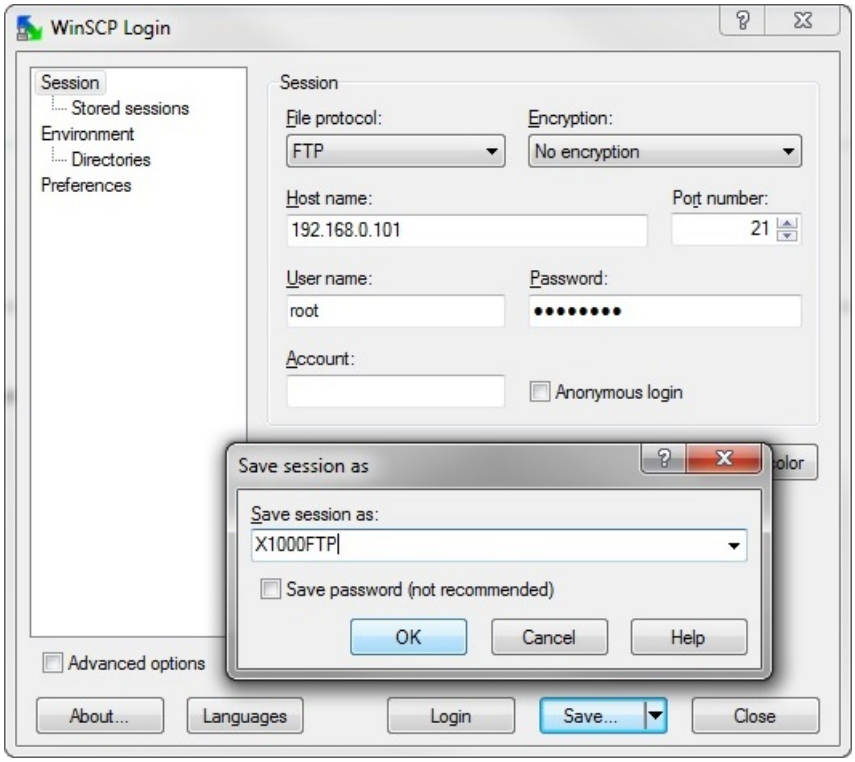
NPE X1000 - Supported Languages

The following document describes programming languages supported by the NPE X1000 platform together with examples of a simple Hello World program in each programming language. Current version has been updated with a description of I/O control using the programming languages and example application codes.

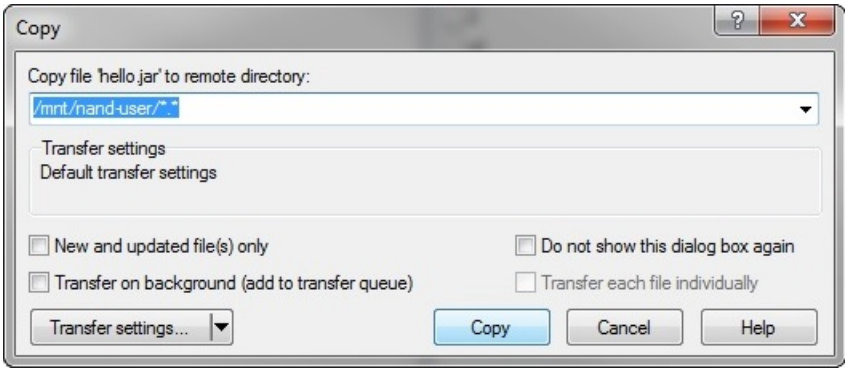
FTP Communication

- Use the *WinSCP* software available at http://winscp.net/download/winscp515.zip. in order to use a ready programme on a computer.
- Click *New* after launching the programme and use the following settings:
- File protocol: *FTP*
- Encryption: *No encryption*
- Host name: *<adres_ip_NPEX1000>* (default 192.168.0.101)
- Port number: *21*
- User name: *root*
- Password: *techbase*
- Click *Save* and choose name for your configuration e.g. X1000FTP.



- Double-click the created configuration in order to establish a connection.
- Choose current directory of the compiled program and the destination folder.

- After dragging the file a window with destination path on the device will pop up. Confirm with *copy*.

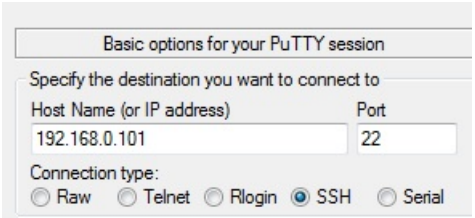


Copy files in binary mode. When copying files directly from Windows it might be necessary to use conversion with the following command:

```
sudo dos2unix file_name
```

SSH Communication

- In order to access the NPE X1000 terminal you can use *Putty* available at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- After launching the device type its IP address into *Host Name(or IP address)*.
- *Connection type* set as *SSH*.



- Confirm configuration with *open*.
- login as: *root*
- password: *techbase*

Java - Hello World

Introduction

Java is a programming language and computing platform released by Sun Microsystems. There are lots of applications and websites that will not work unless you have Java installed

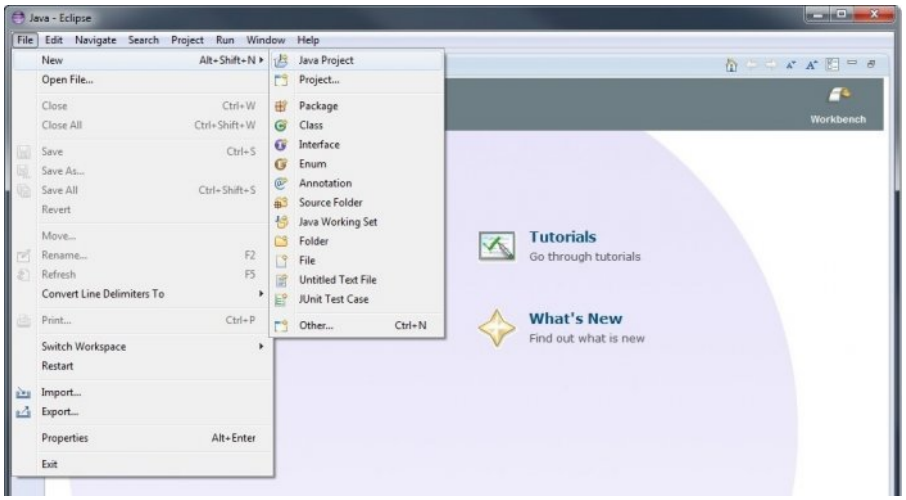
From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Source: http://www.java.com/en/download/faq/whatis_java.xml

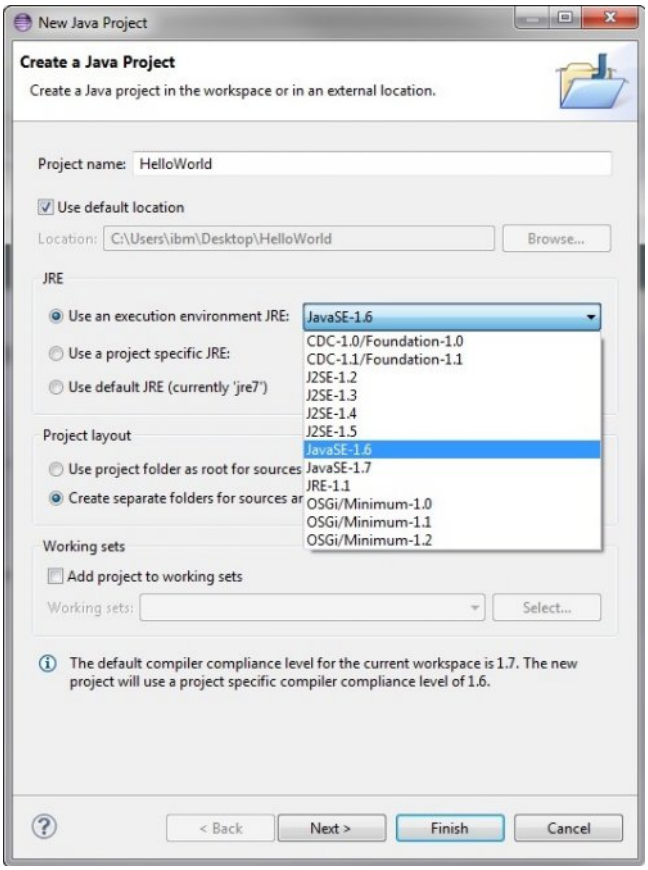
Preparing Language Environment

In order to create applications in Java you can use Eclipse avalable at http://www.eclipse.org/downloads/

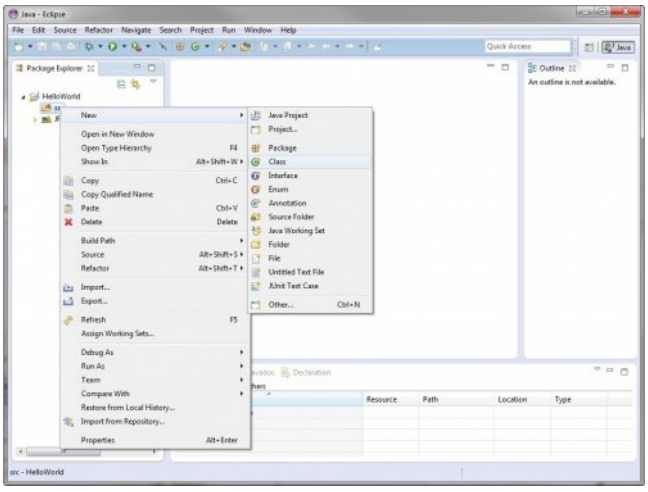
- After launching Eclipse you need to create a new project.



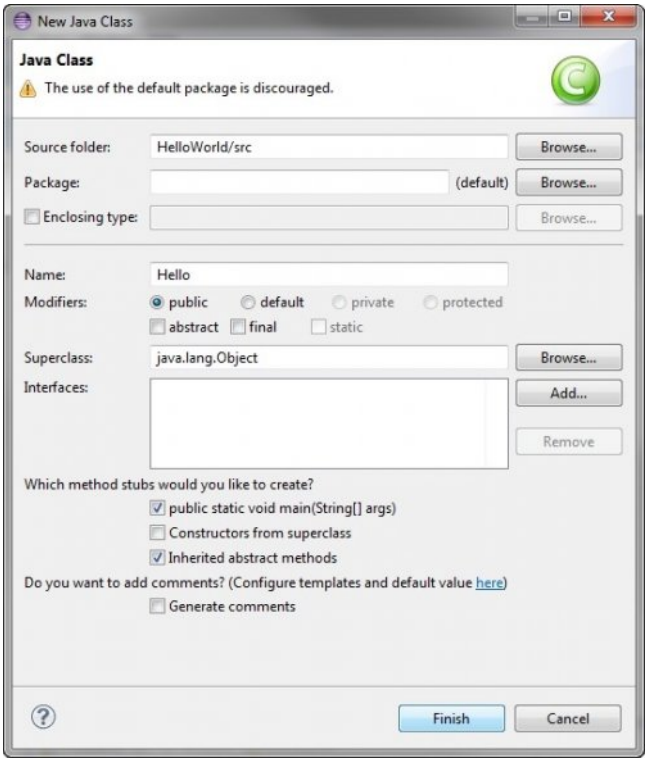
- Name your projects and choose *JRE* environment. In order to run application on NPE X1000 choose *JavaSE-1.6*. Confirm by clicking *Finish*.



- In order to add new file/class right click the *src* directory and choose *new* → *class*.



- Name new class.



- You will find newly created class in the project window of the *src* directory, double-click in order to edit.

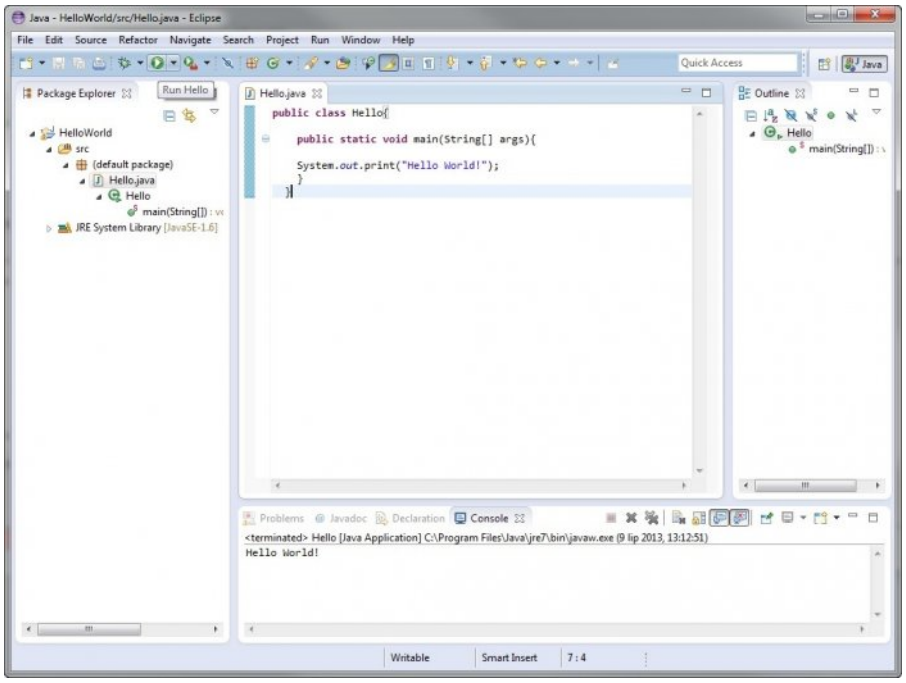
HelloWorld.java

- Example code of an application printing „Hello World!” to the computer screen:

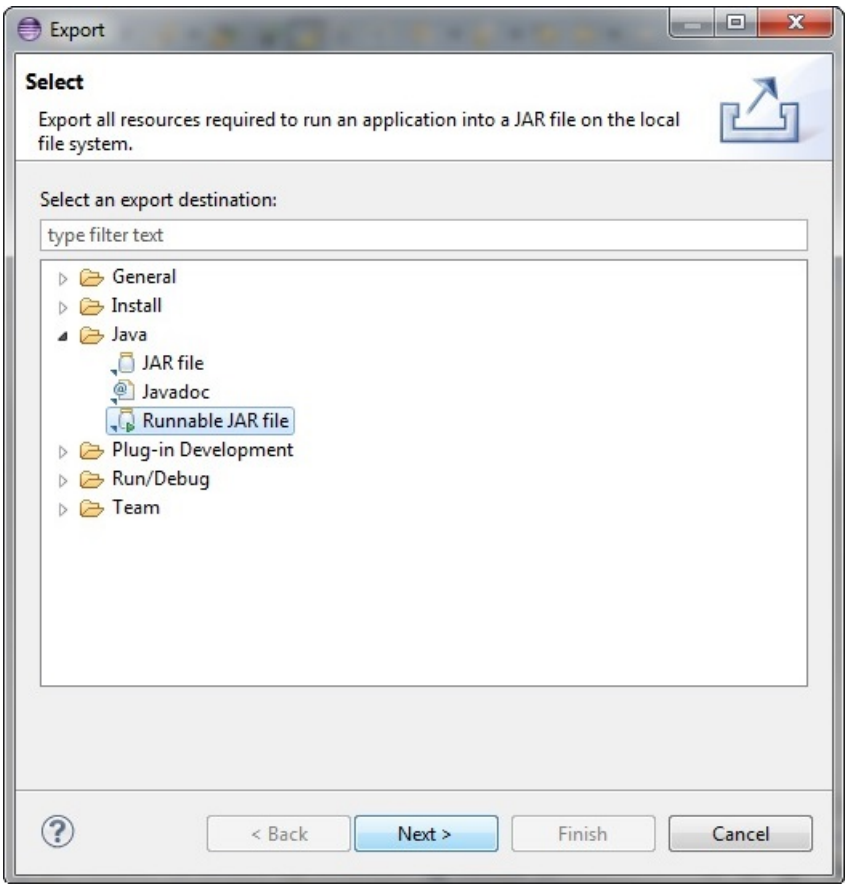
```
public class Hello{

	public static void main(String[] args){

	System.out.print("Hello World!");
	}
}
```

Deployment

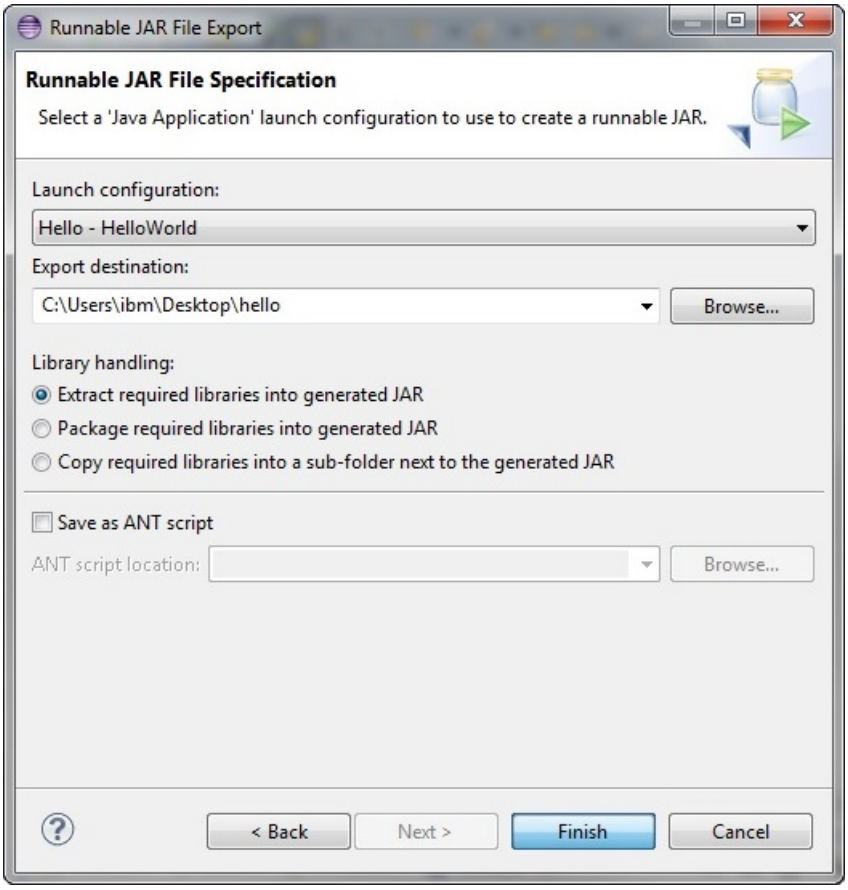
- Choose *Run(ctrl+F11)* from *Run* menu or click the white arrow in a green circle icon in order to test the application.



- Choose *File → Export* and *Java → Runnable JAR file* in order to create executable file.



- When you click *Next* a configuration window with name and location of the output file will appear.



- The received file is ready for launching on the target device.

Verification Procedure

- Get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.jar
```

- run the application with the following command

```
java -jar HelloWorld.jar
```

- Running program:

```
root@npe-x1000:/# cd /mnt/nand-user/
root@npe-x1000:/mnt/nand-user# chmod +x HelloWorld.jar
root@npe-x1000:/mnt/nand-user# java -jar HelloWorld.jar
Hello World!
root@npe-x1000:/mnt/nand-user#
```

Java I/O

You can control input/output using Java. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using Java

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. Read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application Code

```
package npe_demo;

import java.io.*;

public class npe_demo {
        static int value;
        public static void main(String[] args) {

        FileWriter fileWriter = null;

        try{
         File export = new File("/sys/class/gpio/export");
     fileWriter = new FileWriter(export);
     fileWriter.write("72");
     fileWriter.close();
        }catch(IOException ie){};


        try{
        File gpio72dir = new File("/sys/class/gpio/gpio72/direction");
     fileWriter = new FileWriter(gpio72dir);
     fileWriter.write("in");
     fileWriter.close();
        }catch(IOException ie){};

        try{
         File export = new File("/sys/class/gpio/export");
     fileWriter = new FileWriter(export);
     fileWriter.write("106");
     fileWriter.close();
        }catch(IOException ie){};
```

```
        try{
         File gpio106dir = new File("/sys/class/gpio/gpio106/direction");
     fileWriter = new FileWriter(gpio106dir);
     fileWriter.write("out");
     fileWriter.close();
        }catch(IOException ie){};


        try{
        File gpio106val = new File("/sys/class/gpio/gpio106/value");
     fileWriter = new FileWriter(gpio106val);
     fileWriter.write("0");
     fileWriter.close();
        }catch(IOException ie){};

        try{
        File export = new File("/sys/class/gpio/export");
     fileWriter = new FileWriter(export);
     fileWriter.write("128");
     fileWriter.close();
        }catch(IOException ie){};

        try{
         File gpio128dir = new File("/sys/class/gpio/gpio128/direction");
     fileWriter = new FileWriter(gpio128dir);
     fileWriter.write("out");
     fileWriter.close();
        }catch(IOException ie){};

        try{
         File gpio128val = new File("/sys/class/gpio/gpio128/value");
     fileWriter = new FileWriter(gpio128val);
     fileWriter.write("1");
     fileWriter.close();
        }catch(IOException ie){};

        try{
        File gpio128val = new File("/sys/class/gpio/gpio128/value");
     fileWriter = new FileWriter(gpio128val);
     fileWriter.write("0");
     fileWriter.close();
        }catch(IOException ie){};

        try{
        File gpio72val = new File("/sys/class/gpio/gpio72/value");
     FileReader fr = new FileReader(gpio72val);
     value = (char)fr.read();
     fr.close();
        }catch(IOException ie){};

        System.out.println("Value is: "+(char)value);
}
}
```

Verification

Clip presenting application for reading status of DINOUT16:
http://screencast.com/t/Pr977Toee

Python - Hello World

Introduction

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Source: http://python.org/

Preparing Language Environment

- Make sure that current *Python* software package is installed. View list of available packages with the following command:

```
softmgr list
```

- In order to download and install current version use the following command:

```
softmgr update python
```

HelloWorld.py

- In order to create a Python script you just need a text editor.
- Example code of an application printing „Hello World!" to the computer screen:

```
   #!/usr/bin/python
print "Hello, World!";
```

- Save the file with the *.py* extension.
- The received file is ready for launching on the target device.

Verification Procedure

- Get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.py
```

- run the application with the following command

```
python HelloWorld.py
```

- Running program:

Python I/O

You can control input/output using Python. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using Python

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

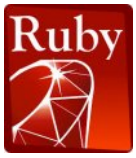
```
#!/usr/bin/python

try:
        file = open('/sys/class/gpio/export', 'w')
        file.write('72')
        file.close()
except: pass


file=open("/sys/class/gpio/gpio72/direction", "w")
file.write("in")
file.close()

try:
        file = open("/sys/class/gpio/export", "w")
        file.write("106")
        file.close()

except:
        pass
file = open("/sys/class/gpio/gpio106/direction", "w")
file.write("out")
file.close()

file = open("/sys/class/gpio/gpio106/value", "w")
file.write("0")
file.close()
try:

        file = open("/sys/class/gpio/export", "w")
        file.write("128")
        file.close()
except:
        pass
file = open("/sys/class/gpio/gpio128/direction", "w")
file.write("out")
file.close()

file = open("/sys/class/gpio/gpio128/value", "w")
file.write("1")
file.close()

file = open("/sys/class/gpio/gpio128/value", "w")
file.write("0")
file.close()

file = open("/sys/class/gpio/gpio72/value", "r")
value = file.read()
file.close()
print("Value is: " + value)
```

Verification

Clip presenting application for reading status of DINOUT16: http://screencast.com/t/6eJhVL4o6cc

Ruby - Hello World

Introduction



Ruby is a language of careful balance. Its creator, Yukihiro “Matz” Matsumoto, blended parts of his favorite languages (Perl, Smalltalk, Eiffel, Ada, and Lisp) to form a new language that balanced functional programming with imperative programming.

Source: http://www.ruby-lang.org/en/about/

Preparing Language Environment

- Make sure that there is a current Ruby package installed on the device. View list of available packages with the following command:

```
softmgr list
```

- Download and install the package using the following command:

```
softmgr update ruby
```

- You will see the following message when the installation process is complete:

```
Checking for missing dependencies for RUBY...
Successfully checked/installed required packages
PACKAGE RUBY: INSTALLED VERSION=0, SERVER VERSION=1307161651, ACTION=update
-------------- Installing RUBY ---------------
Checking if your MAC Address has required privileges.
Access granted
Trying to download ruby_install_1307161651
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 3565k  100 3565k    0     0   335k      0  0:00:10  0:00:10 --:--:--  378k
Downloaded file ruby_install_1307161651 successfully
Creating directory ruby_1307161651
Verifying archive integrity... All good.
Uncompressing ruby..................
Please wait. Installing subpackages...
Package installed successfully
Successfully installed downloaded package (ruby_install_1307161651)
Successfully finished update (It took 148 seconds)
```

HelloWorld.rb

- In order to create a Ruby script you just need a text editor.
- Example code of an application printing „Hello World!” to the computer screen:

```
#!/usr/bin/ruby
puts 'Hello, World!';
```

- Save the file with the *.rb* extension.
- The received file is ready for launching on the target device.

Verification Procedure

- First get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.rb
```

- run the application with the following command

```
ruby HelloWorld.rb
```

- Running program:

Ruby I/O

You can control input/output using Ruby. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using Ruby

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

```
#!/usr/bin/ruby

 File.open("/sys/class/gpio/export", 'w') { |file| file.write("72") }
 File.open("/sys/class/gpio/gpio72/direction", 'w') { |file| file.write("in") }
 File.open("/sys/class/gpio/export", 'w') { |file| file.write("106") }
 File.open("/sys/class/gpio/gpio106/direction", 'w') { |file| file.write("out") }
 File.open("/sys/class/gpio/gpio106/value", 'w') { |file| file.write("0") }
 File.open("/sys/class/gpio/export", 'w') { |file| file.write("128") }
 File.open("/sys/class/gpio/gpio128/direction", 'w') { |file| file.write("out") }
 File.open("/sys/class/gpio/gpio128/value", 'w') { |file| file.write("1") }
 File.open("/sys/class/gpio/gpio128/value", 'w') { |file| file.write("0") }
 File.open("/sys/class/gpio/gpio72/value", 'rb') { |file| $value=file.gets}
 puts "Value is: #$value"
```

Verification

Clip presenting application for reading status of DINOUT16:
http://screencast.com/t/xoX0AFsUkljL

Lua - Hello World

Introduction

Lua is a powerful, fast, lightweight, embeddable scripting language. It combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode for a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping. Lua has a deserved reputation for performance. To claim to be „as fast as Lua" is an aspiration of other scripting languages.

Source: http://www.lua.org/about.html

Preparing Language Environment

- Make sure that there is a current Ruby package installed on the device. View list of available packages with the following command:

```
softmgr list
```

- In order to download and install current version use the following command:

```
softmgr update lua
```

- You will see the following message when the installation process is complete:

```
Software Manager [Version 1306271513] - TechBase Update Utility.
All rights reserved.
Packages Dir: /mnt/nand-user/packages
Very low space on selected dir: /mnt/nand-user Free space:29.6M
Checking for missing dependencies for LUA...
Successfully checked/installed required packages
PACKAGE LUA: INSTALLED VERSION=0, SERVER VERSION=1307161403, ACTION=update
-------------- Installing LUA ---------------
Checking if your MAC Address has required privileges.
Access granted
Trying to download lua_install_1307161403
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  327k  100  327k    0     0   222k      0  0:00:01  0:00:01 --:--:--  227k
Downloaded file lua_install_1307161403 successfully
Creating directory lua_1307161403
Verifying archive integrity... All good.
Uncompressing lua........
Please wait. Installing subpackages...
....
Package installed successfully
Successfully installed downloaded package (lua_install_1307161403)
Successfully finished update (It took 37 seconds)
```

HelloWorld.lua

- In order to create a Lua script you just need a text editor.
- Example code of an application printing „Hello World!" to the computer screen:

```
#!/usr/bin/lua
print ("Hello World!")
```

- Save the file with the *.lua* extension.
- The received file is ready for launching on the target device.

Verification Procedure

- First get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.lua
```

- run the application with the following command

```
lua HelloWorld.lua
```

- Running program:

Lua I/O

You can control input/output using LUA. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using Lua

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

```
#!/usr/bin/lua

file = io.open("/sys/class/gpio/export", "w")
        file:write("72")
        file:close()


file=io.open("/sys/class/gpio/gpio72/direction", "w")
        file:write("in")
        file:close()

file=io.open("/sys/class/gpio/export", "w")
        file:write("106")
        file:close()

file=io.open("/sys/class/gpio/gpio106/direction", "w")
        file:write("out")
        file:close()

file=io.open("/sys/class/gpio/gpio106/value", "w")
        file:write("0")
        file:close()

file=io.open("/sys/class/gpio/export", "w")
        file:write("128")
        file:close()

file=io.open("/sys/class/gpio/gpio128/direction", "w")
        file:write("out")
        file:close()

file=io.open("/sys/class/gpio/gpio128/value", "w")
        file:write("1")
        file:close()

file=io.open("/sys/class/gpio/gpio128/value", "w")
        file:write("0")
        file:close()

file=io.open("/sys/class/gpio/gpio72/value", "r")
value=file:read()
print("Value is: \n" .. value)
file:close()
```

Verification

Clip presenting application for reading status of DINOUT16:
http://screencast.com/t/cHSPbxtjg1iV

Perl - Hello World

Introduction

Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Its major features are that it's easy to use, supports both procedural and object-oriented (OO) programming, has powerful built-in support for text processing, and has one of the world's most impressive collections of third-party modules.

Source: http://perldoc.perl.org/perlintro.html

Preparing Language Environment

- Make sure that there is a current Perl package installed on the device. View list of available packages with the following command:

```
softmgr list
```

- In order to download and install current version use the following command:

```
softmgr update perl
```

- You will see the following message when the installation process is complete:

```
Checking for missing dependencies for PERL...
Successfully checked/installed required packages
PACKAGE PERL: INSTALLED VERSION=1307161524, SERVER VERSION=1307161524, ACTION=no
ne
```

HelloWorld.pl

- In order to create a Perl script you just need a text editor.
- Example code of an application printing „Hello World!” to the computer screen:

```
#!/usr/bin/perl
print "Hello, World!\n";
```

- Save the file with the *.pl* extension.
- The received file is ready for launching on the target device.

Verification Procedure

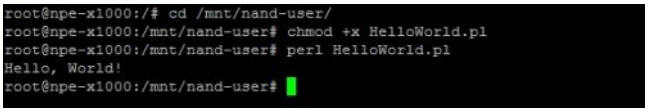
- First get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.pl
```

- run the application with the following command

```
perl HelloWorld.pl
```

- Running program:



Perl I/O

You can control input/output using Perl. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using Perl

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

```
#!/usr/bin/perl

open file, ">", "/sys/class/gpio/export" or die $!;
print file "72";
close file;


open file, ">", "/sys/class/gpio/gpio72/direction" or die $!;
print file "in";
close file;

open file, ">", "/sys/class/gpio/export" or die $!;
print file "106";
close file;

open file, ">", "/sys/class/gpio/gpio106/direction" or die $!;
print file "out";
close file;

open file, ">", "/sys/class/gpio/gpio106/value" or die $!;
print file "0";
close file;

open file, ">", "/sys/class/gpio/export" or die $!;
print file "128";
close file;

open file, ">", "/sys/class/gpio/gpio128/direction" or die $!;
print file "out";
close file;

open file, ">", "/sys/class/gpio/gpio128/value" or die $!;
print file "1";
close file;

open file, ">", "/sys/class/gpio/gpio128/value" or die $!;
print file "0";
close file;

open file, "<", "/sys/class/gpio/gpio72/value" or die $!;
while (my $value = <file>)
{
print "Value is: $value";
}
;
close file;
```

Verification

Clip presenting application for reading status of DINOUT16:
http://screencast.com/t/EWqsGQm38M

PHP - Hello World

Introduction

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

Source: http://pl1.php.net/manual/en/faq.general.php

Preparing Language Environment

- Make sure that there is a current PHP package installed on the device. View list of available packages with the following command:

```
softmgr list
```

- In order to download and install current version use the following command:

```
softmgr update php -b X1000stable
```

- You will see the following message when the installation process is complete:

```
root@npe-x1000:/mnt/nand-user# softmgr update php -b X1000stable
Software Manager [Version 1305121955] - TechBase Update Utility.
All rights reserved.
Packages Dir: /mnt/nand-user/packages
Very low space on selected dir: /mnt/nand-user Free space:47.2M
Checking for missing dependencies for PHP...
Successfully checked/installed required packages
PACKAGE PHP: INSTALLED VERSION=0, SERVER VERSION=1306211235, ACTION=update
-------------- Installing PHP ---------------
Checking if your MAC Address has required privileges.
Access granted
Trying to download php_install_1306211235
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 8636k  100 8636k    0     0   220k      0  0:00:39  0:00:39 --:--:--  210k
Downloaded file php_install_1306211235 successfully
Creating directory php_1306211235
Verifying archive integrity... All good.
Uncompressing php.........
Please wait. Installing subpackages...
......
Package installed successfully
Successfully installed downloaded package (php_install_1306211235)
Successfully finished update (It took 351 seconds)
root@npe-x1000:/mnt/nand-user#
```

HelloWorld.php

- In order to create a PHP script you just need a text editor.
- Example code of an application printing „Hello World!" to the computer screen:

```
<?php
        Echo "Hello, World!\n";
?>
```

- Save the file with the *.php* extension.
- The received file is ready for launching on the target device.

Verification Procedure

- Get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld.php
```

- run the application with the following command

```
php HelloWorld.php
```

- Running program:

PHP I/O

You can control input/output using PHP. Find detailed description of access to I/O in the following chapter:Dostęp do GPIO.

I/O Control Using PHP

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Add *gpio* address to the */sys/class/gpio/export* file
2. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
3. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
4. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
5. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
6. Add *gpio* address for flip clock (*gpio128*)
7. Set *gpio128* direction to *out*
8. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
9. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

```
 <?php
$file = fopen('/sys/class/gpio/export', 'w');
                fwrite($file, '72');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio72/direction', 'w');
                fwrite($file, 'in');
                fclose($file);

$file = fopen('/sys/class/gpio/export', 'w');
                fwrite($file, '106');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio106/direction', 'w');
                fwrite($file, 'out');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio106/value', 'w');
                fwrite($file, '0');
                fclose($file);

$file = fopen('/sys/class/gpio/export', 'w');
                fwrite($file, '128');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio128/direction', 'w');
                fwrite($file, 'out');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio128/value', 'w');
                fwrite($file, '1');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio128/value', 'w');
                fwrite($file, '0');
                fclose($file);

$file = fopen('/sys/class/gpio/gpio72/value', 'r');
                $value = fread($file, 1);
                fclose($file);

                Echo "Value is: $value\n"
?>
```

Verification

Clip presenting application for reading status of DINOUT16:
http://screencast.com/t/GA67VNKWT

ANSI C - Hello World

Introduction

C is a general-purpose programming language. Its design provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, most notably system software like the Unix computer operating system.

Source: http://en.wikipedia.org/wiki/C_%28programming_language%29

Preparing Language Environment

- Make sure that the system used for cross compilation is 64 bit. You can do that using the following command:

```
uname -a
```

- The result of the operation is:

```
karol@karol-VirtualBox:~$ uname -a
Linux karol-VirtualBox 3.8.0-19-generic #29-Ubuntu SMP Wed Apr 17 1
8:16:28 UTC 2013 x86_64 x86_64 x86_64 GNU/Linux
karol@karol-VirtualBox:~$
```

- Next, choose a proper toolchain. Download *angstrom-eglibc-x86_64-armv7a-vfp-neon-v2012.12-toolchain.gz* from http://www.angstrom-distribution.org/toolchains/
- Unzip the archive using the following command:

```
gunzip angstrom-eglibc-x86_64-armv7a-vfp-neon-v2012.12-toolchain.gz
```

- If all goes well the following file appears *angstrom-eglibc-x86_64-armv7a-vfp-neon-v2012.12-toolchain*.
- Grant permission with the following command:

```
sudo chmod +x angstrom-eglibc-x86_64-armv7a-vfp-neon-v2012.12-toolchain
```

- Run with the following command:

```
./angstrom-eglibc-x86_64-armv7a-vfp-neon-v2012.12-toolchain
```

- A question about toolchain target location will appear. Press *enter* to confirm default location: */usr/local/oecore-x86_64*.
- Press *enter* or type *y* and press *enter* in order to confirm the installation.

```
Enter target directory for SDK (default: /usr/local/oecore-x86_64):
You are about to install the SDK to "/usr/local/oecore-x86_64". Proceed[Y/n]?
```

- You will see the following message when the installation process is complete:

```
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

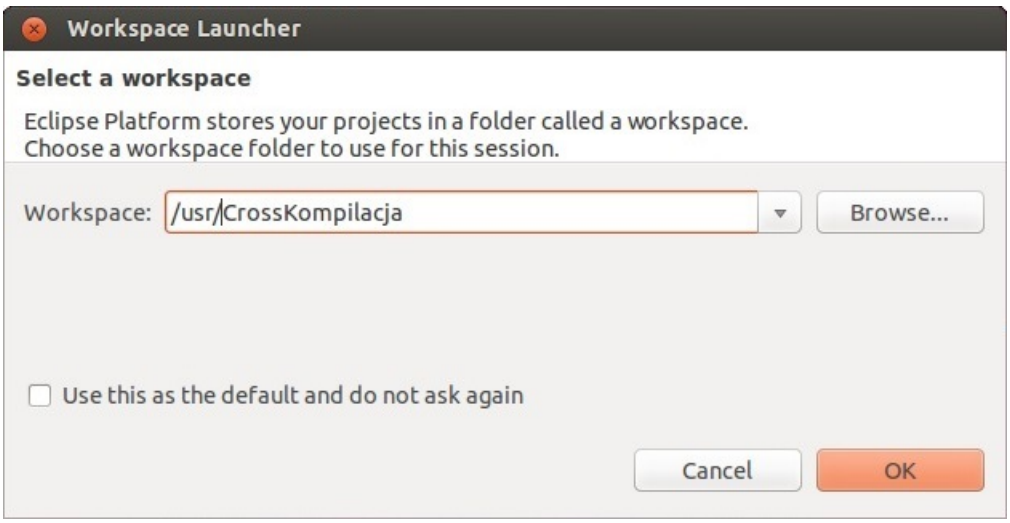
- Next, download and install Eclipse. Use the following command:

```
sudo apt-get install eclipse-cdt
```

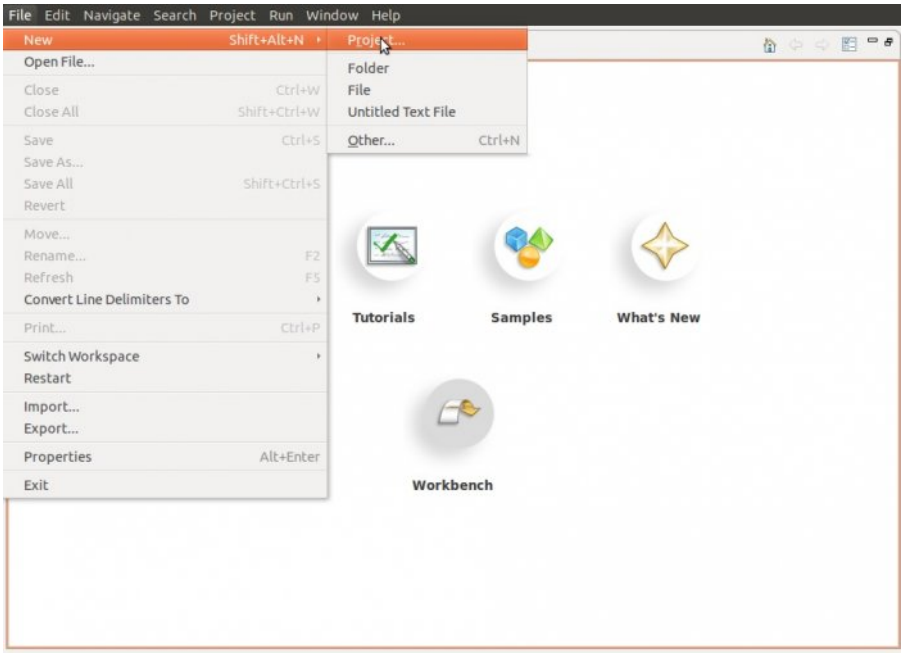
- Launch the software with the following command:

```
eclipse
```

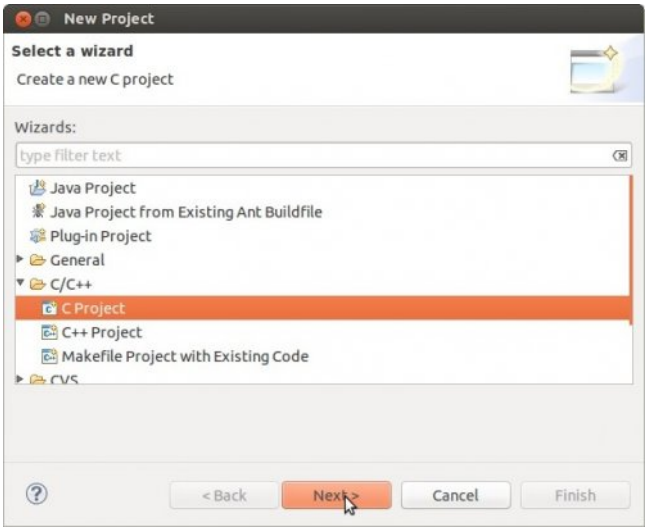
- After launching Eclipse a window with path to your project directory.



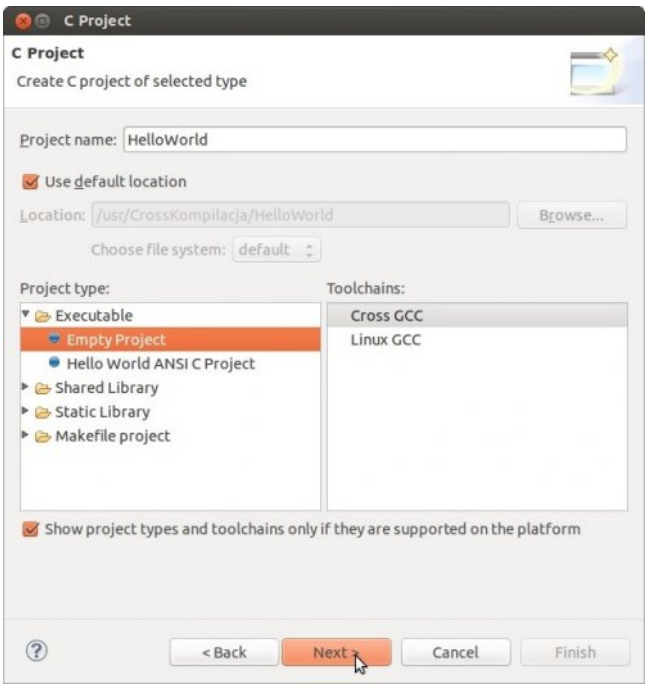
- Choose *File→New→Project*



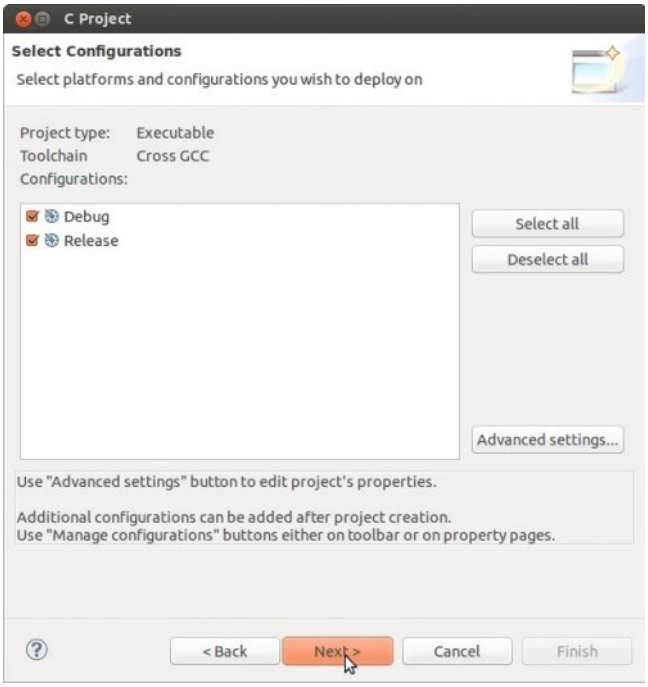
- Next, choose *C/C++* → *C Project*



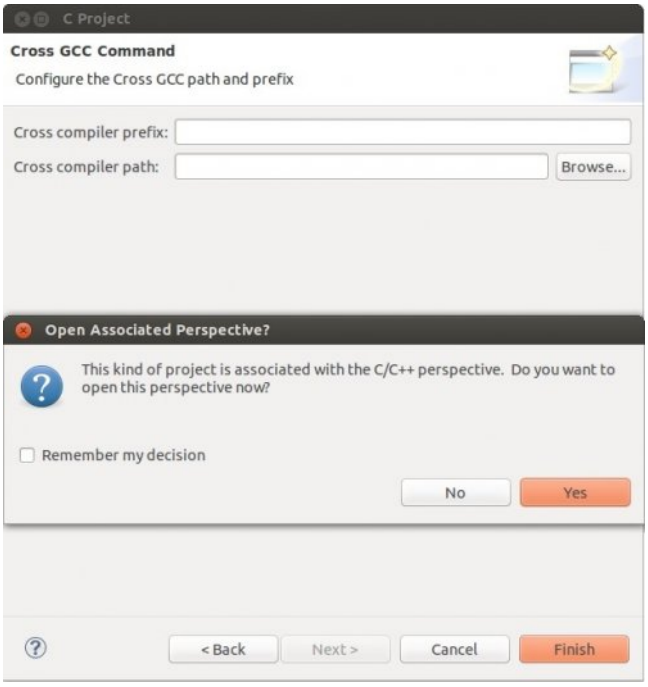
- Name your project e.g. *HelloWorld*, choose *Empty Project*, *CrossGCC* and click *next*.



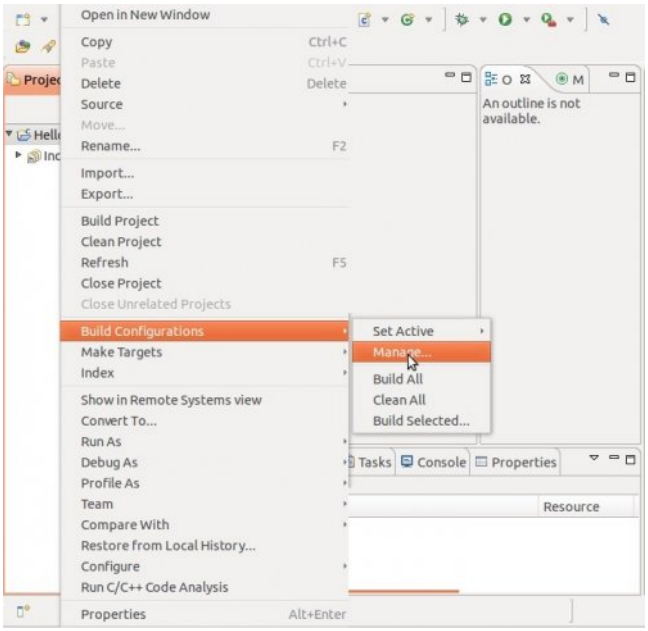
- Click *Next* in the new window.



- Click *Finish* and *Yes*.

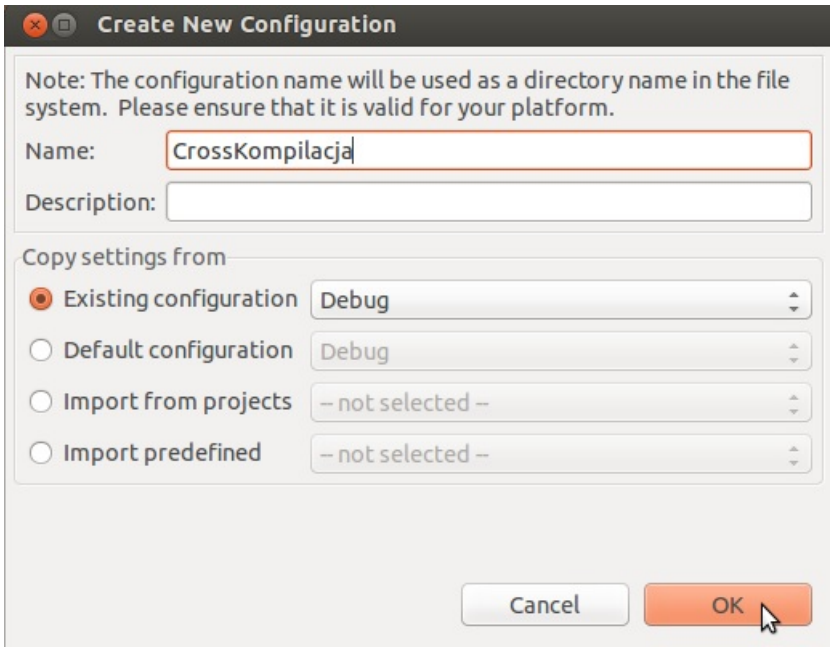


- In *Project Manager* right click *HelloWorld* and choose *Build Configurations → Manage*.

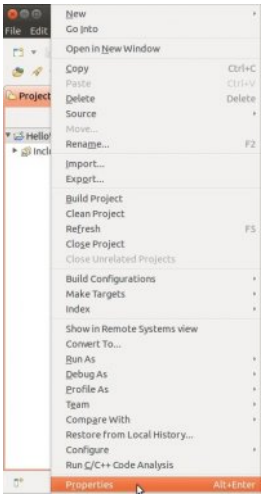


- Click *New* in the new window.

- Type in a name e.g. *CrossKompilacja*, choose *Existing configuration Debug* and accept with *OK*.

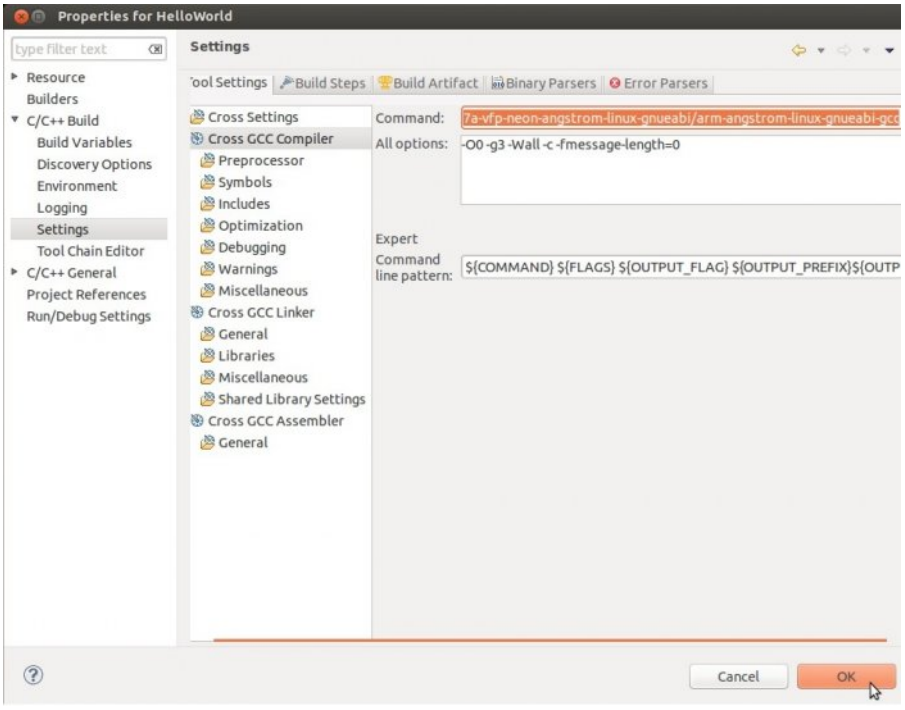


- Select *Project Manager* again, right click *HelloWorld* and choose *Properties*.



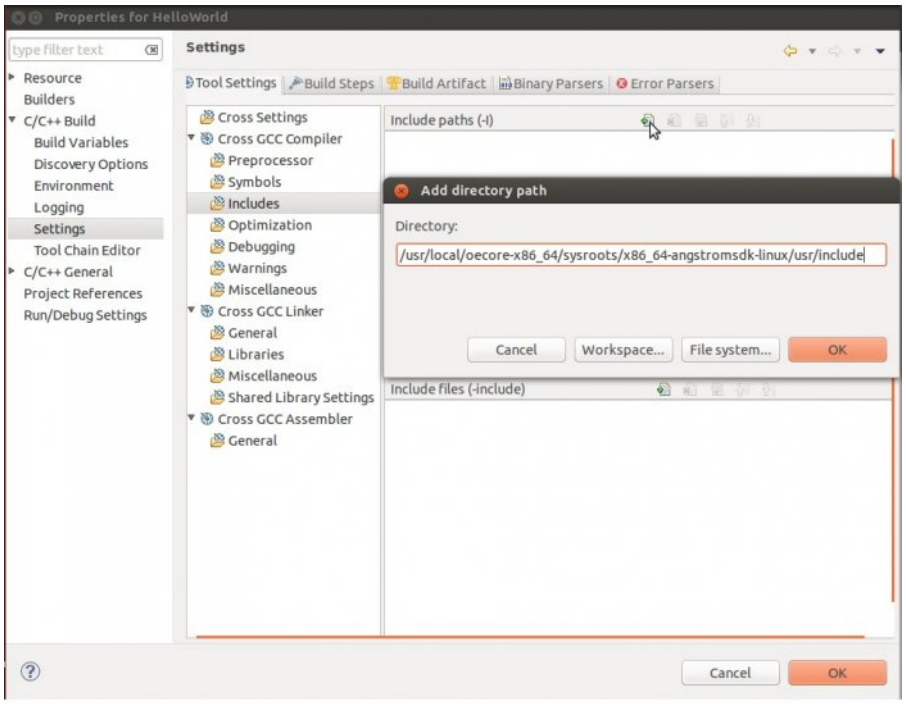
- In *C/C++ Build → Setting* choose *Cross GCC Compiler* and enter access path to our compiler in *Command*:

```
/usr/local/oecore-x86_64/sysroots/x86_64-angstromsdk-linux/usr/armv7a-vfp-neon-anstrom-linux
-gnueabi/arm-angstrom-linux-gnueabi-gcc
```



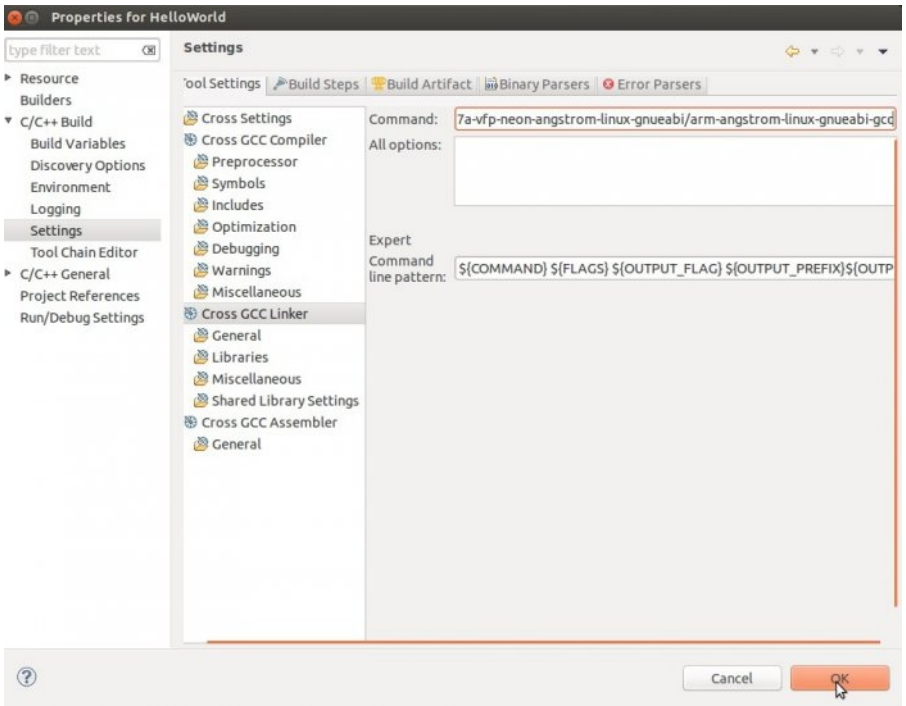
- In *Cross GCC Compiler* choose *Includes*, click *Add* and enter path:

```
/usr/local/oecore-x86_64/sysroots/x86_64-angstromsdk-linux/usr/include/
```



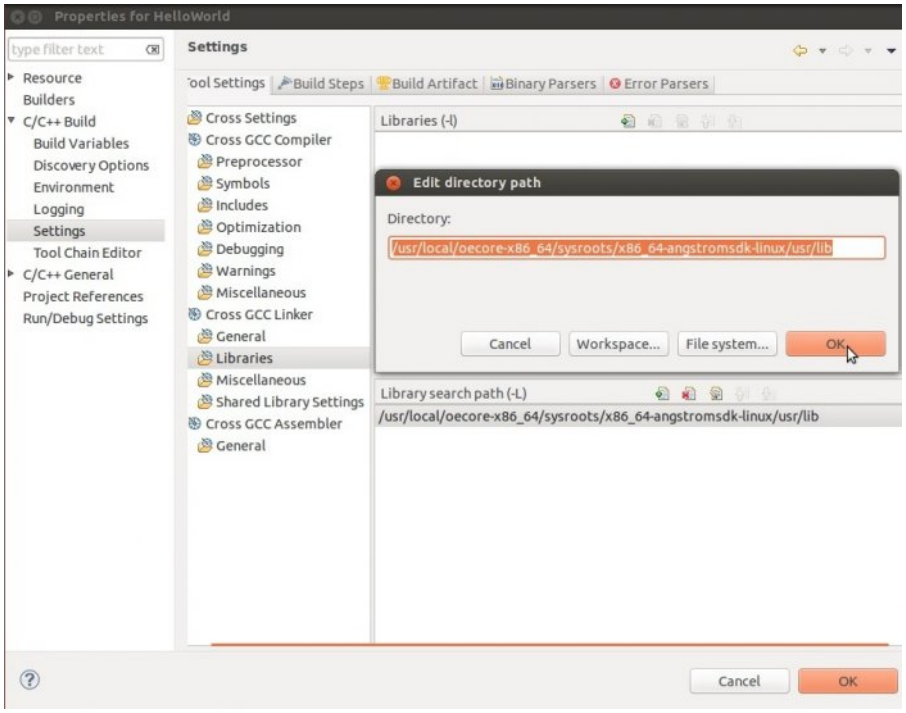
- In *Cross GCC Linker*, *Command* type in the following path:

```
/usr/local/oecore-x86_64/sysroots/x86_64-angstromsdk-linux/usr/armv7a-vfp-neon-anstrom-linux
-gnueabi/arm-angstrom-linux-gnueabi-gcc
```



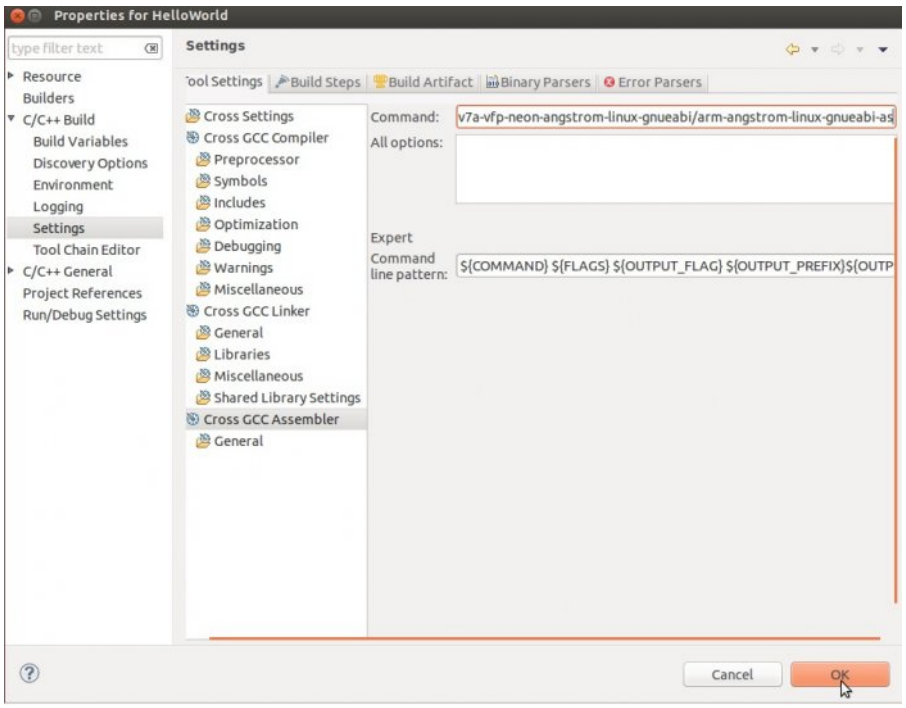
- In *Cross GCC Linker → Libraries* click *Add* in the window under *Library search path* and enter the following path:

```
/usr/local/oecore-x86_64/sysroots/x86_64-angstromsdk-linux/usr/lib/
```

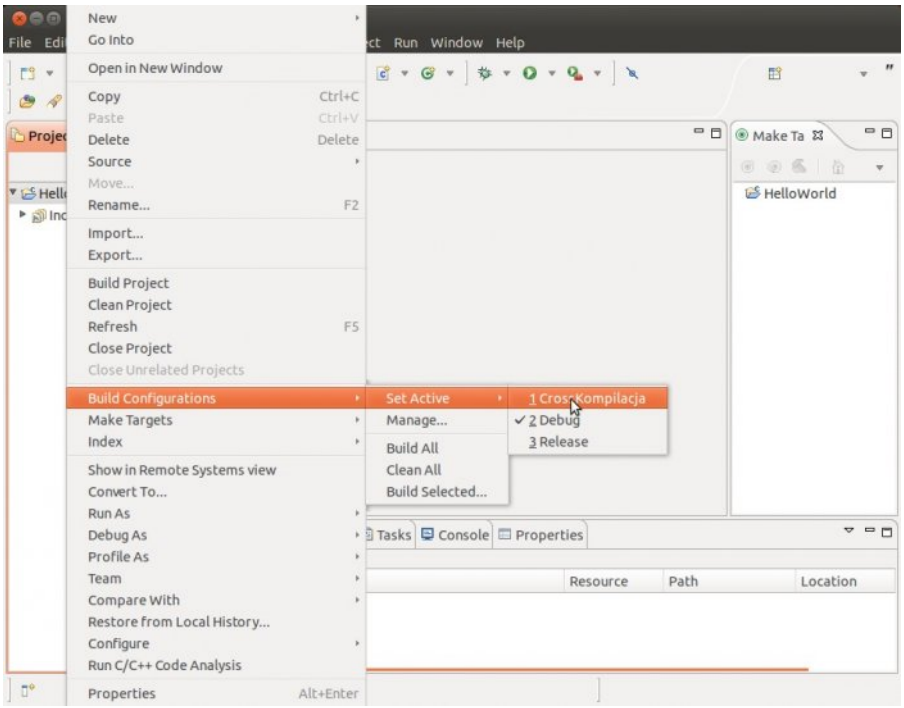


- Set the following path in *Cross GCC Assembler → Command*:

```
/usr/local/oecore-x86_64/sysroots/x86_64-angstromsdk-linux/usr/armv7a-vfp-neon-anstrom-linux
-gnueabi/arm-angstrom-linux-gnueabi-as
```

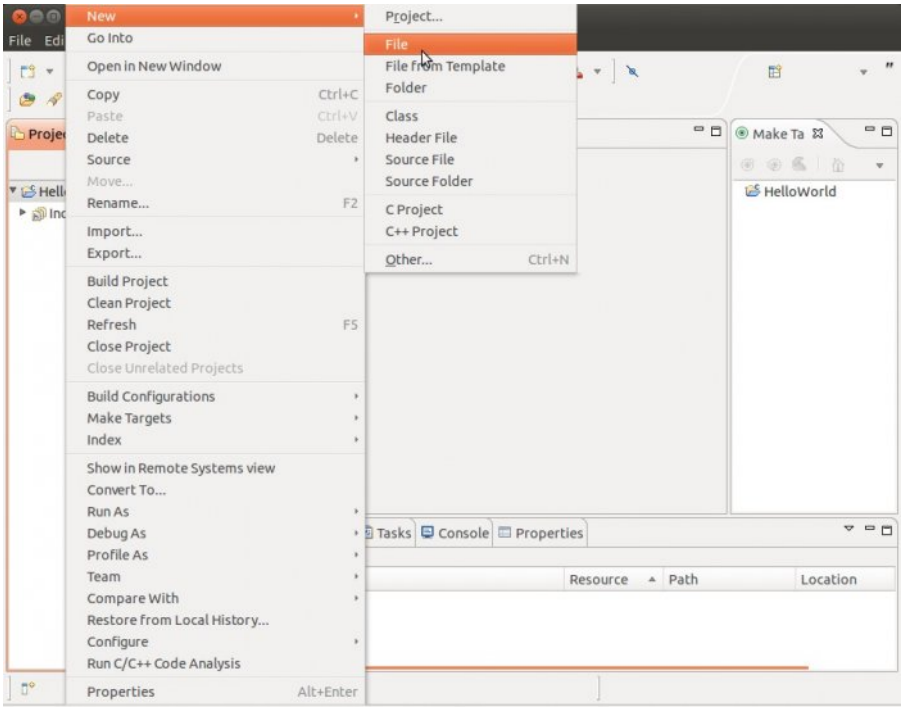


- Press *OK* to confirm.
- Right click the *HelloWorld* in *Project Manager* and choose *Build Configurations → Set Active → CrossKompilacja*.

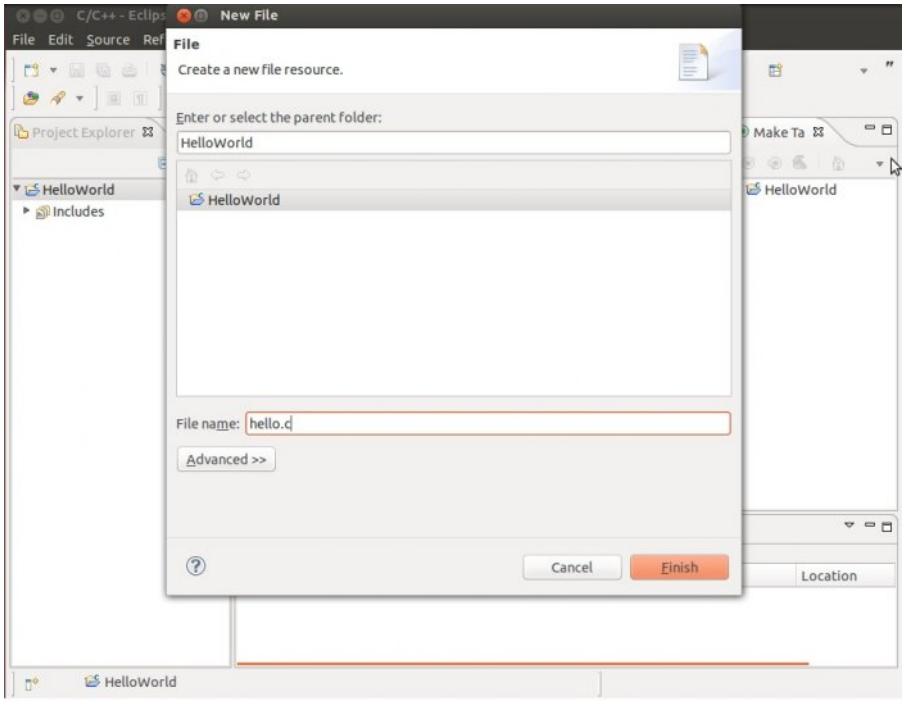


HelloWorld.c

- In order to add a new file in Eclipse you need to right click *HelloWorld* in *Project Manager* and choose *New → File*.

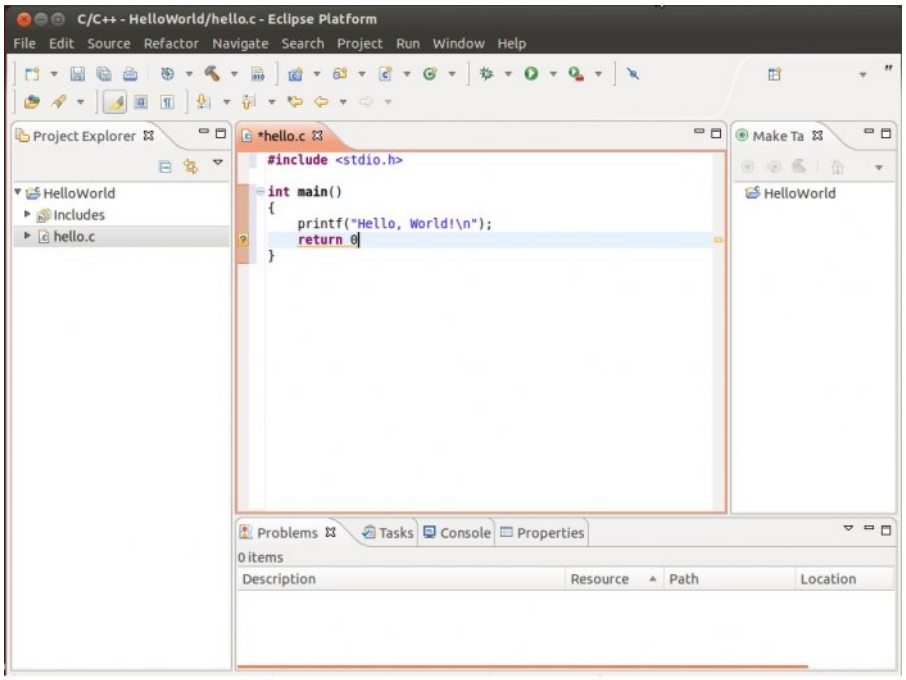


- Name the file e.g. hello.c



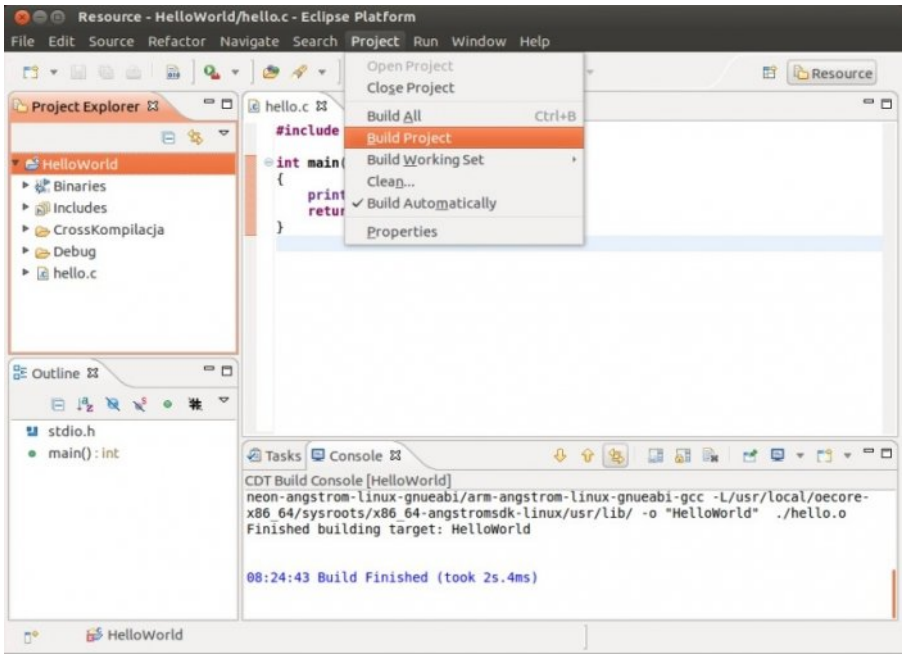
- Example code of an application printing „Hello World!" to the computer screen:

```
   #include <stdio.h>
int main (void)
{
   printf("Hello World!");
   return 0;
}
```

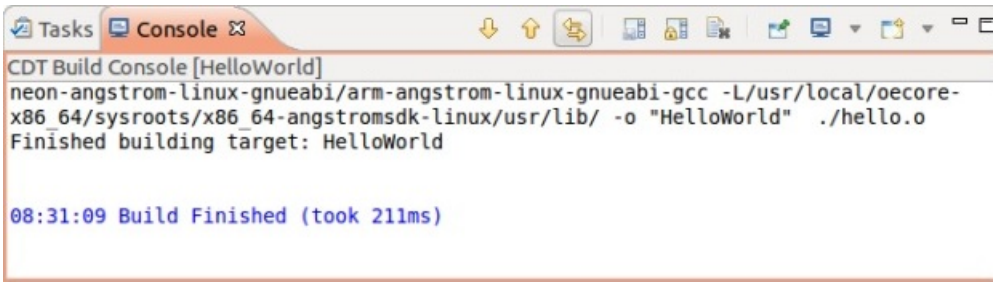


Deployment

- In order to compile the program and receive executable output file you need to click *Project → Build Project*.



- If all goes well the following message appears *Build Finished*



- The HelloWorld file is ready for upload to NPE X1000.

Verification Procedure

- Get access to NPE X1000 terminal
- Find the location of your application and grant permission with the following command:

```
sudo chmod +x HelloWorld
```

- Run the application with the following command

```
./HelloWorld
```

- Running program:

ANSI C I/O

In order to run application in C on NPE, you need to cross compile.
This process is described in details in the following chapter: ansi_c_-_hello_world.
Access to NPE X1000 I/Os is described in Dostęp do GPIO.

I/O Control Using C

Below you will find code of an example application for reading GPIO values. The presented value is read from digital output (*DINOUT16*) which corresponds with *gpio72* address.
In order to set the GPIO value you need to perform the following actions:

1. Declare *FILE* variable
2. Add *gpio* address to the */sys/class/gpio/export* file
3. Set the *gpio* direction to *in* by writing *in* in the */sys/class/gpio/gpio72/direction* file
4. Add *gpio106* address defining the direction to the */sys/class/gpio/export* file
5. Set the *gpio106* direction to *out* by writing *out* in the */sys/class/gpio/gpio106/direction* file
6. Set *0* value of the above *gpio* (*1* for *out*, *0* for *in*)
7. Add *gpio* address for flip clock (*gpio128*)
8. Set *gpio128* direction to *out*
9. Latch value on falling edge of flip flop clock, by writing *1* and *0* in the */sys/class/gpio/gpio128/value* file
10. read *DINOUT16* value, by reading the */sys/class/gpio/gpio72/value* file contents

Application code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE     *file_gpio;
int      value;

int main()
{

file_gpio=fopen("/sys/class/gpio/export", "w");
fprintf(file_gpio, "72");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio72/direction", "w");
fprintf(file_gpio, "in");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/export", "w");
fprintf(file_gpio, "106");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio106/direction", "w");
fprintf(file_gpio, "out");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio106/value", "w");
fprintf(file_gpio, "0");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/export", "w");
fprintf(file_gpio, "128");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio128/direction", "w");
fprintf(file_gpio, "out");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio128/value", "w");
fprintf(file_gpio, "1");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio128/value", "w");
fprintf(file_gpio, "0");
fclose(file_gpio);

file_gpio=fopen("/sys/class/gpio/gpio72/value", "r");
fscanf(file_gpio, "%d", &value);
fclose(file_gpio);
fprintf(stdout, "Value is: %d\n", value);
return 0;
}
```

Verification

Clip presenting application for reading status of DINOUT16: http://screencast.com/t/q30rgGlRsvWx