2.2.11 Tutkinnon osa ammatillisesta perustutkinnosta, 15 osp tai 30 osp</p> <p>2.2.12 Tutkinnon osa ammattitutkinnosta</p> <p>2.2.13 Tutkinnon osa erikoisammattitutkinnosta</p> <p>2.2.14 Tutkinnon osa ammattikorkeakouluopinnoista</p> <p>2.2.15 Työpaikkaohjaajaksi valmentautuminen, 5 osp</p> <p>2.2.16 Yrityksessä toimiminen, 15 osp</p> <p>2.2.17 Huippuosaajana toimiminen, 15 osp</p> <p>2.2.18 Paikallisesti tarjottava tutkinnon osa, 15 osp</p> <p>2.2.19 Tutkinnon osat vapaasti valittavista tutkinnon osista, 5–15 osp</p> <p>(KPEDUn tarjoamat tutkinnon osat on merkitty harmaalla.)</p>

Taulukko 2.3. Elektroniikka- ja ICT-asentajan yhteiset tutkinnon osat

TIETO- JA TIETOLIIKENNETEKNIKAN PERUSTUTKINTO		
3 Yhteiset tutkinnon osat, 35 osp	Pakolliset	Valinnaiset
3.1 Viestintä- ja vuorovaikutusosaaminen, 11 osp	8 osp	3 osp
3.2 Matemaattis-luonnontieteellinen osaaminen, 9 osp	6 osp	3 osp
3.3 Yhteiskunnassa ja työelämässä tarvittava osaaminen, 8 osp	5 osp	3 osp
3.4 Sosiaalinen ja kulttuurinen osaaminen, 7 osp	-	7 osp
	19 osp	16 osp
<p>Tutkinnon perusteissa määrättyjen osa-alueiden (3.1.1–3.4.6) valinnaisten osaamistavoitteiden lisäksi tai niiden sijaan koulutuksen järjestäjä voi päättää erilaajuisia muita valinnaisia osaamistavoitteita. Koulutuksen järjestäjän itse päättämille valinnaisille osaamistavoitteille määritellään osaamisen arviointi ja laajuus osaamispisteinä. Tutkinnon osien valinnaiset osaamistavoitteet voivat olla myös opiskelijan aikaisemmin hankkimaa osaamista, joka tukee kyseisen tutkinnon osan ja sen osa-alueiden osaamistavoitteita.</p>		

Taulukko 2.4. Elektroniikka- ja ICT-asentajan vapaasti valittavat tutkinnon osat

TIETO- JA TIETOLIIKENNETEKNIKAN PERUSTUTKINTO
4 Vapaasti valittavat tutkinnon osat, 10 osp
4.1 Ammattitaitoa syventäviä tai laajentavia ammatillisia tutkinnon osia
4.2 Paikallisiin ammattitaitovaatimuksiin tai osaamistavoitteisiin perustuvia tutkinnon osia
4.3 Yhteisiä tutkinnon osia tai lukio-opintoja
4.4 Jatko-opintovalmiuksia tai ammatillista kehittymistä tukevia opintoja
4.5 Työkokemuksen kautta hankittuun osaamiseen perustuva yksilöllinen tutkinnon osa
5 Tutkintoa yksilöllisesti laajentavat tutkinnon osat

Liite 2

Arduino / ECIO -kysymykset molempien järjestelmien rauta- ja ohjelmistopuolesta, kuinka helppoa niitä oli käyttää ja oliko niistä hyötyä motivaation kannalta.

1. Oliko Arduino / ECIO -laitteen käyttö helppo oppia?
2. Oliko Arduinin / ECION ohjelmointiympäristö (IDE) helppo oppia käyttämään?
3. Oliko Arduino / ECIO -laitteen ohjelmoiminen helppo oppia?
4. Mitä mieltä olet Arduinin / ECION IDE:n Help:istä?
5. Mitä mieltä olet Arduinin / ECION nettisivuista?
6. Antoiko Arduino / ECIO sinulle ohjelmointimotivaatiota, niin että haluat itse opetella lisää?

Liite 3

Kysymykset käytännön harjoitustöiden vaikeusasteesta, kysymykset 1-6.

Valintavaihtoehdot: 1=helppo, 2=helpohko, 3=sopiva, 4=vaikeahko, 5=vaikea

1. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 1: Arduino, 3 tutustumisharjoitusta
2. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 2: Arduino, ovisummeri
3. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 3: Arduino, lämpötilan mittaus
4. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 1: ECIO, ralliautojen starttivalot
5. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 2: ECIO, kytkimen painallukset binäärisesti
6. Mitkä harjoitustyöt olivat vaikeimpia?
Harjoitustyö 3: ECIO, hämäräkytkin

Arduino:n ja ECIO:n väliset käytön vertailukysymykset, kysymykset 7-13.

Valintavaihtoehdot: Arduino on 1, Matrix on 2

7. Kumman laitteiston käytöstä pidit enemmän?
8. Kumman ohjelmointitavan käytöstä pidit enemmän?
9. Kumpaan laitteistoon oli helpompi tehdä harjoituskytkennät?
10. Kumpi laitteisto oli selkeämpi käyttää?
11. Kumman laitteiston/ohjelmiston avulla oli helpompi oppia?
12. Kumman järjestelmän ja ohjelmiston avulla olisi mielestäsi nopeampi ja helpompi rakentaa jokin näiden harjoitusten mukainen, omaan käyttöön tuleva toimiva, su-lautettu järjestelmä?
13. Kumpaa järjestelmää kokonaisuutena käyttäisit mieluummin?

Liite 4

Ensimmäinen ohjelmointiharjoitus laitteelle: Arduino UNO 101. 7.10.2016 / JSAL

Luokka:

Nimi:

Pvm:

ICT Arduino harjoitus 1A.

Tehdään ohjelma, joka vilkuttaa Arduino-kortilla olevaa lediä sekunnin välein.

Keltainen led on kytketty Arduinon digitaaliporttiin 13.

Tässä harjoitellaan pinMode- ja digitalWrite -komentojen käyttöä.

ICT Arduino harjoitus 1B

Tehdään ohjelma, joka lukee kytkimen tilan ja sytyttää kytkimen tilan

mukaan ledin. Kytkin kytketään digitaalinput porttiin 7 ja ledi on

digitaalinput portissa 2.

Tässä harjoitellaan pinMode- ja digitalWrite -komentojen käyttöä

Kytkeä tehdään koekytkentälevylle, käytä ledille 470 ohmin etuvastusta.

Muista tarkistaa ledin napaisuus, jotta se tulee kytkettyä oikein päin.

ICT Arduino harjoitus 1C

Tehdään ohjelma, joka lukee potentiometriltä jännitteen tason ja

tulostaa jännitteen arvon sarjamonitoriin, sekä ohjaa ledin valon voimakkuutta (PWM).

Potentiometri on kytketty analogia-porttiin A3 ja ledi digitaaliporttiin 10.

Tässä harjoitellaan analogRead-komennon käyttöä, sekä Sarjamonitori-ominaisuuden

liittämistä ohjelmaan, jotta voidaan tarkastella ohjelman tuottamaa tietoa ruudulla.

Huomioidaan analogia- ja digitaalisignaalien jännitetasojen määrät.

Kytkeä tehdään koekytkentälevylle, kytke +5V ja GND Arduinosta koekytkentälevylle.

Huomioi myös potentiometrin ja ledin oikeat kytkennät.

Harjoitus 1 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____

Liite 5

Toinen ohjelmointiharjoitus laitteelle: Arduino UNO 101.

26.10.2016 / JSAL

Luokka:

Nimi:

Pvm:

Harjoituksen tarkoitus: Harjoituksessa rakennetaan itse suunniteltu summerilla varustettu toimiston ovivalojärjestelmä.

Harjoituksissa järjestelmää rakennetaan osatehtävien kautta toimivaksi kokonaisuudeksi. Ilmoitukset annetaan valoilla, pyyntö ovelta sisään (Green + Summeri), varattu (Red), vapaa (Green).

Arduino ohjelmointiharjoitus 1A.

Ensin harjoitellaan hieman sitä, miten komponentti kytketään Arduinon porttiin sekä miten ohjelmallisesti ohjataan tiettyä ulostuloporttia.

Tehtävänä on rakentaa kytkentälevylle kytkentä, jossa on Led (Red) kytkettynä Arduinon digitaaliporttiin (Output). Led vilkuttaa valoa 2 sek. välein ohjelman ohjaamana.

Muista n. 470 ohmin etuvastus Ledille ja tarkista, että Led on kytketty oikein päin ja oikeisiin jännitteisiin.

Arduino ohjelmointiharjoitus 1B.

Sitten otetaan lisää komponentteja ja ulostuloportteja käyttöön.

Täydennä kytkentää siten, että siinä on kolme erityyppistä lediä, Red, Green ja Yellow.

Lisää ohjelmaan koodia niin, että kaikki valot syttyvät vuorotellen ja palavat 1 sek. ajan, sekä ohjelma pyörii silmukassa koko ajan.

Arduino ohjelmointiharjoitus 1C.

Lisää kytkentään vihreää valoa varten kytkin (PushButton), eli harjoitellaan lisäksi tiedon lukemista sisään mikrokontrolleriin. Ohjelmaa muutetaan kytkennän mukaisesti. Alussa kaikki Ledit ovat sammuneina. Sitten ohjelmassa luetaan PushButton-kytkimen asento, ja sitä vastaava vihreä ledi syttyy ja palaa sen aikaa, kun kytkintä painetaan.

Kahden kytkimen käytön erot kytkennöissä. PushButton: Ledi palaa sen aikaa, kun kytkintä painetaan. Silloin kun kytkin vapautetaan, niin vastaava ledi sammuu ja pysyy sammuneena. OnOff: Kun kytkin käännetään ON-asentoon, niin Ledi palaa jatkuvasti, ja OFF-asennossa ledi sammuu ja pysyy sammuneena.

Lopullinen kytkentä:

Tämä on omanlainen yksinkertaistettu ovsummerikytkentä, keksitty tätä harjoitusta varten.

Kytkeä ja ohjelmaa muutetaan vastaamaan tätä selostusta. Ledien, sekä summerin ohjaukset toimivat ohjelmallisesti. Kytkiessäsi komponentteja, ole huolellinen Ledien ja kytkimien järjestyksestä, eli mikä tarkoitus niillä on.

Ulkopuolelle ajatellaan kaksi valoa (ovat kytkentälevyllä), punainen ja vihreä sekä yksi PushButton-kytkin. Toimistoon kytketään 2 kytkintä (varattuna Red=OnOff, vapaana Green=PushButton) toimiston käyttöön, joilla ilmoitetaan varaustilanteesta, ja yksi vihreä Led.

Toimistossa sisään-kytkintä painetaan => vihreä Ledi syttyy ovella. Toimistossa varattuna-kytkintä painetaan => punainen ledi syttyy ovella.

Lopuksi liitetään kytkentälevylle vielä Summeri, joka kytketään johonkin ulostuloporttiin.

Sen jälkeen, ovella summeri soi sekä vihreä Led syttyy silloin, kun sisältä toimistosta (Green-PushButton) annetaan lupa tulla toimistoon. Vihreä ledi palaa ja summeri soi niin kauan kuin kytkin on päällä. Ohjelmaa muutetaan tätä vastaavaksi.

Silloin kun oven ulkopuolista kytkintä painetaan, niin vain vihreä Led syttyy ja palaa toimistossa, niin kauan kuin kytkintä painetaan.

Ohjelma tehdään niin, että vain yksi kytkin kerrallaan on käytössä, muut eivät toimi silloin. Saadaan yksinkertaisempi kytkentä koekytkentälevylle.

Silloin kun vihreä tai punainen Led palaa, niin muut Ledit ovat sammuneina, eivätkä muut kytkimet myöskään toimi. Kaksi Ledia ei pala yhtä aikaa.

Kun ”asiakas” on sisällä, kytketään toimistossa punaisen valon kytkin päälle => punainen Led ovella jää päälle.

Ohjelmaa muutetaan tuota toimintaa vastaavaksi.

Lopuksi piirrä kytkentä Fritzing-ohjelmalla.

Harjoitus 2 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____

Liite 6

Kolmas ohjelmointiharjoitus laitteelle: Arduino UNO 101

2.11.2016 / JSAL

Luokka:

Nimi:

Pvm:

Harjoituksen tarkoitus: Harjoituksessa rakennetaan lämpötilan mittauslaitteisto yhdellä lpt-anturilla (DS18B20).

Yleistä: Lämpötilasta annetaan ilmoitus sarjaliikennemonitoriin.

Vinkki: Rakenna kytkentää ja ohjelmaa toiminto kerrallaan -> vähemmän vikamahdollisuuksia!!!

Ohjelmaan tulee liittää kirjastotiedostot lämpötila-anturia varten, eli harjoituksena on myös Arduinon kirjastojen käyttö.

Tehtävänä on rakentaa yhden lpt-anturin (DS18B20) kytkentä kytkentälevylle, ja siitä yhteydet Arduinoon.

Tee lpt-anturin kytkentä ns. parasitic-kytkennällä (eli kahdella johdolla, katso kytkentäohje).

Tee ohjelma, joka ilmoittaa sarjaliikennemonitoriin löytyneiden anturien määrän ja mitatun lämpötilan. Testaa, että sarjamonitorissa lämpötila muuttuu, kun lämmität/kylmennät anturia.

Lopuksi piirrä kytkentä Fritzing-ohjelmalla.

Harjoitus 3 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____

Liite 7

ECIO Ohjelmointiharjoitus 1.

12.3.2017 / JSAL

Luokka:

Nimi:

Pvm:

Harjoituksen tarkoitus: Harjoituksessa ohjelmoidaan ralliautoille starttivalot.

Tässä ohjelmassa harjoitellaan tiedon lukua mikrokontrollerin portista/pinnistä (Input), sekä tiedon syöttämistä kontrollerin porttiin/pinniin (Output).

ECIO kytketään lisäkorttiin (Ecio Base Board), jossa on 5 x 9Pin-D-Sub lisäkorteille. ECIO:lle ja näille lisäkorteille riittää USB-liitäntän jännite, ei tarvita lisäpoweria.

Lisäkortteina käytetään:

- kytkinkorttia (yht. 8 kytkintä) portissa A.
- käyttöjännite (+V jännite) on kytkettävä kytkinkortille, jotta se toimisi
- ground (-V jännite) tulee liittimen rungon kautta
- ledikortti (yhteensä 8 kirkasta lediä) kytketään porttiin D.
- ledikortti ei tarvitse lisäjännitettä

Kun ohjelma käynnistyy alussa, tai ECIO:n Reset-näppäintä painettaessa, syttyvät kaikki 8 lediä 2 sek:ksi.

Lähtölaskenta alkaa siitä, kun painetaan kytkinkortilta kytkintä A0 (fyysisesti 1 kytkin).

Ledi D0 syttyy 1 sekunnin kuluttua ohjelman käynnistymisestä (GetReady).

Tästä 2 sekunnin kuluttua syttyy ledi D1 (Wait), ledi D0 palaa koko ajan.

D1:n jälkeen on 3 sekunnin viive ja sitten syttyy ledi D2 (Go).

Kaikki kolme lediä palavat ja pysyvät päällä 3 sek, jonka jälkeen kaikki ledit sammuvat.

Autot starttaavat silloin kun kaikki kolme lediä palavat.

Harjoitus 1 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____

Liite 8

ECIO Ohjelmointiharjoitus 2.

14.3.2017 / JSAL

Luokka:

Nimi:

Pvm:

Harjoituksen tarkoitus: Harjoituksessa ohjelmoidaan kytkinpainalluksille binäärilaskuri.

Tässä ohjelmassa harjoitellaan tiedon lukua mikrokontrollerin portista/pinnistä (Input), tiedon syöttämistä kontrollerin porttiin/pinniin (Output), sekä ehtolauseiden tekemistä.

ECIO kytketään lisäkorttiin (Ecio Base Board), jossa on 5 porttia (D-Sub 9Pin) lisäkorteille.

ECIO:lle ja näille lisäkorteille riittää USB-liitännän jännite, ei tarvita lisäpoweria.

Lisäkortteina käytetään:

- kytkinkorttia (yht. 8 kytkintä) portissa E.
- käyttöjännite (+V jännite) on kytkettävä kytkinkortille, jotta se toimisi
- ground (-V jännite) tulee liittimen rungon kautta
- ledikortti (yht. 8 kirkasta lediä) kytketään porttiin B.
- ledikortti ei tarvitse lisäjännitettä

Ohjelman käynnistyessä alussa tai ECIO:n Reset-näppäintä painettaessa sammuvat kaikki ledit.

Kytkimen painallusten laskenta alkaa siitä, kun painetaan kytkinkortilta kytkintä E0 (fyysisesti 1 kytkin).

Ledi B0 syttyy ensimmäisellä kytkimen painalluksella, ja sitten seuraavalla B1 ja B0 sammuu jne. binäärilukujen mukaisesti.

Kaikki neljä lediä palavat, kun kytkintä on painettu 15 kertaa ja pian sen jälkeen kaikki ledit sammuvat.

Harjoitus 2 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____

Liite 9

ECIO Ohjelmointiharjoitus 3.

27.3.2017 / JSAL

Luokka:

Nimi:

Pvm:

Harjoituksen tarkoitus: Harjoituksessa rakennetaan hämäräkytkin LDR-vastuksella ja tehdään ohjelma, joka lukee valovastuksen (LDR = Light Dependent Resistor) tilaa, ja sytyttää LED:in silloin kun on riittävän hämärää. Potentiometrillä asetetaan sopiva hämäräarvo.

Tässä ohjelmassa harjoitellaan tiedon lukua mikrokontrollerin portista/pinnistä (Input), tiedon syöttämistä kontrollerin porttiin/pinniin (Output), sekä ehtolauseiden tekemistä. Samalla tutustutaan myös LDR-vastuksen toimintaan, kun sitä sovelletaan hämäräkytkimenä.

ECIO kytketään koekytkentäalustalle, johon kytketään muut komponentit kiinni mikrokontrollerin oikeisiin pinneihin. Kytkentäpiirustus annetaan piirikaaviona.

ECIO:lle riittää USB-liitännän jännite, ei tarvita lisäpoweria.

Lisäkomponentteina käytetään:

- vastus 470 ohm
- LED
- lineaarinen potentiometri, 10 kohm
- LDR

Ohjelman käynnistyy painettaessa ECIO:n Reset-näppäintä. Säädä potentiometriä niin, että LED juuri syttyy, ja sitten takaisin sen verran että LED sammuu. Kokeile peittää LDR-vastus, LED:in pitäisi nyt syttyä. LED sammuu, kun päästät taas valoa LDR:lle.

Elektroniikan kertauksena käydään läpi LDR:n toimintaperiaate ja se, mitä eroa on lineaarisella ja logaritmisella potentiometrillä, sekä niiden käyttökohteita.

Harjoitus 3 loppuu tähän.

Ohjelma ja kytkentä esitelty opettajalle: Pvm: _____ Kuittaus: _____