

Programming with TI-nSpire

A guide on how to use ndless

Julian M.



11

Table of contents

1	Getting started	3
1.1	Disclaimer	3
1.2	Differents: TI-Basic vs. C	3
1.3	What is required?	3
2	Installation	4
2.1	MSYS	4
2.1.1	Adding MSYS to the PATH variable	4
2.2	YAGARTO	4
2.3	7-Zip	4
2.3.1	Adding 7-Zip to the PATH variable	4
2.4	Tortoise SVN	4
3	Building ndless	5
3.1	Check out the source code	5
3.2	Make!	5
4	Emulating the nSpire	7
4.1	Things you need	7
4.2	Setup	7
4.2.1	Create a raw image file	7
4.2.2	Install the OS	7
4.2.3	Preventing crashes	8
4.3	Overview	8
4.4	Installing ndless	9
4.5	Your first program	10

1 Getting started

1.1 Disclaimer

To be done.

1.2 Differents: TI-Basic vs. C

These are some examples I picked to show some differents between TI-Basic and C.

Feature	TI-Basic	C
Functions	~	✓
On-Calc programming	✓	✗
Get pressed keys	✗	✓
Pixelwise drawing on screen	✗	✓
Timers	✗	✓
RS232 Serial port	✗	✓

1.3 What is required?

Basic C knowledge

- No need for explanation; There are much tutorials out in the internet, just google it

ndless sourcecode

- The sourcecode is required to build ndless

nSpire emulator

- The nspire emulator is recommended to test all your programs directly on your PC

Text editor

- I will use Notepad++ in this Tutorial, but you can use every text editor you want

Some extra programs

- MSYS
- YAGARTO
- 7-Zip
- Tortoise SVN

2 Installation

2.1 MSYS

Just follow all instructions and install it to `C:\msys\1.0`

2.1.1 Adding MSYS to the PATH variable

Open the Start menu and choose “Run...”, type `cmd` and hit enter.

Type the following into the console:

```
set %PATH%=%PATH%;C:\msys\1.0\bin
```

2.2 YAGARTO

Follow the instructions and install it to a destination of your choose.

Make sure “Add YAGARTO to the PATH variable” is checked in the “Choose Components”-Window!

2.3 7-Zip

Follow the instructions and install it to a destination of your choose.

2.3.1 Adding 7-Zip to the PATH variable

Open the Start menu and choose “Run...”, type `cmd` and hit enter.

Type the following into the console:

```
set %PATH%=%PATH%;<7-Zip Installation Folder>
```

Replace *<7-Zip Installation Folder>* with your installation folder, for example `C:\Program Files\7-Zip`.

2.4 Tortoise SVN

Tortoise SVN is not especially required; You can use any SVN client you want, but I will use Tortoise in this Tutorial.

Just follow all instructions and install it to a destination of your choose.

3 Building ndless

3.1 Check out the source code

Create a folder called “ndless” on your C: drive and right-click it. Choose “SVN Checkout...” and enter <https://www.unsads.com/scm/svn/nsptools/Ndless/trunk> in “URL of repository”. You should end up like this:

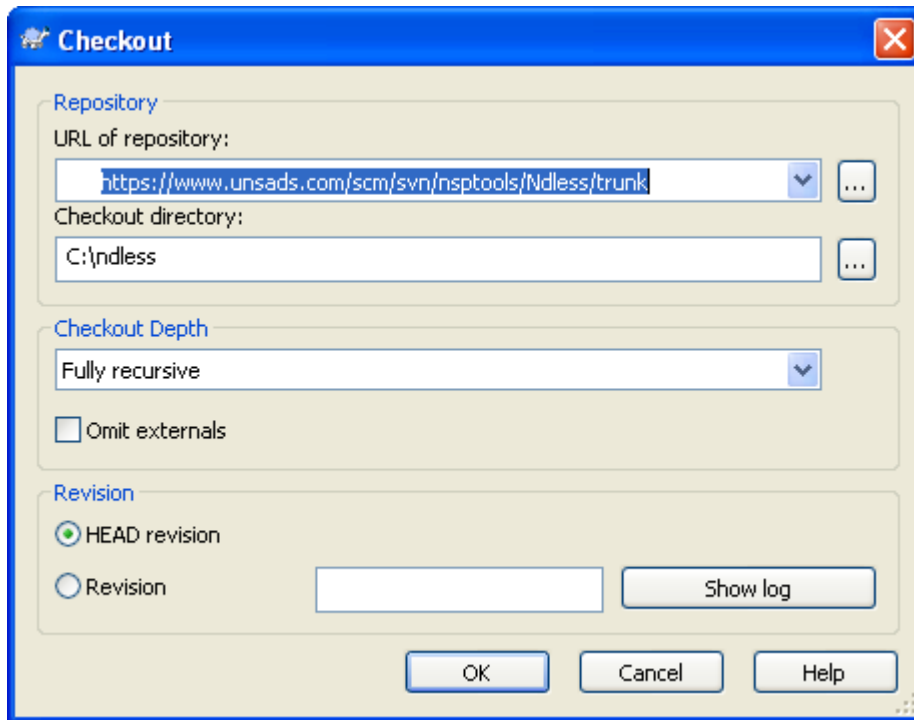


Image 1: SVN Checkout

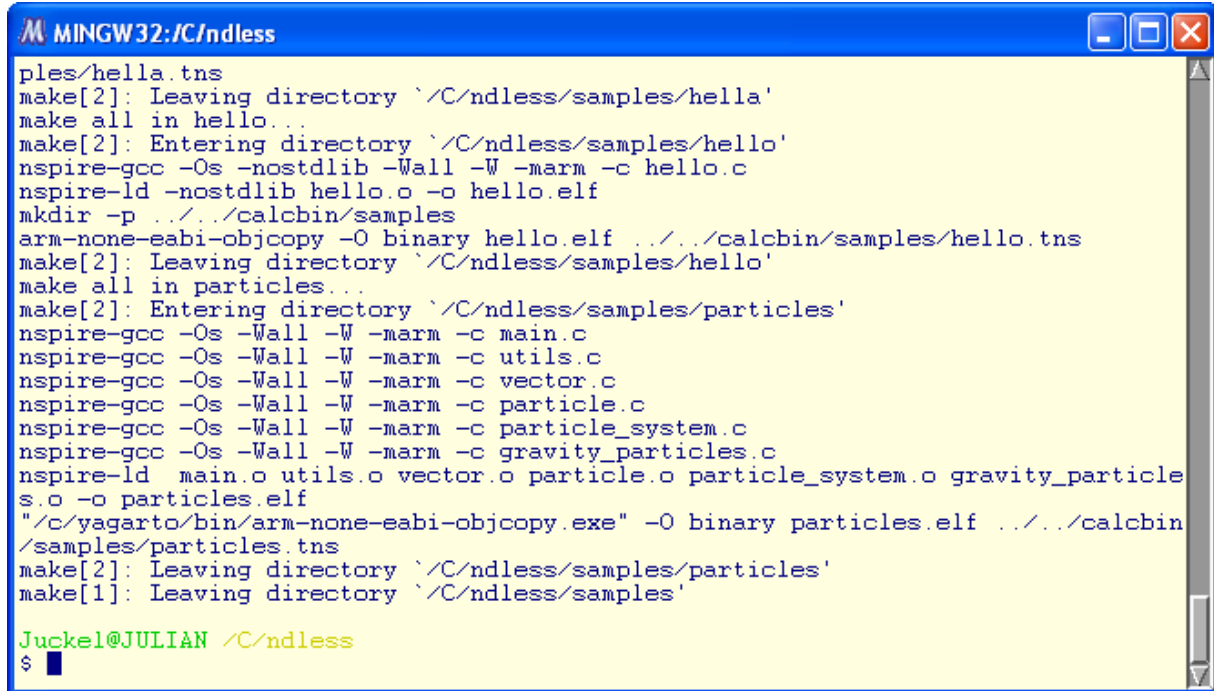
Now just click “OK”. You will be asked for username and password; Use guest for both.

After downloading the source code, you need to add C : \ndless\bin to your PATH variable. Do it like you did before in 2.1.1 and 2.3.1.

3.2 Make!

Open Start→Program Files/All Programs→MinGW→MSYS→msys (rxvt). Type `cd /C/ndless` and press enter. Now type `make`. It may take a while to finish.

If your output looks like this:



```
MINGW32:/C/ndless
ples/hella.tns
make[2]: Leaving directory `/C/ndless/samples/hella'
make all in hello...
make[2]: Entering directory `/C/ndless/samples/hello'
nspire-gcc -Os -nostdlib -Wall -W -marm -c hello.c
nspire-ld -nostdlib hello.o -o hello.elf
mkdir -p ../../calcbin/samples
arm-none-eabi-objcopy -O binary hello.elf ../../calcbin/samples/hello.tns
make[2]: Leaving directory `/C/ndless/samples/hello'
make all in particles...
make[2]: Entering directory `/C/ndless/samples/particles'
nspire-gcc -Os -Wall -W -marm -c main.c
nspire-gcc -Os -Wall -W -marm -c utils.c
nspire-gcc -Os -Wall -W -marm -c vector.c
nspire-gcc -Os -Wall -W -marm -c particle.c
nspire-gcc -Os -Wall -W -marm -c particle_system.c
nspire-gcc -Os -Wall -W -marm -c gravity_particles.c
nspire-ld main.o utils.o vector.o particle.o particle_system.o gravity_particle
s.o -o particles.elf
"/c/yagarto/bin/arm-none-eabi-objcopy.exe" -O binary particles.elf ../../calcbin
/samples/particles.tns
make[2]: Leaving directory `/C/ndless/samples/particles'
make[1]: Leaving directory `/C/ndless/samples'

Juckel@JULIAN /C/ndless
$
```

Image 2: MSYS

You have done everything correctly.

4 Emulating the nSpire

Now that you have built ndless, you should download and set up nspire_emu.

4.1 Things you need

nspire_emu

- The emulator itself; Used to test your C programs.

imgdump

- Used to create raw boot2 images needed by nspire_emu.

OS 2.1 Image file

- The basic OS needed to run on the emulator.

4.2 Setup

4.2.1 Create a raw image file

At first open the downloaded OS file with 7-Zip (Right-click→7-Zip→Open). These files should be included in the image:

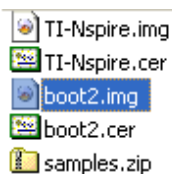


Image 3: OS files

Now extract the selected file (`boot2.img`) to the `imgdump` folder, open a console (Start→Run...→cmd) and type:

```
cd <imgdump_path>
imgdump.exe boot2.img
```

Replace `<imgdump_path>` with the path where your `imgdump` is located. This will produce a file called `boot2.img.raw`. Copy it to your `nspire_emu` folder.

4.2.2 Install the OS

To create a flash image with preinstalled OS, type into the console:

```
cd <nspire_emu_path>
nspire_emu.exe /N /F=flash.bin /PO=<OS_Image_File_name.tno>
```

Replace *<nspire_emu_path>* with the path where *nspire_emu* is located and *<OS_Image_File_name.tno>* with the name of your downloaded OS 2.1 Image file name. Now you are ready to boot your emulated nSpire and install the OS. To make starting the emulator a lot more easier, you should create a batch (*.bat) file in your emulator's directory that contains:

```
nspire_emu.exe /B=boot2.img.raw /F=flash.bin
```

Now you can just double-click this file and the emulator will boot your nSpire. When you start it for the first time, you have to press "I" when you are asked to. After the OS is installed (and you have chosen your language and font size) you need to save your flash to make sure you do not have to install the OS again every time the nSpire boots. To do so, go to File→Save Flash.

4.2.3 Preventing crashes

Every time the calculator tries to go in standby (normally after three minutes of inactivity), the emulator will freeze. To prevent those crashes, you can set the standby time to 30 minutes. Go to Settings & Status→Handheld Setup... and change the Power Standby time. After doing this, save the flash again.

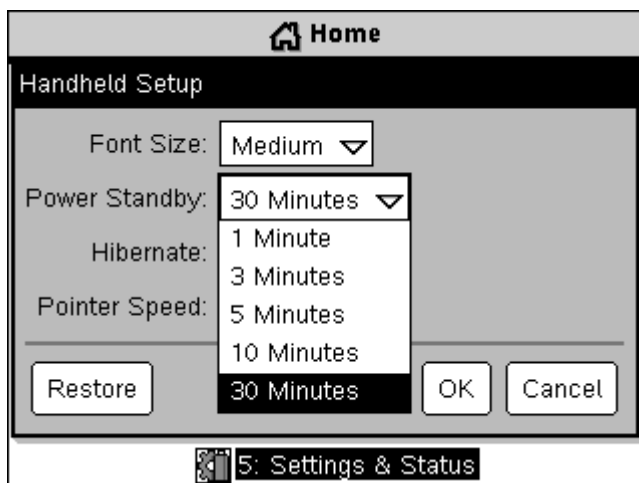


Image 4: Setting the standby time

4.3 Overview

Now you should have two windows. The first, the RS232 console, shows all the stuff the calc sends trough the RS232 interface¹; The second is the calculator itself, including keypad and screen.

¹ I will explain the RS232 interface later in this tutorial.

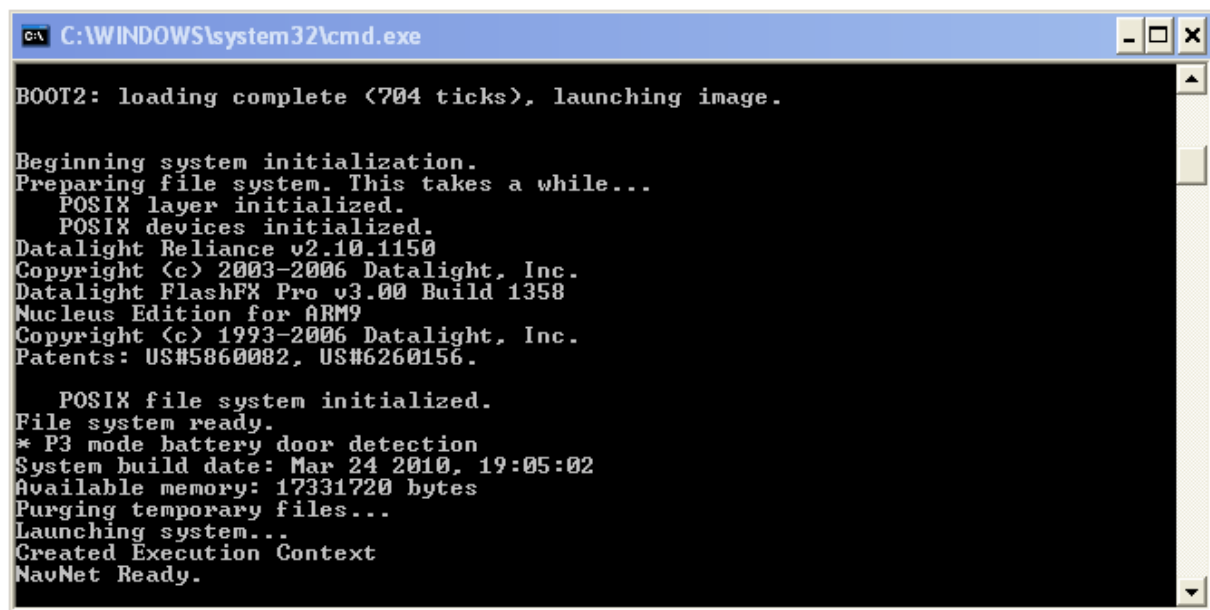


Image 5: RS232 Console

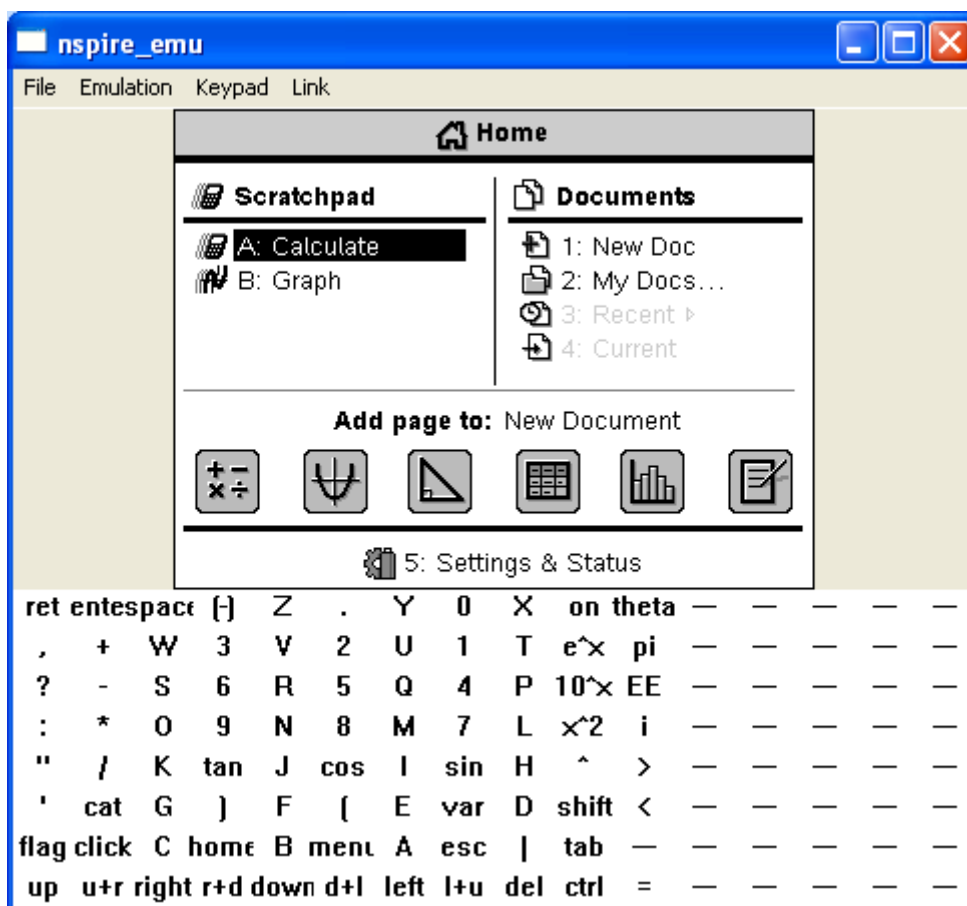


Image 6: Calculator screen

4.4 Installing ndless

To install ndless, you need to send it to your emulated calculator first. Open Link→Connect to “plug the USB cable in”. Your RS232 console should output `usbblink connected`. Now you can send the ndless files by choosing Link→Send Document...; The required files are named

ndless_installer_os-2.1.0.tns and ndless_resources.tns and are located in `<ndless_path>/calcbn`. Replace `<ndless_path>` with the path of your compiled ndless. After both files are sent, save the flash again. Now go to My Documents, select ndless_installer_os-2.1.0 (should be located in Examples) and press Enter. After a successful installation a message will appear:

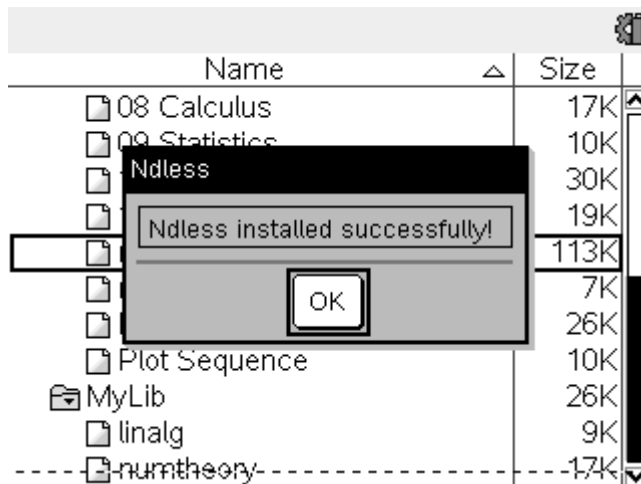


Image 7: ndless installation

If random pixel lines show up and the calculator freezes, the installation failed and you need to try again. If the calculator does not restart itself, you need to reset the CPU (Emulation→Reset CPU). The nSpire will restart then.

4.5 Your first program

Now that you know how to send files to the emulator, you can run your first C program.