

## Desktop

### Introduction

---

There are three examples under this folder.

- 1) Beginner  
A clean and simple example for beginners to start with NSP32, to demonstrate the basic usage of our API.
- 2) ConsoleDemo  
A console program to demonstrate full functionalities of NSP32. Users can operate NSP32 by interactive console commands.
- 3) SpectrumMeter  
A GUI program to visualize the spectrum measured by NSP32.

### API jar Location

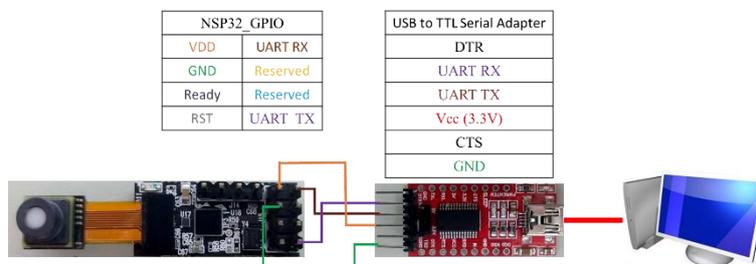
---

Our API jar is located at [/examples/desktop/XXX/libs/NanoLambdaNSP32.jar], where XXX is the name of each example project.

### Hardware Setup

---

If you are going to connect NSP32 through USB port on your desktop, you might need a "USB to TTL Serial Adapter that supports 3.3V".



The following table shows the general pin connections between the NSP32 module and the adapter.

Hardware		NSP32	Adapter
GPIO			
Power	VDD3V3		3V3
	GND		GND
UART	UART TX		UART RX
	UART RX		UART TX

### Software Setup

---

- 1) Runs on

## Desktop

Java SE Development Kit 8 in:

- Windows 7/8/8.1/10
- macOS Mojave 10.14
- Ubuntu 16.04/18.04

### 2) Setup

- i) Install JDK 8 or above to build and run the examples.

For more information on rebuilding and running the examples, please see "Run the Example" section below.

- ii) The examples utilize the following third party libraries (required jar files are already included in the /libs/ folder of each example project).

jSerialComm [<http://fazecast.github.io/jSerialComm/>]

XChart [<https://knowm.org/open-source/xchart/>]

## Run the Example

---

With the "Beginner" example, you need to modify the source code according to the serial port name where your NSP32 is connected to. For your convenience, we have marked these code sections with the title "modify this section to fit your need".

There is a "run.bat" and a "run.sh" file located at the root of each example folder. Windows users can run "run.bat" batch file to rebuild and run each example. Linux or macOS users can run "run.sh" by terminal commands:

```
$ chmod +x run.sh
$ sudo ./run.sh
```

Note: For users who had installed different JDK versions, use the following command to make sure you are running JDK 8 or higher:

```
$ javac -version
```

## Desktop

### 1) Example Snapshot of “Beginner”

```
C:\WINDOWS\system32\cmd.exe
C:\Users\lab321\Downloads\API\nsp32_api_java_android_desktop\examples\desktop\Beginner>mkdir classes
C:\Users\lab321\Downloads\API\nsp32_api_java_android_desktop\examples\desktop\Beginner>javac -cp libs/NanoLambdaNSP32.jar;libs/jSerialComm-2.3.0.jar -d classes src/*.java
C:\Users\lab321\Downloads\API\nsp32_api_java_android_desktop\examples\desktop\Beginner>jar -cvfm Beginner.jar MANIFEST.MF -C classes - libs/*.jar
added manifest
adding: Beginner$1.class(in = 516) (out= 371)(deflated 28%)
adding: Beginner$2.class(in = 866) (out= 543)(deflated 37%)
adding: Beginner$3.class(in = 1637) (out= 858)(deflated 47%)
adding: Beginner$4.class(in = 708) (out= 468)(deflated 33%)
adding: Beginner.class(in = 2060) (out= 1022)(deflated 50%)
adding: libs/jSerialComm-2.3.0.jar(in = 372797) (out= 356068)(deflated 4%)
adding: libs/NanoLambdaNSP32.jar(in = 8801) (out= 7502)(deflated 14%)
C:\Users\lab321\Downloads\API\nsp32_api_java_android_desktop\examples\desktop\Beginner>rmdir /s /q classes
C:\Users\lab321\Downloads\API\nsp32_api_java_android_desktop\examples\desktop\Beginner>java -jar Beginner.jar
*** press ENTER to exit program ***
sensor id = C9-96-6F-46-3F
first element of wavelength = 400
first element of spectrum = 51.38523
```

### 2) Example Snapshot of “ConsoleDemo”

```
C:\WINDOWS\system32\cmd.exe
*****
detected serial ports are:
1) COM25

[NOTE] If the target port doesn't show up, re-plug the cable and press ENTER to try again.

type the index number to connect to NSP32 (0 to exit): 1
*****
1) hello - say hello to NSP32
2) sensorid - get sensor id string
3) wavelength - get wavelength
4) spectrum - start spectrum acquisition and get the result data
5) xyz - start XYZ acquisition and get the result data
6) exit - exit program

type an available command (case sensitive): spectrum
-----
integration time = 32
saturation flag = false
num of points = 121
spectrum =
36.32277
60.08918
87.01769
117.91424
152.80666
188.63972
220.9036
```

## Desktop

### 3) Example Snapshot of “SpectrumMeter”

