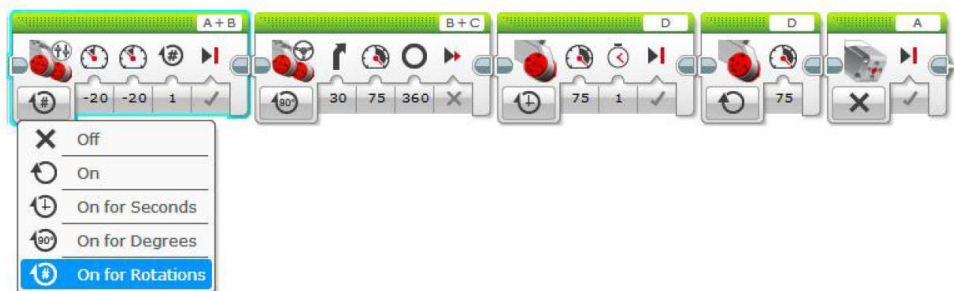


2. Moottorien ohjaaminen, ohjelman silmukat ja haarat

Harjoitukset jatkuvat käyttäen ensimmäisessä harjoituksessa rakennettua robottia. Kerrataan moottori-, viive-, sekä silmukkalohkojen parametointi.

Ensimmäisessä harjoituksessa kokeilimme kahta erilaista kahden moottorin ohjaamiseen tarkoitettua ohjelmalohkoa. Tämän harjoituksen aluksi käydään läpi moottorilohkojen asetukset.

Seuraavassa kuvassa on esitetty viisi moottorilohkoa.



1. ensimmäisenä on Move Tank moottorilohko, jolla käsketään tässä

(moodi = On for Rotations)

- a. moottoreita A ja B pyörimään synkronoidusti
- b. taaksepäin 20% nopeudella (-20)
- c. yhden kokonaisen kierroksen (1) verran ja
- d. asettamaan lopuksi sähköisen jarrunsa päälle (✓),
- e. ennen ohjelman suorituksen siirtymistä eteenpäin.



2. toisena on Move Steering moottorilohko, jolla käsketään tässä
(moodi = On for Degrees)

- robottia kaartamaan 30% jyrkästi (30)
- käyttäen moottoreita B ja C synkronoidusti,
- suhteellisella nopeudella 75% (75),
- nimellisesti 360 asteen (360) verran ja
- jäämään lopuksi vapaapyörintä -tilaan (X),
- ennen ohjelman suorituksen siirtymistä eteenpäin.



3. Kolmantena on Large Motor moottorilohko, jolla käsketään tässä
(moodi = On)

- moottoria D pyörimään
- eteenpäin 75% nopeudella (75)
- yhden sekunnin (1) ajan ja
- asettamaan lopuksi sähköisen jarrunsa päälle (✓),
- ohjelman suoritus siirtyy välittömästi eteenpäin.



4. Neljäntenä on Large Motor moottorilohko, jolla käsketään tässä
(moodi = On for Rotations)

- moottoria D pyörimään
- eteenpäin 75% nopeudella (75)
- määräämättömän ajan verran.
- Ohjelman suorituksen siirtymistä eteenpäin.



5. Viidentenä on Medium Motor moottorilohko (pienempi moottori),
jolla käsketään tässä (moodi = Off)

- moottoria A pysähtymään välittömästi
- asettamaan lopuksi sähköisen jarrunsa päälle (✓),
- ohjelman suoritus siirtyy välittömästi eteenpäin.



Lisää tietoa ja vinkkejä löydät klikkaamalla ohjelmointi ohjelman yläaidasta
Help -> Show EV3 Help -> Programming Blocks -> ja valitse tyyppi valikosta.

Polku

"Polku vie suunnistajaa rastilta rastille,
ohjelmaa käskyltä käskylle."

Intron loppuksi opetellaan vielä ohjelmalohkojen kytkemistä toisiinsa polulla. Tietotekniikan yleiskielisissä (englanti) kirjoituksissa ohjelmalohkoja toisiinsa kytkevästä viivoista käytetään usein sanaa wire, joka voidaan tulkita langaksi, johtimeksi, tai vaijeriksi. Suomenkielistä sanaa polku voidaan kuitenkin pitää tässä asiayhteydessä vähintään yhtä kuvaavana.

Polku on ohjelmassa yksisuuntainen reitti, jolla voidaan yhdistää kaksi ohjelmalohkoa, jota pitkin ohjelman suoritus kulkee. Joskus se haarautuu, mutta aina sillä on alku ja loppu.

Polun piirtäminen

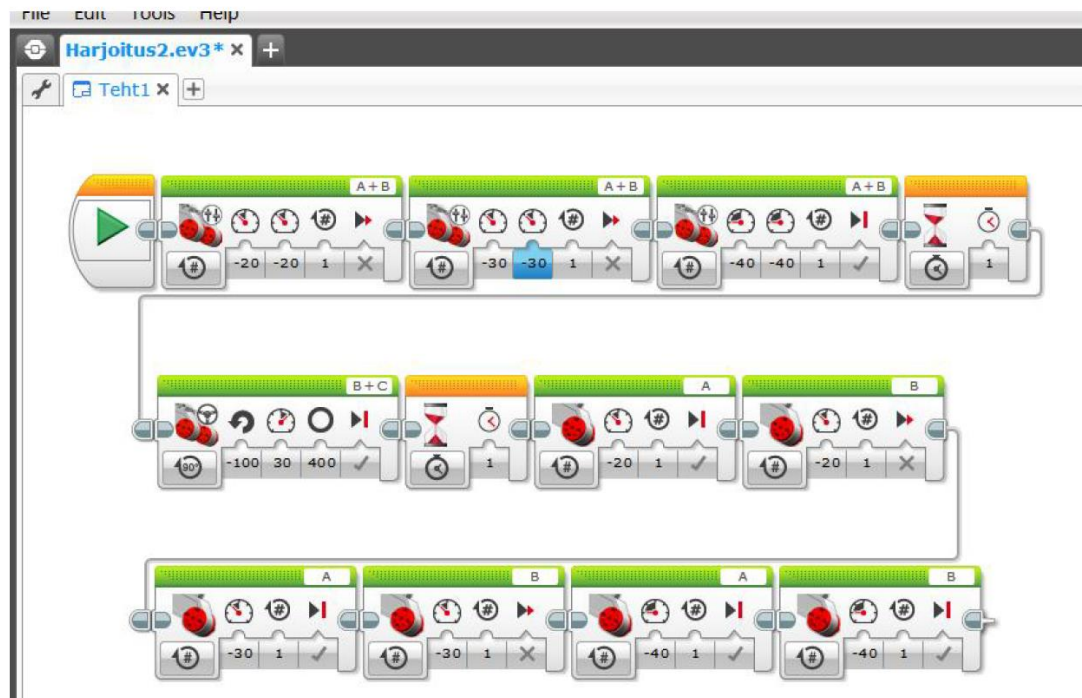
Vie hiiren osoitin ohjelmalohkon oikeassa reunassa olevan "liittimen" päälle (hiiren osoitin vaihtuu lankarullaksi). Paina hiiren vasen painike alas ja vedä lankarulla ohjelmassa seuraavaksi suoritettavan lohkon vasemman reunan liittimeen. Tämä vaatii tarkkaa kättä. Liitos syntyy vasta kun liitin vaihtaa värinsä siniseksi.

- Polun saa pois klikkaamalla sen loppua.
- Ohjelmalohkot saa siirtymään yhteen ja niiden välissä olleen polun katoamaan klikkaamalla polun alkua.

Tehtävä1

Tutustutaan moottorilohkojen toimintaan. Harjoitusta varten lähdemme tekemään ohjelmaa, jossa robotti ajaa ensin asteittain kiihdyttäen noin 50cm, kääntyy ja palaa asteittain kiihdyttäen takaisin. Menomatka ja kääntyminen tehdään kahden moottorin moottorilohkoilla, paluu yhden moottorin moottorilohkoilla.

Tee alla olevan mallin mukainen ohjelma.



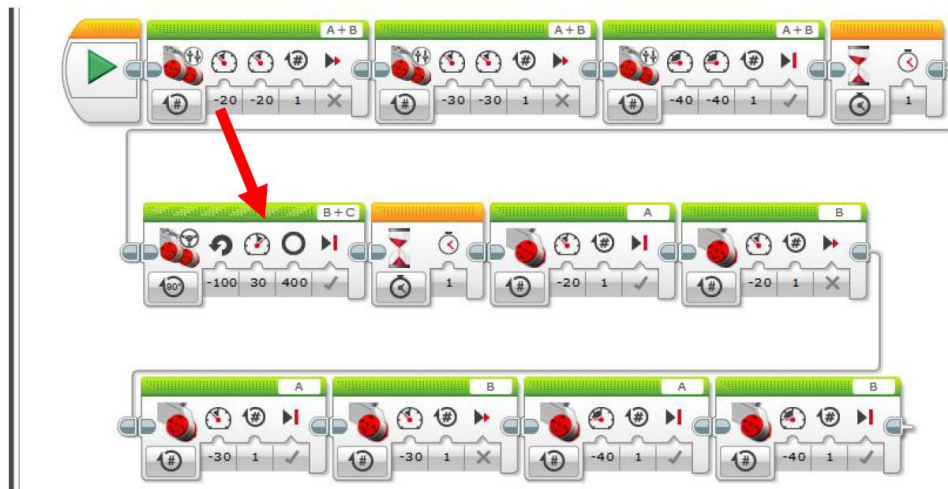
Laske robotti lattialle, lataa ohjelma ja kokeile sitä.

Robotti ei tullut takaisin lähtöpisteeseensä. Ohjelmassa on nyt virhe.

EV3:een liittyvä mahdollisuus tarkastella sen toimintoja reaaliaikaisesti on osoittautunut käteväksi avuksi ohjelmien vikojen etsimisessä ja muutenkin ohjelmien kehittämisessä.

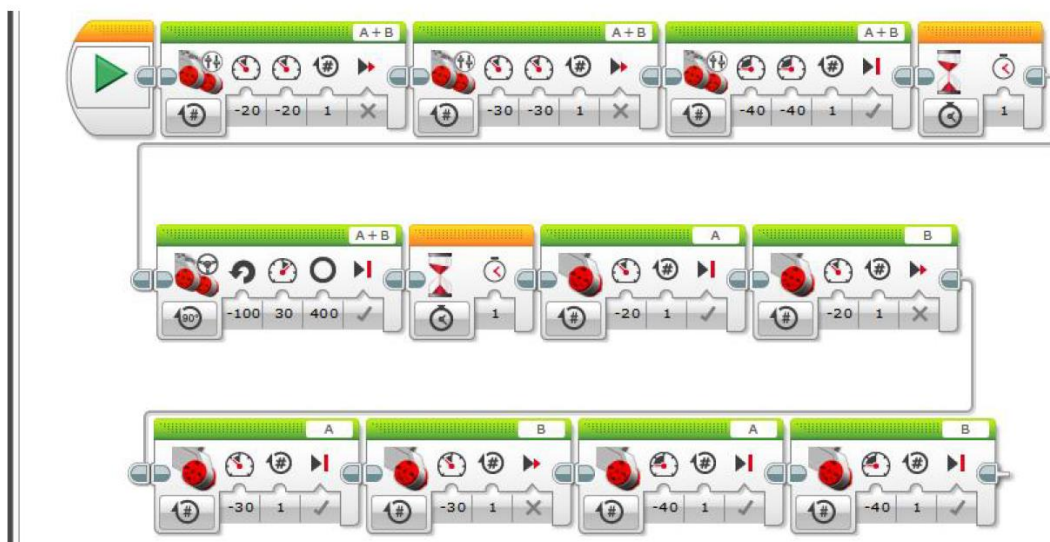
Pysäytetään ohjelman suoritus, kytketään ohjelmointikaapeli takaisin kiinni robottiin ja käynnistetään ohjelma uudestaan. Seuraa nyt ohjelman kulkua tietokoneen näytöltä. Ohjelmalohkon, jota kulloinkin suoritetaan, tunnistat sen yläpalkissa juoksevasta raidoituksesta.

Ohjelman suorituksen pitäisi jumittaa nyt viidennen ohjelmalohkon kohdalla.



Kääntymiseen käytetyn kahden moottorin moottorilohkon suoritus jäi kesken. Se vertailee moottorien pyörimisen keskinäistä suhdetta, ja nyt se ei saanut minkäänlaista pyörimisestä kertovaa tietoa lähtöportista C (käytettävien moottorien määrittely oli tehty virheellisesti).

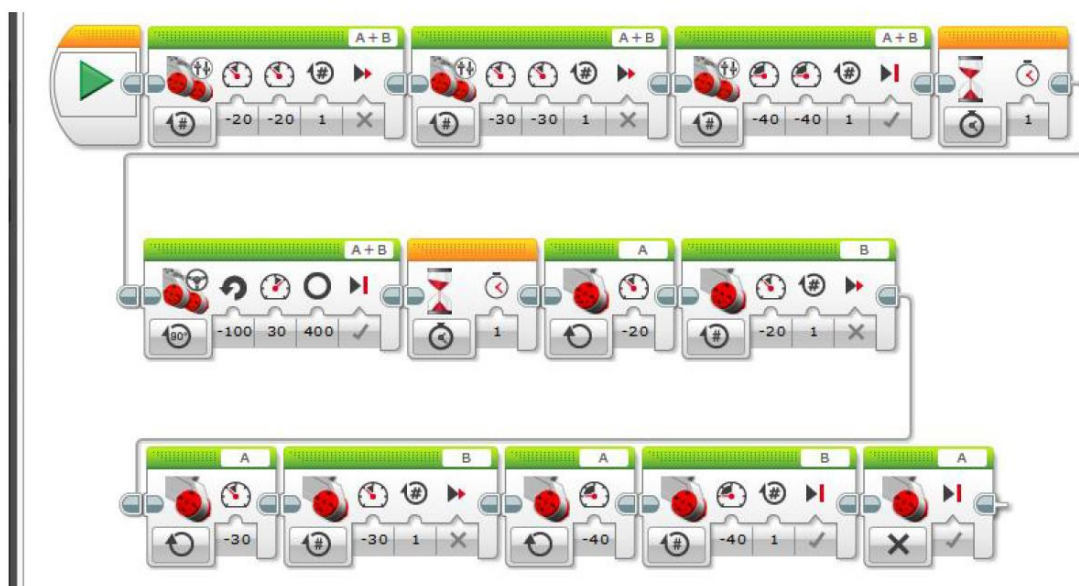
Korjaa käytettävien moottorien määrittely nyt oikeaksi (A ja B).



Kokeillaan jälleen, palaako robotti nyt takaisin lähtöpaikalleen?

Nyt robotti kääntyi, mutta kiemurtelee paluumatkalla, mikä ei ollut tarkoituksena. Kun seurataan ohjelman kulkua, voidaan havaita suorittavan aina A-moottoria koskevan komennon ensin loppuun asti, ennen B-moottorin vastaavaa komentoa.

Muutetaan seuraavaksi ohjelmaa siten, että ohjelma vain käynnistää A-moottorin ja antaa välittömästi B-moottorille käskyn, joka pitää sisällään tiedon myös jakson kestosta. Lopuksi pitää A-moottori vielä pysäyttää erikseen.



Kokeillaan jälleen ohjelman toimintaa. Seuraa robotin toimintaa tarkasti. Miten sen toiminta menomatkalla poikkeaa paluumatkasta?

Hyvät ohjelmointikäytännöt

Näissä harjoituksissa moottorin pyörimissuunta on tarvittaessa käännetty käyttämällä miinusmerkkiä nopeustiedon edessä. Pyörimissuuntaan voi kääntää myös laittamalla miinusmerkin moottorin pyörimistä asteina tai kokonaisina kierroksina määrittävän luvun eteen.

Tässä piilee kuitenkin pieni pulma. Ohjelmoitaessa tulee aina välillä vastaan tilanteita, joissa esimerkiksi robotin kulkema matka on kätevintä määritellä sekunteina, esim. 2.7s (huom. voit käyttää myös sekunnin osia). Mutta aika kulkee vain yhteen suuntaan, eteenpäin.

Pyörimissuunnan kääntäminen käyttämällä miinusmerkkistä nopeustietoa on osoittautunut yhdeksi monista hyvistä ohjelmointikäytännöistä. Yhteistä hyville ohjelmointikäytännöille on niiden taipumus auttaa meitä kirjoittamaan helpolukuisempia ohjelmia ja välttämään virheitä.

”Tiesitkö että kuvakkeiden käyttö ohjelmoinnissa on yksi tärkeimmistä hyvistä ohjelmointikäytännöistä.”