

```

                                di splaySquareAndSquareRoot2_0.nxc
// File: di splaySquareAndSquareRoot2_0.nxc
// Date: 10/01/12 15:43
// Desc: Display number, its square and square root save to file
// Vers: 2.0
// Refs: di splaySquareAndSquareRoot1_0.nxc

// Global variables (for file writing)
unsigned int result; // flag returned when handling files
byte fileHandle; // handle to the data file
short bytesWritten; // number of bytes written to the file
string fileHeader; // column header for data in the file
int fileNumber, filePart; // integers to split up data file names
string fileName; // name of the file
string strFileNumber; // file number e.g myDataFile 1, 2, 3
string strFilePart; // file part e.g. myDataFile1-1, 1-2, 1-3
string text; // string to be written to file i.e. data values

// Create and initialize a file
void InitWriteToFile() {
    fileNumber = 0; // set first data file to be zero
    filePart = 0; // set first part of first data file to zero
    fileName = "squareData.csv" ; // name of data file
    result=CreateFile(fileName, 1024, fileHandle);
    // NXT Guide Section 9.100 pg. 1812 and Section 6.59.2.2 pg. 535
    // returns file handle (unsigned int)

    // check if the file already exists
    while (result==LDR_FILEEXISTS) // LDR_FILEEXISTS returns if file pre-exists
    {
        CloseFile(fileHandle);
        fileNumber = fileNumber + 1; // create new file if already exists
        fileName=NumToStr(fileNumber);
        fileName=StrCat("squareData", fileName, ".csv");
        result=CreateFile(fileName, 1024, fileHandle);
    } // end while

    // play a tone every time a file is created
    PlayTone(TONE_B7, 5);
    fileHeader = "x, x^2, sqrt(x)" ; // header for myData file
    WriteLnString(fileHandle, fileHeader, bytesWritten);
    // NXT Guide Section 6.59.2.43 pg. 554
    // Write string and new line to a file
    // bytesWritten is an unsigned int. Its value is # of bytes written
} // end InitWriteToFile

void WriteToFile(string strTempText) {
    // strTempText stores the text (i.e. ticks and motorRpm to be written to file

    // write string to file
    result=WriteLnString(fileHandle, strTempText, bytesWritten);
    // if the end of file is reached, close the file and create a new part
    if (result==LDR_EOFEXPECTED) // LDR_EOFEXPECTED is flagged when end-of-file
    {
        // close the current file
        CloseFile(fileHandle);
        // NXT Guide Section 6.59.2.1 pg. 535
        // Closes file associated with file handle

        // create the next file name
        filePart = filePart + 1;
        strFileNumber = NumToStr(fileNumber);
        strFilePart = NumToStr(filePart);
    }
}

```

```

        displaySquareAndSquareRoot2_0.nxc
fileName = StrCat("squareData" , strFileNumber,"-", strFilePart , ".csv");

// delete the file if it exists
DeleteFile(fileName);
// NXT Guide Section 6.59.2.5 pg. 537
// Delete the file specified by the string input

// create a new file
CreateFile(fileName, 1024, fileHandle);
// play a tone every time a file is created
PlayTone(TONE_B7, 5);
WriteLnString(fileHandle, strTempText, bytesWritten);
} // end if
} // end WriteToFile

// Close the file
void StopWriteToFile() {
    // close the file
    CloseFile(fileHandle);
} // end StopWriteToFile

task main ()
{
    int x; // integers from 1 to 10
    int xSquared; // square of x
    float xSquareRoot; // square root of x

    string strX;
    string strXSquared;
    string strXSquareRoot;

    // Create a new file that captures time and motor speed
    InitWriteToFile();

    for (x = 1; x <=10; x++) {
        xSquared = x*x;
        xSquareRoot = sqrt(x);

        // TextOut (xPosition, yPosition, string) put string on LCD's x,y position
        // NB: x = y = 0 is lower left corner of LCD; +x goes rights, +y goes up
        // FormatNum is a string with sprintf syntax

        TextOut (10, LCD_LINE4, FormatNum("x = %d" , x));
        TextOut (10, LCD_LINE5, FormatNum("xSquared = %d" , xSquared));
        TextOut (10, LCD_LINE6, FormatNum("sqrt(x) = %3.3F" , xSquareRoot));
        Wait (SEC_2);

        // Create string version of calculated values
        strX = FormatNum("%d" , x);
        strXSquared = FormatNum("%d" , xSquared);
        strXSquareRoot = FormatNum("%3.3F" , xSquareRoot);

        // Concatenate the 3 strings into a single one.
        // Write resulting string to file. The text will be end with a EOL
        text=StrCat(strX, ",", strXSquared, ",", strXSquareRoot, ",");
        WriteToFile(text);
    } // end of for loop

    // Finished computing square and square root, so clean up and quit
    ClearScreen();
    TextOut(0, LCD_LINE2, "Quitting", false);
    StopWriteToFile();
    PlaySound(SOUND_LOW_BEEP); // Beep to signal quitting

```

```
    Wait (SEC_2);
} // end of main
```

displaySquareAndSquareRoot2_0.nxc