

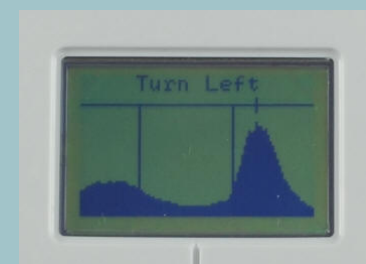
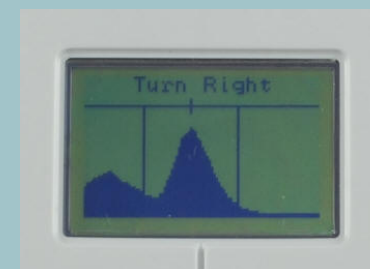
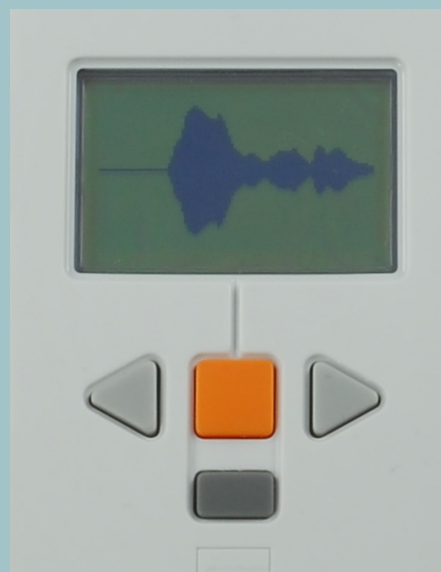
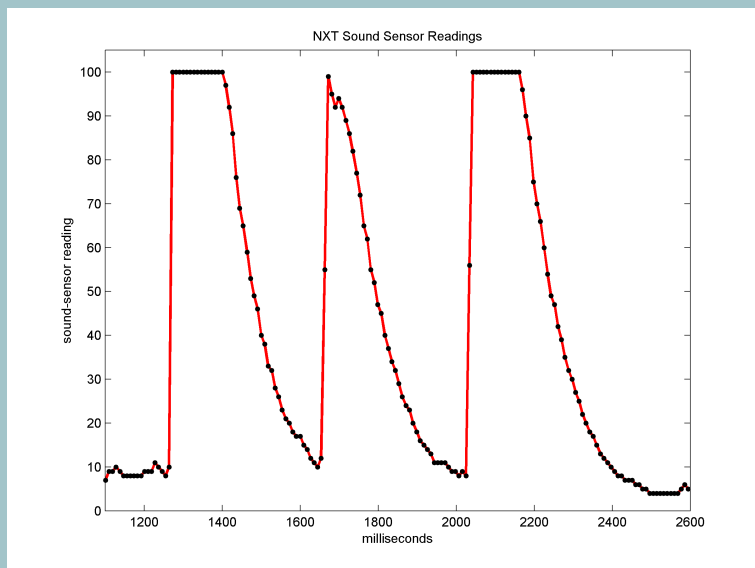
Programozás 6. rész

Mintavételezés

Tárolás tömbökben
Tárolás fájlrendszerben

Mintavételezés

Hang



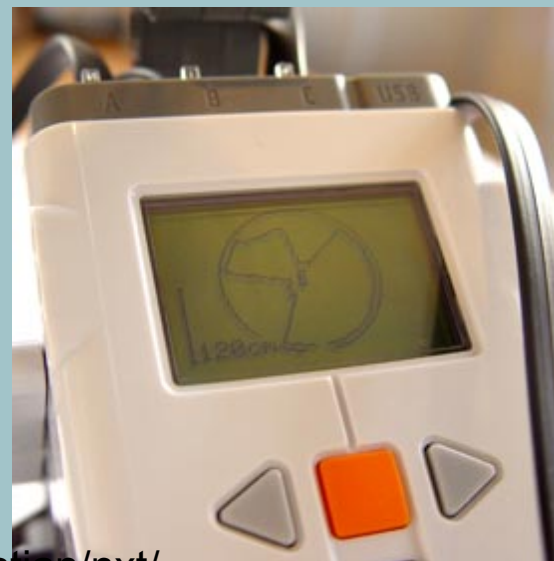
Hangérzékelésnél mindenféleképpen figyelniük kell, hogy minél gyorsabban dolgozzuk fel a szenzortól beérkező jelet. Gyorsaságból adódóan részletgazdagabb mintánk lesz, amivel könnyebben tudunk "Beszéfelismerést" megvalósítani.

Mintavételezés

Radar



<http://kisd.de/~krystian/nxt/>



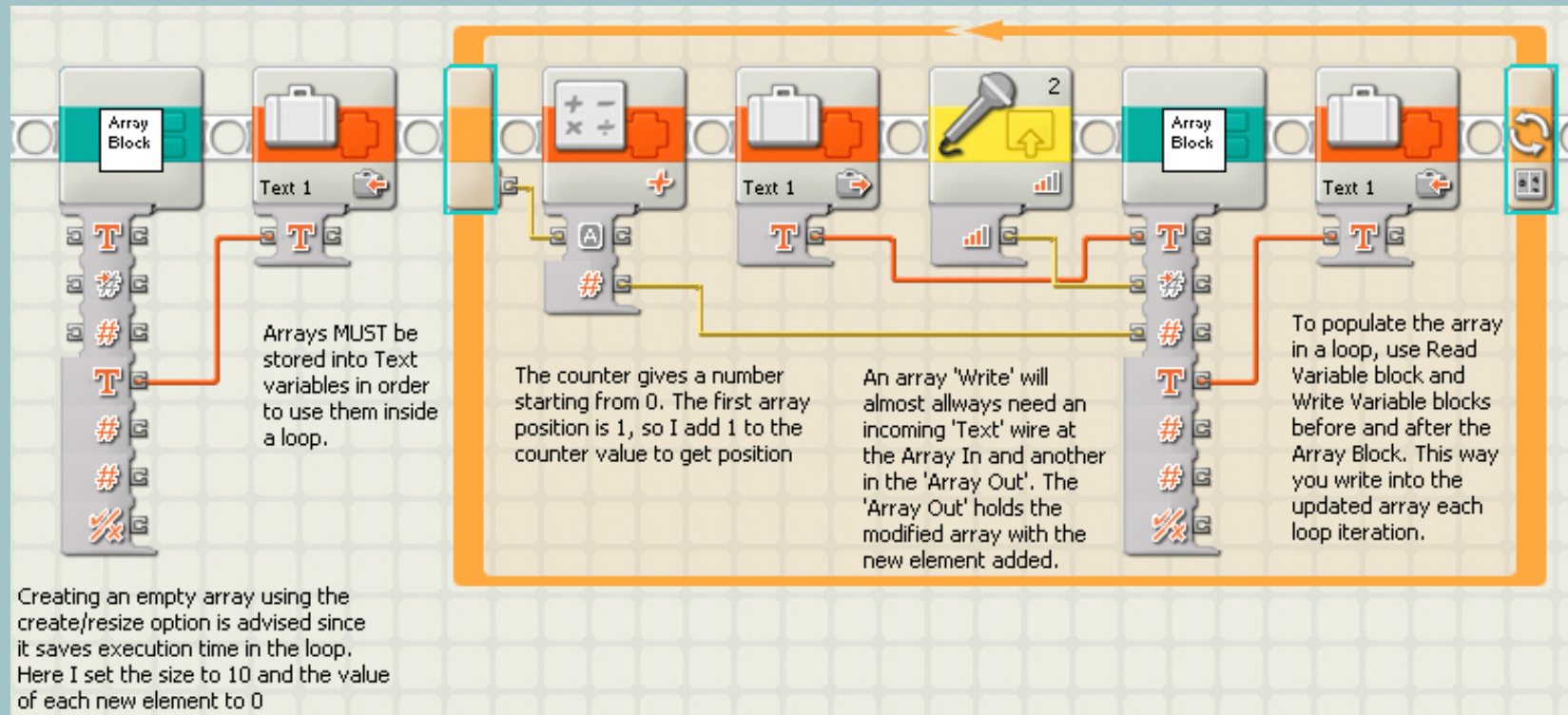
A radarkép elkészítése: a távolságmérőtől kapott értéket megfelelő szögfüggvény segítségével kijelezzük a NXT kijelzőjén.

De mit kell tenni, ha a radarképet nagyítani szeretnénk?

EI kell tárolni a szenzor adatait. Minél több adatot eltárolunk, annál részletesebb képet tudunk készíteni. És ha közben a robot még mozog is jobb ha minél gyorsabb a mintavételezés.

Tárolás tömbökben (I)

NXT-G



Mintavételezési idő nagyon lassú.
Csak egydimenziós tömböt támogat.

Tárolás tömbökben (II)

NXC

Gyorsabb az NXT-G-nél. (kb 20x)
Jelenlegi NXC verzióban maximum 4 dimenziós tömböket hozhatunk létre, viszont kétdimenziós tömbnél már inicializáló parancsra van szükségünk. (ArrayInit)
Csak egydimenziós tömböt támogat.

```
#include "NXCDefs.h"

task main()
{
  int myArray[][];
  int myVector[];
  byte fooArray[][][];

  ArrayInit(myVector, 0, 10); // 10 nulla kerül a myVector-ba
  ArrayInit(myArray, myVector, 10); // 10 vektor kerül a myArray-be
  ArrayInit(fooArray, myArray, 2); // 2 myArrays megy a fooArray-ba

  myVector = myArray[1];
  fooArray[1] = myArray;
  myVector[4] = 34;
  myArray[1] = myVector;

  int ax[], ay[];

  ArrayBuild2(ax, 5, 6); // 2 darab elem betöltése a tömbb
  ArrayBuild4(ay, 2, 10, 6, 43);
  int axlen = ArrayLen(ax);
  ArraySubset(ax, ay, 1, 2); // ax = {10,6}
  if (ax == ay) { // array comparisons supported as of b25
  }
}
```

RobotC

"Beszéd felismeréshez" ideális a gyorsasága miatt.
A tömbökbe törtszámokat is tárolhatunk.

```
// Deklaracional is meg lehet adni az tobbdimenzios
// tomb elemeinek kezdeti erteket
char doubleArray4[3][4] =
{'X','X','X','X', 'X','X','X','X', 'X','X','X','X'};

{
  int index1;
  int index2;

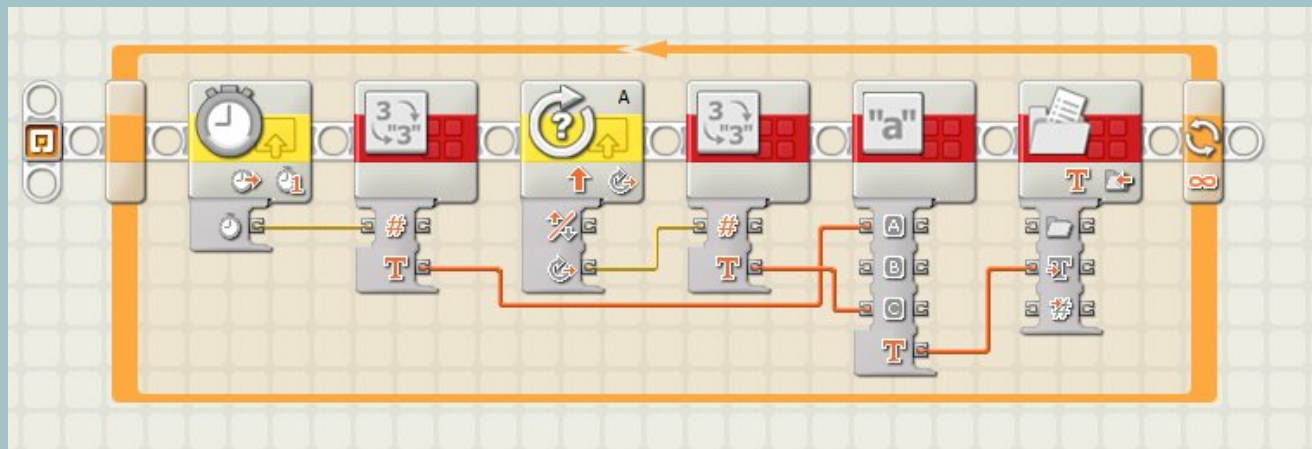
  char doubleArray4[3][4];

  // elemek feltoltese a ketdimenzios tombbe
  for (index1 = 0; index1 < 3; ++index1)
  {
    for (index2 = 0; index2 < 4; ++index2)
      doubleArray4[index1][index2] = 'X';
  }
  return;
}
```

Tárolás fájlokban (I)

Az NXT fájlok nevének meg kell felelni a 15.3 as kritériumnak.
Az NXT egyszerre 4 fájlba tud írni folyamatosan

NXT-G



File Access
Block

Csak az alap fájlkezelő funkciók érhetőek el:
Fájl létrehozás, törlés, írás olvasás

Tárolás fájlokban (II)

NXC

RobotC

```
task main ()
{
    result = CreateFile("data.txt", 1024, handle);
    // Hozzáírásnál használjuk: OpenFileAppend("data.txt", fsize, handle);

    // Megnyitás olvasásra
    result = OpenFileRead("data.txt", fsize, handle);

    // Író és olvasó funkciók
    result = Read(handle, value);
    result = ReadBytes(handle, len, buffer);
    result = Write(handle, value);
    result = WriteString(handle, "testing", count);
    result = WriteBytes(handle, buffer, count);
    result = WriteBytesEx(handle, len, buffer);

    // A sor vége jeleket is használhatjuk
    result = ReadLn(handle, value);
    result = WriteLnString(handle, "testing", count);
    result = WriteLn(handle, value);

    // A fájlt használat után mindig le kell zárni!
    result = CloseFile(handle);

    // A fájlok átnevezhetők ...
    result = RenameFile("data.txt", "mydata.txt");
    // Törölhetőek
    result = DeleteFile("data.txt");
}
```

Az NXC átfogóan támogatja az NXT fájlrendszerét.

NXT-G funkcióin kívül a fájlokat át lehet nevezni és kezeli a sorvége jeleket.

```
#pragma platform(NXT) // This program only works on NXT
const string sFileName = "testFile.dat";
TFileIOResult nIoResult;
TFileHandle hFileHandle;

{
    byte nParmToReadByte = 0;
    short nParmToReadWord = 0;
    long nParmToReadLong = 0;
    float fParmToReadFloat = 0;

    Delete(sFileName, nIoResult);
    hFileHandle = 0;
    OpenWrite(hFileHandle, nIoResult, sFileName, nFileSize);
    WriteByte(hFileHandle, nIoResult, 12);
    WriteShort(hFileHandle, nIoResult, 1234);
    WriteLong(hFileHandle, nIoResult, 1234);
    WriteLong(hFileHandle, nIoResult, 123456789);
    WriteFloat(hFileHandle, nIoResult, 1234);
    WriteFloat(hFileHandle, nIoResult, 12.34);
    WriteString(hFileHandle, nIoResult, "String Constant");

    Close(hFileHandle, nIoResult);

    Delete("testFile2.dat", nIoResult);
    Rename("testFile2.dat", nIoResult, sFileName);

    OpenRead(hFileHandle, nIoResult, "testFile2.dat", nFileSize);
    ReadByte(hFileHandle, nIoResult, nParmToReadByte);
    ReadShort(hFileHandle, nIoResult, nParmToReadWord);
    ReadLong(hFileHandle, nIoResult, nParmToReadLong);
    ReadLong(hFileHandle, nIoResult, nParmToReadLong);
    ReadFloat(hFileHandle, nIoResult, fParmToReadFloat);
    ReadFloat(hFileHandle, nIoResult, fParmToReadFloat);
    Close(hFileHandle, nIoResult);
}
```

További érdekességek

<http://kisd.de/~krystian/nxt/>

<http://nxtasy.org/2006/12/17/array-block-10-beta-version/>

<http://www.g7smy.dyndns.org/robotc/?BlueToothMessaging>

<http://thenxtstep.blogspot.com/2007/09/datalogging-with-nxt-g.html>

<http://mindstorms.lego.com>