

Programozás 7.rész

Optimalizálás NXT-G-ben

Optimalizálás általánosan

Optimalizálás NXT-G-ben (I.)

MOTOR és MOVE blokk



Általánoságban véve ez a két blokk hasonló. Viszont ha mind a kettőt felhasználjuk a programunkban, az NXT kénytelen mindkettőt feltölteni a memóriájába. De ezek a blokkok viszonylag sok memóriát fogyasztanak. Ha tehetjük csak az egyiket használjuk! Egy két tulajdonság, ami segíthet dönteni:

- MOVE blokk szinkronizálja két motor mozgását. A MOTOR viszont csak egy motort képest meghajtani.
- A MOTOR blokknál beállítható, hogy ellenállástól függetlenül, a motor mindig ugyanazon sebességgel mozogjon, így lehetővé válik nagyon alacsony sebességű mozgás is.

Optimalizálás NXT-G-ben (II.)

Hangok


Próbáljunk meg minél rövidebb hangmintákat használni, ugyanis a hangok sok helyet foglalnak az NXT Brick memóriájából. Egy WAV alapú *.RSO fájl hanghullámok folyamatos mintáit tartalmazza. (8 kHz mintavételű 10 másodperces hang 80 kB helyet foglal az NXT memóriájában)

Ezért inkább próbáljunk meg Melody fájlokban (MIDI) tárolt zenéket letölteni az NXT-re. (*.RMD)

A MIDI fájlokban adott szólamok hangzási ideje van letárolva. (pld.: egy szólamnak 10 másodperces hangzása csupán 4 Byte helyet foglal: 2 Byte frekvencia, 2 Byte hangzási hossz. Persze egy zene nem egy szólamból áll...)

Counterpoint on
The First Noel

Traditional English Carol
arranged by Anne Ku
London 18 Nov 2000

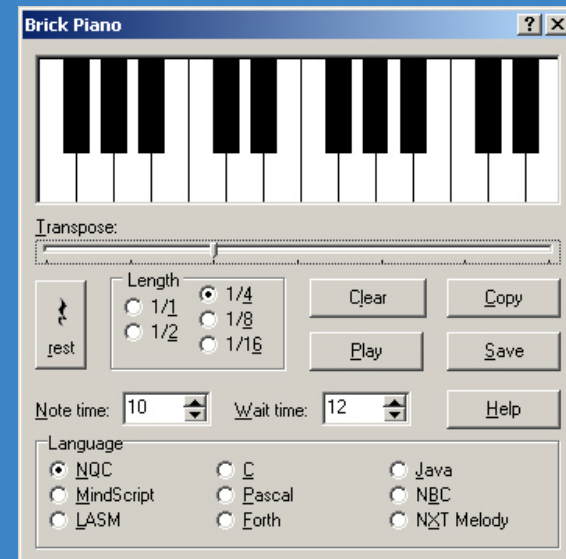


The image shows a musical score for 'The First Noel' in G major. It consists of four systems of two staves each (treble and bass clef). The music is a counterpoint between the two parts. The first system starts with a whole rest in the treble and a half note G in the bass. The second system has a half note G in the treble and a half note G in the bass. The third system has a half note G in the treble and a half note G in the bass. The fourth system has a half note G in the treble and a half note G in the bass. The score is arranged by Anne Ku and is a traditional English carol.

Optimalizálás NXT-G-ben (II.)

Zene

Kevés helyet foglaló zenét a Brick Command Center segítségével is létrehozhatunk.



Vagy komponáljuk meg a saját zenéinket programból!



RobotC

```
PlayTone (523,40) ; wait10Msec (50) ;  
PlayTone (587,40) ; wait10Msec (50) ;  
PlayTone (659,40) ; wait10Msec (50) ;  
PlayTone (523,40) ; wait10Msec (50) ;  
PlayTone (659,40) ; wait10Msec (50) ;  
PlayTone (698,40) ; wait10Msec (50) ;  
PlayTone (784,80) ; wait10Msec (90) ;  
PlayTone (659,40) ; wait10Msec (50) ;  
PlayTone (698,40) ; wait10Msec (50) ;  
PlayTone (784,80) ; wait10Msec (90) ;  
PlayTone (784,20) ; wait10Msec (30) ;  
PlayTone (880,20) ; wait10Msec (30) ;  
PlayTone (784,20) ; wait10Msec (30) ;
```

NXT-G

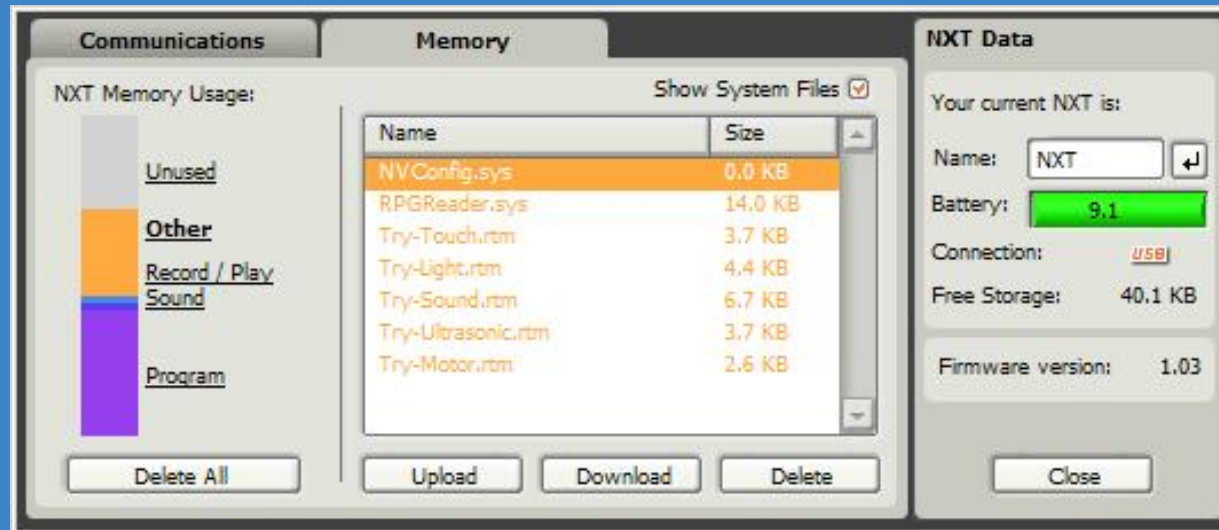
Optimalizálás NXT-G-ben (II.)

Töröljük, amire nincs szükségünk

Az NXT téglában számos olyan fájl található, amire a mi programunknak nincs szüksége.

Töröljük a Demo és Try-Me programokat!

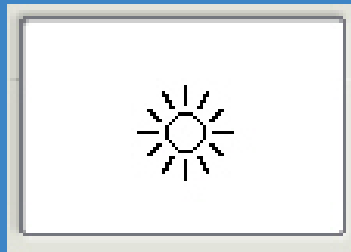
Töröljük a használaton kívüli hangfájlokat! (bejelentezési hang, gombok hangja, stb)



Optimalizálás NXT-G-ben (III.)

Kijelző

A kijelzőre küldött grafika is sok helyet foglal, ezért ott is ajánlatos az ábrát programból legenerálni.



Egy RIC fájl

```
rajz.nxc
task main ()
{
    // Napocska
    LineOut(25, 5, 40, 20, false);
    LineOut(50, 0, 50, 20, false);
    LineOut(75, 5, 60, 20, false);
    LineOut(90, 30, 60, 30, false);
    LineOut(75, 55, 60, 40, false);
    LineOut(50, 60, 50, 40, false);
    LineOut(25, 55, 40, 40, false);
    LineOut(10, 30, 40, 30, false);
    CircleOut(50, 30, 10, false);
    Wait(4000);
}
```

NXC-ben

Ha nagy hangsúlyt fektetünk a fejlesztésben a grafikára, érdemes RobotC-ben írni programunkat, kihasználva ennek a nyelvnek a rajzolási gyorsaságát.

Optimalizálás NXT-G-ben (IV.)

Blokkok frissítése

NXT-G-hez is folyamatosan jelennek meg újabbnál újabb komponensek (blokkok), amikkel optimalizálhatjuk programjainkat. Akár a fent említett főbb blokkjainkat is lecserélhetjük!

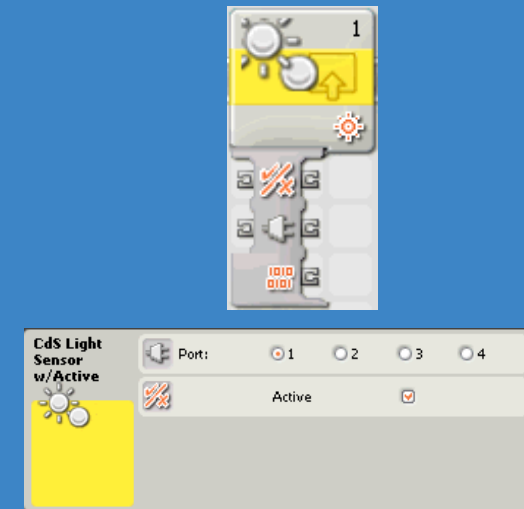
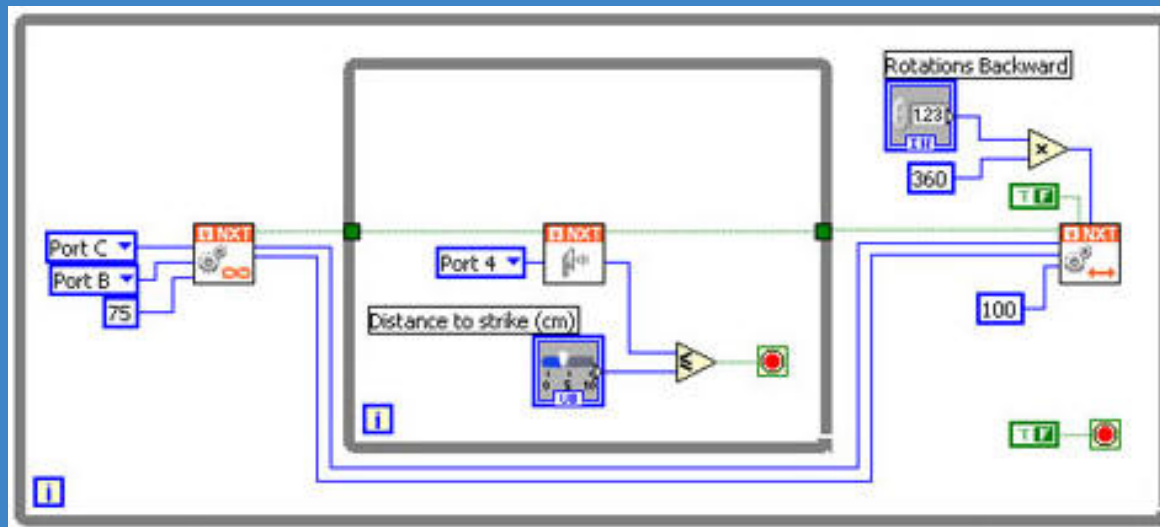


Ilyen csomag például a Mini Blocks

Optimalizálás NXT-G-ben (V.)

Speciális blokkok

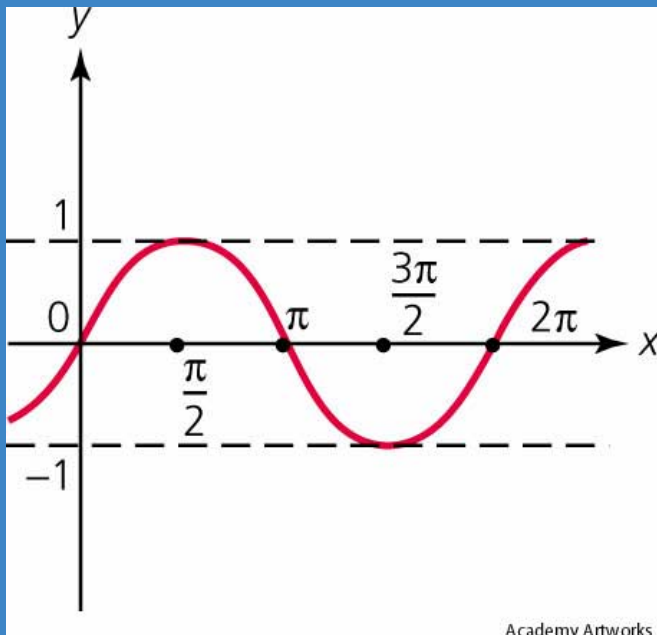
Ha különleges, és gyors megoldásra van igényünk, mi is létrehozhatunk alap NXT-G blokkokat a LabView környezet segítségével.



Fényérzékelő használata, mint egyszerű fényforrás

Gyorsítás tömbök segítségével

Ha viszonylag gyakran használunk trigonometrikus képleteket, akkor célszerű nem kiszámolni a szögfüggvény értékét, hanem egy egy dimenziós tömbben eltárolni az értékeket.
(Szögfüggvény számolása sok idő)



```
/*  
    Ha nagyon sokszor szuksegunk van egy adott fok szinuszara,  
    erdemes a trigonometrikus fuggveny meghivasa helyett a program  
    elejen elmenteni egy tombbe az adott fokokhoz tartozo  
    szinus ertekeket  
*/  
float fArray[360];  
float ertek;  
  
task main ()  
{  
    int i;  
  
    for (i = 0; i < 360; ++i)  
    {  
        fArray[i] = sinDegrees(i);  
    }  
  
    // 27 fok szinusza  
    ertek = fArray[27];  
}
```

További érdekességek

bricxcc.sourceforge.net

forums.nxtasy.org

mindstorms.lego.com/support/updates/

www.legoeducation.com/community/9/blogs/nxt/archive/2006/11/21/mini-block-library.aspx

www.firstlegoleague.org/sitemod/upload/Root/WritingEfficientNXTGPrograms2.pdf

www.nxtasy.org/repository/nxt-g-blocks/

thenxtstep.blogspot.com

www.nxt-mindstorms.com

<http://www.hightechkids.org/?2-2-110-2>