



Using devkit dedicated pads file with NXmap

Table of content

1	INTRODUCTION.....	3
2	OPERATIONS.....	4
3	SUPPORT	6

NANOXPLORE CONFIDENTIAL

1 Introduction

New Python files have been added to the NXmap archive. You can find them in:
`/opt/NanoXplore/NXmap/2.9.5/install/share/python` directory.

Each file lists all the available ports on a specific development kit board (devkit) revision. They can be used to ease the pad assignment in a NXmap project.

This short documentation will show the user how to use these files. NXmap is considered already installed and well configured (FlexLM license).

NANOXPLORE CONFIDENTIAL

2 Operations

First, user has to create a project as described in the others documentations. This short documentation is based on 'demo1' project provided with NXmap.

Looking at the '/opt/NanoXplore/NXmap/2.9.5/install/example/vhdl/demo1/script.py' file, user can see that the pads are assigned in a different way.

```
import NX1H35_EK_V1

kit = NX1H35_EK_V1.Kit()

project.addPads({
    'ck_i' : kit.pad('OSC0')
    , 'sw_i[0]' : kit.pad('SW1')
    , 'sw_i[1]' : kit.pad('SW2')
    , 'sw_i[2]' : kit.pad('SW3')
    , 'sw_i[3]' : kit.pad('SW4')
    , 'sw_i[4]' : kit.pad('SW5')
    , 'sw_i[5]' : kit.pad('SW6')
    , 'rst_i' : kit.pad('PB9')
    , 'led_o[0]': kit.pad('LED1')
    , 'led_o[1]': kit.pad('LED2')
    , 'led_o[2]': kit.pad('LED3')
    , 'led_o[3]': kit.pad('LED4')
    , 'led_o[4]': kit.pad('LED5')
    , 'led_o[5]': kit.pad('LED6')
    , 'led_o[6]': kit.pad('LED7')
    , 'led_o[7]': kit.pad('LED8')
})
```

The first two lines may need to be changed according to your devkit revision. You can find it on the bottom left of the board: "DK625V1" for a devkit v1 or "DK625V2" for a devkit v2.

The 'Kit' object created on the second line provides the following methods:

- **pad(name)**: returns the pad definition dictionary associated to the given pad name.
e.g.: "{ 'location': 'IOB12_D09P', 'type': 'LVCMOS_2.5V_2mA' }" for name 'OSC0'
- **location(name)**: returns the location associated to the given pad name.
e.g.: "IOB12_D09P" for 'OSC0'
- **type(name)**: returns the type associated to the given pad name.
e.g.: "LVCMOS_2.5V_2mA" for 'OSC0'
- **interface(name)**: returns the location of the associated interface.
e.g.: "USER_D0" for 'LED9'

Using these methods is very simple as shown in the following examples:

```
import NX1H35_EK_V1

kit = NX1H35_EK_V1.Kit()

project.addPads({
    'ck_i' : { 'location': kit.location('OSC0'), 'type': 'LVCMOS_2.5V_8mA' }
})
```

```
import NX1H35_EK_V1

kit = NX1H35_EK_V1.Kit()

project.addInterfaces({
    'output1': kit.interface('LED9')
    , 'output2': kit.interface('LED10')
})
```

NANOXPLORE CONFIDENTIAL

3 Support

Do not hesitate to contact NanoXplore support team (support@nanoxplore.com) for any problem or question.

NANOXPLORE CONFIDENTIAL