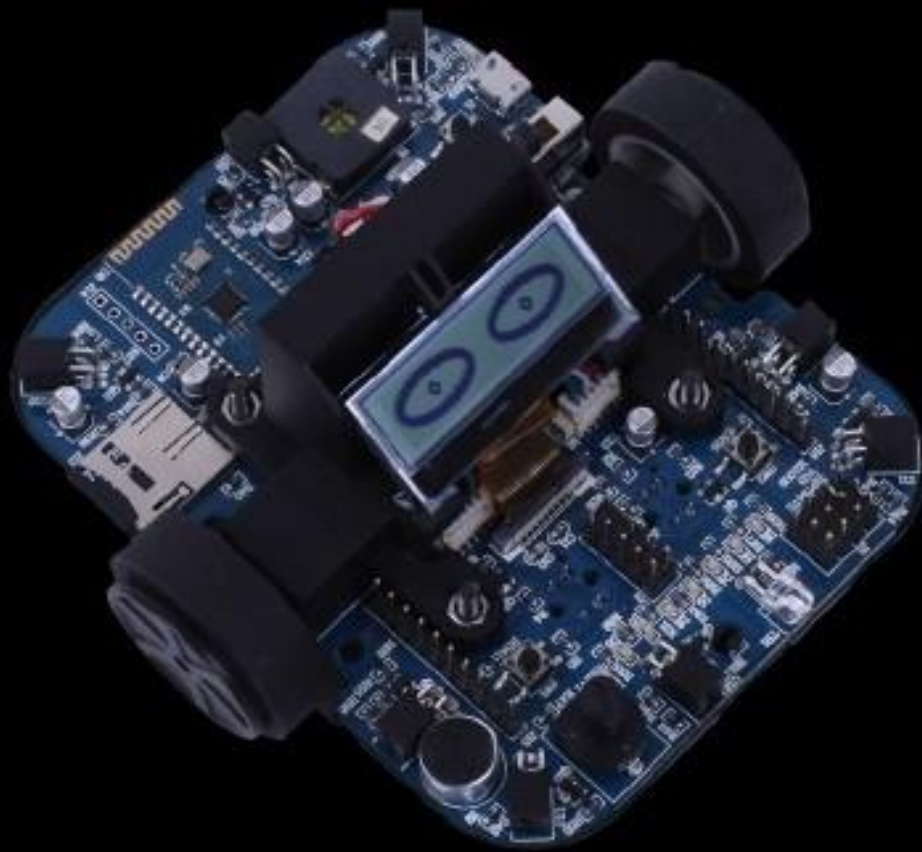


Robotiikkakurssi - opetusopas



CP5894

MATRIX

www.matrixtsl.com

Sisältö

Robotiikan kurssi Ohjeellinen opas

Johdanto

Aloittaminen	3
Tutustulaitteeseen - Robotin osat	4
Kielineutraali - käyttöliittymä	8
Pseudokoodi - Yksinkertainen ohjelmoinnin apuväline	10

Laitteiston asennus

Windows PC	11
Android-tabletti/puhelin	13
Raspberry Pi ja Linux	15

Robotin ohjaaminen

Flowcoden käyttäminen	17
App Inventorin käyttäminen	19
Käyttämällä C++ / C# / VB	22
Pythonin käyttö	24
LabView:n käyttö	26
Scratchin käyttö	27

Työlehdet

Harjoitus 1 - Robotin liikkeet	29
Harjoitus 2 - LED-valot	31
Harjoitus 3 - Kytke se päälle, sammuta se	33
Harjoitus 4 - Hei maailma	35
Harjoitus 5 - Kuinka kirkas tuo valo on?	38
Harjoitus 6 - Valoa etsimässä	40
Harjoitus 7 - Seuraa linjaa	42
Harjoitus 8 - Painikkeet tekevät työn	45
Harjoitus 9 - Näyttöpaneeli	47
Harjoitus 10 - Kallista ja käännä (mobiililaitteen avulla)	49
Harjoitus 11 - Robotti osaa suunnistaa labyrintin läpi	51
Harjoitus 12 - Soita musiikkia	53
Harjoitus13 - Robo-DJ	55
Harjoitus 14 - Vakoilurobotti	57
Harjoitus 15 - Robo-Pop	60
Jatketaan eteenpäin - Haasteet 16-20	62

Liite

API-viitekartta	63
Mikroprosessorin nastaliitännät	67
Nuottien taajuudet	68
Robotin nimen asettaminen	69
AllCode API -laiteohjelmiston uudelleen lataaminen	70

Johdanto

Aloittaminen

Robotiikan kurssi

Ohjeellinen opas

Mistä aloittaa?

Onnittelut. Sinulla on nyt huipputekninen Formula AllCode robotti, joka voi suorittaa kaikenlaisia mielenkiintoisia tehtäviä. Se voi liikkua, navigoida labyrintissa, tanssia, tehdä ääniä, soittaa musiikkia, äänittää ja toistaa ääntä ja paljon muuta. Koska robotilla voi tehdä niin monia asioita, ensimmäinen ongelma, johon saatat törmätä, on löytää paras tapa aloittaa.

Tämä opas on laadittu auttamaan sinua oppimaan mahdollisimman nopeasti, jotta saat robotin tekemään haluamasi asiat.



Hauskaa

Yksi tämän oppaan tärkeimmistä elementeistä on varmistaa, että sinulla on hauskaa. Uuden taidon oppiminen tai hankkiminen on aina paljon palkitsevampaa, kun tehtävään liittyy paljon iloa. Tämän opaskirjan ohjeet on kehitetty antamaan tuntikausia hauskanpitoa ja nautintoa samalla kun opit robotiikan perusteet. Robotin saaminen tekemään ensimmäiset liikkeensä ja äänteleämään ensimmäiset äänensä saattaa olla aluksi haastavaa, mutta sen saavuttamisen pitäisi olla mielenkiintoista ja puhdasta iloa.

Luovuus

Vaikka voit käyttää robottia yksin, se on paljon hauskempaa, kun teet sen yhdessä yhden tai useamman kaverisi kanssa, sillä teillä kaikilla on varmasti ideoita ja kokeilemisen arvoisia asioita. Tulet yllättymään, miten yksinkertainen idea voi muuttua yhteisellä tekemisellä joksikin todella upeaksi!

Rakennuspalikat

Tämän oppaan ohjeet on jäsennelty siten, että niissä käsitellään ensin helppoja asioita, kuten ledien käyttöä tai viestin näyttämistä nestekidenäytöllä, ja sitten siirrytään monimutkaisempiin tehtäviin, kuten sokkelossa liikkumiseen tai polun seuraamiseen. Jokaisessa tehtävässä on värillinen palkki, joka osoittaa ehdotetun taitotason - helppo, keskivaikea tai edistynyt.

Joidenkin tehtävien harjoitukset ovat "itsenäisiä", mikä tarkoittaa, että voit kokeilla niitä haluamassasi järjestyksessä. Toiset ovat "linkitettyjä", ja niissä käytetään rakennuspalikoita ja toimenpiteitä joihin olet törmännyt aiemmassa toiminnassa. Tämä näkyy värillisellä palkilla työarkin yläreunassa.

Työn suunnittelu

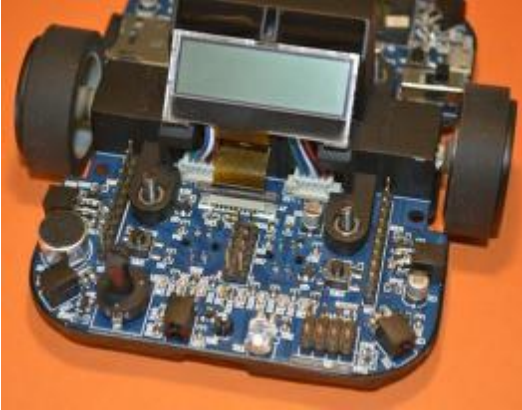
On hyvä aloittaa tekemällä suunnitelma siitä, mitä haluat saavuttaa joka kerta, kun teet harjoituksia.

Tutustu laitteeseen

Robotin osat

Robottiikan kurssi

Ohjeellinen opas

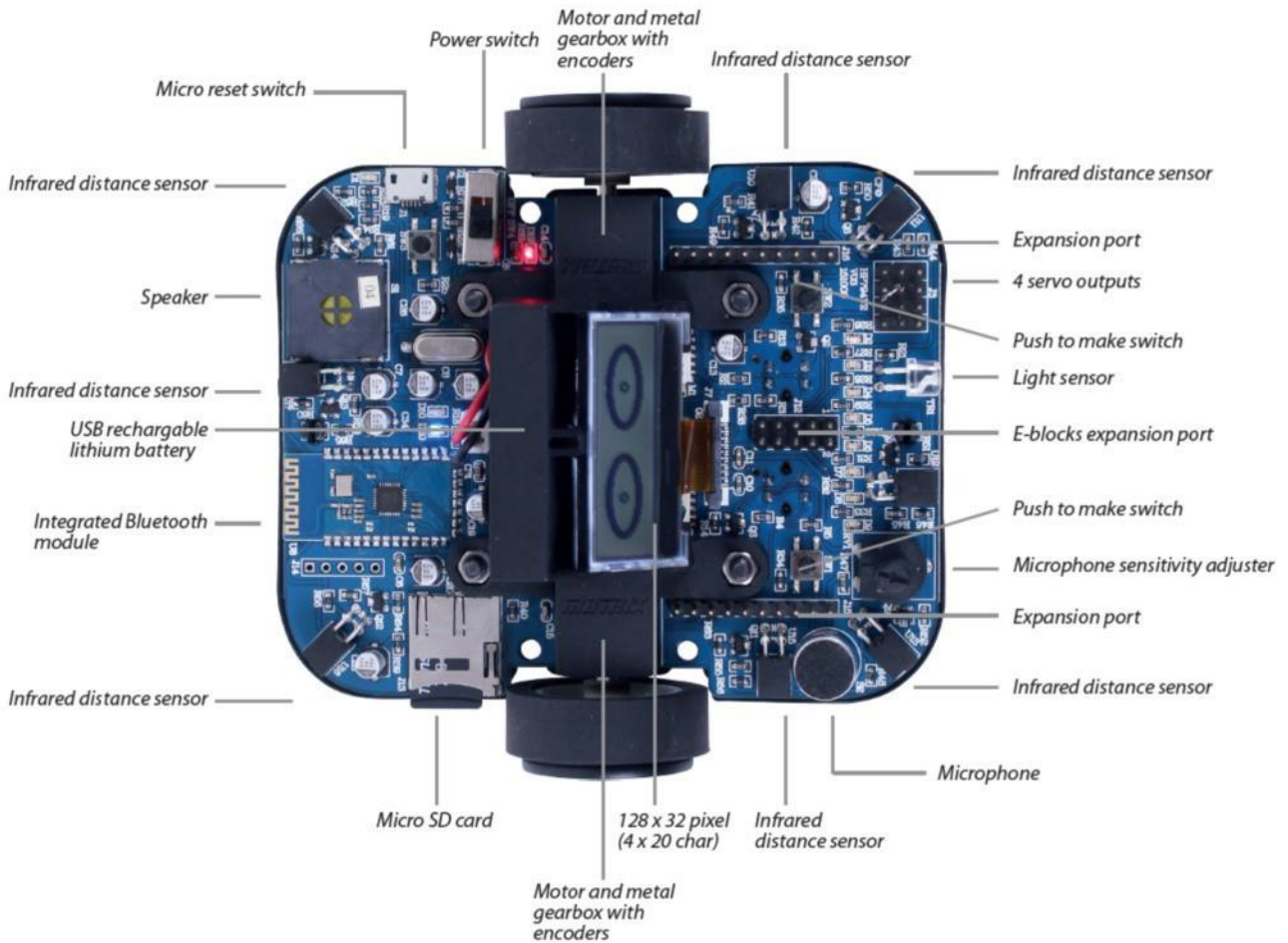


Tässä osassa tarkastellaan robotin eri osia ja selvitetään kaikkien antureiden ja ohjausjärjestelmien perustoiminnot.

Kahden robotin ohjaamiseen tarkoitetun moottorin lisäksi robotissa on valo- ja etäisyysanturit sekä äänentallennuslaitteisto. Nämä yhdessä tarjoavat tekemistä tuntikausiksi. Voit palata tälle sivulle kun teet harjoituksia ja haluat katsoa osien sijoittelua.

Robotin tärkeimmät osat

Alla olevassa lohkokaaviossa näkyvät Formula AllCode -robotin osat.



Tutustuminen

Robotin osat

Robottiikan kurssi Ohjeellinen opas

Digitaalinen signaalihojain, mikroprosessori

Robotin sydämenä on digitaalinen signaalisäädin (dsPIC®). Sen valmistaja on Microchip Technology Inc. Tämä suorituskykyinen laite suorittaa moottorinohjauksen (PWM), kehittyneitä analogisia toimintoja, useita tietoliikennetoimintoja, mukaanlukien äänen tallennus, äänen käsittely ja toisto. Prosessori pystyy tekemään nopeita laskutoimenpiteitä. .



Moottorit

Robotissa on kaksi voimakasta kestmagneettimoottoria, niissä on alennusvaihteisto vääntömomentin lisäämiseksi. Moottorit käyttävät kumipyöriä. Erittäin tehokas elektroninen ohjausjärjestelmä varmistaa, että akun virrankulutus on mahdollisimman pieni. Tämä tarkoittaa sitä, että robotin kanssa voi pitää pitkäänhauskaa ennen kuin litiumioniakku on ladattava. Molempia moottoreita voidaan käyttää toisistaan riippumatta. Näin robotti pystyy kääntymään omalla akselillaan ja ohittamaan tiukat tai hankalat tilanteet.

Moottoreissa olevien takaisinkytkentäkoodereiden ansiosta dsPIC® "tietää", tarkalleen, kuinka paljon kukin pyörä on kääntynyt. Robotti voi liikkua tietyn matkan eteenpäin/ taaksepäin tai kääntyä tietyssä kulmassa.



LCD-paneeli

Robotin keskellä on graafinen LCD-paneeli, jota voidaan käyttää viestien ja/tai yksinkertaisten graafisten muotojen näyttämiseen.



LEDit

Robotin etureunassa on rivi vihreitä LEdejä. Niitä voidaan ohjata robotin ohjelman komennoilla.

Valoanturi

Robotin etureunassa on valoanturi. Sitä voidaan käyttää mittaamaan robottiin osuvan valon määrää. Robotin ohjelma voi käyttää valoanturin tietoa.

Jos valoanturin tieto yhdistetään moottoreiden ohjaukseen, robotti löytää tai välttää valonlähteen.



Tutustuminen

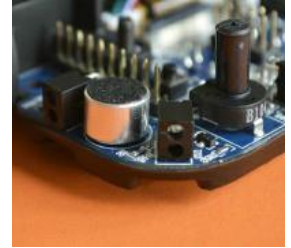
Robotin osat

Robotiikan kurssi

Ohjeellinen opas

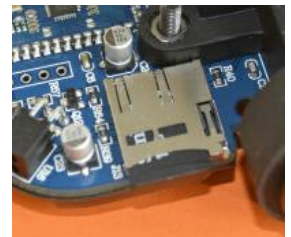
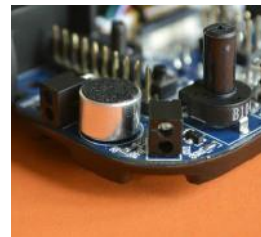
Etäisyysanturit

Robotin neljälle sivulle ja neljään kulmaan on asennettu infrapunälähtettimet ja -vastaanottimet. Ne mittaavat ja raportivat ohjelmalle robotin etäisyyden mahdollisesta esteestä. Kutakin kahdeksasta IR-lähtetimestä voidaan käskä lähettämään infrapunavalonsäde. Jos säde kohtaa esteen, se heijastuu takaisin robottiin, jossa IR-vastaanotin havaitsee sen.



Mikrofoni, microSD-korttipaikka ja minikaiutin

Nämä kolme osatekijää toimivat yhdessä, jotta äänet voidaan nauhoittaa (mikrofonin avulla), tallentaa microSD-kortille ja toistaa robotin sisäisen kaiuttimen avulla.



Ennalta nauhoitetut äänet ja musiikki, joka on tallenttu laitteeseen, voidaan toistaa kaiuttimesta.

Robotin tuottamat nuotit voidaan soittaa ja yhdistää toisiinsa, näin voidaan tehdä musiikillisia ääniä.



Micro USB-liitin

TMicro USB-liittimellä on kaksi toimintoa. Sitä voidaan käyttää robotin litiumakun lataamiseen liittämällä siihen laturi, jollaista luultavasti käytät matkapuhelimesi kanssa. Se voidaan myös liittää tietokoneeseen/kannettavaan tietokoneeseen/tablettiin USB-Johdolla, jolloin robottiin voidaan ladata tiedostoja tietokoneelta

Sisäinen Bluetooth-moduuli

Tämä on pieni komponentti, joka on asennettu laitteen takareunaan. Se tarjoaa toisen tavan ladata ohjelmia robottiin.

Yksi sen tärkeimmistä käyttötarkoituksista on mahdollistaa langattoman yhteyden muodostaminen robottiin ja sen jälkeen vuorovaikutus robotin sisäisten antureiden kanssa, moottoreiden ohjaaminen, viestien kirjoittaminen nestekidenäytölle, LEDien käyttäminen ja muut vastaavat toiminnot



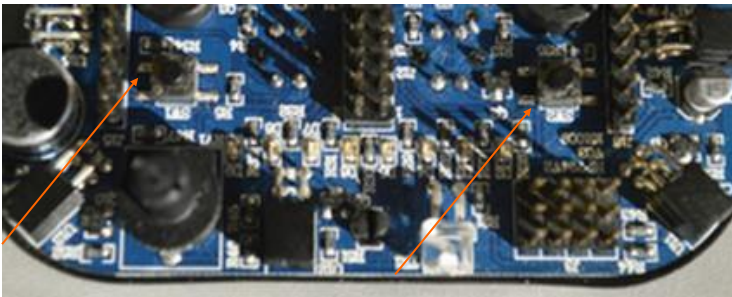
Tutustuminen

Robotin osat

Robotiikan kurssi Ohjeellinen opas

Painikkeet

Robotissa on kolme painiketta, jotka on merkitty SW1, SW2 ja SW3. Tietokoneohjelmasi voi käyttää kahta ensimmäistä painiketta käyttäjän määrittelemän tehtävän käynnistämiseksi. SW3 suorittaa laitteiston nollauksen ja käynnistää sen uudelleen.



Liukukytin

Tämä on merkitty nimellä SW4, ja se on robotin pääkytkin On/Off.



Linjatunnistimet

Jos käänät robotin ympäri, näet kaksi viivatunnistinta. Kumpikin yksikkö lähettää valonsäteen. Riippuen siitä minkälainen pinta (heijastava tai heijastamaton) robotin alla on, määräytyy kuinka suuri osa sädestä palaa takaisin

Viivatunnistimia käytetään ensisijaisesti siihen, että robotti seuraa paperille piirrettyä tai matolle tulostettua polkua.

Kielineutraali

Sovellusohjelman käyttöliittymä

Robottiikan kurssi Ohjeellinen opas



Kieliriippumaton, agnostinen, kielineutraali, alustariippumaton - mitä nämä termit tarkoittavat?

Periaatteessa se tarkoittaa, että sinun ei ole pakko käyttää tiettyä ohjelmointikieltä tai -alustaa robotin ohjaamiseksi.



Tämä johtuu siitä, että Formula AllCode tarjoaa sovelluksen (API), jonka avulla voit olla vuorovaikutuksessa robotin kanssa käyttämällä yksinkertaisia käskyjä tai protokollia.

Ohjelmointirajapinta API, Application Programming Interface

Yksi tapa selittää tämä on se, miten television kaukosäädin toimii. Vaikka televisioissa on kosketusnäppäimiä tai kosketuksella toimivia alueita näytön reunoilla, niin monien mielestä on kätevämpää käyttää kaukosäädintä television kytkemiseen päälle/pois, kanavan vaihtamiseen, äänenvoimakkuuden tai kirkkauden säätöön jne.

TV:n kaukosäädin on hyvin yksinkertainen. Se koostuu näppäimistöstä ja infrapunavalonsäteestä. Kun painiketta painetaan, sen arvo koodataan ja sitä käytetään lähettämään infrapunäsäteen kautta binäärikuvio televisioon.

Televisio purkaa vastaanotetun kuvion ja suorittaa halutun toiminnon.

Jos jostain syystä television kaukosäädin ei toimisi, olisi - helppo korvata se uudella, tai jopa ostaa yleiskaukosäädin (jos sinulla on useita laitteita ohjattavana).

On olemassa sovelluksia, joiden avulla älypuhelimista voi tehdä television kaukosäätimen. Saatavilla on myös laitteita, jotka voivat lähettää television kaukosäätimen koodeja. Ainoa kriittinen asia on lähettää oikeanlainen komento kun sitä tarvitaan.

Televisiossa on siis API, jonka avulla kaukosäädin (missä muodossa tahansa) voi ohjata television toimintoja tai elektronisia ohjausjärjestelmiä.

Formula AllCode -robotissa käytetty sovellusliittymä tarjoaa saman alustariippumattomuuden ja kieliriippumattomuuden kuin edellä kuvattu TV-esimerkki. Erona on lähetys, robotin välineenä on Bluetooth eikä infrapunäsäde. Tämä tarkoittaa sitä, että jos järjestelmässäsi on Bluetooth, voit käyttää sitä robottisi kanssa.

Voit käyttää robotin ohjaamiseen Bluetooth-yhteensopivaa matkapuhelinta. Vaihtoehtoisesti voit ottaa Bluetooth-yhteyden PC/Mac/Raspberry Pi® tai Matrix moniohjelmointikortin Bluetooth-moduulin avulla.



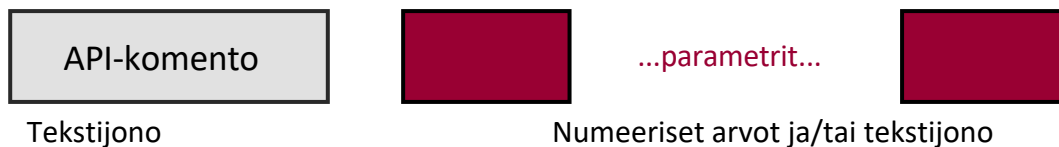
Kielineutraali

Sovellusohjelman käyttöliittymä

Robotiikan kurssi Ohjeellinen opas

Koska API reagoi yksinkertaiseen tekstipohjaiseen protokollaan, sinulla on vapaus käyttää mahdollisia ohjelmointikieliä, kuten C, C#, C++ tai Python, tai graafisia tai muita ohjelmointikieliä sekä kuvakepohjaisia kieliä, kuten Flowcode, App Inventor tai LabView.

Toinen asia, joka on syytä huomioida API:ssa, on se, että jotkin komennot ovat kaksisuuntaisia. Tämä tarkoittaa, että robotille lähetetty komento voi johtaa paluuviestiin. Hyvä esimerkki tästä ovat robottiin asennetut infrapunaetäisyysanturit. API-komento voitaisiin lähettää määritetylle anturille, jolloin se antaisi paluuviestinä numeerisen arvon robotin ja esteen välisestä etäisyydestä.



Jokainen komento alkaa merkkijonolla, joka määrittää, mitä robotin pitäisi tehdä. Tätä voi seurata yksi tai useampi parametri. Riippuen siitä, mitä robotilla yritetään tehdä, nämä parametrit voivat olla numeerisia tai tekstimuotoisia tai niiden sekoitus.

Jos haluat esimerkiksi lähettää arvon robotin etupuolella oleville kahdeksalle LED-valolle, voit käyttää seuraavia komentoja:

LEDWrite <arvo>

Koska LEDit on ryhmitelty yhteen ja muodostavat 8-bittisen rivin, niitä voidaan ohjata lähettämällä niihin binääriluku. Parametri <arvo> voi siis olla arvon 0 ja 255 välillä.

On huomattava, että tietyn kielen API-komennoilla voi olla erilaisia ominaisuuksia. Esimerkiksi Pythonissa käytetään "fa.LEDWrite(20)", kun taas C#:ssa käytetään "FA_DLL.FA_LEDWrite(20);", ja App Inventorissa, Flowcodessa ja LabView:ssä valitaan sopiva kuvake.

Tässä on toinen esimerkki, jossa näytetään, miten robotin moottoreita ohjataan.

Forward <etäisyys>

Parametrin <etäisyys> arvo voi olla välillä 1-1000, ja se määrittää etäisyyden, jonka robotti kulkee (millimetreinä) suorassa linjassa.

Jos haluat ohjata kahta moottoria itsenäisesti, voit käyttää tätä komentoa.

Setmotors <left> <right>

Parametrit <vasen> ja <oikea> voivat saada arvoja -100-100, jolloin kunkin moottorin nopeus ja suunta voidaan määrittää, jolloin robotti voi liikkua eri suuntiin.

Pseudokoodi

Yksinkertainen ohjelmoinnin apuväline

Robotiikan kurssi
Ohjeellinen opas

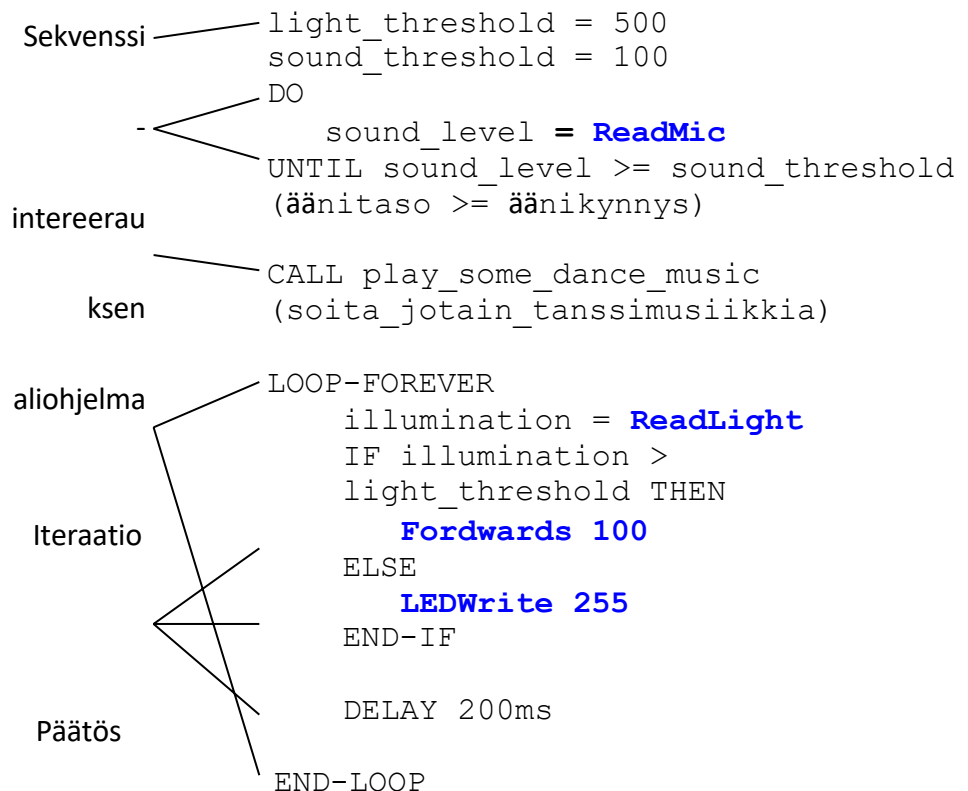
Useimmat tässä oppaassa esiintyvät esimerkit on kirjoitettu pseudokoodilla.

Saatat kysyä, miksi emme käyttäneet vuokaavioita tai jotakin Formula AllCode Robotin kanssa käytettävistä ohjelmointikielistä.

Vastaus on kaksitahoinen. Ensinnäkin tämä opas olisi valtava, jos kaikki esimerkit olisi kirjoitettu jokaisella robotin tukemalla kielellä ja käyttäisit vain yhtä niistä.

Toiseksi, jos käytät vuokaavioita, saattaisit luoda rakenteita, jotka eivät sovellu kolmeen tärkeimpään valvontarakenteeseen esim. sekvenssi, päätös ja toistuvat toiminnot (iteraatio).

, joita formaalit ohjelmointikieliet tukevat. Tässä on esimerkki ohjelmasta:



Sekvenssi on ryhmä yhtä tai useampaa lausetta, jotka seuraavat toisiaan. Tässä pseudokoodissa on kaksi esimerkkiä iteroinnista: ensimmäinen toistuu, kunnes tiettyyn kohtaan tapahtuma tapahtuu ja toinen toistuu ikuisesti. Päätös, jota joskus kutsutaan haaraksi, voi kulkea kahta polkua riippuen kysymykseen annetusta vastauksesta.

Tässä esimerkissä näkyy myös CALL, joka hyppää aliohjelmaan. Tätä käytetään tekemään koodista helpommin luettavaa piilottamalla joitakin tarpeettomia yksityiskohtia

Tummansinisellä korostetut kohdat ovat API-kutsuja, joiden avulla voit olla vuorovaikutuksessa robotin kanssa.

Näiden kutsujen nimet vastaavat Flowcoden, App Inventorin, Pythonin ja muiden Formula AllCode Robotin tukemien kielten makroja. Käyttämällä pseudokoodia voit ottaa ensimmäisen askeleen ideoiden toteuttamiseksi ilman, että joudut opiskelemaan virallisen ohjelmointikielen yksityiskohtia.

Laitteiston asennus

Windows PC

Robottiikan kurssi
Ohjeellinen opas



Monet Windows-laitteissa, erityisesti kannettavissa tietokoneissa ja tableteissa on sisäänrakennettu Bluetooth-toiminto. Jos tietokoneessasi ei ole, sinun on käytettävä Bluetooth USB donglea.

Windows-käyttöjärjestelmän eri versiot käyttävät hieman erilaisia tapoja yhdistää Bluetooth-laitteet, mutta kaikki noudattavat samoja ohjeita.

Sinun tarvitsee tehdä tämä prosessi vain kerran

1) Ota Bluetooth käyttöön

Bluetooth on usein oletusarvoisesti käytössä, ja voit yleensä jättää tämän vaiheen huomiotta. Jos se ei kuitenkaan ole, se voidaan ottaa käyttöön Windowsin asetuksista ja/tai ohjauspaneelistä. Hyvin harvoin Bluetooth on kytkettävä päälle erityisellä kytkimellä tai toimintonäppäimellä. Tietokoneen tai Windowsin ohjeesta löydät lisätietoja.

Manage Bluetooth devices

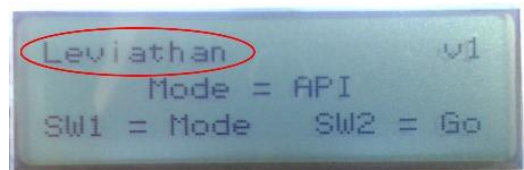
Bluetooth



2) Yhdistä robotti

Kytke robotti ensin päälle - sen nimi näkyy näytössä. näytön vasemmassa yläkulmassa.

Jälleen kerran, pariliitos toimii hieman eri tavoin eri Windows-versioissa, joten on vaikea antaa tarkkoja ohjeita. Windowsin ohjeessa ja verkkosivustolla on oppaita, joissa kerrotaan, miten yhteys muodostetaan



Kun muodostat pariliitoksen, näyttöön tulee luettelo käytettävissä olevista Bluetoothlaitteista. Valitse laite, jolla on robotin nimi, ja valitse Seuraava tai Pariliitos.

Sinua pyydetään syöttämään yhdistämiskoodi. Formula AllCode -robotti käyttää oletuskoodia 1234, mutta tämä voidaan vaihtaa toiseen koodiin jos haluat varmistaa ettei kukaan muu voi muodostaa pariliitosta robotin kanssa.

Kun koodi on syötetty, Windows vahvistaa, että se on yhdistetty robotin kanssa.

Laitteiston asennus

Windows PC

Robottiikan kurssi
Ohjeellinen opas

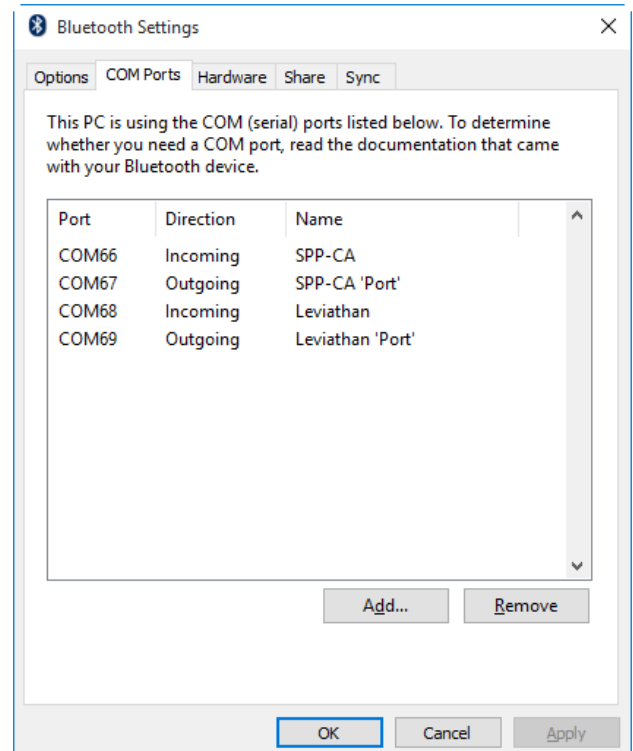
3) Määritä COM-portin numero

Tehtäväpalkkiin pitäisi ilmestyä ponnahtusikkuna, joka ilmoittaa, että laite on käyttövalmis. Jos napsautat tätä ennen kuin se häipyä, saat selville robotille määritetyn COM-portin. Tätä COM-portin numeroa käytetään, kun kommunikoidaan Formula AllCode -robotin kanssa, tämä COMportin numero pysyy samana niin kauan kuin et poista tai irrota pariliitosta Windowsista.

Jos COM-portti ei näkynyt robotin pariliitoksen muodostamisen yhteydessä, löydät sen Bluetooth-asetukset-ikkunasta, kuten oikealla näkyy.

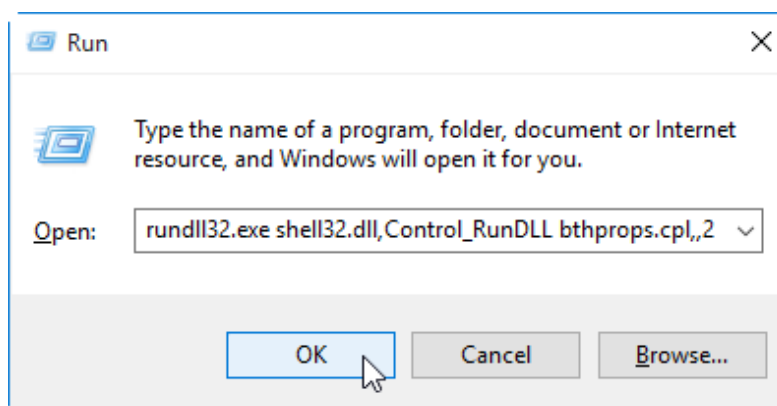
Kullekin robotille on lueteltu kaksi COMporttia. Varmista, että käytät aina "lähtevän" portin numeroa.

Tätä ikkunaa voi olla hieman vaikea löytää joistakin Windows-versioista. Esimerkiksi Windows 10:ssä tämä löytyy Bluetooth-asetusnäytön "Lisää Bluetooth-asetuksia" -linkistä.



Onneksi on olemassa helppo tapa avata tämä ikkuna kaikissa Windows-versioissa versiosta 7 alkaen. Avaa "Suorita..."-ikkuna pitämällä Windows-näppäintä painettuna ja painamalla R-näppäintä, sitten kirjoita (tai kopioi ja liitä) seuraava komento ruutuun ja paina "OK":

rundll32.exe shell32.dll,Control_RunDLL bthprops.cpl,,2



Nyt kun olet yhdistänyt robotin ja määrittänyt COM-portin numeron, voit käyttää mitä tahansa monista Windowsissa saatavilla olevista ohjelmointikielistä Formula AllCode -robotin ohjaamiseen

Laitteiston asennus

Android-tabletti/puhelin

Robotiikan kurssi

Ohjeellinen opas



Android-puhelimet ja -tabletit tarjoavat helppokäyttöisen ohjelmointiohjelmiston, kuten App Inventorin, joka antaa innostavan alustan Formula AllCode -robotin ohjaamiseen.

Näissä laitteissa on lähes aina sisäänrakennettu Bluetooth.

Muiden laitteiden tapaan Formula AllCode -robotti on yhdistettävä puhelimen tai tabletin kanssa, ennen kuin sitä voidaan käyttää.

Jos puhelimesi tai tabletissasi on jo Bluetooth-toiminto, sinun on ehkä ensin otettava se käyttöön napsauttamalla Asetukset -> Yhteydet.

Bluetooth kytketty pois päältä



Bluetooth



Bluetooth kytketty päälle



Bluetooth



Kun Bluetooth on otettu käyttöön, sinun on yhdistettävä Formula AllCode -robotti puhelimeesi, jotta sovellukset näkevät laitteen.

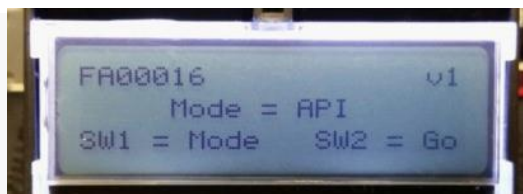
Aloita napsauttamalla Bluetooth-vaihtoehtoa kohdassa Asetukset -> Yhteydet.

Varmista seuraavaksi, että robotti on päällä, napsauta Android-laitteen Scan-painiketta uusien Bluetooth-laitteiden tarkistamiseksi. Huomaa, että saatat joutua selaamaan alaspäin nähdäksesi tulokset osoitteesta

Available devices



FA00016



Formula AllCodon nimi näkyy LCD-näytön vasemmassa yläkulmassa.

Laitteiston asennus

Android-tabletti/puhelin

Robottiikan kurssi
Ohjeellinen opas

Kun laitteen nimi on ilmestynyt näkyviin, napsauta laitteen nimeä, ja sinua pyydetään tekemään laitteiden paritus.

Oletus PIN-koodi on 1234.

Bluetooth pairing request

To pair with:
FA00016

Type the device's required PIN:

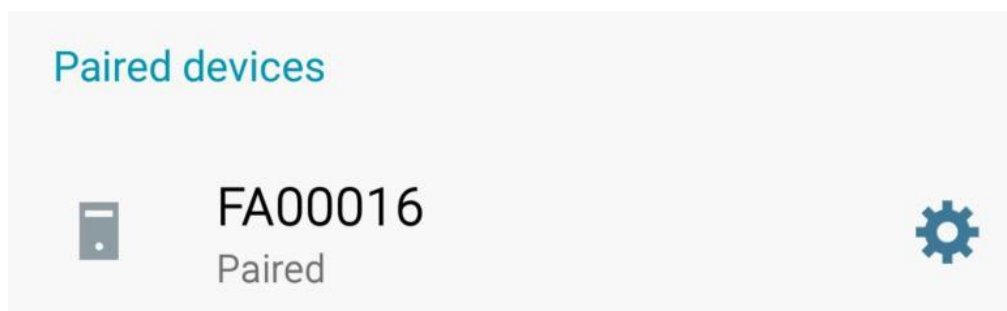
....|

(Try 0000 or 1234)

PIN containing letters or symbols

CANCEL OK

Kun laite on pariliitetty, se näkyy luettelossa muiden pariliitettyjen Bluetooth-laitteiden kanssa ja on valmis käytettäväksi lataamiesi tai luomiesi Formula AllCode -sovellusten kanssa.

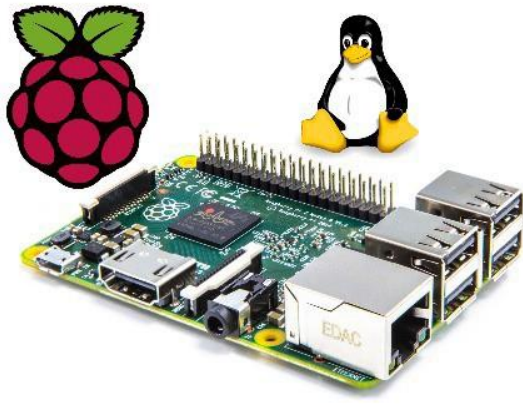


Huomaa: Tämä saattaa olla hieman erilainen Android-laitteessasi. Puhelimesi tai tablettisi erityispiirteitä varten katso Bluetooth-laitteiden pariliitoksen muodostaminen kyseistä laitetta varten.

Laitteiston asennus

Raspberry Pi ja Linux

Robottiikan kurssi
Ohjeellinen opas



Raspberry Pi on suosittu yhden piirilevyn tietokone.

Yleisin käyttöjärjestelmä, jota käytetään Raspberry Pi on Linux-versio nimeltä Raspbian.

Tässä olevat ohjeet Formula AllCode -robotin yhdistämiseen eivät rajoitu Raspberry Pi -robottiin, vaan niitä pitäisi soveltaa useimpiin Linux-pohjaisiin tietokoneisiin.

Bluetoothin määrittäminen on suhteellisen helppoa Raspberry Pi -laitteessa, ja se voidaan tehdä monella eri tavalla. Seuraavat vaiheet ovat ehkä monimutkaisempi tapa määrittää se, mutta sen pitäisi toimia kaikissa tilanteissa. Huomaa, että Pi tarvitsee Bluetooth-USB-donglen.

Vaihe 1 - Hae Bluetooth-asetukset

Avaa komentorivin päätte ja kirjoita komento "hciconfig". Tämä tuo esiin luettelon RPi:ssäsi käytettävissä olevista Bluetooth-laitteista. Tärkeää on huomioida tunnisteen nimi Bluetooth moduuli - minun tapauksessa se on "hci0"

```
pi@raspberrypi / $ hciconfig
hci0: Type: BR/EDR Bus: USB
      BD Address: 00:15:83:15:A3:10 ACL MTU: 339:8 SCO MTU: 128:2
      UP RUNNING PSCAN ISCAN
      RX bytes:1420 acl:0 sco:0 events:53 errors:0
      TX bytes:452 acl:0 sco:0 commands:46 errors:0
```

Vaihe 2 - Hae laite AllCode

Kytke robotti päälle ja kirjoita "hcitool scan". Kun tein tämän, se näytti kaksi laitetta. Omani oli jälkimmäinen ("API_B"), ja sinun on otettava huomioon 6 heksadesimaalipari numerot jotka ovat MAC osoite, yksilöllinen tunniste robotille - tässä tapauksessa

```
pi@raspberrypi / $ hcitool scan
Scanning ..
00:BA:55:23:1C:16      FormAllCode
00:BA:55:23:1C:20      API_B
```

Vaihe 3 - Pariliitos

```
pi@raspberrypi / $ sudo bluez-simple-agent hci0 00:BA:55:23:1C:20
RequestPinCode (/org/bluez/2342/hci0/dev_00_BA_55_23_1C_20)
Enter PIN Code: 1234
Release
New device (/org/bluez/2342/hci0/dev_00_BA_55_23_1C_20)
```

Laitteiston asennus

Raspberry Pi ja Linux

Robotiikan kurssi
Ohjeellinen opas

Vaihe 4 - Muutoksen muuttaminen pysyväksi

Viimeinen vaihe on saada tämä pariliitos tapahtumaan automaattisesti, kun RPi:tä käytetään seuraavan kerran. Tämä voidaan tehdä muokkaamalla tiedostoa "/etc/bluetooth/rfcomm.conf" (esim. nano-toiminnolla) ja kirjoittamalla seuraava koodi. Jälleen kerran sinun on varmistettava, että käytät oikeaa MAC-osoitetta, joka löytyi aiemmin.

```
pi@raspberrypi / $ sudo nano /etc/bluetooth/rfcomm.conf
```

Sinun on lisättävä rfcomm.conf-tiedostoon seuraavan kaltainen osio:

```
rfcomm1 {
# Sitoo laitteen automaattisesti
käynnistytksen yhteydessä bind yes;

# Laitteen Bluetooth-osoite laite
00:BA:55:23:1C:20;

# RFCOMM-kanava yhteyden kanavalle
1;

# Yhteyden kuvaus kommentti "Formula
AllCode";
}
```

Kolme punaista tekstiä voidaan mukauttaa - voit käyttää vaiheessa 2 löytämääsi MAC-osoitetta, ja voit käyttää nimeä "kommentti"-kentässä.

Jos sinulla on useampi kuin yksi robotti, voit lisätä useita osioita - nimeä vain kukin osio. "rfcomm1", "rfcomm2" jne.

Vaihe 5 - Yhteyden testaaminen

Kun yhteys on muodostettu, voit testata yhteyden käyttämällä komentorivipäätteessä seuraavaa komentoa:

```
echo "PlayNote 100,100\n" > /dev/rfcomm1
```

Jos kaikki sujuu hyvin, Formula AllCode antaa äänimerkin.

Jos tämä ei toimi ja saat "permission denied" -viestin, sinun on ehkä lisättävä seuraavat tiedot "dialout"-ryhmään. Voit tarkistaa, onko näin, käyttämällä komentoa "id", jonka parametrina on käyttäjänimesi, ja tarkistamalla, mihin ryhmään kuulut. Jos ryhmä "dialout" ei ole luettelossa, voit lisätä itsesi ryhmään seuraavalla komennolla (muista korvata "username" käyttäjänimelläsi!):

```
sudo usermod -a -G dialout käyttäjänimi
```

Tämän jälkeen sinun on kirjauduttava ulos ja kirjauduttava takaisin sisään nähdäksesi tämän muutoksen, ja "PlayNote"

Robotin ohjaaminen

Flowcoden käyttäminen

Robottiikan kurssi Ohjeellinen opas

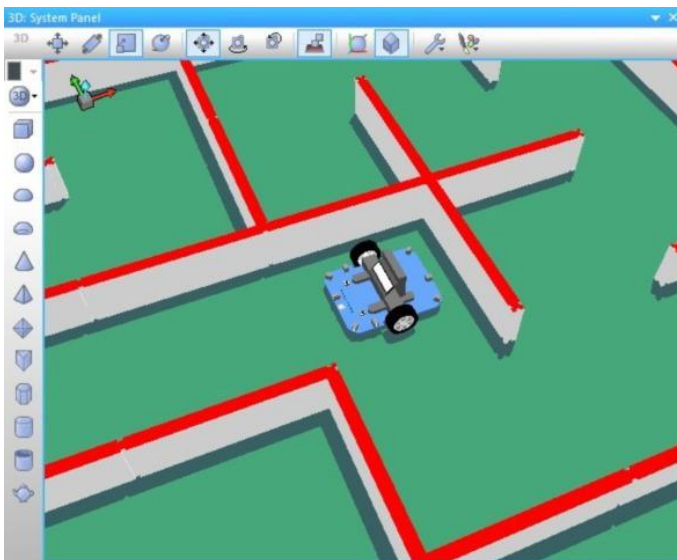


Tässä osassa kerrotaan, miten Flowcodea käytetään robotin ohjaamiseen.

Kuten luultavasti tiedät, Flowcode tarjoaa komponenttimakroja kaikille monimutkaisille laitteille, kuten CAN-väylälle, ZigBee:lle ja robotille. Tämä tarkoittaa, että voit alkaa oppia robottiikasta ja siitä, miten ohjata Formula AllCode Robotia hyvin nopeasti ja helposti. Matrixin verkkosivustolla on ohjelmia, jotka antavat ohjeita

Tässä osassa oletetaan, että tunnet Flowcoden käytön perusteet. On olemassa kaksi tapoa käyttää Flowcodea Formula AllCode -robotin ohjaamiseen. 1) ohjelmoimalla robotin laiteohjelmisto uudelleen ja 2) käyttämällä sisäänrakennettua API-toimintoa.

Flowcode 6.1.3:ssa ja sitä uudemmissa versioissa mekatroniikkavalikosta on käytettävissä kaksi komponenttia, joiden avulla voit valita halutun toimintatilan, kuten oikealla on esitetty.



Koodin lataaminen robottiin

Formula AllCode -komponentin avulla voimme luoda koodia, joka toimii seuraavissa ohjelmissa: Formula-mikrokontrolleri AllCode.

Komponentin mukana toimitetaan täysin toiminnallinen simulointi, jonka avulla voimme luoda ohjelmia, jotka seuraavat viivoja tai ratkaisevat sokkeloita ilman, että koodia tarvitsee jatkuvasti kääntää ja ladata robotille.

Makrofunktioiden määrän vähentämiseksi Formula AllCode -komponentin sisällä voidaan koodi käyttää servomoottorin ulostulojen, SD-kortin ja kiihtyvyyssanturin ohjaamiseen. Saatavilla on erillisiä komponentteja, joiden avulla tämä voidaan tehdä.

Huomautus: Koodin lataaminen tämän komponentin avulla poistaa API-toiminnot robotista. Ohjeet API:n palauttamiseen, jotta robotti saadaan takaisin alkuperäiseen tehdasasetukseen löytyvät myöhemmin tästä ohjeesta..

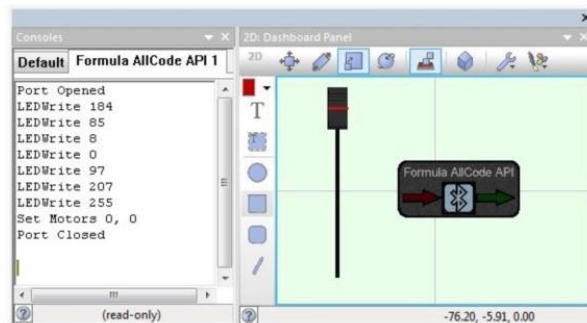
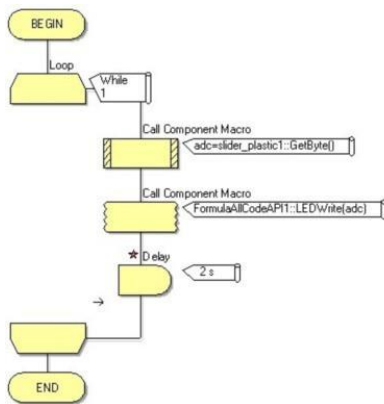
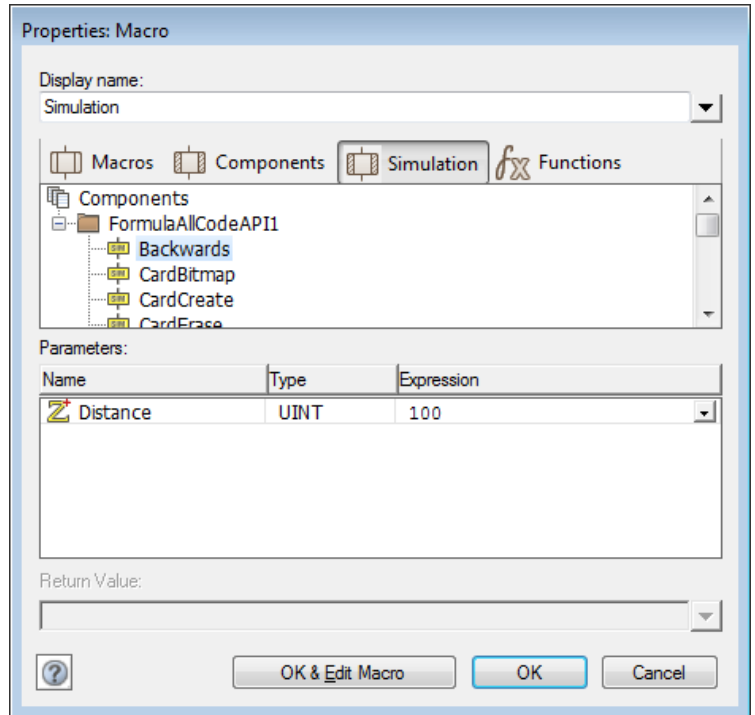
Robotin ohjaaminen

Robottiin kurssi Ohjeellinen opas

Suora ohjaus Flowcodesta API:n avulla Toisen komponentin ("Formula AllCode API") avulla voimme ohjata Formula AllCode-robottia Flowcodesta ilman lataamista.

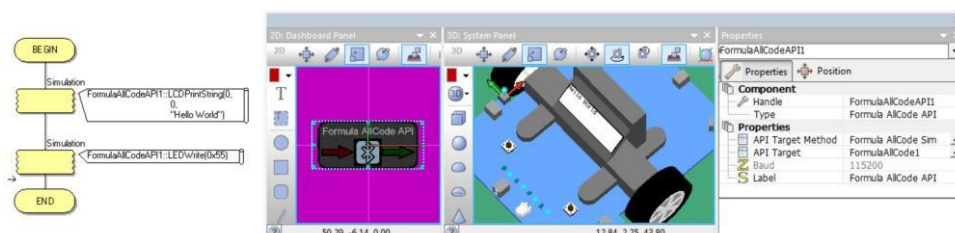
Huomaa, että komponenttimakrotoiminnot näkyvät Simulointi-välilehdellä tavallisen Komponentit-välilehden sijasta. Tämä korostaa sitä, että koodi ei ole ladattavissa Formula AllCodeen.

Tässä ohjataan LEDien arvoa lukemalla arvo simuloidusta analogisesta signaalista liukupotentiometrilta. Konsoli näyttää nyt luettelon API-funktiokutsuista, parametreista ja paluuarvoista.



Valittu tietoliikenneportti avataan automaattisesti, kun käynnistät simulaation. Moottoreiden nopeus asetetaan automaattisesti arvoon 0 ennen portin sulkemista, kun lopetat simuloinnin.

Siirtämällä Formula AllCode -komponentin paneeliin voit myös ohjata simulaatoribotttiaAPI makrojen kautta.



Robotin ohjaaminen

App Inventorin käyttäminen

Robottiikan kurssi Ohjeellinen opas



[App Inventor](#)



[Malli](#)

Tässä osassa kerrotaan, miten pääset alkuun App Inventor -nimisellä koodauskielellä, jonka avulla voit käyttää Android-laitetta robotin ohjaamiseen.

Nämä QR-koodit ja hyperlinkit auttavat nopeuttamaan asennusta, joten voit aloittaa koodauksen.

App Inventor

App Inventor on vapaasti saatavilla oleva graafinen ohjelmointikieli, jota ylläpidetään yhdessä

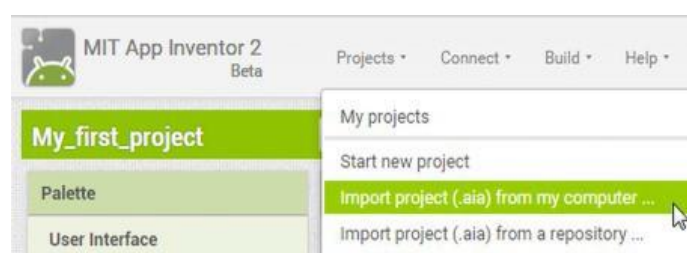
Massachusetts Institute of Technologyn (MIT) pilvipalvelu- ja tallennusjärjestelmissä Yhdysvalloissa. Android-matkapuhelimen tai -tabletin sovellusten kehittämisen aloittamiseen tarvitaan vain verkkoselain ja Google-tili. App Inventor käyttää värikoodattuja kuvakkeita, jotka ovat palapelin palojen muotoisia, ja sovelluksen voi luoda yhdistämällä palat toisiinsa. Järjestelmä estää virheiden tekemisen varmistamalla, että vain tietyt samanväriset palat voidaan liittää yhteen. Tämä tekniikka kannustaa kaikenikäisiä ihmisiä nauttimaan "koodaamisesta" ja kehittämään itseluottamustaan ja taitojaan tietokoneohjelmoinnissa.

App Inventorin määrittäminen

Tärkeimmät tarvitsemasi välineet ovat pöytäkone tai kannettava tietokone (jossa on moderni selain, kuten Chrome tai Firefox) ja Android-käyttöjärjestelmää käyttävä puhelin tai tabletti. Tarvitset myös QR-lukijan, joten olisi hyvä ladata sellainen kännykkään tässä vaiheessa.

Seuraa näitä yksinkertaisia ohjeita, niin saat laitteet toimintakuntoon todella nopeasti.

1. Luo Google-tili (jos sinulla ei vielä ole).
2. Siirry App Inventorin verkkosivustolle skannaamalla QR-koodi tai napsauttamalla yllä olevaa hyperlinkkiä ja kirjaudu sisään Google-tililläsi.
3. Seuraa verkko-ohjeita, mukaan lukien MIT AI2 Companion App -sovelluksen asentaminen Androidlaitteeseesi.
4. Sinun on yhdistettävä verkkopohjainen App Inventor puhelimeen tai tablettiin. Tee tämä valitsemalla App Inventorin Connect-valikosta AI Companion.
5. Lataa Formula AllCode -malli tietokoneellesi skannaamalla QR-koodi tai napsauttamalla hyperlinkkiä. Muista, minne olet tallentanut ne työpöydälläsi/kannettavassa tietokoneessa.



Robotin ohjaaminen

App Inventorin käyttäminen

Robotiikan kurssi

Ohjeellinen opas

Ensimmäinen ohjelmasi

Aina kun haluat aloittaa uuden projektin, toimi seuraavasti:

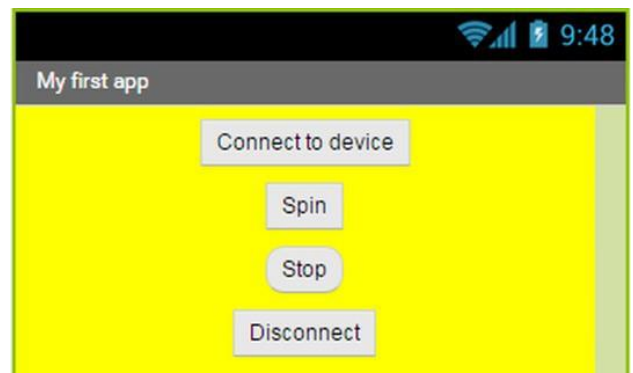
1. Lataa mallitiedosto napsauttamalla App Inventor -valikosta 'My Projects' ja valitsemalla 'FA_Template'-projekti.



2. Tallenna tämä malli uudeksi tiedostoksi valitsemalla "Tallenna projekti nimellä..." "Projektit"-valikosta ja antamalla sitten nimeksi sopiva nimi projektillesi.

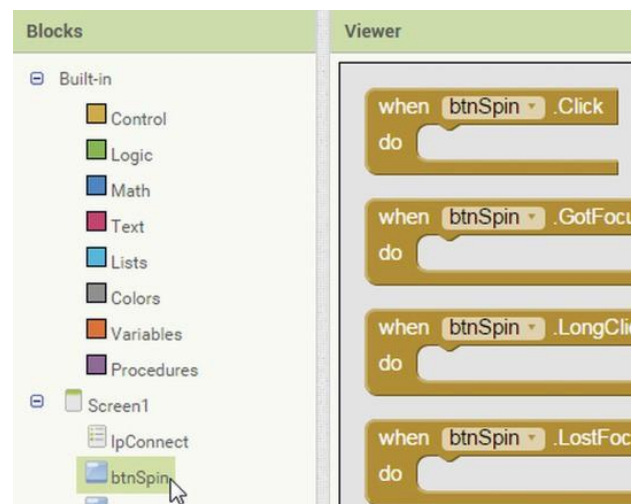
3. Napsauta 'Screen1' 'Components' -ruutua ja aseta 'AppName' ja 'Title' 'Properties' -ikkunassa. paneeli johonkin sopivaan.

4. Vedä painike käyttöliittymäpaneelistä Viewer-näyttöön ja muuta sen teksti muotoon "Spin". Nimeä painike myös uudelleen niin, että siinä lukee "btnSpin". Tee sama uudelleen luodaksesi toisen painikkeen nimeltä "Stop", jonka nimi on "btnStop".



5. Siirry 'Blocks'-tilaan ja napsauta 'btnSpin'-objektia - näkyviin tulee luettelo kuvakkeista. Vedä "when btnSpin.Click"-kuvake ohjelmaasi.

Tee sama uudelleen, mutta tällä kertaa napsauta kohtaa 'btnStop'-objekti.



Robotin ohjaaminen

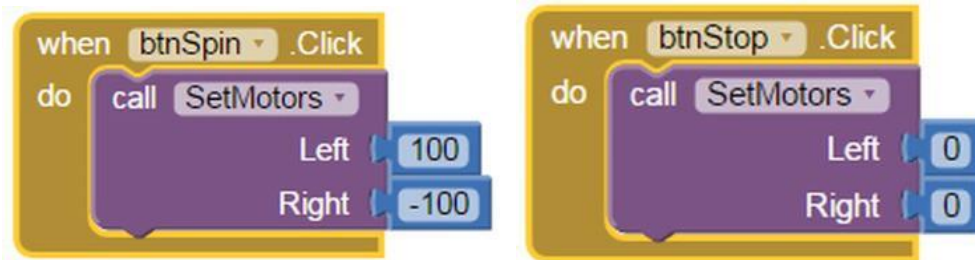
App Inventorin käyttäminen

Robottiikan kurssi

Ohjeellinen opas

6. Napsauta "Proseduurit" "Sisäänrakennetut" -luettelosta ja vedä "call SetMotors" -kuvake "when btnSpin.Click" ja "when btnStop.Click" -kuvakkeiden keskelle. Lisää kaksi literaaliarvoa 'Math'-ohjelmasta SetMotors-koodilohkojen vasemman ja oikean puolen

parametreiksi. Kahden lohkon pitäisi näyttää tältä:



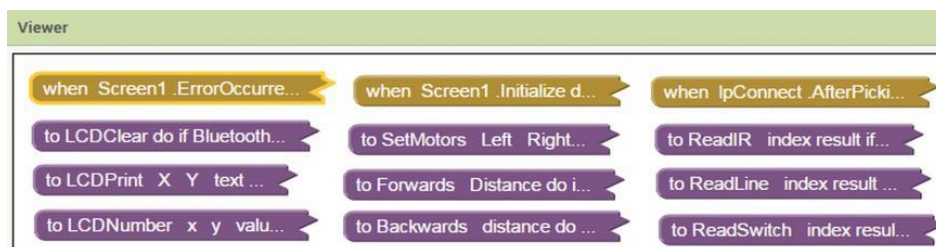
7. Nyt sinun pitäisi rakentaa projekti. Valitse "App (anna .apk-tiedoston QR-koodi)" "Build"-valikosta. Kun tämä on valmis, suorita "MIT AI2 Companion" -sovellus ja skannaamalla QR-koodi Android-laitteeseesi "Scan QR Code" -painikkeella.

Huom:

Jos käyt App Inventorin verkkosivustolla, löydät ohjeita muista menetelmistä, joilla voit siirtää ohjelmasi Android-laitteeseen.

8. Voit nyt suorittaa ohjelmasi Android-laitteellasi. Napsauta "Yhdistä laitteeseen" ja valitse Formula AllCode Robot luettelosta. Napsauttamalla "Spin"-painiketta robotin pitäisi pyöriä ja "Stop"-painikkeella sen pitäisi pysähtyä.

Olet ehkä huomannut App Inventorissa useita kuvakkeita näytön yläreunassa. Nämä määrittelevät menettelyt Formula AllCode -robotin kanssa kommunikoidemiseksi ja joitakin vakiotoimintoja, joiden avulla Bluetooth-yhteys voidaan määrittää.



Ruskeaväriset kuvakkeet kuvaavat **tapahtumia**, kuten kun painiketta napsautetaan, kun ajastin käynnistyy tai kun tapahtuu virhe.

Violetinväriset kuvakkeet liittyvät tiettyihin **menettelyihin** tai **aliohjelmiin**, jotka on suunniteltu suorittamaan tiettyjä tehtäviä puolestasi. Sinun ei pitäisi muuttaa niitä, ellei ole kokenut App Inventorin käyttäjä.

Robotin ohjaaminen

Käyttämällä C++ / C# / VB

Robottiikan kurssi
Ohjeellinen opas



Yleinen ohjelmointityökalu on Microsoftin Visual Studio.

Tässä luvussa esittelemme erilaisia menetelmiä, joilla voit kommunikoida Formula AllCode -robotin kanssa käyttäen joitakin laajalti tunnettuja ohjelmointikieliä, kuten C++, C# ja Visual Basic.

Formula AllCode:n käyttäminen Visual Studion kanssa Visual C++, Visual C# tai Visual Basic -ohjelmointikielillä on melko suoraviivaista, ja se koostuu DLL:n käyttämisestä MatrixTSL:n tarjoaman kirjaston ja siihen liittyvien tiedostojen avulla kommunikoidakseen robotin kanssa.

Kuten muillakin kielillä, sinun on käytettävä COM-portin numeroa, johon robotti on liitetty. Esimerkkejä on Matrix TSL -sivuston Formula AllCode -sivuilla täällä:

<https://www.matrixtsl.com/allcode/resources/>.

C#:n käyttö

Oikealla oleva ohjelma näyttää perusohjelman C#-kielellä.

Sinun tulisi käyttää nimiavaruutta "FormulaAllCode" ja sijoittaa FA_DLL-kirjastotiedosto samaan kansioon kuin projektisi. DLL-tiedosto itse on oltava samassa kansioon kuin luomaasi EXE:tä.

Huomaat, että Formula AllCode API -komennot ovat muotoa etuliitteenä merkit

"FA_", ja niille on myös lähetettävä COM-portti joka kerta ensimmäisenä parametrina.

Muista muuttaa liitteessä olevia API-komentoja, kun käytät niitä.

Muista myös sulkea COM portti ohjelmasi lopussa!

```
using System;

namespace FormulaAllCode
{
    class MyProgram
    {
        static void Main(string[] args)
        {
            //Assign portin numero
            char iPort = (char) 9;

            //Avaa COM-portti
            FA_DLL.FA_ComOpen(iPort)
            ;
            //Soita sävel
            FA_DLL.FA_PlayNote(iPort, 523,
            400);
            FA_DLL.FA_PlayNote(iPort, 659,
            400);
            FA_DLL.FA_PlayNote(iPort, 784,
            800);

            //Sulje COM-portti
            FA_ComClose(iPort);
        }
    }
}
```

Robotin ohjaaminen

C++ / C# / VB:n käyttö

Robottiikan kurssi
Ohjeellinen opas

VB:n käyttö

Oikealla näkyy sama ohjelma, tällä kertaa Visual Basicilla.

Näet, että ohjelma on hyvin samankaltainen kuin C#-ohjelma, ja sen syntaksissa on vain pieniä eroja. Formula AllCode API:n avat samkutsut anloaivnaten s.e ura

FA_API.vb-tiedosto on lisättävä osoitteeseen projektisi.

Muista myös laittaa "FASlave.DLL"-tiedosto samaan kansioon kuin luotu EXE.

Module Moduuli1

```
Sub Main()
    'Assing the port number
    Dim iPort As Byte
    iPort = 9

    'Open COM-port
    FA_ComOpen(iPort)

    'Paly a tune
    FA_PlayNote(iPort, 523, 400)
    FA_PlayNote(iPort, 659, 400)
    FA_PlayNote(iPort, 784, 800)
    'Close COM-port
    FA_ComClose(iPort)
End Sub
```

End Module

C++:n käyttö

Näytetään hieman erilainen ohjelma C++:lle, tällä kertaa piirtämällä tasasivuinen kolmio.

Jotta voit käyttää DLL:ää C++:n kanssa, sinun täytyy viitata funktioihin sisällyttämällä "FA_API.h"-otsikkotiedosto. Sinun on myös lisättävä "FASlave.lib"-tiedosto Visual Studio -projektiisi.

Laita myös DLL samaan kansioon kuin EXE, jonka luot.

Muiden kielten tapaan Formula AllCode API:n kutsut ovat hyvin samankaltaisia, joten robottia on erittäin helppo käyttää eri kielillä - olettaen, että osaat joka tapauksessa kyseisen kielen perusteet!

```
#include "stdafx.h"
#include "FA_API.h"

int main()
{
    //Assign and open COM-port
    char iPort = 9;
    FA_ComOpen(iPort);

    //Draw a triangle
    for (int i=0; i<3; i++)
    {
        FA_Forwards(iPort, 500);
        FA_Left(iPort, 120);
    }

    //Close COM-port
    FA_ComClose(iPort);
    ;

    return 0;
}
```

Robotin ohjaaminen

Pythonin käyttö

Robotiikan kurssi

Ohjeellinen opas



Python on laajalti käytetty tietokoneohjelmointikieli, joka on saatavilla monissa järjestelmissä. Se on ilmainen, helppo oppia ja hauska käyttää.

Tässä osiossa näytetään, miten Python asetetaan käytettäväksi Formula AllCode'n kanssa. Oletuksena on, että sinulla on Pythonin perustiedot. Jos ei, niin internetissä on monia hyviä oppaita kielen oppimiseen.

Asetukset

Ensimmäiseksi sinun on varmistettava, että Python on asennettu tietokoneellesi. Se on yleensä asennettu oletusarvoisesti Raspberry Pi -laitteeseen, mutta Windows- ja muihin laitteisiin sinun on todennäköisesti ladattava ja asennettava se osoitteesta <http://www.python.org>.

Pythonista on kaksi versiota, 2 ja 3, ja kumpaakin voidaan käyttää Formula AllCode -robotin ohjaamiseen, mutta kannattaa varmistaa, että käytössäsi on uusin versio.

Pythonin lisäksi sinun on asennettava myös PySerial-kirjasto. Tämä löytyy GitHubista: <https://github.com/pyserial/pyserial> tai se voidaan ladata Linux-pohjaiseen laitteeseen käyttämällä seuraavaa komentoa terminaali-ikkunassa:

```
sudo apt-get install python-serial
```

Nyt kun Python ja PySerial-kirjasto on asennettu, sinun pitäisi ladata Formula AllCode Python -kirjasto täältä: <https://www.matrixtsl.com/allcode/resources/>.

Tältä sivulta löydät esimerkkejä ja muita resursseja, joiden avulla voit ohjata robottia Pythonilla ja monilla muilla kielillä.

Ensimmäinen Python-ohjelmani

```
import FA # Import the Formula AllCode library
fa = FA.Create() # Create an instance of the API
fa.ComOpen(5) # Open the COM port
fa.Forwards(100) # Move forward 100mm
fa.ComClose() # Close the COM port
```


Robotin ohjaaminen

Pythonin käyttö

Robottiikan kurssi

Ohjeellinen opas

Tämä hyvin yksinkertainen ohjelma ajaa robottia 10 cm eteenpäin API-komennolla `Forwards`, mutta ennen ja jälkeen komennon on useita muita koodirivejä, jotka saattavat vaatia enemmän selitystä.

Kolme ensimmäistä koodiriviä tuovat kirjaston, jotta voit käyttää API-komentoja, sitten luodaan API:n instanssi ja avataan viestintäkanava siihen. Numero "5" edustaa COM-porttia, joka luotiin, kun robotti paritettiin.

On tärkeää sulkea COM-portti, ja se on tehtävä ohjelman lopussa kutsumalla `ComClose` API-komentoa.

Useiden robottien ohjaaminen Luomalla

useita instansseja

API, voimme itse asiassa hallita

useampaa kuin yhtä samanaikaisesti.

Oikealla oleva ohjelma näyttää, miten tämä voidaan tehdä.

Aivan kuten ensimmäisessä ohjelmassa, aloitamme tuomalla FA-kirjaston (huomaa, että tuomme myös "time"-kirjaston). Sen jälkeen luomme 2 API:n instanssia ja avaamme niiden COM-portit.

Neliön piirtämisen pitäisi olla helppoa.

Lopuksi molemmat COM-portit suljetaan.

Teoriassa useita robotteja voidaan ohjata samanaikaisesti, mutta valitettavasti tietokoneen Bluetooth-laitteen kapasiteetti asettaa käytännön rajoituksia. Olen havainnut, että 3 tai 4 on realistinen maksimi.

Jatketaan eteenpäin

Olemme esittäneet vain muutaman lyhyen esimerkin siitä, miten Formula AllCode - robottia ohjataan Pythonilla. Jos tarkastelet tämän asiakirjan lopussa olevaa API-viitettä, löydät paljon käytettävissä olevia komentoja.

```
# Import the libraries
import FA
import time
#Create and open 2 robots
fa1 = FA.Create()
fa2 = FA.Create()
fa1.ComOpen(5)
fa2.ComOpen(6)
#Draw a square
loop = 0
while loop > 0:

    fa1.Forwards(100)
    fa2.Forwards(100)
    time.sleep(1)
    fa1.Right(90)
    fa2.Right(90)
    time.sleep(1)
    loop = loop - 1

# Close the COM-ports
fa1.ComClose()
fa2.ComClose()
```

Robotin ohjaaminen

Labview:n käyttö

Robottiikan kurssi
Ohjeellinen opas



LabVIEW on kehitysympäristö, jonka avulla voidaan luoda mukautettuja sovelluksia, jotka ovat vuorovaikutuksessa reaali maailman tietojen tai signaalien kanssa esimerkiksi tieteen ja tekniikan aloilla.

Sitä voidaan käyttää myös Formula AllCode -robotin

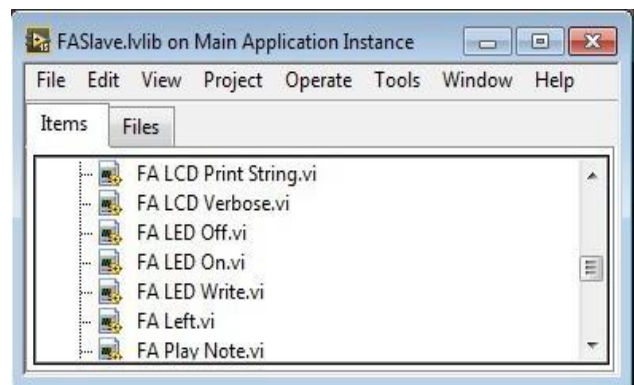
ohjaamiseen. Tässä osiossa selitetään, miten pääset alkuun

Robotin käyttäminen Labview:n kanssa on melko helppoa, ja se koostuu MatrixTSL:n tarjoaman kirjaston käytöstä. Lataa ensin kirjasto (joka koostuu DLL:stä ja LabView-kirjastotiedostosta) Matrix TSL:stä.

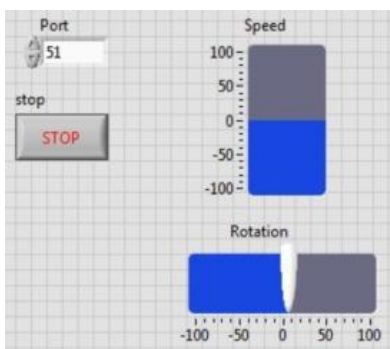
verkkosivusto:

<https://www.matrixtsl.com/allcode/resources/>

Aloita luomalla uusi tyhjä VI ja avaa sitten VI:n "FASlave.lvlib"-tiedostoon, joka ladattiin aiemmin. Tämä sisältää kaikki funktiokutsut Formula AllCode API:lle, kuten oikealla näkyy.



Alla on esimerkkiohjelma, jossa käytetään kahta liikusäädintä robotin ohjaamiseen. Aseta "Port"-ruutuun numero COM-portin numeroksi, joka luotiin robotille, kun se paritettiin.

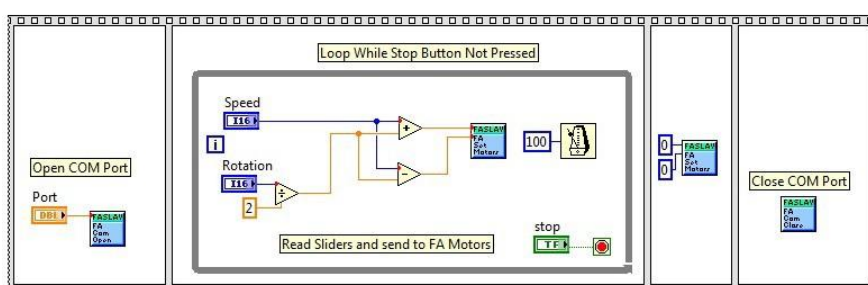


Sivun alareunassa oleva ohjelma näyttää rakenteen kaaviona, joka varmistaa, että ohjelman eri osia kutsutaan vuorotellen.

Vasen ikkuna suoritetaan ensin, se avaa COM-portin.

Keskimmäinen ikkuna pyörii, kunnes painetaan "stop". Se ottaa liikusäätimien arvon ja lähettää sen SetMotors API -komennolle joka 100ms.

Seuraava ikkuna pysäyttää moottorit, kun silmukka on päättynyt, ja viimeinen, oikeanpuoleisin ikkuna sulkee COM-portin.



Robotin ohjaaminen

Scratchin käyttö

Robotiikan kurssi

Ohjeellinen opas



Scratch on erittäin suosittu ohjelmointityökalu, jonka avulla voidaan esitellä koodaustekniikoita kouluikäisille lapsille.

Vaikka sitä käytetäänkin pääasiassa sellaisten ohjelmien luomiseen, joilla ohjataan näytön hahmoja, voit käyttää sitä myös ulkoisten laitteistojen, kuten Formula AllCode -robotin, ohjaamiseen.

Tässä osassa kuvataan, mitä tarvitset aloitukseen.

Tätä kirjoitettaessa Scratchista on olemassa useita versioita ja johdannaisia - sekä verkossa että offline-tilassa. Tässä annetut ohjeet koskevat Scratch 2 Offline Editoria.

Scratch ei luonnostaan tue ulkoista laitteistoa, joten olemme kirjoittaneet ulkoisen "apusovelluksen", joka kääntää Scratchin ja Formula-ohjelman välisen kommunikaation. Mukana on myös mallitiedosto, jonka avulla voit aloittaa robotin käytön. Voit ladata tämän apusovelluksen ja mallitiedoston täältä:

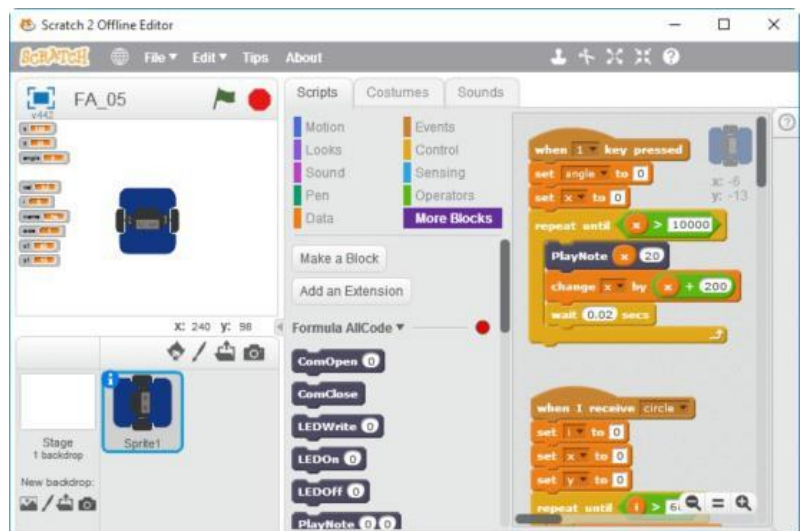
<https://www.matrixtsl.com/allcode/resources/>

Ennen kuin käynnistät Scratchin, suorita apusovellus ja valitse luettelosta robotti, johon haluat muodostaa yhteyden.

Voit nyt avata Scratchin. Avaa ensin mallitiedosto aloittaaksesi uuden projektin. Tämä lataa kaikki API-komennot Scratchin "More Blocks" -alueelle. Se antaa sinulle myös Formula AllCode sprite ja perus esimerkkiohjelmat.

Omien ohjelmien luomisen pitäisi nyt olla helppoa.

Esimerkiksi robotin liikuttamiseksi, kun nuolinäppäimiä painetaan, alla olevat kuvakkeet voivat toimia seuraavasti kun niitä käytetään.



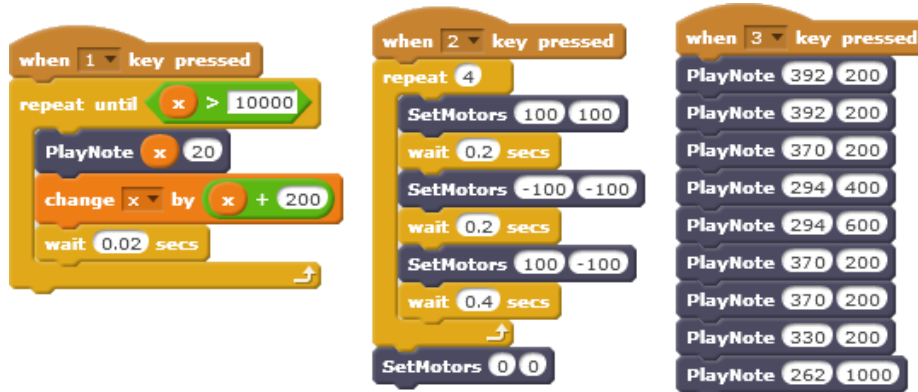
Robotin ohjaaminen

Scratchin käyttö

Robottiikan kurssi

Ohjeellinen opas

Scratchilla voi kirjoittaa monia muitakin ohjelmia. Tässä on kolme esimerkkikoodia:

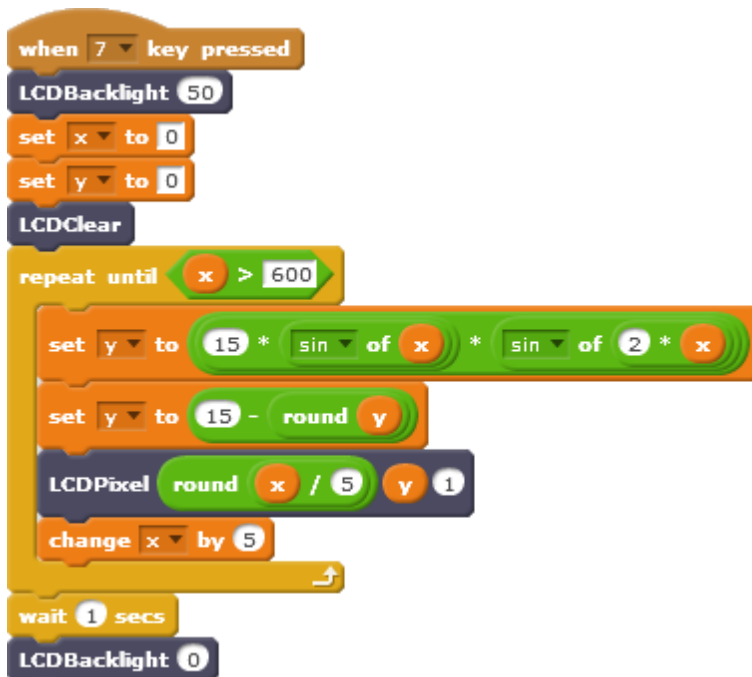


Ensimmäinen antaa mielenkiintoisen äänen, toinen tanssittaa robottia ja kolmas soittaa tuttua sävelmää!

Voit myös linkittää näytön toimintoja, kuten oikealla näkyy.

Tässä tietokone kysyy nimesi ja näyttää sen Formula AllCode -robotin nestekidenäytöllä.

Sen jälkeen vilkkuu LCD-taustavalo huomion kiinnittämiseksi



Huomio.

Myös monimutkaisempia ohjelmia voidaan kirjoittaa. Vasemmalla oleva ohjelma piirtää mielenkiintoisen trigonometrisen yhtälön. Voit myös lukea anturin arvot robotista.

Harjoitus 1

Robotin liikkeet

Robotiikan kurssi

Ohjeellinen opas



Tässä harjoituksessa kerrotaan, miten robotti saadaan liikkumaan eteen- ja taaksepäin ja kääntymään tietyn kierroksen kautta. Kuvio, jotta se voi seurata on geometrisen muotinen, kuten neliö, kolmio tai talon mallinen.

Toiminta

Itsenäinen

Taitotaso

Helppo

Moottorin ohjaukset

Robotin moottoreita voidaan ohjata kahdella tavalla. Ensimmäisessä menetelmässä käytetään API-komentoa **SetMotors**, jolla asetetaan kummankin moottorin nopeus ja pyörimissuunta. Tätä komentoa käsitellään yksityiskohtaisesti myöhemmässä harjoituksessa. Toinen ohjausmenetelmä perustuu Logo-liikkeisiin.

Logon liikkeet

Logo on kieli, joka suunniteltiin vuosia sitten tietokoneohjelmoinnin opettamiseen oppilaille, jotta he voisivat ohjata laitteita, kuten pientä robottia, yksinkertaisten komentojen avulla.

Eteenpäin ja taaksepäin

Suoraviivaisen liikkeen ohjaamiseen on kaksi API-komentoa.

Forwards <distance> ja **Backwards <distance>**

Arvo <distance> määrittää etäisyyden millimetreinä, jonka robotin on kuljettava, ja se voi vaihdella välillä 0-1000. Tässä on hyvin yksinkertainen ohjelma, jolla robotti saadaan liikkumaan 100 mm eteen- ja taaksepäin.

```
// Pseudokoodilla kirjoitettu esimerkkiohjelma
LOOP-FOREVER
```

Tärkeää näissä kahdessa komennossa

on se, että robotti kulkee suoraa linjaa, koska on olemassa kunkin moottorin antama palaute tehon säätöä ja vaihtelua varten.

moottoreihin. Seuraavaksi robotti saadaan liikkumaan sivuttain.

```
Forwards 100 //siirrä 100 mm eteenpäin
DELAY 250ms
Backwards 100 //siirrä taaksepäin 100
mm DELAY 250ms
END-LOOP
```

Vasen ja oikea

Nämä ovat kaksi API-komentoa kulmaliikkeen hallintaan.

Left <angle> ja **Right <angle>**

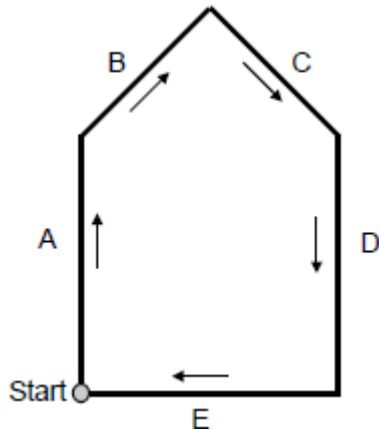
Arvo <kulma> määrittää kulman asteina, jota robotin tulisi kiertää, kulma-arvo voi olla 0-360

Harjoitus 1

Robotiikan kurssi Ohjeellinen opas

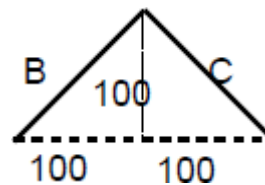
Talon muodon piirtäminen

Seikkaillaan hieman rohkeammin ja annetaan robotin merkitä talon julkisivu.



Sivujen A, D ja E ehdotettu pituus on 200 mm.

Kuten jäljempänä on esitetty, kattorakenne koostuu kahdesta suorakulmaisesta kolmiosta, joiden pohjan ja kohtisuoran pituus on 100 mm.



Sinun vuorosi

(a) Pythagoraan lauseen avulla voit laskea sivujen B ja C pituudet. pystyä myös päättämään kulmat.

Syötä sitten laskemasi arvot (kysymysmerkkien kohdalla) käyttäen alla olevaa ohjelmaa ja katso, piirtääkö robotti oikean muodon.

```
// Pseudokoodilla kirjoitettu esimerkkiohjelma
Forwards 200 //Draw side A
Right ??
Forwards ??? //Draw apex B
Right 90
Forwards ??? //Draw apex C
Right ??
Forwards 200 //Draw side D
Right 90
Forwards 200 //Draw base of house, side E
```

(b) Yritä saada robotti piirtämään seuraavat geometriset muodot: tasasivuinen kolmio, suorakulmainen kolmio, kuusikulmio, vinoneliö. Mitä muita muotoja voit piirtää?

(c) Haasta yksi kavereistasi piirtämään muoto ja ohjelmoi robottiasi tekemään se.

Yhteenveto

Tässä osassa on selitetty, miten API-komentojen avulla voit ohjata robotin moottoreita niin, että ne liikkuvat tarkasti eteen- tai taaksepäin tai kääntyvät vasemmalle tai oikealle tietyssä kulmassa.

Harjoitus 2 LED-valot

Robottiikan kurssi Ohjeellinen opas



Tässä harjoituksessa kerrotaan, miten Formula AllCode -robotin etureunassa olevia kahdeksaa vihreää LED-valoa käytetään.

Niiden avulla voidaan tuottaa mielenkiintoisia ja luovia visuaalisia tehosteita, jotka lisäävät nautintoa

Toiminta

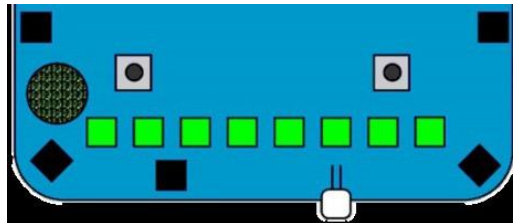
Itsenäinen

Taitotaso

Helppo

LEDit

Robotin etuosassa on kahdeksan LED-valoa. Jokaisella on viitenumero(LED-7:stä alaspäin LED-0:aan), kuten alla näkyy



Seuraavien kahden API-komennon avulla voit kytkeä yksittäisen LED-valon päälle tai pois päältä.

LEDon <index> ja **LEDOff** <index>

<Index> voi ottaa arvon 0 ja 7 väliltä määritelläkseen LEDin, jota haluat käsitellä. Jos

haluat esimerkiksi kytkeä LED-7 päälle, käytä: **LEDon 7** ja sammuttaaksesi sen:

LEDOff 7

Huom:
Nämä kaksi komentoa vaikuttavat vain yhteen LEDiin.

Muiden seitsemän LEDin tila pysyy muuttumattomana.

Tämä ohjelma saa LED-7 vilkkumaan 4 kertaa sekunnissa.

Jos haluat asettaa kaikkien LEDien tilan samanaikaisesti, voit käyttää API-komentoa **LEDWrite** <value>, jossa <value> voi olla arvo välillä 0-255.

Yhdessä LEDit voidaan esittää 8-bittisenä binäärilukuna (eli arvona 0:n ja 255:n välillä). LED-0 on vähiten merkitsevä bitti, ja sen arvo on 1. LED-7 on merkitsevin bitti, ja sen arvo on 128.

Jos haluat asettaa kaikkien LEDien tilan samanaikaisesti, sinun on vain laskettava kokonaisarvo ja käytettävä **LEDWrite** <overall value> .

```
// pseudokoodiesimerkki
```

```
LOOP-FOREVER
  LEDOn 7
  DELAY 125ms
  LEDOff 7
  DELAY 125ms
END-LOOP
```

Harjoitus 2

LED-valot

Robotiikan kurssi Ohjeellinen opas

Jos haluat esimerkiksi sytyttää vain LED-2:n, LED-1:n ja LED-0:n, sinun on lähetettävä parametrina arvo 7 (eli $4 + 2 + 1 = 7$). Tätä varten käytetään seuraavaa API-komentoa.

`LEDWrite 7`

Sinun vuorosi

Olet ehkä kuullut Knight Riderista, amerikkalaisesta televisiosarjasta, siinä oli auto, jossa oli ajovalot joita käytettiin animaationsarjan luomiseen. Tunnetuimmassa jaksossa valot liikkuvat kohti keskustaa ja sitten takaisin ulos.

Sinun pitäisi pystyä luomaan Knight Rider -kuvio robotin kahdeksan ledin avulla. Ensimmäinen vaihe on selvittää LEDEihin kirjoitettavat lukuarvot.

Ensimmäisessä kuviossa LED-7 ja LED-0 syttyvät yhdessä. Sinun on selvitettävä mikä

Arvo tätä edustaa. Seuraavassa kuviossa LED-6 ja LED-1 palavat. Jälleen sinun on laskettava sen arvo. Jatka tätä, kunnes olet saanut kaikki arvot selville, ja syötä sitten nämä arvot ohjelmaasi API-komentoina. Todennäköisesti huomaat, että sinun on lisättävä API-komentojen väliin viive, jotta siirtyminen olisi sujuvaa ja haluttu *Knight Rider* -efekti saavutettaisiin.

Muita efektejä kokeiltavaksi

Voit kokeilla monia muita tehosteita. Tässä on kaksi ehdotusta:

Käärmekuvio - ensimmäinen LED syttyy (arvo 1) ja siirtyy sitten yhden paikan vasemmalle, jolloin toinen LED syttyy (arvo 2). Tämä jatkuu, kunnes merkittävin LED syttyy (arvo 128). Sitten sekvenssi jäljittää askeleensa takaisin alkuun. Tuloksena syntyvän efektin pitäisi näyttää käärmeeltä. liikuttaa tai heiluttaa päätään tai vartaloaan puolelta toiselle.

Toinen efekti, jota saatat haluta kokeilla, on Egg-kuvio. Tämä kuvio alkaa kuten Snake-efekti, mutta kun saavutat ylimmän numeron (arvo 128), LED jää palamaan, kun kuvio palaa takaisin alkuun. Jakso toistuu uudelleen, kunnes LED-6 saavutetaan. Tämä LED jää palamaan (kuten sekä LED-7), kun kuvio kulkee takaisin.

Jotkin luotavat kuviot saattavat noudattaa matemaattista sarjaa, mikä tarkoittaa, että voit tulostaa arvon nimetystä muuttujasta alla olevan kuvan mukaisesti.

`LEDWrite <named_variable>`

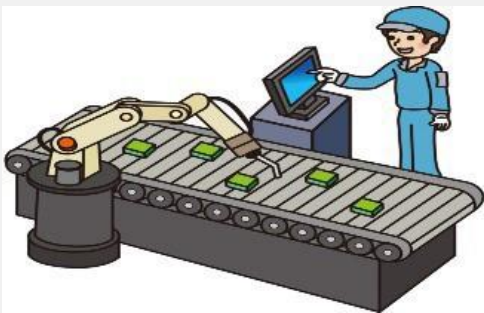
Yhteenveto

Tässä jaksossa on näytetty, miten API-komennoilla voidaan käsitellä robotin LED-riviä lähettämällä arvo, jolla voidaan ohjata kaikkia kahdeksaa LEDiä yhdessä tai erikseen.

Harjoitus 3

Kytke se päälle, sammuta se

Robotiikan kurssi
Ohjeellinen opas



Tässä harjoituksessa selitetään, miten piirilevyn painikkeet ja LEDit toimivat, jotta voit ymmärtää, miten tulot ja lähdöt toimivat yksinkertaisessa digitaalisessa järjestelmässä.

Toiminta

Itsenäinen

Taitotaso

Helppo

Painikkeet

Robotin etupuolella, moottorien kiinnikkeiden vieressä, on kaksi painiketta, jotka on merkitty SW1 ja SW2 ja jotka ohjelma voi tunnistaa ja joita voidaan käyttää tehtävän käynnistämiseen.

Mikro-USB-liittimen vieressä on painike, joka on merkitty SW3:lla ja joka suorittaa järjestelmän nollaustoiminnon aina, kun sitä painetaan.

API-komento SW2:n tai SW1:n tilan lukemiseksi on:

ReadSwitch <index>

ReadSwitch API-kutsu käyttää yhtä "index"-parametria (0 vasemmalle painikkeelle, 1 oikealle painikkeelle) ja palauttaa arvon 1 (painettu) tai 0 (ei painettu).

Tätä kutsua käytetään osoittamalla tulos nimettyyn muuttujaan (esim. SW_1).

SW_1 = ReadSwitch 0

Jos haluat vain suorittaa toiminnon tai yksinkertaisen tehtävän, kun painiketta painetaan, sinun tarvitsee vain testata, onko nimetyn muuttujan arvo suurempi kuin nolla - ja sitten suorittaa seuraavat toimenpiteet

```
// Pseudokoodilla kirjoitettu
esimerkki LOOP-FOREVER
SW_1 = ReadSwitch 0
IF SW_1 > 0 THEN // Detect if SW1 is pressed
CALL SW_1_TASK // User routine
END-IF
SW_2 = ReadSwitch 1
IF SW_2 > 0 THEN // Detect if SW2 is pressed
CALL SW_2_TASK // User routine
END-IF
END-LOOP
```

Harjoitus 3

Kytke se päälle, sammuta se

Robotiikan kurssi
Ohjeellinen opas

Sinun vuorosi

Nyt kun tiedät, miten painikkeet SW1 ja SW2 toimivat, sinun kannattaa yrittää kirjoittaa ohjelma, joka suorittaa sarjan tehtäviä peräkkäin.

(a) Kun SW1:tä painetaan, annetaan Knight Rider -kuviobotin LEDeihin. Kun SW_2 on painettuna, LED-valoihin tulee Snake-kuvio.

Tämä tarkoittaa, että sinun täytyy kirjoittaa koodia, jotka on merkitty nimellä SW_1_Task ja SW_2_Task haluttujen kuvioiden tuottamiseksi.

Jotta asiat pysyisivät yksinkertaisina tässä vaiheessa, huomaat, että ohjelmasi suorittaa yllä olevan sekvenssin kerran, ellei painiketta pidetä painettuna.

Kun vapautat painikkeen, nykyinen kuviokuva pysähtyy, kun se tulee jaksonsa loppuun.

Myöhemmässä harjoituksessa saat selville, miten kirjoitetaan ohjelma, joka muistaa, mikä painike painettiin, joten sinun ei tarvitse pitää sormeaa painikkeella.

(b) Laita vasen painike piirtämään neliö ja oikea painike piirtämään ympyrä.

(c) Tutustu liitteessä oleviin API-komentoihin ja katso, voitko saada muita asioita tapahtumaan, kun painat painiketta.

Yhteenveto

Tässä jaksossa on näytetty, miten voit lukea painikkeiden SW1 ja SW2 tilaa ja voit suorittaa tietyn tehtävän.

Toinen mahdollinen käyttötapa SW1:lle tai SW2:lle on toimia liipaisimena, jolloin kun painiketta painetaan, se lähettää signaalin toiselle ohjelmalle, joka käskää sitä aloittamaan toiminnan.

Voit esimerkiksi järjestää, että ääni (joka on aiemmin tallennettu mikro-SD-kortille) toistetaan tai robotti tanssii, kun painiketta painetaan.

Harjoitus 4

Hei maailma

Robotiikan kurssi
Ohjeellinen opas

Nyt kun tiedät, miten paneeli on järjestetty, voit kirjoittaa viestin API-kutsun avulla.

```
LCDPrint <x> <y> <text>
```

Hei maailma

Tässä on esimerkki viestistä, jonka näet useimmissa ohjelmointia käsittelevissä kirjoissa.

```
// Pseudokoodilla kirjoitettu esimerkki
LCDPrint 0 0 "Hei maailma"
```

Tässä on esimerkki monirivinen viesti keskitettynä LCD-paneeliin.

```
// Pseudokoodilla kirjoitettu esimerkki

LCDPrint 0 0 " Hei ja tervetuloa "
LCDPrint 0 8 " "
LCDPrint 0 16 " Robotiikan "
LCDPrint 0 24 " Käyttöoppaaseen "
```

Viestin jokainen rivi keskitetään merkitsemällä tekstijono välilyönneillä molemmin puolin. Tarvittavien välilyöntien määrä lasketaan vähentämällä kunkin viestin pituus seuraavasti 21:stä (LCD-paneelin merkinleveys) ja jakamalla tulos kahdella. Joskus tulos on pariton luku, jolloin viestin toisella puolella on ylimääräinen välilyönti.

Sinun vuorosi

Tee yllä olevasta ohjelmasta makro tai aliohjelma keräämällä pseudokoodi yhteen. ja anna sille nimi kuten MESSAGE_1

```
// Pseudokoodilla kirjoitettu
esimerkki

LOOP-FOREVER
  LCDClear
  CALL MESSAGE_1
  DELAY 5s
  LCDClear
  CALL MESSAGE_2
  DELAY 5s
END-LOOP
```

Harjoitus 4

Hei maailma

Robottiikan kurssi
Ohjeellinen opas

Seuraavassa on esimerkki siitä, miten nimetyn muuttujan 'etäisyys' arvo voidaan näyttää.

```
LCDNumber 0 0 distance
```

Jotta ulkoasusta tulisi käyttäjäystävällisempi, numeerista arvoa voitaisiin merkitä tekstillä, jotta tiedät, mitä se tarkoittaa.

```
// Pseudokoodilla kirjoitettu esimerkki
```

```
LCDClear
LCDPrint 0 0 "Distance =
" LCDNumber 66 0 distance
```

Taustavalon kirkkaus

Kuten edellisellä sivulla mainittiin, paneelin taustavalon kirkkautta voidaan muuttaa katseluolosuhteiden mukaan. API-komento tätä varten on: **LCDBacklight <value>**, jossa <arvo> voi vaihdella 0:sta (taustavallo pois päältä) 100:aan (täysi kirkkaus).

LCDOptions ei-grafisessa tilassa

Seuraavalla API-kutsulla voidaan ohjata tekstin näyttämistä nestekidenäytössä.

```
LCDOptions <foreground> <background> <transparent>
```

Kaksi ensimmäistä parametria voivat saada arvon 0 tai 1 määritelläkseen vastaavasti valkoisen tai mustan. Tämä tarkoittaa, että voit näyttää valkoista tekstiä mustalla taustalla tai mustaa tekstiä valkoisella taustalla. Kolmas parametri voi myös saada arvon 0 (tekstin näyttäminen etualalla ja taustalla) tai 1 (tekstin näyttäminen vain etualalla).

Aiheeseen liittyvät LCD-komennot

Seuraavat API-komennot vaikuttavat LCD-paneelin graafiseen puoleen ja niitä käsitellään myöhemmässä harjoituksessa.

LCDLine, LCDRect ja LCDPixel

- LCD-paneelin ulkoasu on kuin kaavioitu paperi.
- X-akseli kulkee yläreunan poikki ja Y-akseli sivussa alaspäin.
- X- ja Y-akselit numeroidaan nolasta alkaen (ei yhdestä).

Sinun vuorosi (jälleen)

Tässä jaksossa on näytetty, miten voit käyttää eri API-komentoja näyttämään ja muotoilemaan tietoa (teksti ja numeeriset arvot) LCD-paneelissa.

(a) Laita robotti näyttämään vitsin repliikit. Sinun on pidettävä tauko ennen kuin näytät vitsin repliikin!

(b) Näytä suosikkijalkapallojoukkueesi tulokset.

Harjoitus 5

Kuinka kirkas tuo valo on?

Robotiikan kurssi
Ohjeellinen opas



Tässä harjoituksessa selitetään, miten robotin etureunassa olevaa valotunnistinta käytetään.

Ensimmäinen harjoitus mittaa valotason ja näyttää sen LCD-paneelissa.

Toisessa harjoituksessa LEDien valaistusta säädetään ympäristön valoisuuden mukaan.

Toiminta

Linkitetty

Taitotaso

Helppo

Robotin etureunan keskellä on valoanturi. Vaikka se näyttää aivan tavalliselta LEDiltä, se on itse asiassa valotransistori, joka reagoi siihen kohdistuvan valon määrään. Transistorin läpi kulkevan virran määrä kasvaa valon lisääntyessä.

Sijoittamalla laite yksinkertaiseen potentiaalinjakajaan saadaan jännite vaihtelevaan valon määrään muuttuessa. Robotti voi aistia tämän ja muuntaa sen lukuarvoksi.

API-komento valotason lukemiseen on: [ReadLight](#), ja sitä käytetään osoittamalla tulos nimettyyn muuttujaan (esim. valaistus).

```
valaistus = ReadLight
```

Ensimmäiseksi sinun on selvitettävä valotransistorin tuottamien numeeristen arvojen vaihteluväli eri valaistusolosuhteissa. Tässä yksinkertaisessa harjoituksessa näytetään, miten valotaso mitataan ja arvo näytetään LCD-paneelissa.

```
// Pseudokoodilla kirjoitettu esimerkki
LCDClear
LCDPrint 0 0 "Light value = "

LOOP-FOREVER
    illumination = ReadLight
    LCDNumber 84 0 illumination
    DELAY 100ms
END-LOOP
```

Vaikka valotransistori voi teoriassa tuottaa arvoja 0-4095, niitä ei välttämättä saavuteta ilman, että robotti sijoitetaan pilkkopimeään koteloon tai sitä pidetään kirkkaan sähkölampun vieressä.

Kirjoita ylös joitakin tuloksia, joita saat, kun sijoitat robotin huoneen eri osiin tai viet sen ulkoilmaan. Merkitse muistiin, onko kyseessä aurinkoinen vai pilvinen päivä.

Voit myös kokeilla valaista käsilampulla valotunnistinta ja katsoa, millaisia lukemia saat kun siirryt lähemmäs tai kauemmas robotista.

Harjoitus 5

Kuinka kirkas tuo valo on?

Robotiikan kurssi
Ohjeellinen opas

Seuraava harjoitus on hyvin yksinkertainen ohjaus- tai palautejärjestelmä. Tarkoituksena on mitata ympäristön valaistustaso ja säätää tämän arvon avulla, kuinka monta robotin etuosan LED-valoa syttyy. Tarkoituksena on kytkeä enemmän LED-valoja päälle, kun valon määrä vähenee, ja sammuttaa niitä, kun valon määrä kasvaa.

Jos tätä ajatusta sovellettaisiin käytännön tilanteessa (käyttämällä tehokkaampia valoja), huoneen valaistus pitäisi pystyä pitämään vakiona valaistustasosta riippumatta. Alla oleva pseudokoodi osoittaa, miten tämä saavutetaan.

```
// Pseudokoodilla kirjoitettu

esimerkki LOOP-FOREVER
    illumination = ReadLight
    valaistus = Round(valaistus / 16) LEDit =
    255 - valaistus
    LEDWrite = LEDs
    DELAY 100ms
END-LOOP
```

Ohjelma alkaa lukemalla valoanturin arvon nimettyyn muuttuun nimeltä *'illumination'* ja jakaa sitten tämän arvon 16:lla. Tämä skaalaa valaistuslukeman siten, että se on alueella 0-255 eikä 0-4091.

Varmistaaksemme, että tulos on kokonaisluku, käytämme Round()-funktioita. Valitsemasi ohjelmointikielessä on tämä funktio, mutta sen nimi saattaa olla jokin muu ja se saattaa vaatia ylimääräisen parametrin pyöristystason määrittämiseksi.

Ohjelman seuraavassa osassa luodaan kääntäen verrannollinen valaistustehoste vähentämällä valaistusarvo arvosta 255. Kun valaistusarvo kasvaa, nimimuuttujan 'LEDs' arvo pienenee.

Ohjelman viimeinen osa kirjoittaa nimettyyn muuttuun 'LEDs' annetun arvon robotin etuosan LEDeihin, minkä jälkeen koko sarja toistetaan lyhyen viiveen jälkeen.

Sinun vuorosi

Tässä osiossa on näytetty, miten voit käyttää API-kutsua valoisuusanturin arvon lukemiseen. Kokeile kahta tässä kuvattua harjoitusta saadaksesi kokemusta valoisuusanturin käytöstä ja vahvistaaksesi tietojasi LCD-paneelin ja LEDien käytöstä.

Toinen harjoitus ei toimi odotetulla tavalla - jos muuttujan "LEDit" arvo on 127, 7 LEDiä syttyy, mutta jos arvo on 128, vain yksi syttyy. Yritä muuttaa ohjelmaa niin, että syttyvien LEDien määrä kasvaa, kun valotaso laskee.

Harjoitus 6

Valoa etsimässä

Robottiikan kurssi

Ohjeellinen opas



Tässä harjoituksessa kootaan yhteen parissa aiemmassa harjoituksessa käsitellyjä asioita todellisen tehtävän ratkaisemiseksi.

Tehtävä, joka on ratkaistava, on se, että robotti seuraa käsikäyttöisen taskulampun valonlähdettä. Kuulostaa yksinkertaisemmalta kuin luuletkaan!

Toiminta

Linkitetty

Taitotaso

Keskivaikea

Strategia

Ensimmäinen asia, joka sinun on tehtävä edellä kuvatun ongelman ratkaisemiseksi, on laatia strategia mahdollisten tilanteiden käsittelemiseksi. Jos robotti esimerkiksi havaitsee taskulampun valonsäteen, se siirtyy tietyn matkan eteenpäin ja tarkistaa tilanteen uudelleen. Melko itsestään selvää, mutta mitä robotin pitäisi tehdä jos se kadottaa valonsäteen näkyvistä? Sen on pysähdyttävä ja yritettävä löytää valonlähde.

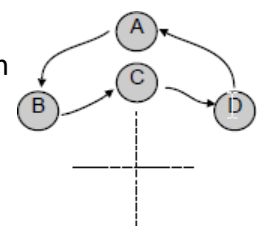
Yksi tapa selvittää, mitä robotin pitäisi tehdä, on laittaa itsensä samaan tilanteeseen ja miettiä, mitä itse tekisi. Käyttämällä silmiäsi voisit luultavasti nähdä valonsäteen. Sinä saatat joutua siirtämään silmiäsi hieman vasemmalle tai oikealle, jotta näet säteen uudelleen. Jos säde on muuttanut dramaattisesti sijaintiaan, saatat joutua liikuttamaan päätäsi. vasemmalle tai oikealle, koska silmiesi liikkumismahdollisuudet ovat fyysisesti rajalliset.

Robotin anatomia on hieman erilainen kuin meillä. Sillä saattaa olla vain yksi silmä (eli valoanturi), joka on kiinteässä asennossa (robotin etureunassa), mutta se pystyy liikuttamaan päätään 360 asteen kierroksen verran. Se on jotain, mihin me emme pysty!

Tämä on strategia, jolla yritetään löytää valonsäde. Tällä hetkellä robotti on paikallaan, joten se saa kääntyä vasemmalle tietyllä kulmalla, esimerkiksi 5 astetta, ja tarkistaa, onko valonsäde näkyvissä. Jos se onnistuu löytämään säteen, se alkaa liikkua eteenpäin. Jos se ei onnistu, se siirtää päätään uudelleen oikealle samassa kulmassa ja tarkistaa valon uudelleen. Jos nämä kaksi manööveriä eivät ole onnistuneet, kulmaa suurennetaan, esimerkiksi 10 asteeseen, ja menettely toistetaan.

Seuraavassa on sekvenssikaavio, joka osoittaa havainnollisesti, miten strategia voitaisiin toteuttaa.

Piste A on robotti, joka osoittaa suoraan eteenpäin. Nuoli pisteestä A pisteeseen B osoittaa kulmaliikkeen vasemmalle. B:stä C:hen tarkoittaa, että robotti palaa takaisin eteenpäin. Nuoli C:stä D:hen osoittaa kulmaliikkeen oikealle, kun taas D:stä A:han on paluuliike eteenpäin.



Valonsäteen olemassaolo tarkistetaan kaikissa neljässä pisteessä. Siirtymä D:stä A:han on paikka, jossa havaintokulmaa kasvatetaan ennen sarjan toistamista.

Harjoitus 6

Valoa etsimässä

Robotiikan kurssi

Ohjeellinen opas

Tässä on esimerkki valonhakuohjelman rakenteesta. Ohjelman lukemisen helpottamiseksi API-komennot on korostettu **sinisellä**.

```
// Pseudokoodilla kirjoitettu esimerkkiohjelma

threshold = 2000 //Experimental value
pos = "A"
movement = 5 //Initial 5 degrees
LOOP-FOREVER
  illumination = ReadLight
  IF illumination > threshold THEN
    pos = "A"
    movement = 5
    Forwards 100
  ELSE
    IF pos == "A" THEN
      pos = "B" Left movement
    ELSE-IF pos == "B" THEN

      pos = "C"
      Right movement
    ELSE-IF pos == "C" THEN
      pos = "D"
      Right movement
    ELSE //Must be at point D
      pos = "A" Left movement
      IF movement < 180 THEN
        movement = movement +
          5 ELSE

          movement = 5
      END-IF
    END-IF
  END-IF
  DELAY 50ms //Short delay (see
text) END-LOOP
```

Vaikka rakenne vastaa hyvin pitkälti edellisellä sivulla annettua tekstimuotoista kuvausta, on muutama asia, jotka on syytä huomioida. Kynnysarvo saatiin kokeilemalla pientä käsikäyttöistä taskulamppua huoneessa, jossa on hämärä valaistus. Sinun on pyrittävä tilanteeseen, jossa käytettävissä olevan valon määrä ja taskulampun valo eivät ylitä valotunnistimen ylärajaa (esim. 4095).

Pääsilmukka sisältää lyhyen viiveen vasemman/oikean liikkeiden ajoituksen tasoittamiseksi. Voit kokeilla muuttaa nimetyn muuttujan *movement* arvoa ja katsoa, miten tämä vaikuttaa robotin käyttäytymiseen ja sen kykyyn paikantaa valonsäde.

Harjoitus

Seuraa linjaa

Robotiikan kurssi
Ohjeellinen opas



Tässä harjoituksessa kootaan yhteen joitakin edellisissä jaksoissa käsiteltyjä tehtäviä todellisen ongelman ratkaisemiseksi.

Tehtävä, joka on ratkaistava, on se, että robotti seuraa ajoneuvon antureiden avulla mustaa viivaa valkoisella pohjalla. Kuulostaa yksinkertaisemmalta kuin luuletkaan!

Toiminta

Linkitetty

Taitotaso

Keskivaikea

Jos et ole katsonut robotin alle, nyt on aika tehdä se. Näet kaksi anturia, jotka havaitsevat pinnasta heijastuvan valon määrän. Kumpikin anturi sisältää infrapunasäteilijän ja valotransistorin, joka pystyy tunnistamaan valkoisen ja mustan pinnan eron kohteen kontrastin ja heijastusominaisuuksien perusteella.

API-komento linja-anturin lukemiseen on: `ReadLine <index>`, ja sitä käytetään osoittamalla tulos nimettyyn muuttujaan (esim. `line_reading`).

Parametri `<index>` voi saada arvon 0 tai 1 oikean anturin valitsemiseksi.

Tässä on esimerkki `rivin anturi-0` lukemisesta, joka on robotin yläpuolelta katsottuna vasen anturi. siten, että gLCD-paneelin pinta osoittaa sinusta poispäin.

```
left = ReadLine 0
```

Harjoituksena voit kirjoittaa yksinkertaisen ohjelman, joka lukee linja-anturit ja näyttää niiden arvot nestekidenäytöllä. Näin voit kokeilla ja nähdä, millaisia arvoja anturi näyttää

```
// Pseudokoodilla kirjoitettu esimerkki
LCDClear
LCDPrint 0 0 " Left Right "

LOOP-FOREVER
  left = ReadLine 0
  right = ReadLine 1
  LCDNumber 24 0 left
  0 left
  LCDNumber 72 0 right
END-LOOP
```

Yllä olevan ohjelman pitäisi tuottaa selkeästi muotoiltu näyttö, jossa linja-anturin arvot näkyvät otsikoissa Vasen ja Oikea. Näin voit tallentaa heijastusarvot erityyppisille pinnoille. Huomaa mustan ja valkoisen pinnan lukemat.

Lukemat, jotka sinun pitäisi saada, ovat luultavasti 0 mustalle ja 200 valkoiselle.

Harjoitus 7

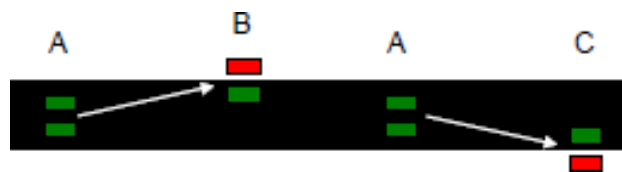
Seuraa linjaa

Robotiikan kurssi Ohjeellinen opas

Testiradan perustaminen ja tehtävän ratkaiseminen

Jos sinulla on robotin mukana tullut matto, voit käyttää sitä, tai voit tehdä oman radan käyttämällä 25 mm leveää mustaa teippiä, joka on kiinnitetty jäykkään valkoiseen kartonkiin. Ovaali tai kahdeksikon muotoinen rata ovat hyviä lähtökohtia. Varmista, että mutkat eivät ole liian jyrkkiä, sillä muuten robotin on vaikea seurata niitä.

Kun käytät Logo-liikkeitä, robotin pitäisi kulkea suorassa, mutta on kuitenkin olemassa tekijöitä, kuten liukas tai rasvainen pinta, joka voi aiheuttaa robotin renkaiden luistamista.



Yllä oleva kaavio osoittaa, mitä tapahtuu, kun robotti poikkeaa radalta. Pisteessä B ja C vasemman ja oikean linjan anturin lukemat ovat muuttuneet. Se osoittaa, että robotti on radan reunalla.

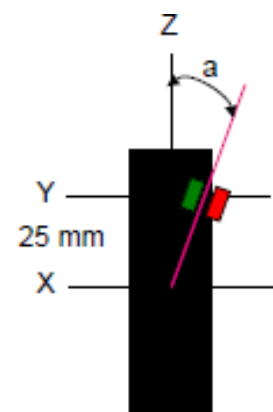
Alla olevassa kaaviossa robotti liikkuu suorassa linjassa, kun radan suunta muuttuu. Aivan kuten edellä, tämäkin tilanne voidaan havaita havaitsemalla muutos linja-antureissa



Saadakseen robotin takaisin raiteilleen sen on tehtävä oikealle suuntautuva liike pisteessä B ja vasemmalle suuntautuva liike pisteessä C.

Kulma, jonka kautta robotin on liikuttava, voidaan määrittää kokeilemalla tai soveltamalla yksinkertaista geometriaa ja trigonometriaa, kuten alla on esitetty.

Piste X edustaa robotin kahden pyörän keskilinjaa. Piste Y on 25 mm robotin etuosaa kohti, ja se on linja-antureiden normaali paikka robotin liikkuessuoraan eteenpäin. Piste Z on robotin keskilinja edestä taaksepäin ja myös mustan nauhan keskikohta. Kahden anturin välinen etäisyys on 17 mm, mikä mahtuu mukavasti 25 mm leveän mustan nauhan päälle. Kulma "a" voidaan laskea yksinkertaisella trigonometrialla tai laatimalla mittakaavapiirustus. Kummallakin tavalla saat tulokseksi seuraavan arvon noin 10-12 astetta. Tämä on kulman korjauskerroin.



Harjoitus 7

Seuraa linjaa

Robotiikan kurssi Ohjeellinen opas

Jos yhdistät kaikki nämä asiat, voit luoda ohjelman tehtävän ratkaisemiseksi.

```
// Pseudokoodilla kirjoitettu esimerkkiohjelma
black = 0 //Reflection value from a black surface
white = 200 //Reflection value from a white surface
correction = 10 //10 degrees
LOOP-FOREVER
  left_sensor =ReadLine
  0
  right_sensor = ReadLine 1 IF
  left_sensor >= white THEN
    Right correction
  ELSE-IF right_sensor >= white THEN
    Left correction
  ELSE
    Forwards 10 //Both sensors detecting a black surface
  END-IF
END-LOOP
```

IF-THEN-ELSE-rakenteella tarkistetaan, missä robotti sijaitsee. Jos se on mustan teipin päällä, sen pitäisi liikkua eteenpäin. Jos se on osittain valkoisen alueen päällä, on sovellettava korjauskerrointa robotin palauttamiseksi takaisin keskelle.

Sinun vuorosi

Yllä olevaa ohjelmaa apuna käyttäen yritä saada robotti seuraamaan viivaa Logo-liikkeillä.

Ohjelma käyttää globaaleja muuttujia, jotka asetetaan ohjelman alussa. Voit muuttaa arvoja, jos pinnan heijastusominaisuudet ovat erilaiset.

Kannattaa myös kokeilla korjausarvon muuttamista, jotta näet, miten se vaikuttaa seurantatarkkuuteen. Sitten voit tutkia toista tapaa ohjata robottia API-kutsun **SetMotors** avulla. Tämä kutsu käyttää kahta moottoria jatkuvasti seuraavasti.

SetMotors <left> <right>

Parametrit <left> ja <right> voivat saada arvon -100-100, jossa -100 tarkoittaa suurinta pyörimisnopeutta vastapäivään ja 100 tarkoittaa täyttä nopeutta myötäpäivään. Esimerkkinä API-komento **SetMotors 50 50** ajaa robottia eteenpäin puolen nopeudella, ja **SetMotors -50 -50** ajaa robottia taaksepäin, jälleen puolen nopeudella.

Muokkaa edellä esitettyä esimerkkiohjelmaa, jotta se toimii SetMotors-komennolla. Moottoreiden nopeuden asettaminen on jonkinlainen kompromissi sen välillä, että radalla liikutaan hyvää vauhtia ja ettei liikuta liian kauas raiteelta ennen kuin radan anturit havaitsevat ja korjaavat tilanteen.

Käyttämällä edellä mainittuja ohjelmia lähtökohtana voit tehdä kokeita, joissa selvitetään, kuinka nopeasti saat robotin kiertämään radan. Haasta ystäväsi kilpailemaan, kenellä on nopein robotti

Harjoitus 8

Painikkeet tekevät työn

Robotiikan kurssi

Ohjeellinen opas



Tässä harjoituksessa selitetään, miten ajoneuvon painikkeita voidaan käyttää tietyn tehtävän suorittamiseen.

Harjoituksessa on esimerkki siitä, miten luodaan ylös/alas-binäärilaskuri, jota ohjataan painikkeilla.

Toiminta

Linkitetty

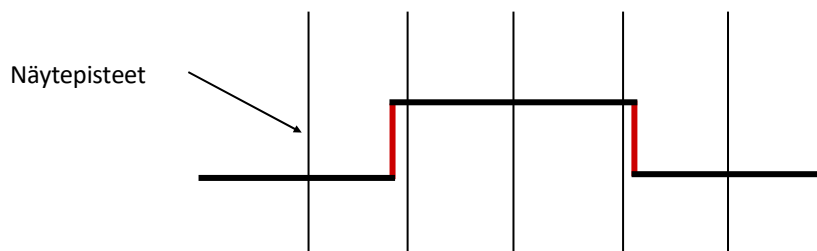
Taitotaso

Keskivaikea

Edellisessä työlehdessä selitettiin, miten API-kutsua `Read <switch>` käytetään SW1:n tai SW2:n tilan havaitsemiseen ja tehtävän suorittamiseen, kun painiketta painetaan. Koska ohjelma oli hyvin yksinkertainen, siinä oli useita rajoituksia. Tärkein niistä oli se, että sormi oli pidettävä painikkeella, jos haluttiin, että tehtävä tai sekvenssi toistuu jatkuvasti. Entä jos haluaisit, että tehtävä suoritettaisiin vain kerran riippumatta siitä, kuinka kauan painat painiketta?

Se on kuin näppäimen painaminen matkapuhelimessa tai laskimessa. Jos näppäintä pidetään painettuna, se halutaan havaita vain kerran, ei toistuvasti, koska muuten saataisiin sarja toistuvia numeroita.

Sinun on havaittava todellinen hetki, jolloin painiketta painetaan tai vapautetaan, eikä se, että painiketta on painettu. Nämä pisteet on värjätty punaisella



Vaikka näiden reunojen havaitsemiseen on monia tapoja, tässä kuvattu menetelmä perustuu näytteiden tai tilannekuvien ottamiseen ajan suhteen. Jokaista näytettä verrataan edelliseen. Jos ne ovat samat, painike ei ole liikkunut.

```
// Pseudokoodilla kirjoitettu esimerkki SW1:n nousevan reunan
// havaitsemiseksi SW1_previous = 0

LOOP-FOREVER
  SW1 = ReadSwitch 0
  IF SW1 > SW1_previous THEN// Detect rising-edge (nouseva reuna)

    CALL specific_task //SW1:n rising-edge task
  END-IF
  SW1_previous = SW1
END-LOOP
```

Harjoitus 8

Painikkeet tekevät työn

Robotiikan kurssi Ohjeellinen opas

Sinun vuorosi

Sinulla on nyt yksinkertainen ohjelma, joka havaitsee nousevan reunan, kun SW1:tä painetaan. Muokkaa ohjelmaa niin, että robotti kääntyy 90 astetta oikealle aina, kun SW1:tä painetaan.

Kun olet tehnyt tämän tehtävän, laajenna ohjelmaa niin, että robotti kääntyy 90 astetta vasemmalle aina, kun SW2:ta painetaan.

Jos mietit käyttämiäsi tuotteita/laitteita, joissa on painikkeita tai näppäimistö, huomaat, että useimmat niistä tekevät jotain, kun painat painiketta. Entä jos halusit laitteen tekevän jotain, kun vapautat painikkeen?

Ratkaisu on hyvin yksinkertainen. Sinun tarvitsee vain muuttaa alkuperäistä ohjelmaasi niin, että se havaitsee kun painike vapautetaan

```
// Pseudokoodilla kirjoitettu esimerkki SW1:n laskevan reunan
// havaitsemiseksi SW1_previous = 0

LOOP-FOREVER
    SW1 = Readswitch 0
    IF SW1 < SW1_previous THEN // Detect falling-edge

        CALL specific_task      //SW1 falling-edge task

    END-IF
    SW1_previous = SW1 //Copy new sample to previous one
END-LOOP
```

Osaat nyt kirjoittaa ohjelman, joka tunnistaa, milloin painiketta painetaan ja milloin se vapautetaan. Riippuen siitä, mitä yrität saavuttaa, saatat haluta havaita molemmat reunat tai vain yhden. Yleisin tilanne on havaita nouseva reuna.

Toinen tehtävä sinulle

Kirjoita ohjelma, joka käyttää robotin etureunassa olevia LEDejä kuin binäärinen ylös/alas-laskin. Ohjelmaa ohjataan SW1:llä ja SW2:lla. Aina kun SW1:tä painetaan, binäärilaskuri kasvaa ja kun SW2:ta painetaan, se pienenee.

Voit itse päättää, mikä reuna on aktiivinen reuna, koska kyseessä on sinun harjoituksesi

Tarvitset ohjelmassasi muun muassa seuraavia asioita:

Nimetty muuttuja (esim. *counter*), jota käytetään binäärilaskurin arvon säilyttämiseen. IF-lohko, joka tunnistaa, milloin SW1:tä painetaan.

IF-lohko, joka tunnistaa, kun SW2:ta painetaan.

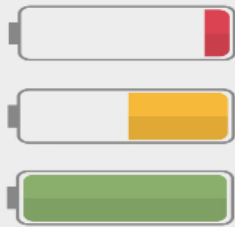
API-komento, jolla kirjoitetaan laskurin arvo robotin LEDeihin.

Harjoitus 9

Näyttöpaneeli

Robottiikan kurssi

Ohjeellinen opas



Tähän harjoitukseen on koottu useita oppaan aiemmissa osissa käsiteltyjä tehtäviä.

Tavoitteena on havaita robotin antureiden tila ja näyttää arvot robotin näytöllä graafinen LCD-paneeli (gLCD) tai matkapuhelin.

Toiminta

Linkitetty

Taitotaso

Keskivaikea

Suunnittelustrategia

Koska gLCD-näyttö on melko pieni, on vaikeaa näyttää paljon tietoja samanaikaisesti, joten yksi ratkaisu on näyttää tietyn tyyppisen anturin tiedot muutaman sekunnin ajan ja siirtyä sitten seuraavaan anturiin. Näin voit ensin suunnitella jokaisen asetelun ja yhdistää ne sitten yhteen niin, että ne näytetään syklisesti. Tämän tehtävän lopussa on ohjeet siitä, miten prosessista voisi tehdä käyttäjälähtöisen eikä vapaasti suoritettavan.

Ensimmäisen pohjapiirroksen suunnittelu

Ensimmäinen tilapaneeli, johon on puututtava, on valoanturin arvon näyttö. Tässä on luonnos miten esitetään teksti tai numero käyttämällä graafisia viivoja.

Valoanturi

123

Ensimmäinen rivi on yksinkertainen staattinen tekstiotsikko, joka koostuu viidestä kirjaimesta, välilyönnistä ja sen jälkeen 6 kirjaimesta.

kirjaimia, yhteensä 12 merkkiä. Tämä vastaa 72 pikseliä (eli 12 * 6 pikseliä). Saadaksesi otsikon keskitettyksi sinun on vähennettävä 128:sta 72 ja jaettava tulos kahdella (eli 28 pikselillä). Tämä tarkoittaa, että otsikko alkaa X,Y-koordinaatista 28,0. Tässä on API-kutsu tätä varten.

```
LCDPrint 28 0 "Light sensor"
```

Seuraavaksi tarvitset jakolinjan, joka kulkee koko paneelin pituudelta. Sen sijaan, että sijoittaisit sen suoraan tekstin alle, kannattaa jättää tyhjä pikselirivi, jotta teksti erottuu. Tässä on API-komento, jolla viiva sijoitetaan riville 9 (kymmenes rivi).

```
LCDLine 0 9 127 9
```

Valoanturista saatu arvo näkyy seuraavalla rivillä. Koska arvo voi on enintään 4095 (eli 4 numeroa), tämä tarkoittaa, että numeron leveys on 24 pikseliä. Tämänkin voit keskittää paneeliin vähentämällä 128:sta 24 ja jakamalla tuloksen kahdella (eli 52 pikselillä). Tässä on API-komento, jolla näytetään nimetyn muuttujan arvo.

```
LCDNumber 52 11 <named_variable>
```

Harjoitus 9

Näyttöpaneeli

Robottiikan kurssi

Ohjeellinen opas

Sinun vuorosi

Ensimmäiseksi sinun on luotava yksinkertainen ohjelma valotunnistimen tilaruutua varten.

```
// Pseudokoodilla kirjoitettu esimerkki
// Tilapaneeli - Valoanturi
LCDClear
LCDPrint 28 0 "Light Sensor"
light_level = ReadLight
LCDLine 0 9 127 9
LCD-number 52 11 light_level
LCDLine 0 19 127 19
```

Sinun kannattaa siistiä asioita laittamalla yllä oleva ohjelma makroon. (Esim. *paneeli_1*) Muokkaa sitten juuri *paneelille_1* luotua koodia ja tee *paneeli_2* kahdelle linja-anturille. Jatka ja tee lisäpaneelit etäisyysanturien sarjaa varten. Yhdistä ne sitten yhteen 5 sekunnin viiveen kanssa, jotta saat aikaan liikkuvan näytön aivan kuten *Knight Riderin* kanssa. Juuri luomasi ohjelma näyttää jokaisen tilapaneelin 5 sekunnin ajan ja siirtyy sitten automaattisesti seuraavaan. Vaikka tämä on jo hyvä, siitä voidaan tehdä paljon hyödyllisempi ja käyttäjäystävällisempi lisäämällä muutama lisäkohde.

Line-sensor	
Left	Right
10	46

Voit esimerkiksi käyttää kahta painiketta (SW1 ja SW2) siirtyäksesi eteenpäin tai taaksepäin eri paneelien tietojen näyttämiseksi. Tämä vaikutus voidaan helposti saavuttaa käyttämällä nousevan reunan tunnustustekniikkaa, jota käsiteltiin aiemmassa harjoituksessa.

Tämän lähestymistavan etuna on, että kun tilapaneeli on valittu, sen tiedot pysyvät näytöllä, kunnes käyttäjä painaa painiketta. Tämä on erittäin hyödyllistä, kunon paljon näytettävää tietoa, kuten kahdeksan etäisyysanturin tiedot.

Toinen tapa navigoida paneeleilla on havaita ääni, joka tulee robotin mikrofonista. Jos esimerkiksi taputat käsiäsi, toiminto voidaan havaita ja sitä voidaan käyttää seuraavan paneelin valitsemiseen kiertosarjassa. Tämän oppaan lopussa olevassaharjoituksessa kerrotaan, miten tämä voidaan toteuttaa.

Yhteenveto

Tässä osiossa on koottu yhteen useita aiemmissa harjoituksissa käsiteltyjä aiheita todella hyödyllisen tilapaneelin luomiseksi. Olet myös saanut selville, miten gLCD-paneelissa voidaan näyttää tietoja kiinteän ajan perusteella tai robotin painikkeilla.

Harjoitus 10

Kallistaminen ja kääntäminen

Robotiikan kurssi Ohjeellinen opas



Tähän harjoitukseen on koottu useita oppaan aiemmissa osissa käsiteltyjä tehtäviä.

Tavoitteena on kehittää sovellus, joka ohjaa robottia matkapuhelimen tai tabletin kallistusanturin avulla.

Toiminta

Linkitetty

Taitotaso

keskivaikea

Tässä harjoituksessa kuvataan, miten kiihtyvyyssanturin tuloksia käytetään robotin ohjaamiseen, joten sitä on mahdollista käyttää vain mobiililaitteella, jossa on nämä ominaisuudet.

Useimmissa matkapuhelimissa ja tableteissa on 2- tai 3-akselinen kiihtyvyyssanturi sisäänrakennettuna. Se tunnistaa, kun kallistat laitetta sivulta toiselle tai kallistat sitä eteen- tai taaksepäin.

Näitä kutsutaan X- ja Y-akseleiksi. Kolmas akseli, Z-akselitunnistaa liikkeen, kun laitetta nostetaan tai lasketaan. Laite mittaa itse asiassa liikkeen muutosnopeutta, sitä kutsutaan kiihtyvyydeksi.

Tämä ohjelma käyttää X- ja Y-kiihtyvyyssarvoja robotin ohjaamiseen.

Kun esimerkiksi kallistat mobiililaitetta sivulta toiselle, robotti liikkuu sivulle, oikealle tai vasemmalle. Kun kallistat sitä eteen- tai taaksepäin, tämä ohjaa moottoreiden nopeutta ja suuntaa. Eteenpäin kallistaminen lisää nopeutta eteenpäin ja taaksepäin kallistaminen lisää nopeutta vastakkaiseen suuntaan. Mobiililaitteen pitäminen vaakatasossa pysäyttää moottorit.

Kiihtyvyyssanturin ominaisuudet

Kiihtyvyyssanturin X-ulostulo (nimetään se $xAccel$) antaa arvon 0, kun laite on levossa tasaisella pinnalla, positiivisen arvon, kun se on kallistettu oikealle (eli sen vasen puoli on koholla), ja negatiivisen arvon, kun se on kallistettu vasemmalle (eli sen oikea puoli on koholla). Y-ulostulo (nimetään se $yAccel$) antaa arvon 0, kun laite on levossa tasaisella pinnalla, positiivisen arvon, kun sen alaosa nostetaan, ja negatiivisen arvon, kun sen yläosa nostetaan.

Nopeuden ja suunnan laskeminen

Kun X- ja Y-kiihtyvyyssarvot on saatu, ne on muunnettava robotin kahden moottorin nopeus- ja suunta-arvoiksi. Voit tehdä tutkimusta seuraavista aiheista

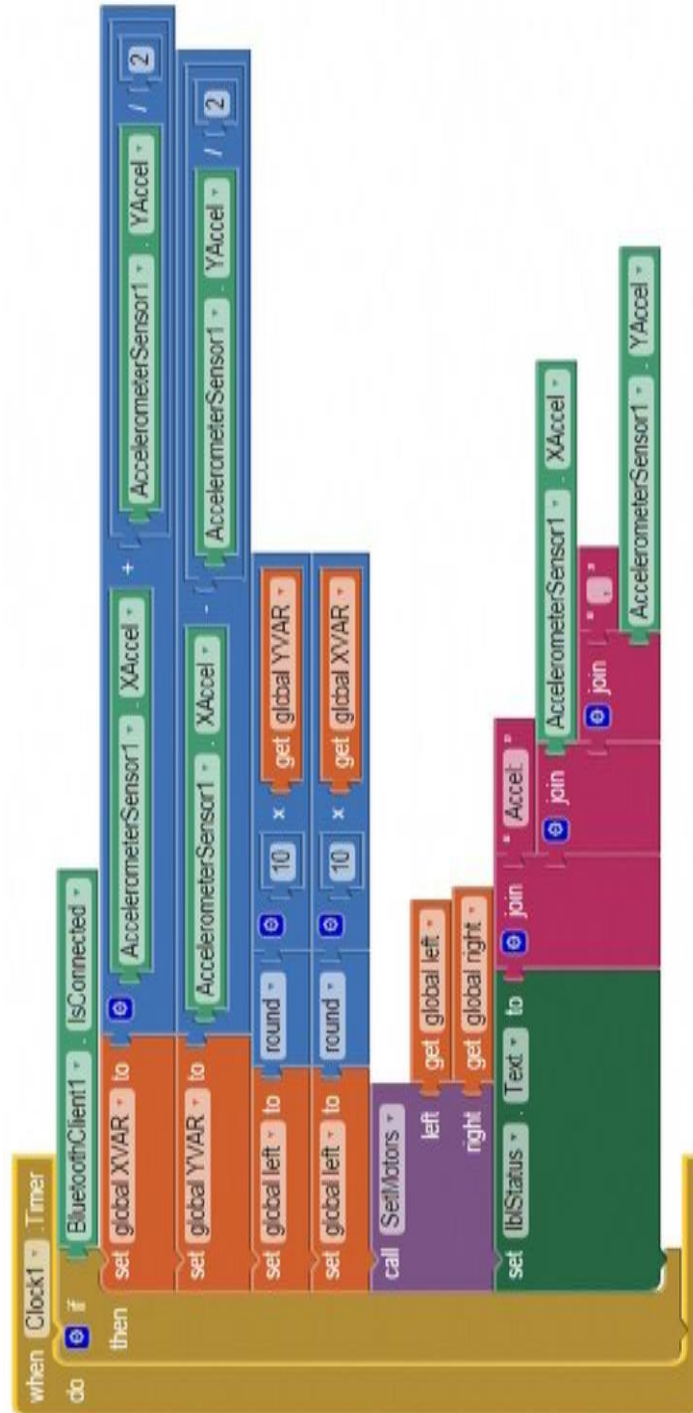
Internetissä ja tutkia eri menetelmiä tämän laskennan suorittamiseksi. Jotta asiat pysyisivät yksinkertaisina, tässä vaiheessa käytetty menetelmä perustuu kuitenkin X- ja Y-arvojen summan ja erotuksen keskiarvon määrittämiseen.

Alla on App Inventor -ohjelma, joka soveltaa keskiarvotekniikkaa ja käyttää tuloksena saatuja arvoja parametreina API-kutsussa `SetMotors <left> <right>`.

Harjoitus 10

Kallistaminen ja kääntäminen

Robotiikan kurssi
Ohjeellinen opas



Sinun vuorosi

Kun olet kokeillut tätä ohjelmaa, sinulla voi olla hauskaa ajaa robottia etänä mobiililaitteella. Voit perustaa esteiden kiertämistä varten kiertoradan?

Harjoitus 11

Robotti osaa suunnistaa labyrintin läpi

Robotiikan kurssi
Ohjeellinen opas



Tähän harjoitukseen on koottu useita oppaan aiemmissa osissa käsiteltyjä tehtäviä.

Tavoitteena on kehittää ohjelma, jonka avulla robotti pystyy navigoimaan yksinkertaisessa sokkelossa käyttäen vasemman seinän seuraamistekniikkaa.

Toiminta

Linkitetty

Taitotaso

Keskivaikea

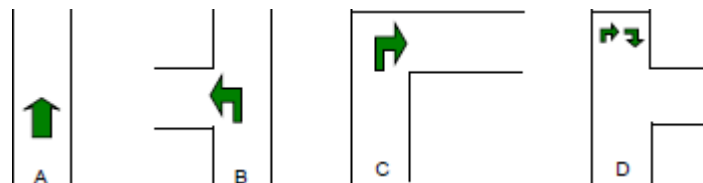
Tämä harjoitus on luultavasti yksi haastavimmista ja palkitsevimista asioista, joita voit tehdä robotin kanssa. Saat sen liikkumaan ja navigoimaan sokkelossa ilman, että siihen kosketaan.

Tämä vetoaa useimpiin ihmisiin heidän iästään riippumatta. Se tarjoaa myös mahdollisuuden kilpailla ystävien kanssa ja nähdä, kenellä on ketterin ja nokkelin robotti.

Jos teet hieman tutkimusta Internetissä, löydät joitakin niistä monista erilaisista menetelmistä, joita ihmiset ovat kehittäneet pakenemaan labyrintin keskeltä tai navigoimaan sen läpi. Yleisin menetelmä on vasemmanpuoleista seinää seuraava algoritmi. Se on algoritmi, menettely tai kaava ongelman ratkaisemiseksi. Se voidaan kirjoittaa epävirallisesti sanoina tai lausekkeina tai strukturoidusti, kuten pseudokoodina tai muodollisena algoritmina, ohjelmointikielellä.

Aloitetaan olettamalla, että robotti on sijoitettulabyrintin sisäänkäynnille, ja katsotaan, miten toimiva algoritmi voitaisiin kehittää. Tässä vaiheessa voi olla hyvä ideatehdä tämä harjoitus ystävän kanssa, sillä voitte vaihtaa ehdotuksia keskenänne.

Robotissa on joukko etäisyysantureita (sijoitettu neljään kulmaan ja neljään reunaan), joiden avulla voidaan havaita, onko tiellä esteitä. Robotin on tarkistettava vasemmanpuoleisesta etäisyysanturista, onko seinää. Sen on myös käytettävä robotin etupuolella olevaa etäisyysanturia varmistaakseen, että etenemispolku on vapaa, ennen kuin jatkaa matkaa.



Jos (vasemmalla puolella oleva) seinä katoaa, mikä osoittaa, että sokkelossa on aukko, robotin on käännettävä vasemmalle ja jatkettava seinän seuraamista. Tilanne (B) kuvassa.

Harjoitus 11

Vasemmalle ja ulos labyrintistä

Robotiikan kurssi Ohjeellinen opas

Jos robotti on joutunut umpikujaan, kuten tilanteessa (D), sen on pysähdyttävä ja kierrettävä 90 astetta myötäpäivään. Kun robotti arvioi tilanteen uudelleen, se huomaa, että sen etenemispolku on edelleen tukossa, joten sen on jälleen pyörittävä 90 astetta myötäpäivään.

Robotti on kääntynyt 180 astetta, ja se voi seurata reittiään ulos umpikujasta.

Yksi tapa tarkastella tätä on, että tilanteen (D) ratkaisemiseksi sovelletaan tilannetta (C) kahdesti. Algoritmi on lähes valmis, toiminnasta on tehtävä silmukka, jotta robotti toistaa edellä mainitut vaiheet syklistä. Tämä ratkaisu on oikeastaan ohjausohjelma, aivan kuten ohjelmat, joita käytetään teollisuusprosessien ohjaamiseen.

Seuraavassa on esimerkki sokkelossa liikkumisen ohjelmarakenteesta. Ohjelman lukemisen helpottamiseksi API-komennot on korostettu [sinisellä](#).

Kuten aiemmin mainittiin, jos robotti joutuu umpikujaan, ohjelma soveltaa tilannetta (C) koskevaa sääntöä kahdesti peräkkäisissä ohjelmasilmukoissa.

```
// Pseudokoodilla kirjoitettu esimerkki
// Navigoi sokkelossa etäisyysantureiden DS-0 ja DS-2 avulla.

clear = 100 //Path ahead is clear or there is no wall
wall = 10 //Wall detected
LOOP-FOREVER
left = ReadIR 0 //Read distance sensor DS-0 (left-hand sensor)
front = ReadIR 2 //Read distance sensor DS-2 (forward sensor)
IF left <= wall AND front == clear THEN //Situation A
SetMotors 50 50 //See note below
ELSE-IF left == clear THEN //Situation B

SetMotors 0 0
left 90
ELSE-IF left <= wall AND front <= wall THEN //Situation C
SetMotors 0 0
Right 90
END-IF
END-LOOP
```

Sinun vuorosi

Luo edellä kuvattu ohjelma, jotta saat kokemusta etäisyysantureiden käytöstä kahden moottorin nopeuden ja suunnan ohjaamiseen.

Kun olet saanut robotin liikkumaan sokkelossa, sinun pitäisi:

(a) Kokeile muuttaa SetMotors-arvoja ja katsoa kuinka nopeasti saat robotin liikkumaan sokkelossa törmäämättä seinään.

(b) Yritä selvittää, pystytkö kehittämään muita ohjausalgoritmeja sokkelonavigointia varten. Voit esimerkiksi hyödyntää robotin oikealla puolella olevaa etäisyysanturia.

Harjoitus 12

Soita musiikkia

Robotiikan kurssi
Ohjeellinen opas



Tässä harjoituksessa selitetään, miten ajoneuvossa olevaa kaiutinta ja digitaalinen signaaliprosessoria käytetään nuottien soittamiseksi.

Harjoituksen toisessa osassa näytetään, miten tehdään yksinkertainen musiikintulkki, jonka avulla voit muodostaa nuotteja ja saada robotin soittamaan.

Toiminta

Itsenäinen

Taitotaso

Helppo

Äänentoistolaitteisto

Robotti voi luoda digitaalisia äänimerkkejä, jotka toistetaan pienen sisäisen kaiuttimen avulla. Äänien taajuus voi olla 1-10 000 Hz ja kesto 1-10 000 ms. Tämä tarkoittaa, että robotti kattaa suuren osan ihmisen kuultavasta äänialueesta. Voit tehdä tämän käyttämällä seuraavaa API-komentoa.

`PlayNote <note> <time>`

Esimerkiksi soittaaksesi keskimmäistä C:tä, jonka taajuus on 262 Hz, yhden sekunnin ajan käytät:

`PlayNote 262 1000`

Jos haluat soittaa nuottien sarjan, toistat vain API-kutsun ja korvaat sopivat arvot.

Jos haluaisit soittaa musiikkikappaleen, tästä menetelmästä tulisi nopeasti isotöinen, joten tässä on menetelmä, jolla voit tehdä hyvin yksinkertaisen musiikintulkin. Se perustuu siihen, että käytetään kirjaimia nuottien esittämiseen ja numeroita kunkin nuotin keston merkitsemiseen.

Alkaen keskimmäisestä C-kirjaimesta kirjainten järjestys olisi C, D, E, F, G, A, B. Numero 1 tarkoittaa yhtä aikayksikköä, 2 kaksinkertaista aikaa jne. Voit päättää, kuinka monta millisekuntia kukin aikayksikkö edustaa. Tässä vaiheessa murtonuotit, sävelkorkeudet ja -korkeudet jätetään huomiotta, ja vain ensimmäinen oktaavi katetaan. Voit laajentaa tulkin ominaisuuksia myöhemmin.

Ensimmäinen vaihe on selvittää oktaavin nuottien taajuus alkaen keskimmäisestä C:stä.

Keskimmäinen C on 262 Hz, D on 294 Hz. Voit selvittää muiden nuottien taajuudet.

Musiikki tulkki

Soittaaksesi musiikkikappaleen sinun on otettava nuotti sivu ja selvitettävä jokaisen nuotin nimi ja kesto. Esimerkiksi "D" 2 tarkoittaa, että nuotti D soitetaan kaksi kertaa pidempään.

Sitten voit syöttää tiedot seuraavalla sivulla olevaan ohjelmaan.

Ohjelma koostuu makrosta, joka purkaa jokaisen nuotin nimen ja keston. Esimerkiksi nuotti "D" korvataan taajuusarvolla 294 Hz. Vastaavalla tavalla numeeriset arvot korvataan numerolla, joka määrittää, kuinka monta millisekuntia nuottia soitetaan.

Harjoitus 12

Soita musiikkia

Robotiikan kurssi Ohjeellinen opas

```
// Pseudokoodilla kirjoitettu esimerkki
// Musiikin tulkki

play(mynote duration)                                //Macro to decode your music
  IF mynote == "C" THEN note = 262
  ELSE-IF mynote == "D" THEN note = 294
  ELSE-IF mynote == "E" THEN note = ???
  ELSE-IF mynote == "F" THEN note = ???
  ELSE-IF mynote == "G" THEN note = ???
  ELSE-IF mynote == "A" THEN note = ???
  ELSE-IF mynote == "B" THEN note = ???
  END-IF

  IF duration == 1 THEN time = 200 //Time unit is 200ms
  ELSE-IF duration == 2 THEN time = 400
  ELSE-IF duration == 3 THEN time = 600
  ELSE-IF duration == 4 THEN time = 800
  END-IF
  PlayNote note time

END-play
CALL play("A" 1) //Can you work out the title of this music?
CALL play("G" 1)

CALL play("A" 1)
CALL play("C" 1)

CALL play("A" 1)

CALL play("G" 1)
CALL play("A" 1)
CALL play("C" 1)
CALL play("C" 1)

CALL play("D" 1)
CALL play("E" 1)
CALL play("D" 1)
CALL play("E" 1)

CALL play("D" 3)
CALL play("E" 1)
```

Sinun vuorosi

(a) Vaikka tämä on hyvin yksinkertainen musiikintulkki, sitä voidaan laajentaa ja parantaa. Täytä ensin ohjelma luetelluille nuotteille.

(b) Seuraavaksi lisätään ylimääräistä dekoodausta, jotta voidaan käsitellä 'sharps', 'flatia', 'lepoja' ja muita oktaaveja.

(c) Lisää lopuksi koodia tempon käsittelyä varten. Joten sen sijaan, että sinulla olisi 200 ms:n perusaika, voisit käyttää tempo skaalauskerroina, jotta musiikki kuulostaisi realistisemmalta.

Tässä jaksossa on selitetty yksinkertainen PlayNote-komento ja näytetty hieman musiikin koodaamista

Harjoitus 13

Robo-DJ

Robotiikan kurssi
Ohjeellinen opas



Tässä harjoituksessa selitetään, miten microSD-kortti esinauhoitettua materiaalia toistetaan robotin kaijuttimesta.

Harjoituksen toisessa osassa näytetään, miten voit rakentaa yksinkertaisen äänentoistolaitteen.

Toiminta

Itsenäinen

Taitotaso

Helppo

Tässä yksinkertaisessa harjoituksessa kopioit suosikkimusiikkisi microSD-kortille ja saat robotin soittamaan sen. Aika helppoa. Todella hauskaa.

Ensimmäinen vaihe on etsiä valmiiksi alustettu microSD-kortti. Nämä ovat pieniä muistikortteja, jotka sopivat älypuheliin ja tabletteihin. Seuraava vaihe on asettaa kortti tilapäisesti mobiililaitteeseen ja kopioida musiikkikappale sille. Jos musiikkisi sijaitsee pöytäkoneessa tai kannettavassa tietokoneessa, saatat tarvita SD-korttisovittimen.

Robotin tukema tallennusmuoto on Waveform Audio File Formaatti, joka tunnetaan yleisesti nimellä WAV sen tiedostopäätteen vuoksi. Varmista, että äänikopiointipakettisi on asetettu tallentamaan .wav-muodossa käyttäen jotakin seuraavista näytteenottotaajuus-/bit-syvyyasetuksista.

8KHz @ 8-bittinen, 8KHz @ 16-bittinen, 16KHz @ 8-bittinen tai 16KHz @ 16-bittinen. Varmista, että robotti on kytketty pois päältä, poista microSD-kortti kopiointilaitteesta ja aseta se robotin microSD-korttipaikkaan.

Olet nyt valmis saamaan robotin soittamaan musiikkia seuraavien API-komentojen avulla.

CardInit ja **CardPlayback <filename>**

Ensimmäinen komento alustaa robotin sisäisen elektroniikan, joka ohjaa SD-korttia. Parametri <filename> määrittää toistettavan .wav-tiedoston nimen. Tämä komento itse asiassa palauttaa arvon 0 tarkoittaa OK, 239 tarkoittaa, että tiedostoa ei löydy, ja 255 tarkoittaa virhettä.

Tässä on esimerkki ennalta nauhoitetun wav-tiedoston toistamisesta.

status = CardPlayback raita1.wav

Täyspitkässä ohjelmassa voit tehdä testin, jolla selvitetään, onko nimetyn muuttujan *status* arvo muu kuin nolla, ja jos näin on, merkitään virhe vastaavasti.

Sinun vuorosi

Olet nyt siinä vaiheessa, että voit tallentaa sarjan "wav"-raitoja microSD-kortille ja kirjoittaa sitten todella yksinkertaisen ohjelman yksittäisen raidan toistamiseksi robotin kaijuttimen kautta.

Harjoitus 13

Robo-DJ

Robotiikan kurssi
Ohjeellinen opas

Äänentoistolaite

Alla olevassa esimerkissä näytetään raitojen valintamenetelmä, jossa käytetään SW1:tä käytettävissä olevien raitojen valitsemiseen ja SW2:ta valitun raidan soittamiseen. Jos aiot käyttää mobiililaitteiden tehtävien suorittamiseen voit käyttää laitteen näytön kosketuspainikkeita.

```
// Pseudokoodilla kirjoitettu esimerkki
SW1_prev = 0
SW2_prev = 0
raita = 1
LCDPrint 0 0 "Track =
" LCDNumber 54 0 track
CardInit

LOOP-FOREVER

  SW1 = ReadSwitch 0 // This is the track selector button
  > IF SW1 > SW1_prev THEN // Detect rising-edge
  > SW1_prev = SW1
  IF track < 10 THEN
    track = track +1
    LCDNumber 54 0
    track
  ELSE track = 1
  END-IF
  ELSE-IF SW1 < SW1_prev THEN // Detect falling-edge
  SW1_prev = SW1
  END-IF

  SW2 = ReadSwitch 1 // This is the execute button
  IF SW2 > SW2_prev THEN // Detect rising-edge
  SW2_prev = SW2
  IF track == 1 THEN
    CardPlayback track1.wav
  ELSE-IF track == 2 THEN
    CardPlayback track2.wav
  ELSE-IF track == 3 THEN
    CardPlayback track3.wav
  ELSE-IF track == 4 THEN
    Kortin toisto
    track4.wav
  etcetera
  END-IF
  ELSE-IF SW2 < SW2_prev THEN // Detect falling-edge
  SW2_prev = SW2
  END-IF

END-LOOP
```

Sinun vuorosi (jälleen)

Vaikka tämä on hyvin yksinkertainen äänentoistolaite, sitä voidaan laajentaa ja parantaa.

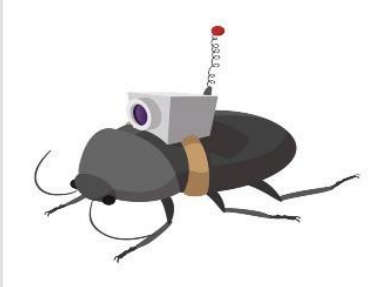
Lisää ylimääräistä dekoodausta, jotta voit valita raidat eteen- ja taaksepäin sen sijaan, että joudut kiertämään raitoja yhteen suuntaan.

Harjoitus 14

Vakoilurobotti

Robotiikan kurssi

Ohjeellinen opas



Tässä harjoituksessa on koottu yhteen joitakin edellisissä jaksoissa esiteltyjä tehtäviä tyypillisen reaali maailman ongelman ratkaisemiseksi.

Ongelma, joka on ratkaistava, on luoda vakoilulaite. Robotti ajaa paikalle, nauhoittaa puhetta, ajaa takaisin ja toistaa nauhoitetun puheen. Se

Toiminta

Linkitetty

Taitotaso

Keskivaikea

Tämän harjoituksen tavoitteena on kehittää sovellus, joka ohjaa robottia kallistusanturin avulla ja käyttää myös äänentoistolaitteistoa puheen tallentamiseen ja toistamiseen. Vaikka tässä esimerkissä käytetään App Inventoria, voidaan käyttää muitakin ohjelmaa, joka vastaa kehitys- ja kohdealustojasi.

Älä lannistu. Tämä harjoitus kuulostaa vaikeammalta kuin se todellisuudessa on. Jos analysoit ongelman, huomaat pian, että olet jo ratkaissut osan tehtävistä. Osoitteessa Esimerkiksi *Tilt and turn* -työselostuksessa selitettiin, miten robottia ohjataan mobiililaitteella. Selvitit myös, miten äänentoistoa käytetään Robo-DJ-työlehdessä, joten ratkaisusta puuttuu vain äänen tallentamisen opettelu.

Äänen tallentaminen robotin avulla on hieman monimutkaisempaa, mutta ei vaikeaa, sillä seuraavat API-kutsut ovat käytettävissä ja tekevät asioista melko yksinkertaisia.

Ensimmäiseksi sinun on alustettava robotin sisäinen elektroniikka, joka ohjaa microSD-korttia. Tämä tehtävä suoritetaan API-kutsulla **CardInit**. Tämä kutsu palauttaa arvon, joka ilmaisee järjestelmän tilan. Numero 0 tarkoittaa, että järjestelmä on kunnossa, 254 tarkoittaa, että järjestelmässä on virhe, ja 255 tarkoittaa, että korttia ei ole havaittu microSD-korttipaikassa. Seuraavaksi aloitetaan tallennus sisäisestä mikrofoniin microSD-kortille.

Tämän tehtävän suorittamiseen käytetään seuraavaa API-komentoa.

CardRecordMic <bitdepth> <amplerate> <time> <filename>

Parametrit <bitdepth> ja <amplerate> on asetettava vastaamaan jotakin robotin neljästä tuetusta tallennustilasta. Nämä ovat: 8-bit@8KHz, 16-bit@8KHz, 8-bit@16KHz tai 16-bit@16KHz.

Ensimmäinen asetus antaa pienimmän tiedostokoon SD-kortille, kun taas neljäs asetus antaa suurimman tiedostokoon SD-kortille ja tuottaa parhaan tallennuslaadun (mutta vie suuremman tiedostokoon). Sen sijaan, että todellisten arvojen syöttämiseen käytetään alkuperäisiä arvoja, käytetään numeroita alla olevien taulukoiden mukaisesti.

	8-bittinen	16-bittinen		8KHz	16KHz
bittisyvyys	0	1	Näytteenotto	0	1

Harjoitus 14

Vakoilurobotti

Robotiikan kurssi

Ohjeellinen opas

Tässä on esimerkki .wav-muotoisen äänitteen tekemisestä parhaalla mahdollisella laadulla 10 sekunnin ajan.

`CardRecordMic 1 1 10 track1.wav`

Tämä API-kutsu palauttaa arvon, joka ilmaisee komennon tilan. Esimerkiksi: palauttaa numeron 0, joka osoittaa, että järjestelmä on kunnossa, 1 tarkoittaa, että tiedosto on jo olemassa kortilla, ja 255 tarkoittaa, että on tapahtunut virhe.

Jos haluat poistaa tallenteen kortilta, voit käyttää seuraavaa komentoa.

`CardDelete <filename>`

Seuraava vaihe on käyttöliittymän luominen mobiililaitteelle.

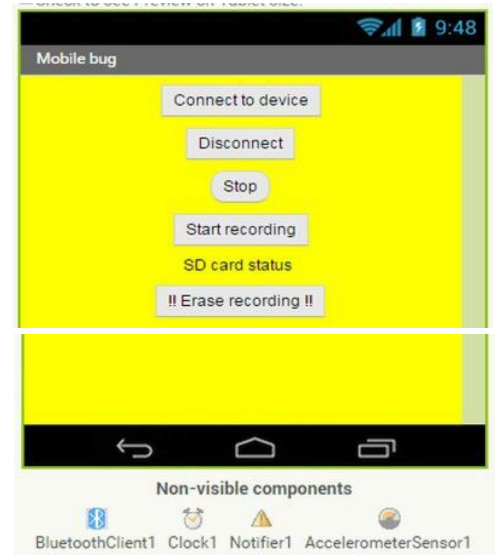
Tässä on esimerkki yksinkertaisesta asetelusta, joka näkyy Appin Designer-näkymässä App Inventor

Kaksi ensimmäistä kosketuspainiketta koskevat robotin Bluetooth-viestintäominaisuutta.

Stop-painike pysäyttää robotin liikkeen asettamalla moottoreiden nopeuden nolnaan.

Aloita tallennus -painike antaa API-komennon CardRecordMic (kuten edellä on kuvattu).

Poista tallennus -painike poistaa tallennuksen kortilta.



Samalla voit näyttää mobiililaitteessa tai robotin gLCD-näytössä viestin, joka näyttää....

"Tämä tallenne tuhoutuu itsestään 15 sekunnin kuluttua."

Seuraavalla sivulla on muutamia esimerkkejä, joiden avulla pääset alkuun lohkokoodauksen kanssa ja jotka osoittavat App Inventorin lohkonäkymässä käynnistyvät tapahtumat.

Sinun vuorosi

Sinun pitäisi muuttaa *Tilt and turn* -sovellus *Mobile bugiksi*. Kun olet tehnyt sen, voit miettiä lisätoimintojen lisäämistä.

Kirjoita esimerkiksi koodia niin, että joka kerta, kun *Tallennuksen käynnistys* -painiketta painetaan, sen tietueet tekevät eri tiedostonimeen. Esim. *track1.wav*, sitten *track2.wav* jne.

Toinen hyödyllinen toiminto olisi jonkinlainen lukitus, joka estää tallenteen toistamisen, kunnes robotti on turvallisesti palannut tukikohtaansa: laajenna ohjelmaa siten, että tietty koodi on syötettävä, ennen kuin toisto voidaan aktivoida.

Harjoitus 14

Vakoilurobotti

Robottiin kurssi Ohjeellinen opas

Kuvakaappaus tapahtumista, jotka käynnistyvät painamalla *Start recording (Aloita tallennus)* -painiketta.

```

when BTN_start_recording .Click
do
  if 0 =
    call CardRecordMic
      bitdepth 0
      samplerate 0
      time 10
      filename "track1.wav"
    then
      set LBL_sd_card_status .Text to "Recording"
    else
      set LBL_sd_card_status .Text to "!! Card problems !!"
  
```

Kuvakaappaus jossa tapahtuu toiminto, kun !!! Poista tallennus !! -painiketta painetaan.

```

when BTN_erase_recording .Click
do
  if 0 =
    call CardDelete
      filename "track1.wav"
    then
      set LBL_sd_card_status .Text to "Deleting sound track"
    else
      set LBL_sd_card_status .Text to "!! Card problems !!"
  
```

Harjoitus 15

Robo-Pop

Robotiikan kurssi Ohjeellinen opas



Tähän harjoitukseen on koottu joitakin edellisissä osioissa käsiteltyjä tehtäviä. Näiden avulla robotti saadaan tanssimaan musiikin tahtiin, kun mikrofoni havaitsee äänen (kuten käsien taputuksen).

Tanssimisen sijasta voisit ohjelmoida robotin niin, että se "juhlii" kuin jalkapalloilija olisi tehnyt maalin.

Toiminta

Linkitetty

Taitotaso

Keskivaikea

Älä lannistu. Tämä harjoitus kuulostaa vaikeammalta kuin se todellisuudessa on. Jos analysoit ongelman, huomaat pian, että olet jo hoitanut suurimman osan tehtävistä. Esimerkiksi *Let's move it, move it* -harjoituksessa selitettiin, miten robotin moottoreita ohjataan. Selvitit myös, miten äänen toistoa käytetään Robo-DJ-työlehdessä, joten ratkaisusta puuttuu vain äänen havaitsemisen opettelu.

API-kutsu äänen havaitsemiseksi on [ReadMic](#). Tämä kutsu palauttaa arvon, joten sitä käytetään osoittamalla se nimettyyn muuttujaan (esim. `sound_level`), kuten alla on esitetty.

```
sound_level = ReadMic
```

Palautettu arvo voi vaihdella välillä 0-4095 ja osoittaa äänen amplitudin. Saatat joutua tekemään joitakin kokeita määrittääksesi kynnyksarvon, joka vastaa "ääntä", jota aiot käyttää "laukaisimena".

Tämä esimerkki näyttää miten ääni yhdistetään aikaisemmin tehtyihin harjoituksiin

```
// Pseudokoodilla kirjoitettu esimerkki
// Robo-pop, joka käynnistyy äänen havaitsemisen perusteella.

threshold = 100 //Set this according to your situation
DO
  sound_level = ReadMic //Clap your hands to make a noise
UNTIL sound_level >= threshold
CALL play_some_dance_music //Use the code from "Robo-DJ"
CALL perform_some_groovy_moves //Use the code from "Let's move it"
```

Harjoitus 15

Robo-Pop

Robotiikan kurssi Ohjeellinen opas

Nyt robottisi tanssii ja soittaa musiikkia. Mitä voisi siis puuttua? Valoja, lasersäteitä - jotain, joka saa yleisön innostumaan. Vaikka robotissa ei ole lasersäteitä, siinä on LCD-paneeli, joka voi näyttää graafisia muotoja - viivoja, suorakulmioita tai neliöitä käsittelemällä yksittäisiä pikseleitä. Käyttämällä robotin kykyä piirtää suorakulmioita voit tehdä tee-se-itse (DIY) valonsäde-esityksen.

DIY-valonsäde-esitys

Seuraava API-komento käskää graafista LCD-paneelia piirtämään suorakulmion.

```
LCDRect <x1> <y1> <x2> <y2>
```

Parametrit <x1> ja <y1> määrittävät suorakulmion vasemman yläkulman ja <x2> ja <y2> oikean alakulman. Seuraavassa on esimerkki 10 x 10 pikselin kokoisen neliön piirtämisestä koordinaatistossa 0,0.

```
LCDRect 0 0 0 10 10
```

Sinun vuorosi (jälleen)

Nyt kun olet nähnyt, miten musta neliö piirretään, tehdään siihen hieman animaatiota. Tehtävänäsi on saada toinen tai molemmat mustista neliöistä, jotka näkyvät alla olevassa kuvassa, liikkumaan LCD-paneelin poikki ja takaisin (kuin valonsäde diskon tanssilattialla).



Jos neliön A lähtöpiste on koordinaatissa 0,4 ja neliön B koordinaatissa 116,18, niin sitten molemmilla ruuduilla on tilaa liukua toistensa päälle.

Seuraavaksi sinun on pohdittava, miten neliöt siirretään.

Yksinkertaisin tapa on käyttää paria nimettyä muuttujaa, jotka toimivat laskureina ja antavat kunkin neliön sijainnin offsetin, ja käyttää niitä kutsussa `LCDRect`

```
LCDRect counterA 0 bottom_right_A 13
```

Esimerkiksi neliön A laskuri alkaa 0:sta. Voit siirtää neliötä oikealle seuraavasti: lisää laskuria (sopivalla arvolla, esimerkiksi 5) ja kutsu `LCDRectiä` uudelleen. `Bottom_right` on toinen nimetty muuttuja, jonka arvo on: `bottom_right_A = counterA + 10`.

Sinun on lisättävä lyhyt viive ennen sekvenssin toistamista, jotta musta neliö liikkuu tasaisesti ruudun poikki. Ohjelmasi on tarkistettava, milloin laskuriA saavuttaa LCD-paneelin oikean reunan - ja päätettävä, nollataanko se lähtöpisteeseen vai jäljitetäänkö sen liike (kuten kuvassa).

Jatketaan eteenpäin

Haasteet 16-20

Robotiikan kurssi
Ohjeellinen opas



Viimeiset 5 haastetta on koottu tänne niille, jotka haluavat mennä pidemmälle.

Emme anna tässä paljon ohjeita, koska suurin osa tästä työstä perustuu edellisiin harjoituksiin ja jos olet päässyt näin pitkälle, olet varmasti jo ohjelmoinnin asiantuntija!

Haaste 16 - Tietojen kirjaaminen

Tässä on paljon mahdollisuuksia: SD-kortin API-komentojen avulla voit kirjata anturitiedot matkan aikana tai valotason auringon laskiessa illalla. Voisit myös pyöriä samalla kun kirjaat valotason, jolloin saat 360°:n kuvaajan huoneen valosta.

Haaste 17 - Servojen käyttö

Kiinnitä servomootorit laajennusporttiin ja tee robotista kaivuri tai anna sille "jalka". potkia palloa.

Robotin etuosassa on neljä servomoottoriliitäntää, ja niitä ohjaavat komennot on lueteltu API-viitteessä.

Haaste 18 - Pimp my ride, tuunaa robotti

Suunnittele Formula AllCode -robotille kuori, joka tekee robotistasi erottuvan. Ehkä voit käyttää 3D-tulostinta luodaksesi robotille todella ainutlaatuisen kuoren.

Kiinnitä kuori robotin alustaan pyörien lähellä olevien 4 reiän avulla.

Haaste 19 - Räätelöity elektroniikka

Laajenna robottisi yksilöllisyyttä lisäämällä sen laajennusportteihin LEDejä ja muuta mukautettua elektroniikkaa.

Muista kytkeä elektroniikka oikein, jotta et vahingoita mikrokontrolleria!

Haaste 20 - Kiihtyvyys ja suunta

Tiesitkö, että laitteessa on sisäänrakennettu kiihtyvyyssmittari? Sen avulla voit mitata, kuinka paljon AllCode-robotti kiihtyy, millaisia voimia on olemassa, kun se kääntyy kulmaan, ja jopa sitä, missä suunnassa se on ylöspäin (Einstein opetti meille, että painovoima ja kiihtyvyys ovat pohjimmiltaan sama asia).

API-komento kiihtyvyyssanturin lukemiseen on [ReadAxis](#).

Liite 1

API-komennot

Robottiikan kurssi Ohjeellinen opas

Tässä osiossa API-kutsut luetellaan toimintoryhmittäin, jotta etsimäsi komento on helppo löytää. Kutsuja on erityyppisiä - jotkin niistä palauttavat arvon, jotkin eivät. Jotkin vaativat yhden tai useamman parametrin, toiset eivät.

Esimerkiksi API-kutsu **SetMotors** vaatii kaksi parametria kahden moottorin nopeuden määrittämiseksi, kun taas **ReadLight-kutsu** palauttaa vain arvon (eli valoanturin arvon). On muitakin kutsuja, jotka vaativat yhden parametrin ja palauttavat myös arvon.

Hyvä esimerkki on **ReadIR** API-kutsu. Tämä kutsu tarvitsee parametrin, jolla määritetään, mikä IR-anturi halutaan tarkistaa. Paluuarvo on kyseisen anturin etäisyysmittaus.

Yhteys			
Paluu	Komento	Parametri(t)	Kuvaus
tila	ComOpen	satama	Avaa COM-portti Portti= 1 - 255 Status= 0(OK) tai 255(virhe).
tila	ComClose		Sulje portti Status= 0(OK) tai 255(virhe).
versio	GetAPIVersion		Palauttaa API:n versionumeron Versio+ 1 - 65535

Anturit			
Paluu	Komento	Parametri(t)	Kuvaus
arvo	ReadSwitch	indeksi	Lue kytkimen arvo Indeksi= 0(vasen) tai 1(oikea) Arvo= 0(väärä) tai 1(tosi).
arvo	ReadIR	indeksi	Lukee IR-anturin Indeksi= 0-7 Arvo= 0-4095
arvo	ReadLine	indeksi	Lukee linja-anturin Indeksi = 0 (vasen) tai 1 (oikea).
arvo	ReadLight		Lukee valoanturin Arvo= 0-4095
arvo	ReadMic		Lukee mikrofonianturia Arvo= 1 - 4095
arvo	ReadAxis	indeksi	Lukee kiihtyvyyksianturin akselin Indeksi= 0 (x), 1 (y) tai 2 (z). Arvo= -32768-32768

Liite 1

API-komennot

Robottiikan kurssi

Ohjeellinen opas

Moottorit			
Paluu	Komento	Parametri(t)	Kuvaus
	SetMotors	Vasen oikea	Aseta moottoreiden nopeus Vasen = -100 - 100 Oikea = -100 - 100
	Forwards	etäisyys	Siirry eteenpäin etäisyys (mm) Etäisyys = 0 - 1000
	Backwards	etäisyys	Siirry taaksepäin (mm) Etäisyys = 0 - 1000
	Left	kulma	Käännä vasemmalle (asteina) Kulma = 0-360 asetta
	Right	kulma	Käännä oikea kulma (asteina) Kulma = 0 - 360

LED / kaiutin			
Paluu	Komento	Parametri(t)	Kuvaus
	LEDWrite	arvo	Kirjoita arvo LEDeihin Arvo = 0 - 7
	LEDOOn	indeksi	LEDin sytyttäminen Indeksi = 0-7
	LEDOff	indeksi	Kytke ja LED pois päältä Indeksi = 0-7
	PlayNote	muis tiinp ano aika	Lähtöääni (Hz) ajalle (ms) Huomautus = 1 - 1000 Aika = 1-1000

Servo			
Paluu	Komento	Parametri(t)	Kuvaus
	ServoEnable	indeksi	Servokanavan käyttöönotto Indeksi = 0-3
	ServoDisable	indeksi	Servokanavan poistaminen käytöstä Indeksi = 0-3
	ServoSetPos	Indeksi n sijainti	Aseta servoasento Indeksi = 0-3 Asento = 0-255
	ServoAutoMove	Indeksi n sijainti	Automaattinen siirtyminen servoasentoon Indeksi = 0 - 3 Asento = 0-255
	ServoMoveSpeed	nopeus	Aseta servon automaattisen liikkeen nopeus Nopeus = 1 - 50

Liite 1

API-komennot

Robottiikan kurssi Ohjeellinen opas

LCD			
Paluu	Komento	Parametri(t)	Kuvaus
	LDClear		
	LCDPrint	X Y teksti	Tekstin tulostaminen nestekidenäytölle X = 0 - 127 Y = 0-31 Teksti = <merkkijono>
	LCDNumber	X Y arvo	Kokonaislukuarvon tulostaminen nestekidenäytölle X = 0 - 127 Y = 0-31 Arvo = -32768-32767
	LCDPixel	X Y valtio	Piirtää pikselin nestekidenäyttöön X = 0 - 127 Y = 0-31 Tila = 0 (pois päältä) tai 1 (päällä).
	LCDLine	X1 Y1 X2 y2	Piirrä viiva nestekidenäyttöön X1= 0-127 Y1 = 0-31 X2 = 0-127 Y2 = 0-31
	LCDRect	X1 Y1 X2 y2	Piirrä suorakulmio nestekidenäyttöön X1 = 0 - 127. Y1 = 0-31 X2 = 0-127 Y2 = 0 o 31
	LCDBacklight	Arvo	Asettaa nestekidenäytön taustavalon kirkkauden Arvo = 0- 100
	LCDOptions	Etuala Tausta läpinäkyvä	Asettaa vaihtoehdon LCD-näytölle piirtämistä varten Etuala = 0 (valkoinen) tai 1 (musta) Tausta = 0 (valkoinen) tai 1 (musta) Läpinäkyvä = 0 (väärä) tai 1 (tosi).

Liite 1

API-komennot

Robottiikan kurssi Ohjeellinen opas

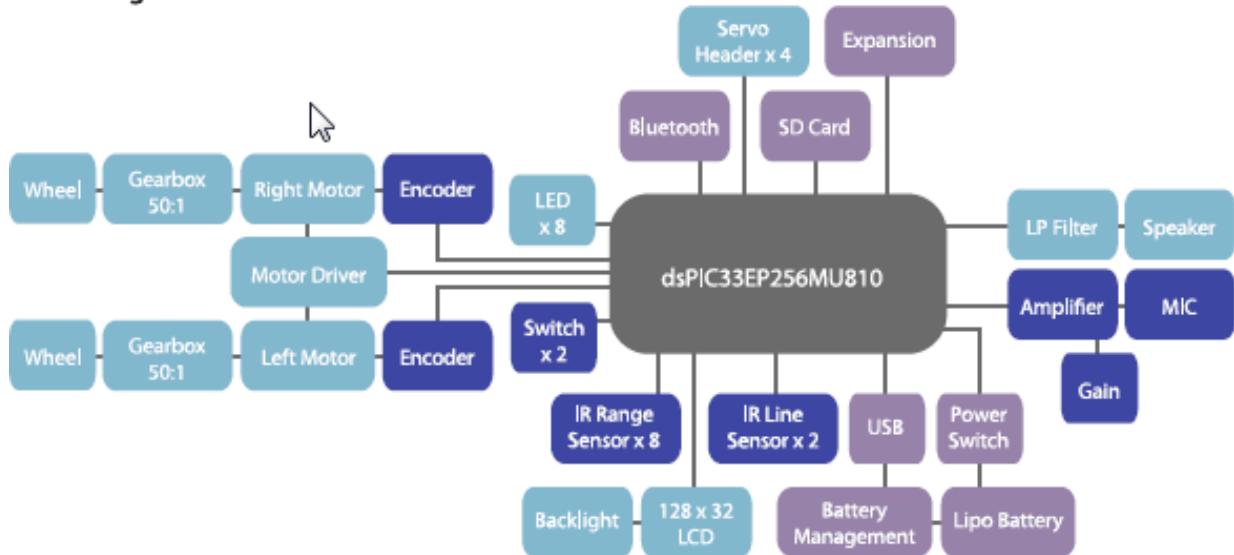
SD-kortti			
Paluu	Komento	Parametri(t)	Kuvaus
tila	CardInit		SD-kortin alustaminen Tila = 0 (OK), 254 (virhe) tai 255 (ei korttia).
tila	CardCreate	tiedostonimi	Luo uusi tiedosto Tiedostonimi = <merkkijono> Status = 0 (OK), 1 (tiedosto on olemassa) tai 255 (virhe).
tila	CardOpen	tiedostonimi	Avaa olemassa oleva tiedosto Tiedostonimi = <merkkijono> Status = 0 (OK), 239 (tiedostoa ei löydy) tai 255 (virhe).
tila	CardDelete	tiedostonimi	Tiedoston poistaminen Tiedostonimi = <merkkijono> Tila = 0 (OK) tai 255 (virhe).
tila	CardWriteByte	tiedot	Kirjoittaa tavu dataa avoimeen tiedostoon Data = 0-255 Tila = 0 (OK) tai 255 (virhe).
tiedot	CardReadByte		Lue tavu dataa avoimesta tiedostosta Data = 0-255
tila	CardRecordMic	Bittisyvyys Näytteenottotaajuus Aika tiedostonimi	Tallenna mikrofoni Bittisyvyys = 0 (8bit) tai 1 (16bit) Näytteenottotaajuus = 0 (8K) tai 1 (16K) Tiedostonimi = <merkkijono>. Status = 0 (OK), 239 (tiedosto on olemassa) tai 255 (virhe).
tila	CardPlayback	tiedostonimi	Toista äänitiedosto Filename = <merkkijono> Status = 0 (OK), 239 (tiedostoa ei löydy) tai 255 (virhe).
tila	CardBitmap	X Y tiedostonimi	Kuvan näyttäminen nestekidenäytössä X = 0 - 127 Y = 0-31 Tiedostonimi = <merkkijono> Status = 0 (OK), 239 (tiedostoa ei löydy) tai 255 (virhe).

Liite 8

Mikroprosessori

Robotiikan kurssi Ohjeellinen opas

Block Diagram



Microcontroller Connections

Motors		IR Range		Expansion J15		Bluetooth		SD Card	
Left Speed Control	D1	IR1 Enable	D8	1	VBATT	TX	F12	MOSI	D4
Left Direction Control	D9	IR1 Signal	B4	2	V93	RX	F8	MISO	D5
Left Encoder Feedback	B11	IR2 Enable	D11	3	G12	Pair Status	F0	SCK	D6
Right Speed Control	D2	IR2 Signal	B5	4	G13	Reset	F13	CS	D7
Right Direction Control	D3	IR3 Enable	D12	5	G14	Servo Headers		USB	
Right Encoder Feedback	B12	IR3 Signal	B14	6	G15	Servo1	A7	USB Detect and LED	B13
LCD		IR4 Enable	D13	7	D9	Servo2	A9	D+	G2
Data	G8	IR4 Signal	B15	8	D10	Servo3	A10	D-	G3
Clock	G6	IR5 Enable	D16	9	D15	Servo4	A14	Switches	
A0	A9	IR5 Signal	B8	10	GND	E-Header Expansion		SW1	A8
CS	A2	IR6 Enable	A5	Expansion J16		1	E0	SW2	A1
Reset	A4	IR6 Signal	B9	1	A15	2	E1		
LEDs		IR7 Enable	A6	2	C1	3	E2		
D0	E0	IR7 Signal	B10	3	C2	4	E3		
D1	E1	IR8 Enable	C4	4	C3	5	E4		
D2	E2	IR8 Signal	B8	5	C13	6	E5		
D3	E3	Outputs		6	C14	7	E6		
D4	E4	Microphone	B3	7	G1	8	E7		
D5	E5	Speaker	F3	8	F1	9	GND		
D6	E6	Light Sensor	B2	9	F2	10	GND		
D7	E7	Battery Voltage	E9	10	GND				

Liite 9

Nuottien taajuudet

Robotiikan kurssi Ohjeellinen opas

Note	Freq (Hz)
C ₁	33
C [#] ₁ /D ^b ₁	35
D ₁	37
D [#] ₁ /E ^b ₁	39
E ₁	41
F ₁	44
F [#] ₁ /G ^b ₁	46
G ₁	49
G [#] ₁ /A ^b ₁	52
A ₁	55
A [#] ₁ /B ^b ₁	58
B ₁	62

Note	Freq (Hz)
C ₂	65
C [#] ₂ /D ^b ₂	69
D ₂	73
D [#] ₂ /E ^b ₂	78
E ₂	82
F ₂	87
F [#] ₂ /G ^b ₂	93
G ₂	98
G [#] ₂ /A ^b ₂	104
A ₂	110
A [#] ₂ /B ^b ₂	117
B ₂	123

Note	Freq (Hz)
C ₃	131
C [#] ₃ /D ^b ₃	139
D ₃	147
D [#] ₃ /E ^b ₃	156
E ₃	165
F ₃	175
F [#] ₃ /G ^b ₃	185
G ₃	196
G [#] ₃ /A ^b ₃	208
A ₃	220
A [#] ₃ /B ^b ₃	233
B ₃	247

Note	Freq (Hz)
C ₄	262
C [#] ₄ /D ^b ₄	277
D ₄	294
D [#] ₄ /E ^b ₄	311
E ₄	330
F ₄	349
F [#] ₄ /G ^b ₄	370
G ₄	392
G [#] ₄ /A ^b ₄	415
A ₄	440
A [#] ₄ /B ^b ₄	466
B ₄	494

Note	Freq (Hz)
C ₅	523
C [#] ₅ /D ^b ₅	554
D ₅	587
D [#] ₅ /E ^b ₅	622
E ₅	659
F ₅	698
F [#] ₅ /G ^b ₅	740
G ₅	784
G [#] ₅ /A ^b ₅	831
A ₅	880
A [#] ₅ /B ^b ₅	932
B ₅	988

Note	Freq (Hz)
C ₆	1047
C [#] ₆ /D ^b ₆	1109
D ₆	1175
D [#] ₆ /E ^b ₆	1245
E ₆	1319
F ₆	1397
F [#] ₆ /G ^b ₆	1480
G ₆	1568
G [#] ₆ /A ^b ₆	1661
A ₆	1760
A [#] ₆ /B ^b ₆	1865
B ₆	1976

Note	Freq (Hz)
C ₇	2093
C [#] ₇ /D ^b ₇	2217
D ₇	2349
D [#] ₇ /E ^b ₇	2489
E ₇	2637
F ₇	2794
F [#] ₇ /G ^b ₇	2960
G ₇	3136
G [#] ₇ /A ^b ₇	3322
A ₇	3520
A [#] ₇ /B ^b ₇	3729
B ₇	3951

Note	Freq (Hz)
C ₈	4186
C [#] ₈ /D ^b ₈	4435
D ₈	4699
D [#] ₈ /E ^b ₈	4978
E ₈	5274
F ₈	5588
F [#] ₈ /G ^b ₈	5920
G ₈	6272
G [#] ₈ /A ^b ₈	6645
A ₈	7040
A [#] ₈ /B ^b ₈	7459
B ₈	7902

Note	Freq (Hz)
C ₉	8372
C [#] ₉ /D ^b ₉	8870
D ₉	9398
D [#] ₉ /E ^b ₉	9956
E ₉	10548
F ₉	11176
F [#] ₉ /G ^b ₉	11840
G ₉	12544
G [#] ₉ /A ^b ₉	13290
A ₉	14080
A [#] ₉ /B ^b ₉	14918
B ₉	15804

Liite 4

Robotin nimen asettaminen

Robottiikan kurssi Ohjeellinen opas

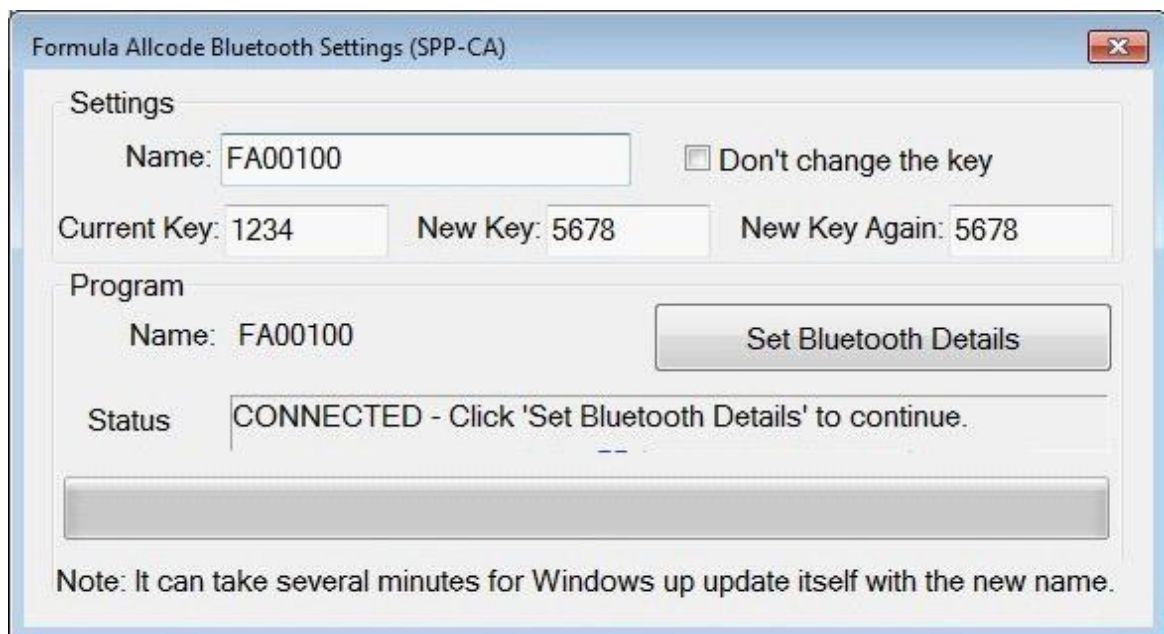
Formula AllCoden Bluetooth-nimi voidaan muuttaa käyttämällä PC:tä, johon on asennettu mLoader-ohjelmisto. mLoader-ohjelmiston voi ladata MatrixTSL:n verkkosivuilta.

<http://www.matrixtsl.com/formula-allcode/>

Kun olet ladannut mLoader-ohjelmiston, ohjelmistokansiossa on tiedosto nimeltä Formula AllCode Rename. Kaksoisnapsauta tätä tiedostoa käynnistääksesi mLoader-ohjelmiston Bluetooth-uudelleen nimeämistilassa.

Liitä Formula AllCode -robotti tietokoneeseen mukana toimitetulla USB-kaapelilla ja paina robotin reset-painiketta, jotta mLoader-ohjelmisto näkee robotin.

Kun olet tehnyt tämän, voit käyttää mLoader-ohjelmistoa Bluetooth-nimen ja pariliitosavaimen (jota kutsutaan joskus salasanaksi) asettamiseen syöttämällä tarvittavat tiedot tekstiin.



Suosittellemme pitämään avaimen oletusarvona "1234", ellei työskentele samassa tilassa muiden käyttäjien kanssa ja haluat estää muita muodostamasta yhteyttä robottiisi. Jos vaihdat salasana-avainta, sinun on syötettävä nykyinen avain, joten yritä muistaa se!

Jos avain on vaihdettu etkä muista sitä, ota yhteyttä Matrixiin saadaksesi ohjeet, miten miten se nollataan.

Liite 5

AllCode API -laiteohjelmiston uudelleen lataaminen

Robotiikan kurssi Ohjeellinen opas

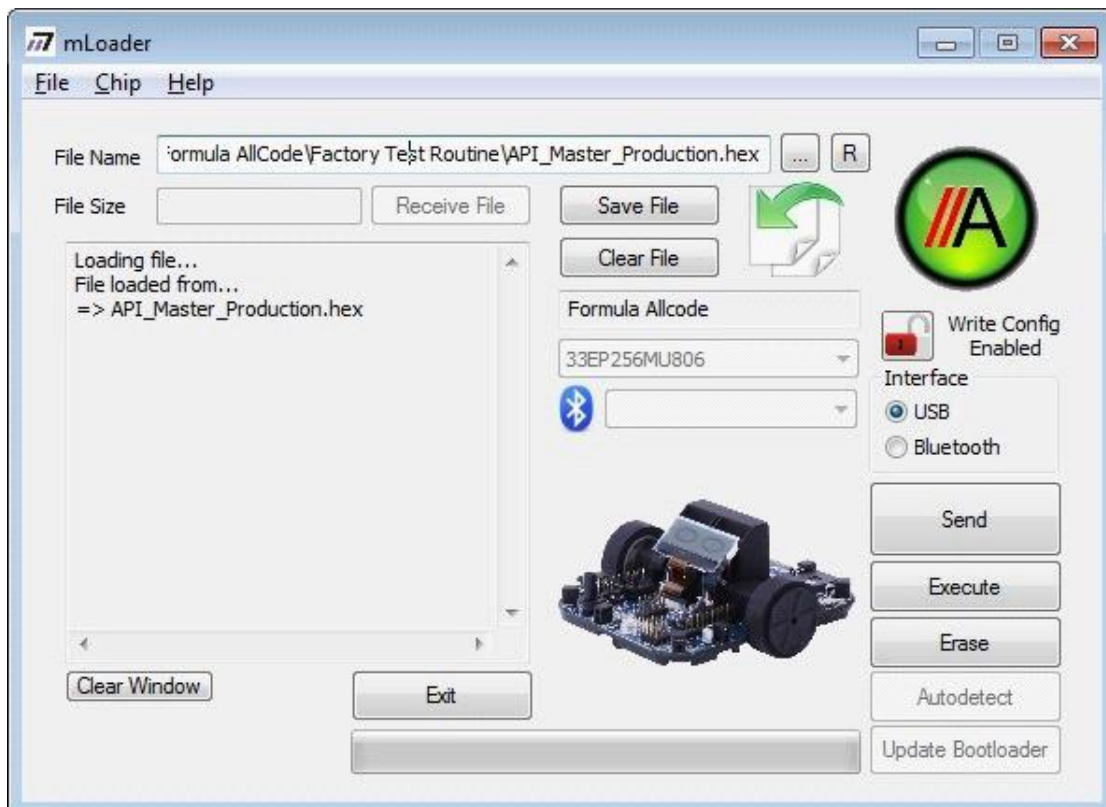
Oletusarvoisesti Formula AllCode -robotti toimitetaan API-firmaohjelmisto valmiiksi asennettuna. Koska robotti on ohjelmitavissa uudelleen käyttämällä kieliä, kuten Flowcodea tai C:tä, on mahdollista korvata tehtaan AllCode API-toiminnot. On myös mahdollista, että API-firmaohjelmistosta julkaistaan uudempia versioita, ja saatat haluta päivittää sen uusimpaan versioon.

API-firmware on ladattavissa heksatiedoston muodossa osoitteesta Formula MatrixTSL:n verkkosivuston AllCode-sivu. Jotta voit ladata heksatiedoston Formula AllCode -ohjelmistoon, tarvitset uusimman mLoader-ohjelmiston, joka on saatavana myös verkkosivuilta:

<https://www.matrixtsl.com/allcode/resources/>

Voit liittää Formula AllCodeen tietokoneeseen mukana toimitetulla USB-kaapelilla tai käyttämällä Bluetooth-datayhteys. Avaa mLoader-ohjelmisto ja paina robotin reset-painiketta, jotta mLoader-ohjelmisto näkee robotin.

Napsauta "..."-kuvaketta File Name -tekstikentän vieressä valitaksesi heksatiedoston, joka lähetetään osoitteeseen AllCode.



Napsauta Lähetä-painiketta siirtääksesi heksatiedoston Formula AllCodeen ja palauttaaksesi API-ohjelmiston. Muista painaa execute-painiketta, kun laiteohjelma on lähetetty, jotta Formula AllCode -laiteohjelma voi toimia.

Ilmoitukset

Robotiikan kurssi Ohjeellinen opas

Tavaramerkit

PIC, PICMicro ja dsPIC ovat Microchip Technologyn rekisteröityjä tavaramerkkejä. Raspberry Pi on Raspberry Pi Foundationin rekisteröity tavaramerkki.

Muut tässä asiakirjassa esiintyvät tuotenimet voivat olla omistajiensa tavaramerkkejä.

Julkaisutiedot

Versio	Julkaisupäivä	Huomautukset
Versio 1	25/02/2016	Ensimmäinen julkaisu
Versio 2	04/05/2016	Muunnettu Publisher-tiedostoksi
Versio 3	04/08/2017	Korjatut linkit eri resursseihin