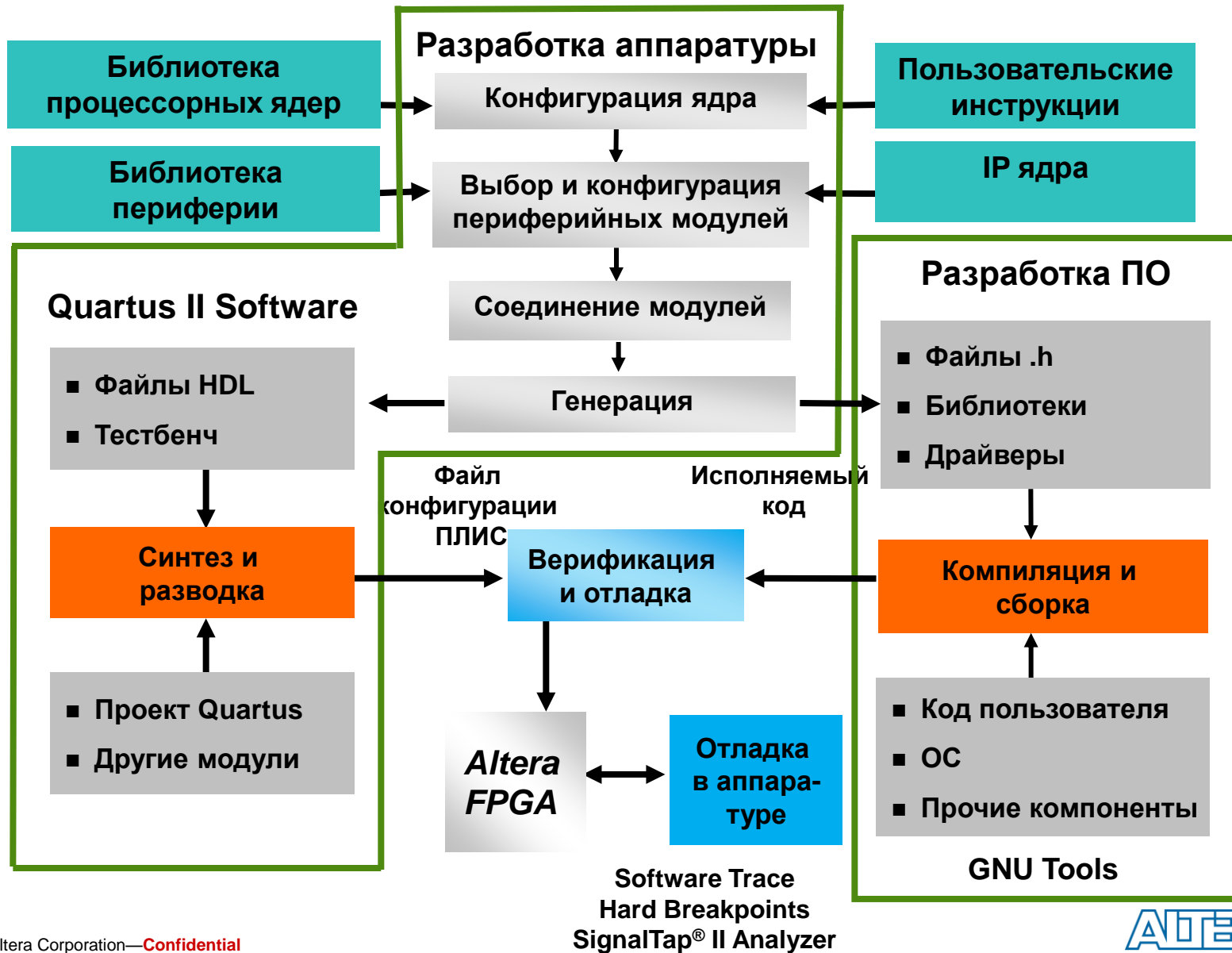


# Разработка программного обеспечения для процессора Nios II

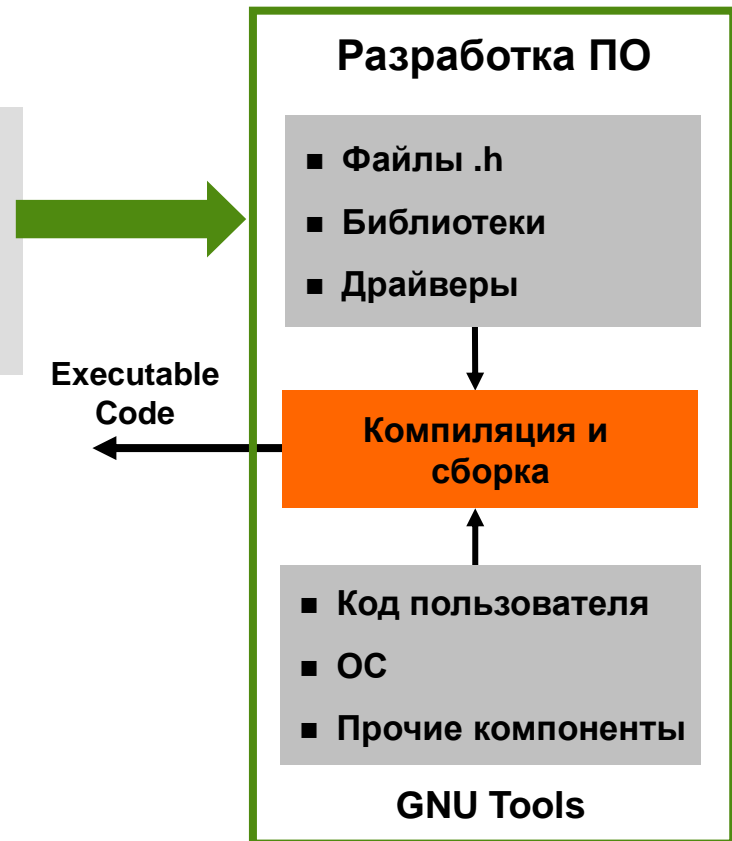
# Маршрут проектирования для Nios II



# Процесс проектирования ПО для Nios II

## ■ Необходимо для разработки ПО

- Файл .sopcinfo
  - Описывает аппаратную платформу для средств разработки ПО
- Файл .sof
  - Используется для конфигурации ПЛИС



# Nios II Embedded Design Suite (EDS)

- Используется для разработки ПО для процессора Nios II
- Основные компоненты
  - Nios II Software Build Tools (SBT)
  - Nios II GCC 4 Tool-chain

# Nios II Software Build Tools

## ■ Что такое Nios II Software Build Tools?

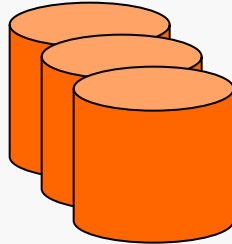
- Набор команд, утилит и скриптов
- Позволяет управлять параметрами сборки для приложений, пакетов поддержки платы (BSP) и библиотек
- `nios2-bsp`
- `nios2-app-generate-makefile`

## ■ Что такое Nios II SBT для Eclipse™?

- Графический интерфейс для разработки и отладки ПО
- Реализован как набор плагинов для среды Eclipse
- Использует команды и утилиты Nios II SBT

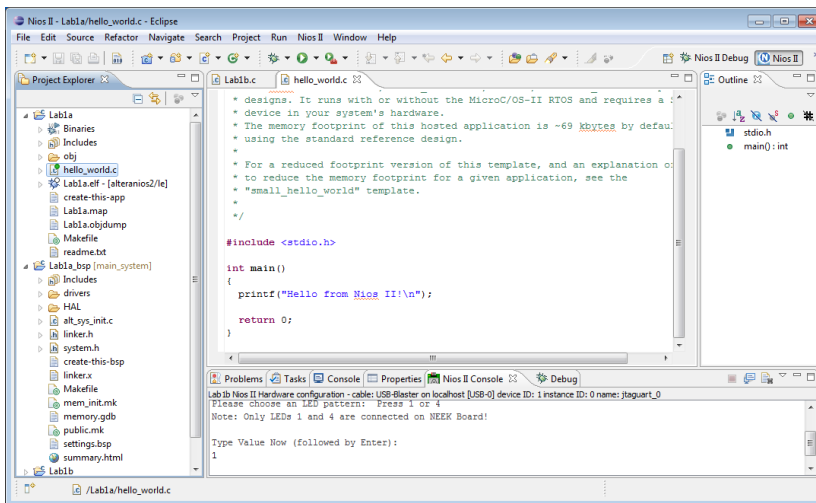
# Nios II Software Build Tools

# Nios II Software Build Tools

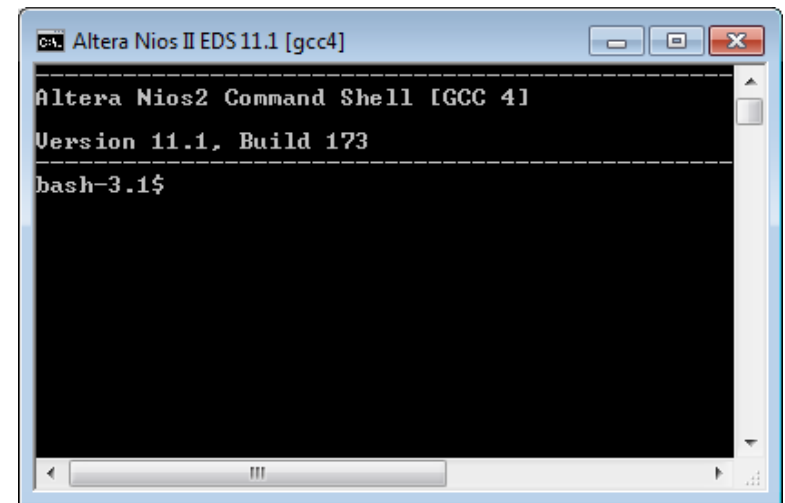


- Application make
- BSP make
- Library make
- Настройка проекта ПО

# Nios II Software Build Tools для Eclipse



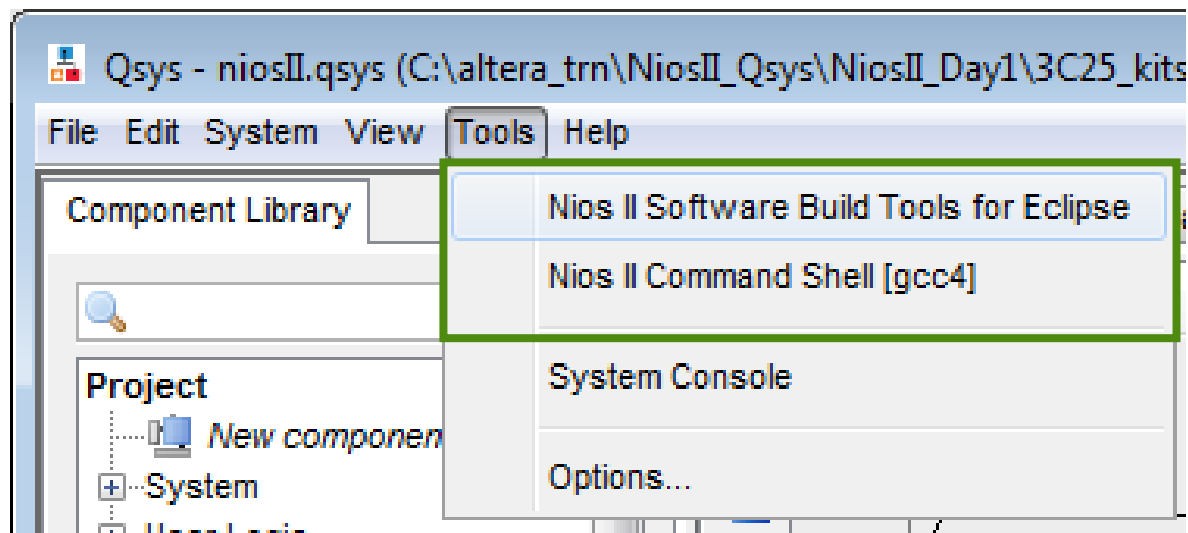
## Nios II Software Build Tools для Nios II Command Shell



**Взаимно совместимы , так как используют одни средства сборки и вспомогательные утилиты**

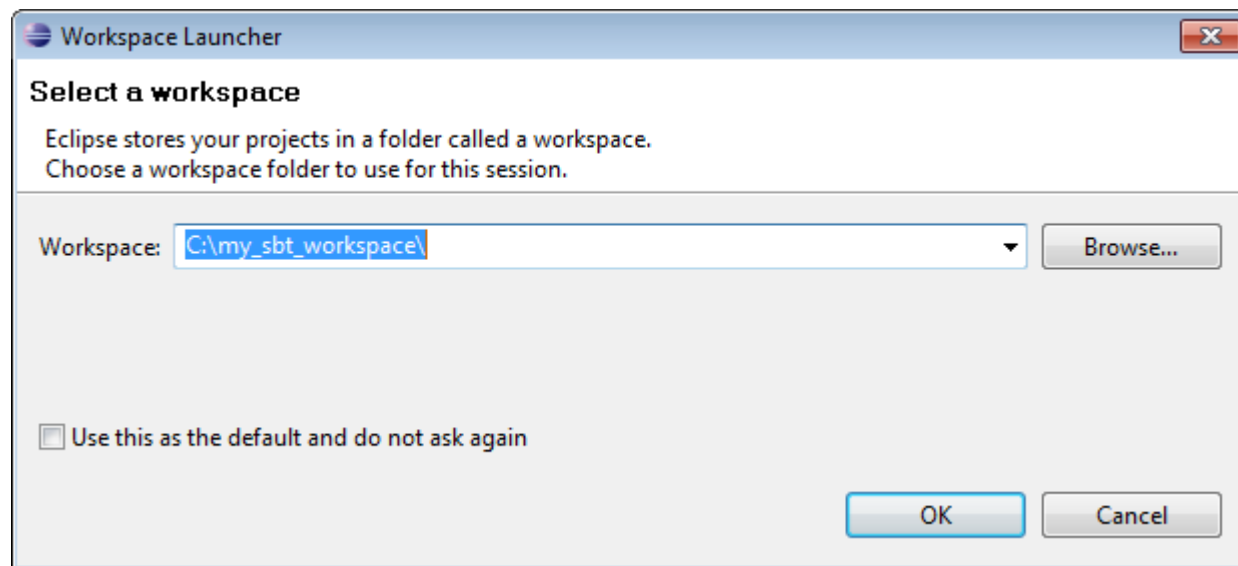
# Запуск Nios II SBT

- Из меню Start в Windows
- Из меню Tools в Qsys

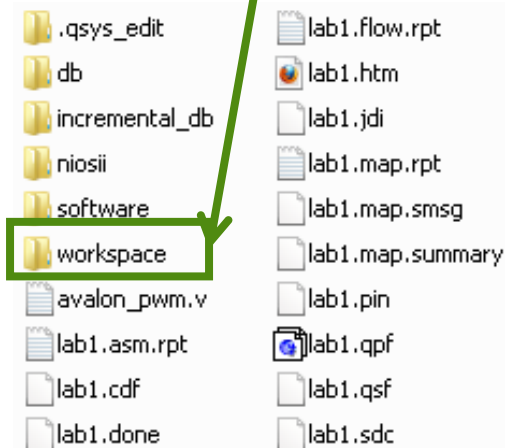


# Workspace Launcher

- Позволяет выбрать рабочее пространство для проектов ПО
- Рабочее пространство хранит информацию о проектах
- Обычно связано с проектом аппаратуры системы на кристалле

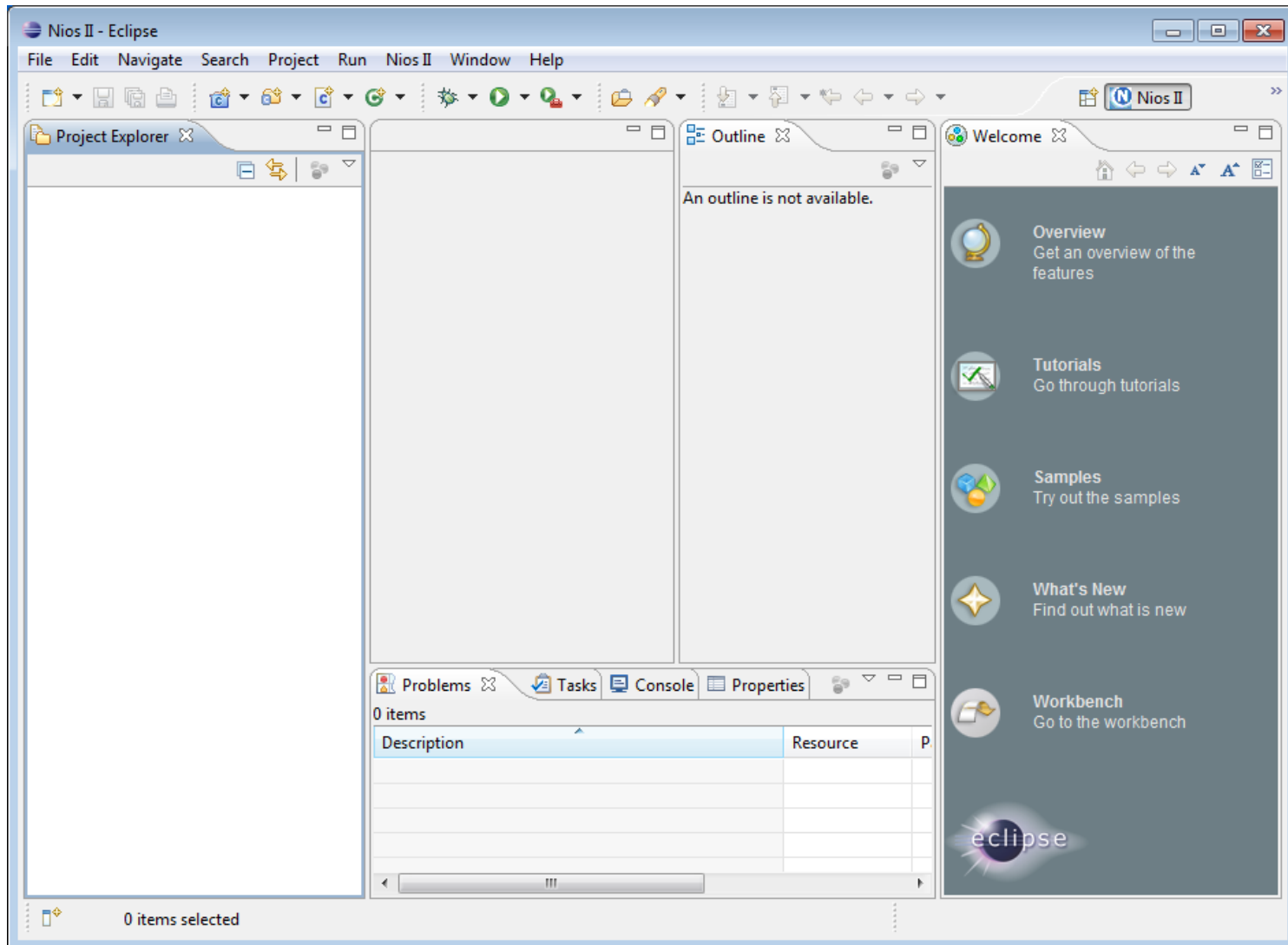


Не размещайте в папке с проектами ПО



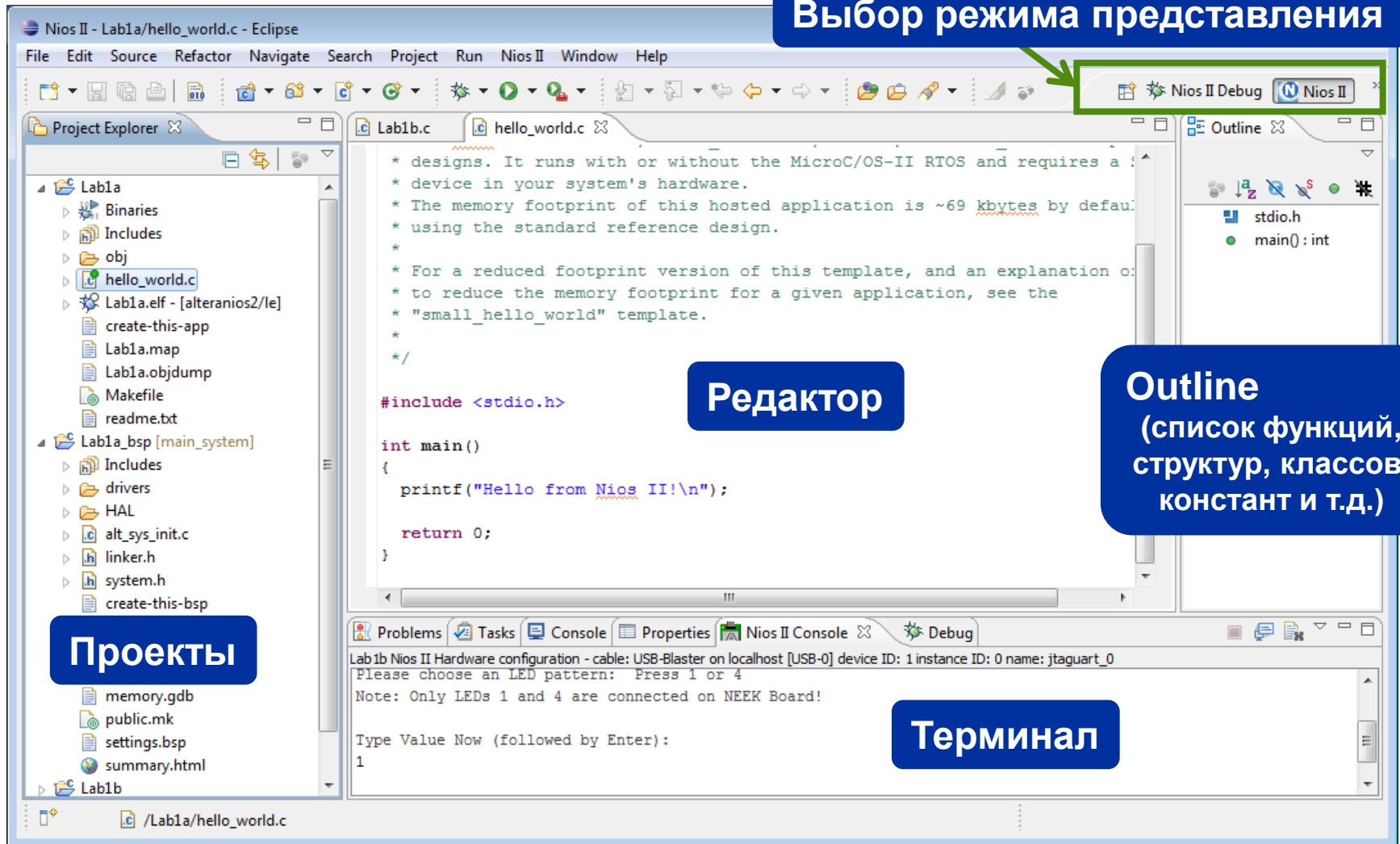


# Новое пустое рабочее пространство

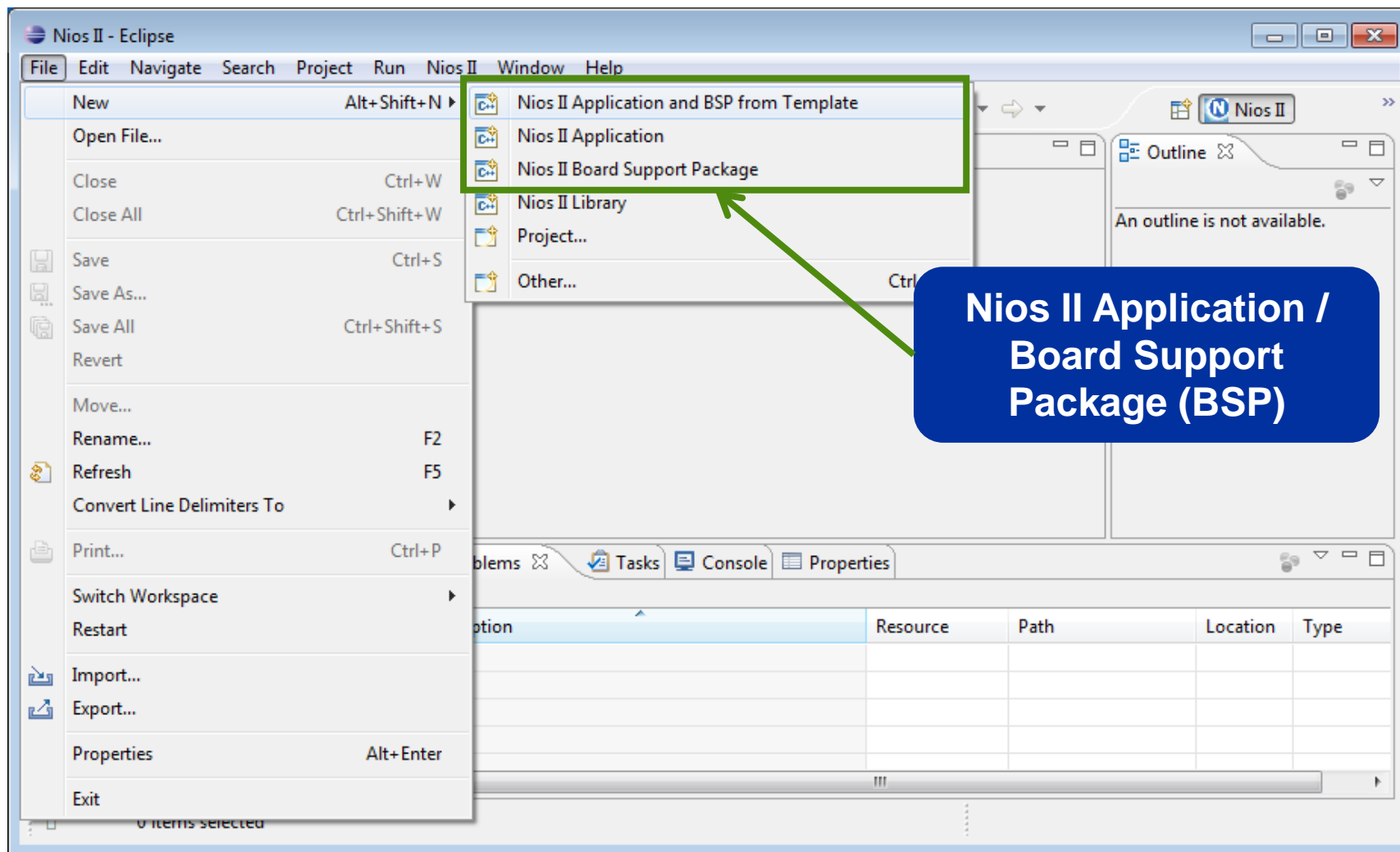


# Рабочее пространство с проектами

Выбор режима представления



# Создание нового проекта ПО



# Проекты приложений и BSP

- Проект приложения - Application Project
  - Собирается как исполняемый файл
  - Требует наличия связанного проекта BSP
- Проект пакета поддержки платы - BSP Project
  - Предоставляет интерфейс к аппаратуре
  - Содержит драйверы компонент (Hardware Abstraction Layer)
  - Может использовать OCPB (MicroC/OS-II)
  - Дополнительные программные компоненты
    - Interniche TCP/IP stack
    - Read Only Zip File System

# Создание приложения и BSP по шаблону

Nios II Application and BSP from Template

**Nios II Software Examples**  
Create a new application and board support package based on a software example template

Target hardware information

SOPC Information File name: C:\altera\_trn\Developing\_Software\_For\_NiosII\NII\_SW\_Dev\working ...

CPU name: cpu

Application project

Project name: my\_proj

☒ Use default location

Project location: C:\altera\_trn\Developing\_Software\_For\_NiosII\NII\_SW\_Dev\working\_dir\soft ...

Project template

Templates

- Blank Project
- Board Diagnostics
- Count Binary
- Hello Freestanding
- Hello MicroC/OS-II
- Hello World**
- Hello World Small
- Memory Test
- Memory Test Small
- Simple Socket Server
- Simple Socket Server (RGMII)
- Web Server
- Web Server (RGMII)

Template description

Hello World prints 'Hello from Nios II' to STDOUT.

This example runs with or without the MicroC/OS-II RTOS and requires an STDOUT device in your system's hardware.

For details, click Finish to create the project and refer to the readme.txt file in the project directory.

The BSP for this template is based on the Altera HAL operating system.

For information about how this software example relates to

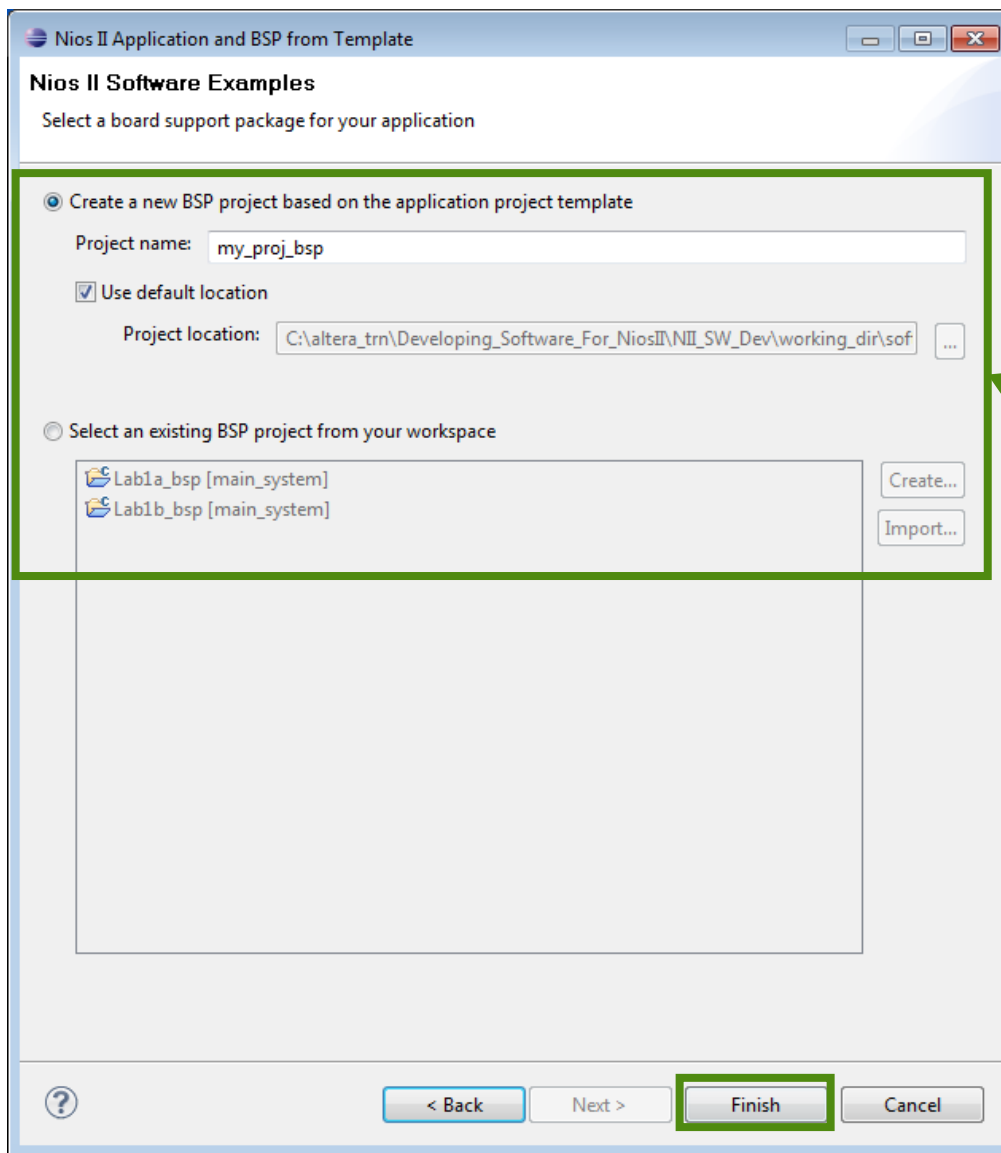
< Back Next > Finish Cancel

Выберите файл  
.sopcinfo

Задайте имя  
проекта ПО

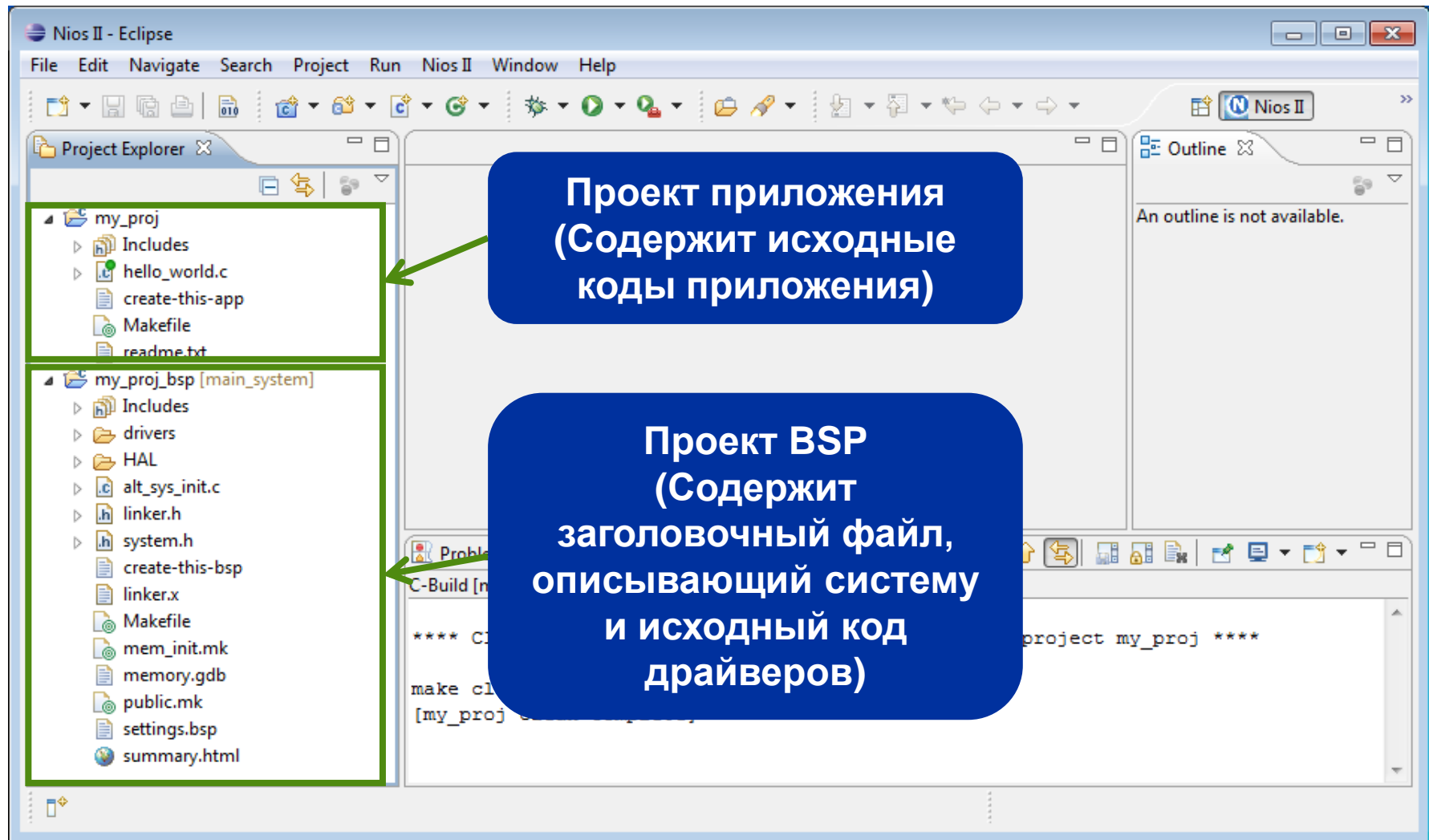
Выберите шаблон  
или пустой  
проект Blank  
Project

# Создание приложения и BSP по шаблону

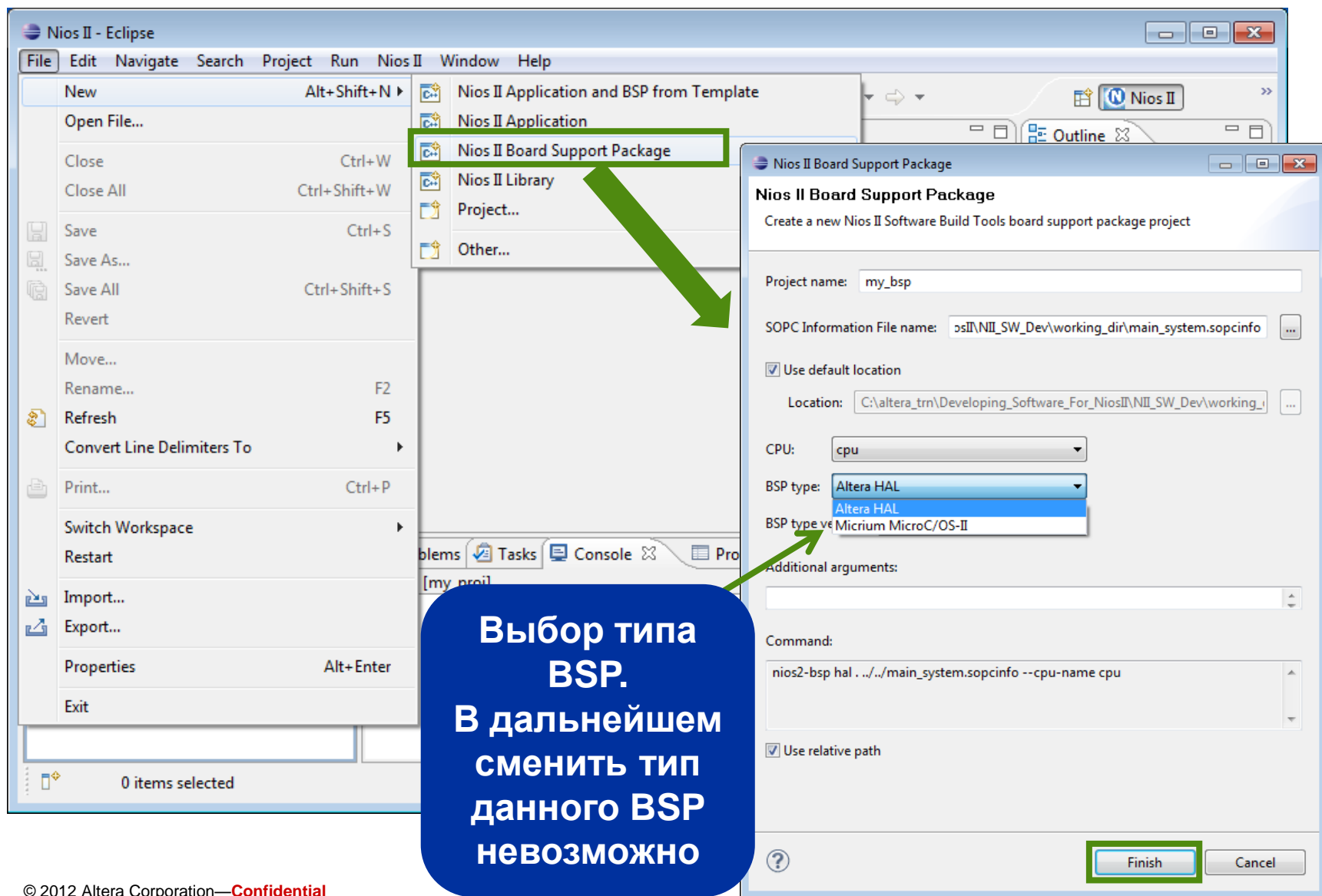


Создайте новый  
BSP или выберите  
существующий

# Папки приложения и BSP

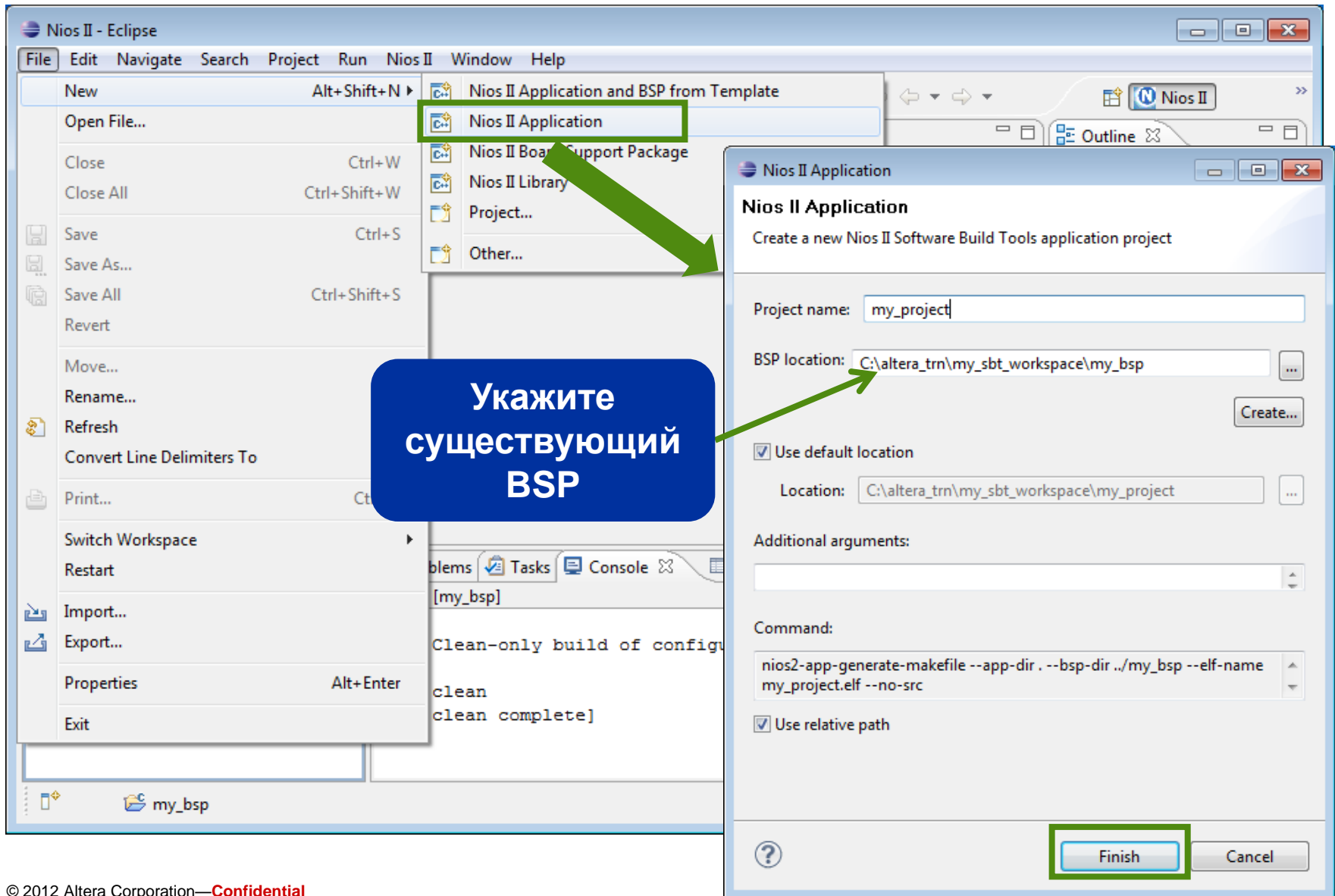


# Создание BSP

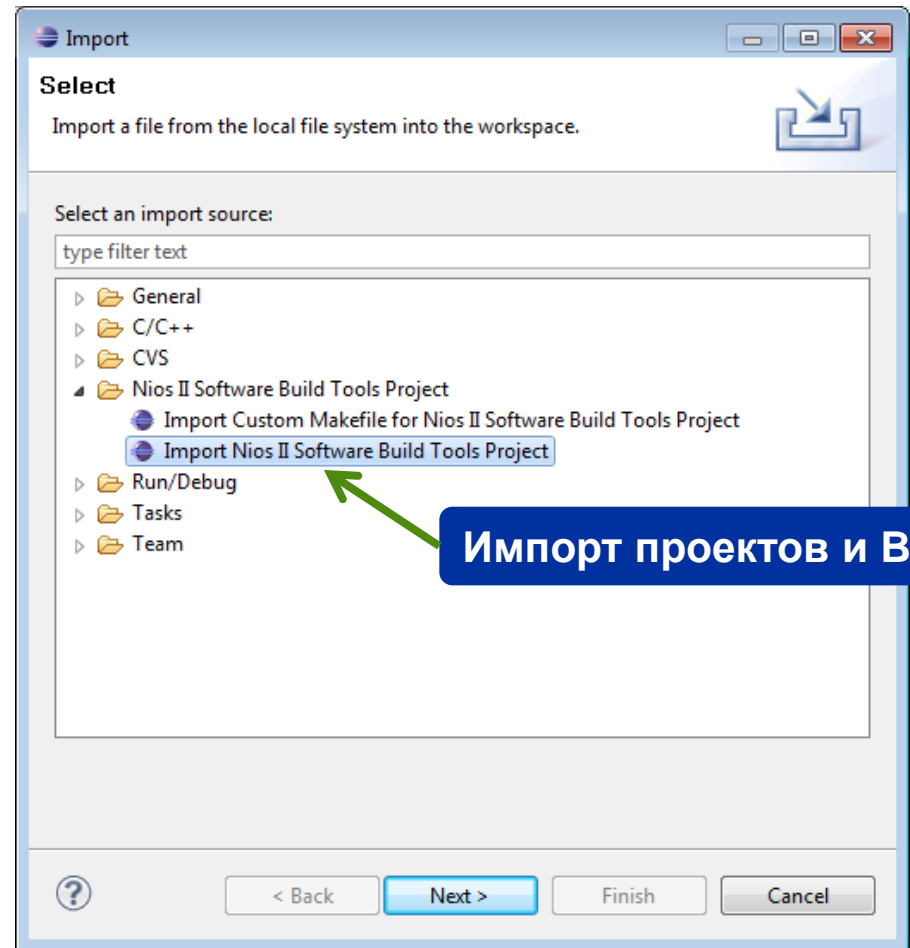
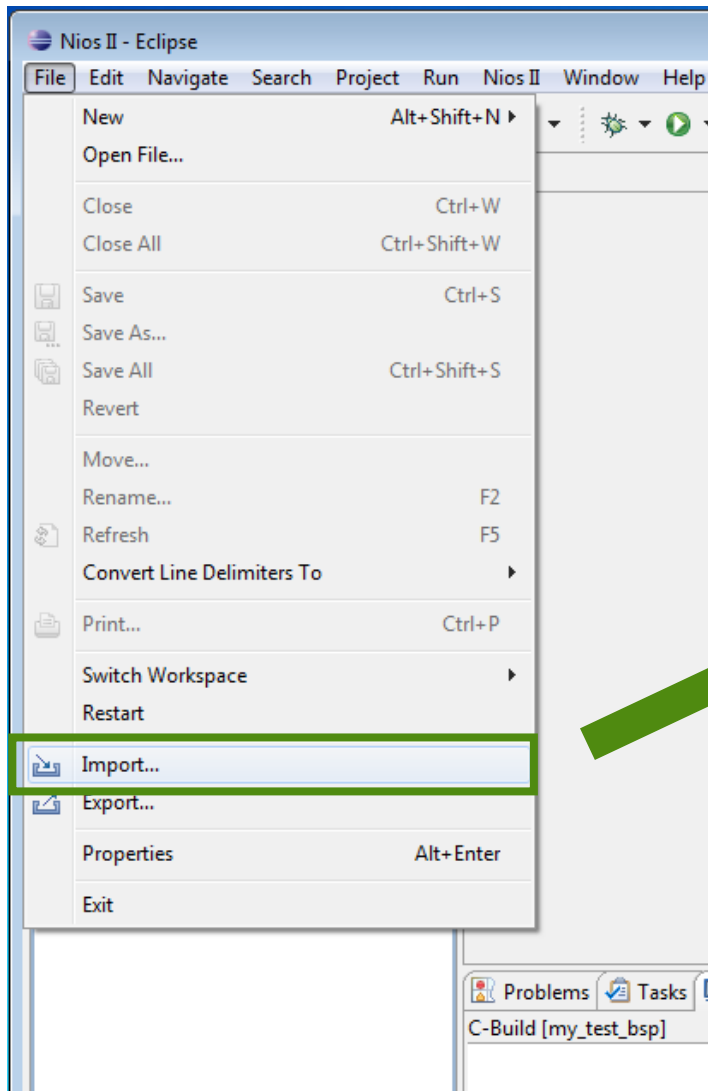




# Создание приложения

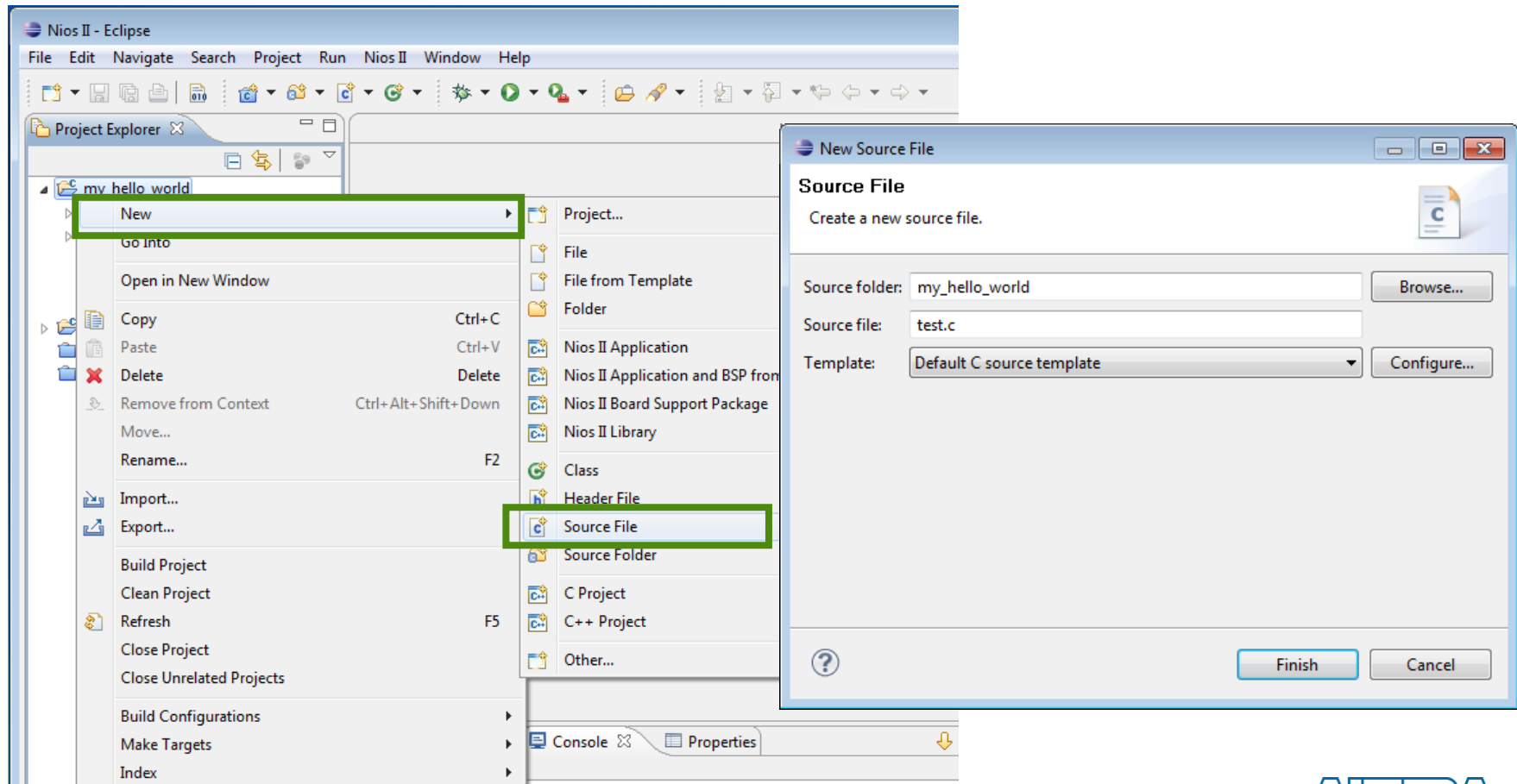


# Импорт проектов в рабочее пространство



# Создание нового исходного файла

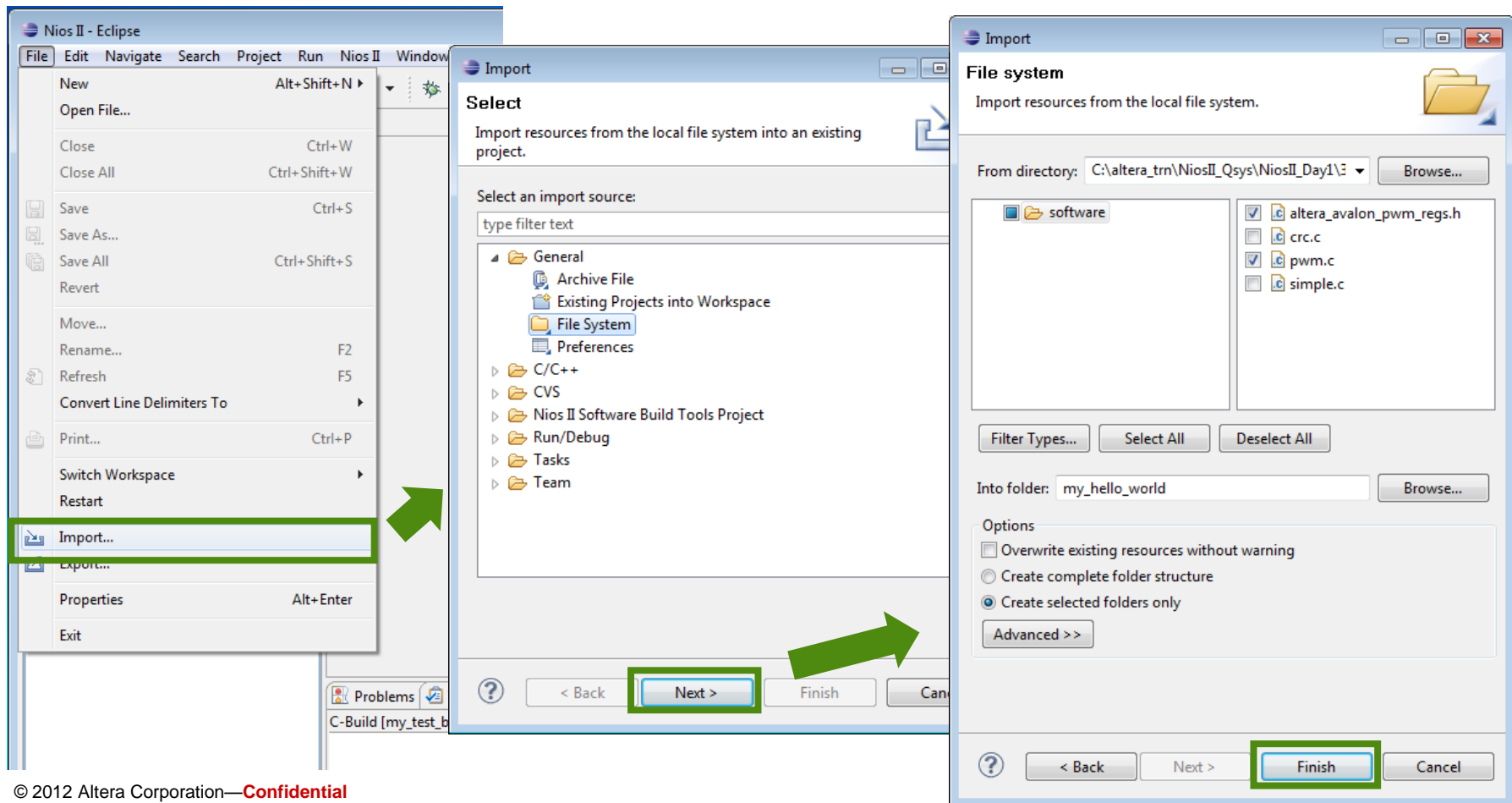
- Выберите New > Source File из контекстного меню, вызванного на имени проекта
  - Укажите папку для сохранения, имя файла и шаблон



# Импорт исходных файлов в проект

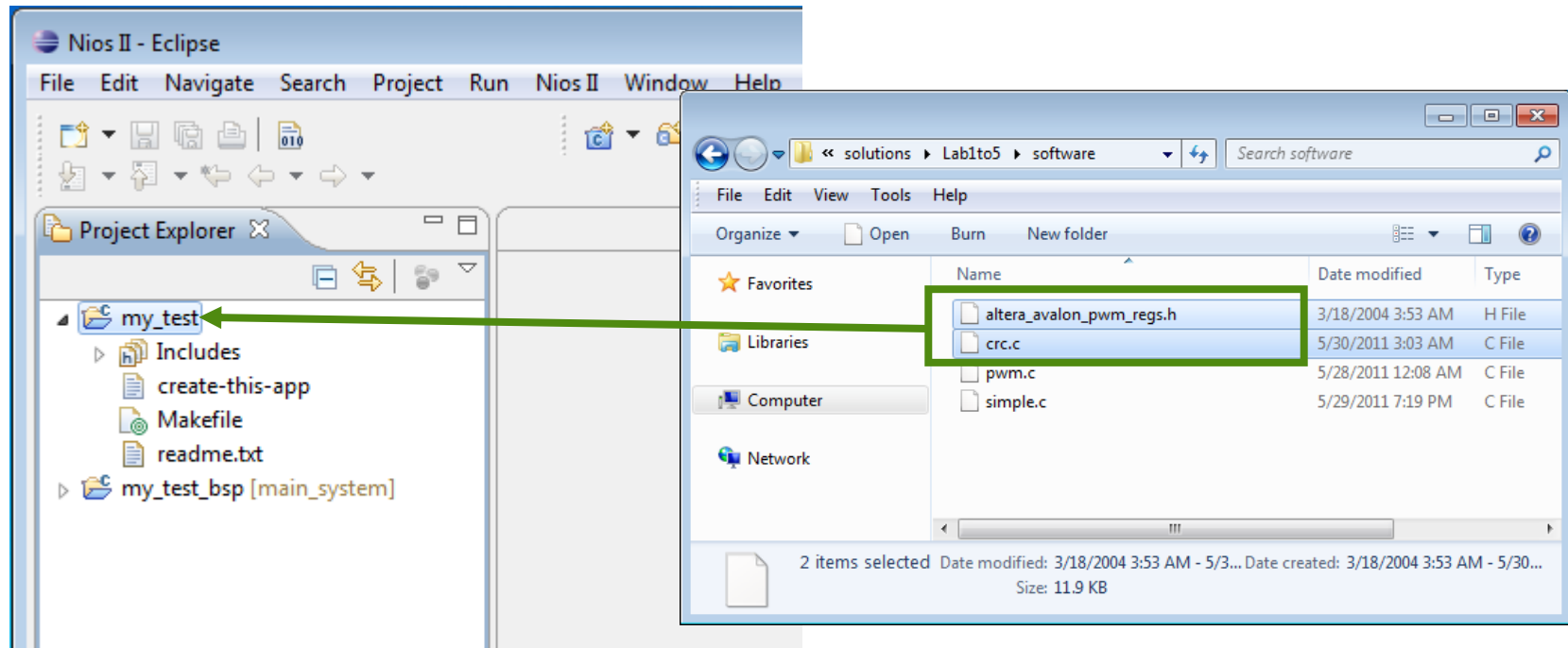
## ■ Выберите **File** → **Import...**

- Укажите папку проекта приложения и выберите файлы



# Импорт перетаскиванием файлов

- Перетащите файлы из проводника в проект приложения
  - Выберите, копировать файлы в папку проекта (**Copy files**) или создать ссылку на исходное расположение (**Link to files**)
  - Проект и makefile будут обновлены автоматически



# Включение автосохранения

The screenshot shows the Eclipse IDE interface. The 'File' menu is open, and the 'Preferences' option is highlighted. The 'Preferences' dialog box is open, showing the 'Workspace' section. The 'Save automatically before build' checkbox is checked and circled in green. The 'Workspace' section also includes options for 'Build automatically', 'Refresh automatically', and 'Always close unrelated projects without prompt'. The 'Workspace save interval (in minutes)' is set to 5. The 'Workspace name (shown in window title)' is empty. The 'Open referenced projects when a project is opened' section has 'Prompt' selected. The 'Text file encoding' is set to 'Default (Cp1252)'. The 'New text file line delimiter' is set to 'Default'. The 'Restore Defaults', 'Apply', 'OK', and 'Cancel' buttons are visible at the bottom of the dialog.

По умолчанию файлы не сохраняются автоматически при запуске сборки

```
/*  
 * "Hello World" example  
 *  
 * This example prints  
 * the Nios II 'standa  
 * designs. It runs wi  
 * device in your syst  
 * The memory footprint  
 * using the standard  
 *  
 * For a reduced footp  
 * to reduce the memo  
 * "small_hello_world  
 *  
 */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("Hello from Me from Nios II!\n");  
  
    return 0;  
}
```

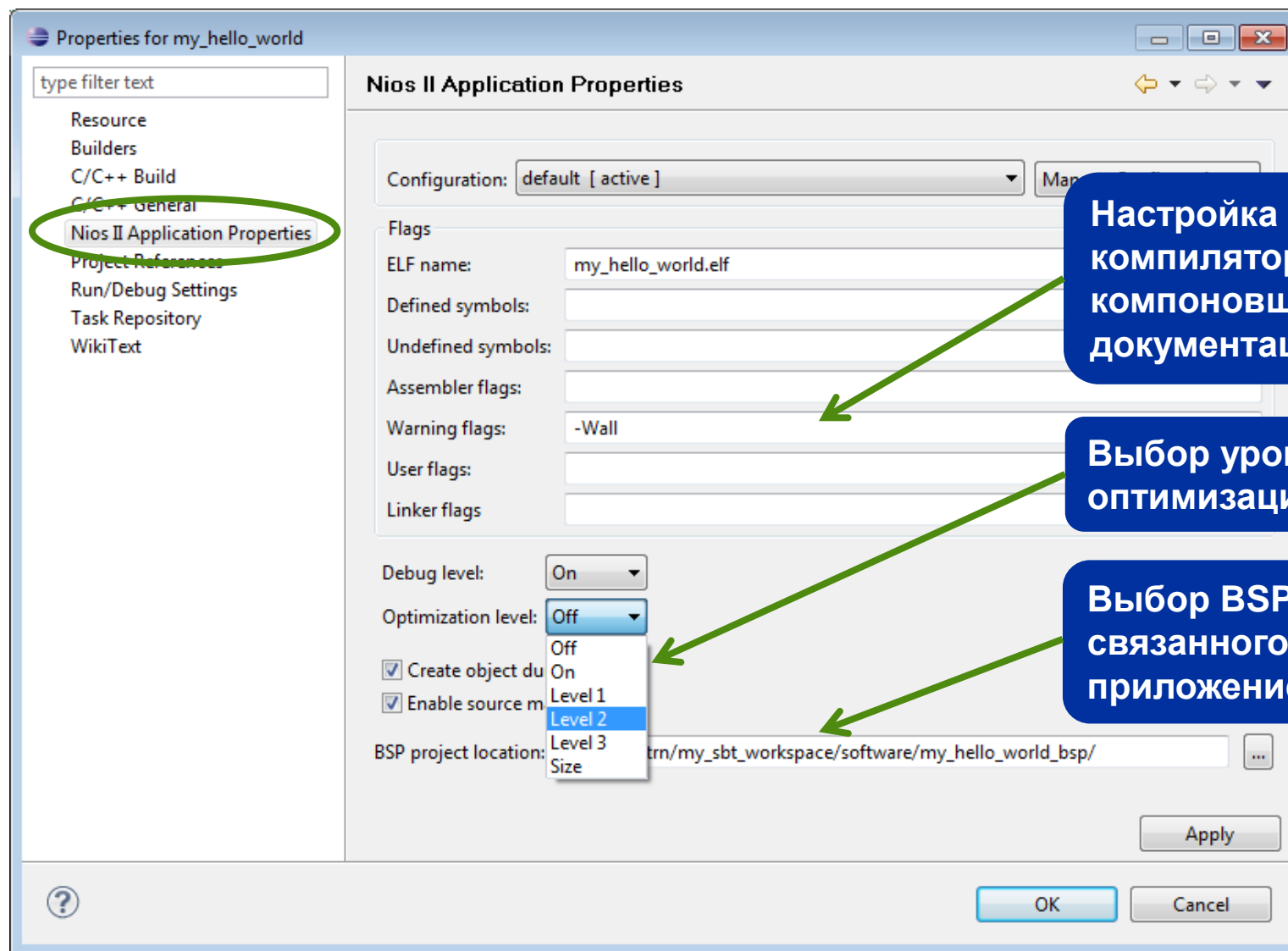
C-Build [my\_hello\_world]  
Info: (my\_hello\_world.elf) 46 KBytes program  
Info: 32721 KBytes free  
Info: Creating my\_hello\_world.objdump  
nios2-elf-objdump --disassemble --syms --all  
>my\_hello\_world.objdump  
[my\_hello\_world build complete]

# Настройка свойств проекта

The screenshot shows the Eclipse IDE interface. The 'Project' menu is open, displaying options: Open Project, Close Project, Build All (Ctrl+B), Build Project, Build Working Set, Clean..., Build Automatically, and Properties. The 'Properties' option is circled in green. In the Project Explorer on the left, the 'my\_test\_bsp' project is selected, with a green arrow pointing from a blue callout box to it. The callout box contains the text: 'Или вызовите контекстное меню Properties на проекте приложения или BSP'. The bottom of the IDE shows the 'Properties' tab for 'C-Build [my\_hello\_world]' with build output logs.

Или вызовите  
контекстное  
меню  
Properties на  
проекте  
приложения  
или BSP

# Свойства проекта приложения



Настройка флагов компилятора и компоновщика (см. документацию GCC)

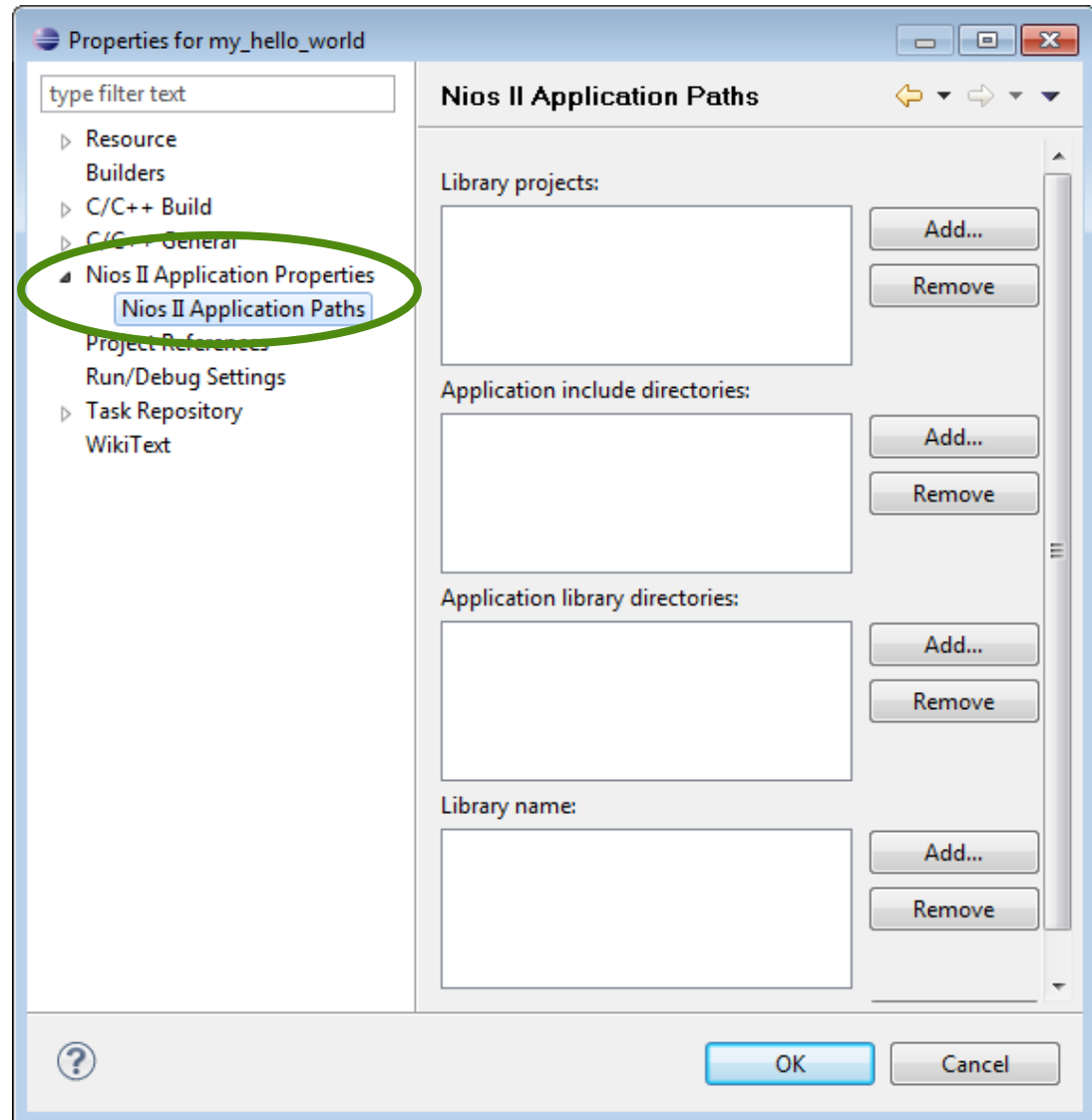
Выбор уровня оптимизации GCC

Выбор BSP, связанного с приложением

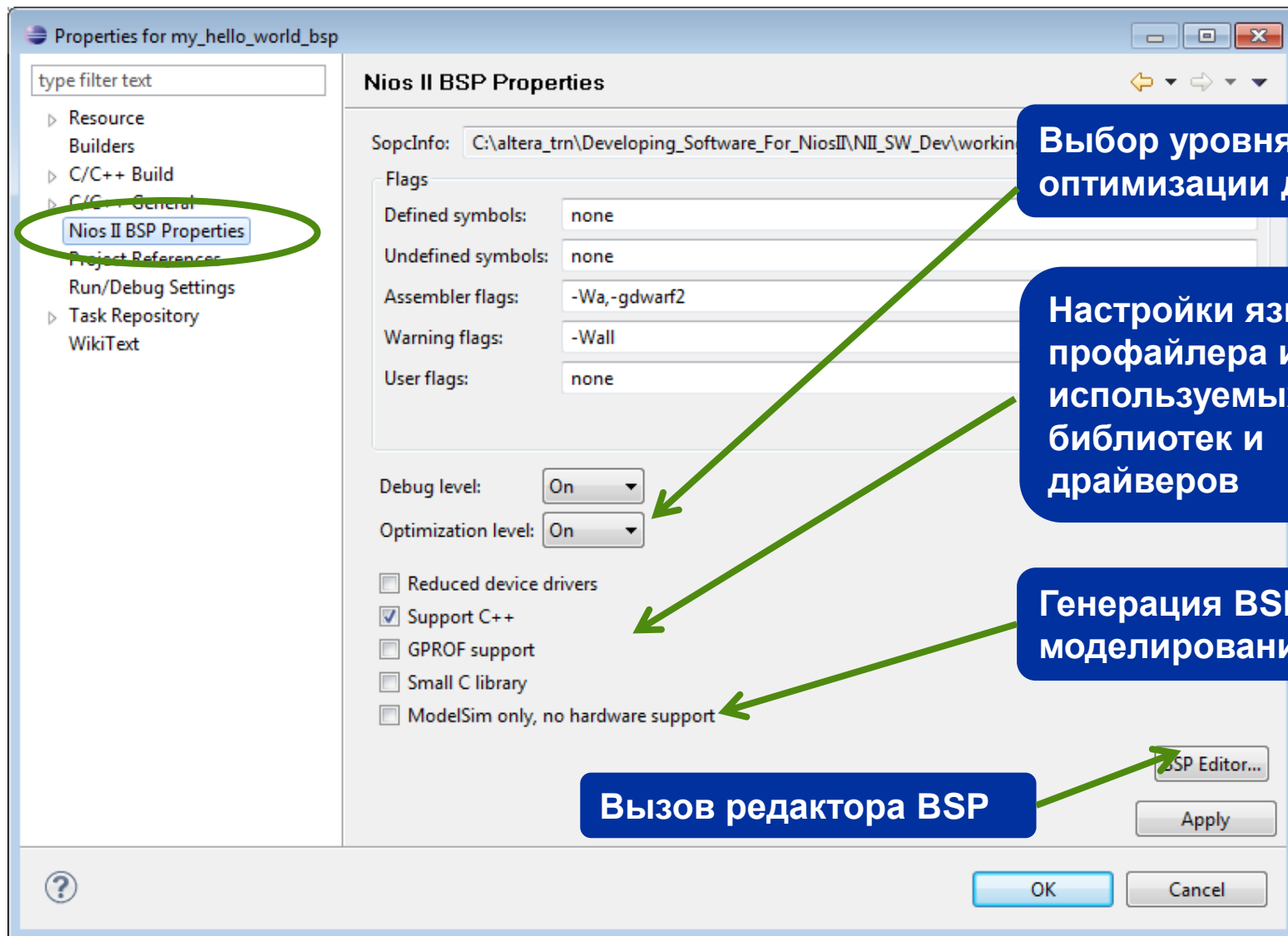


# Добавление библиотек к проекту приложения

- Можно включать папки и файлы
- Включение по полному или относительному пути



# Свойства проекта BSP



Выбор уровня  
оптимизации для BSP

Настройки языка,  
профайлера и типа  
используемых  
библиотек и  
драйверов

Генерация BSP для  
моделирования

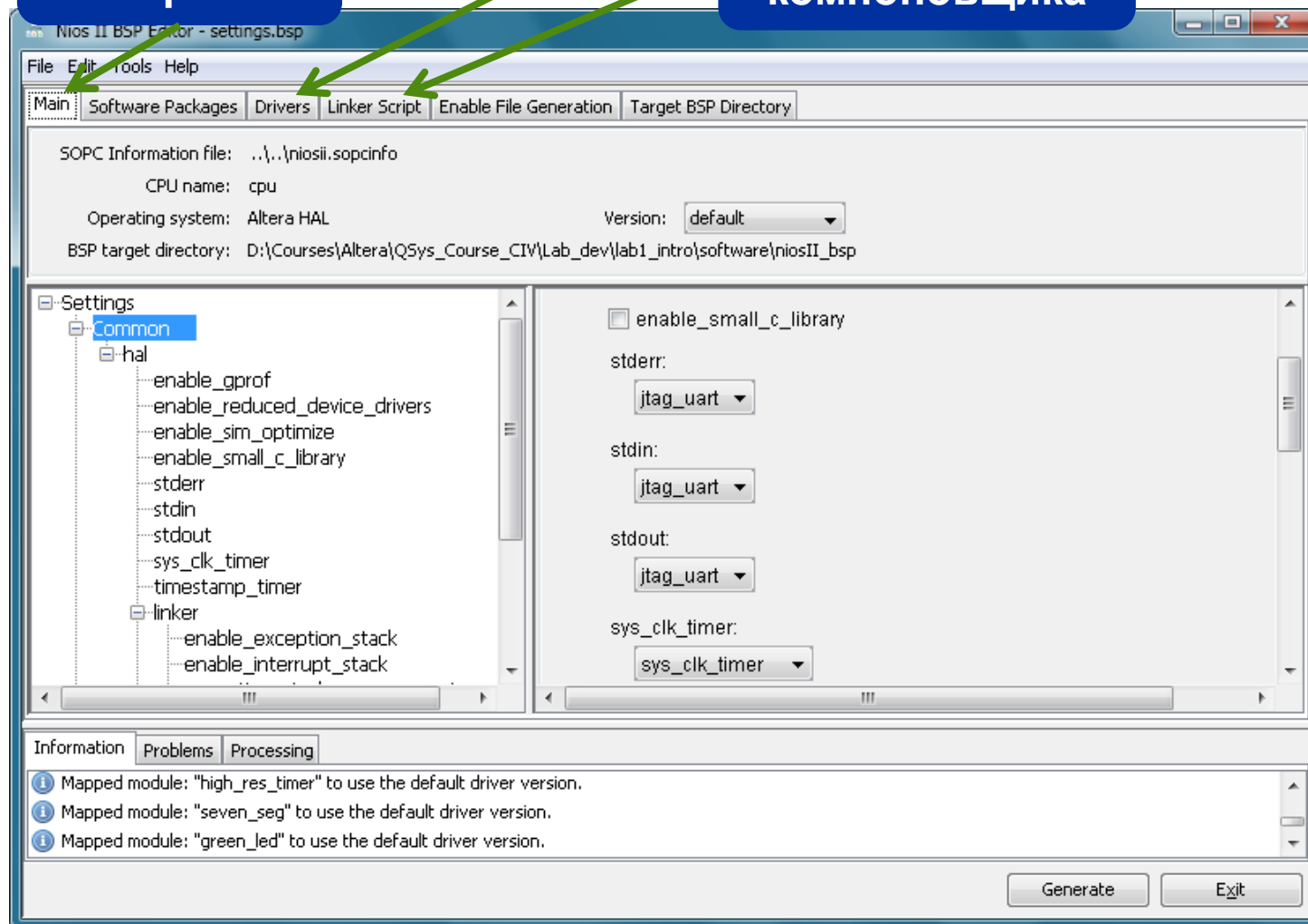
Вызов редактора BSP

# BSP Editor

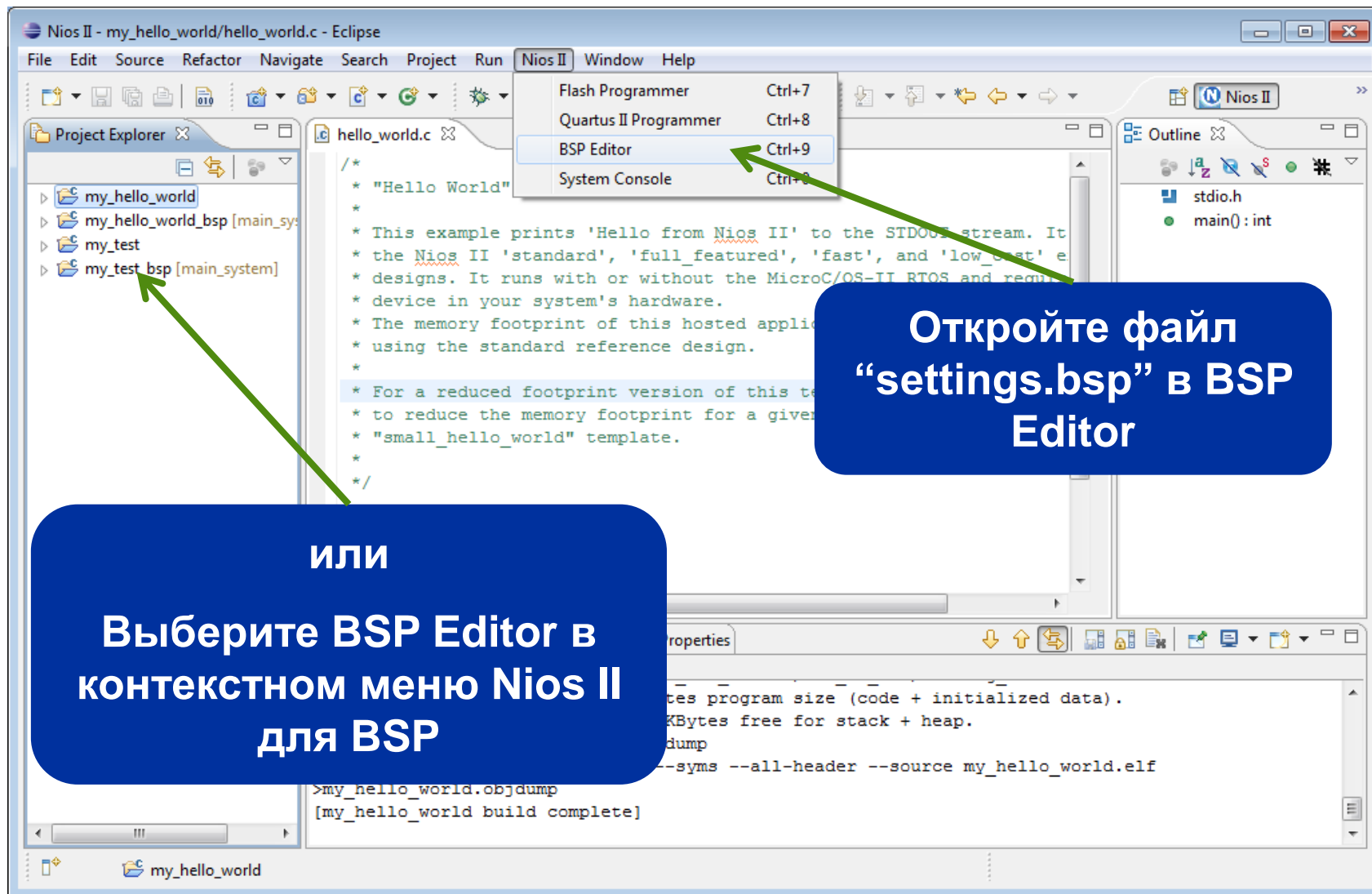
Основные  
настройки

Драйверы

Настройки  
компоновщика

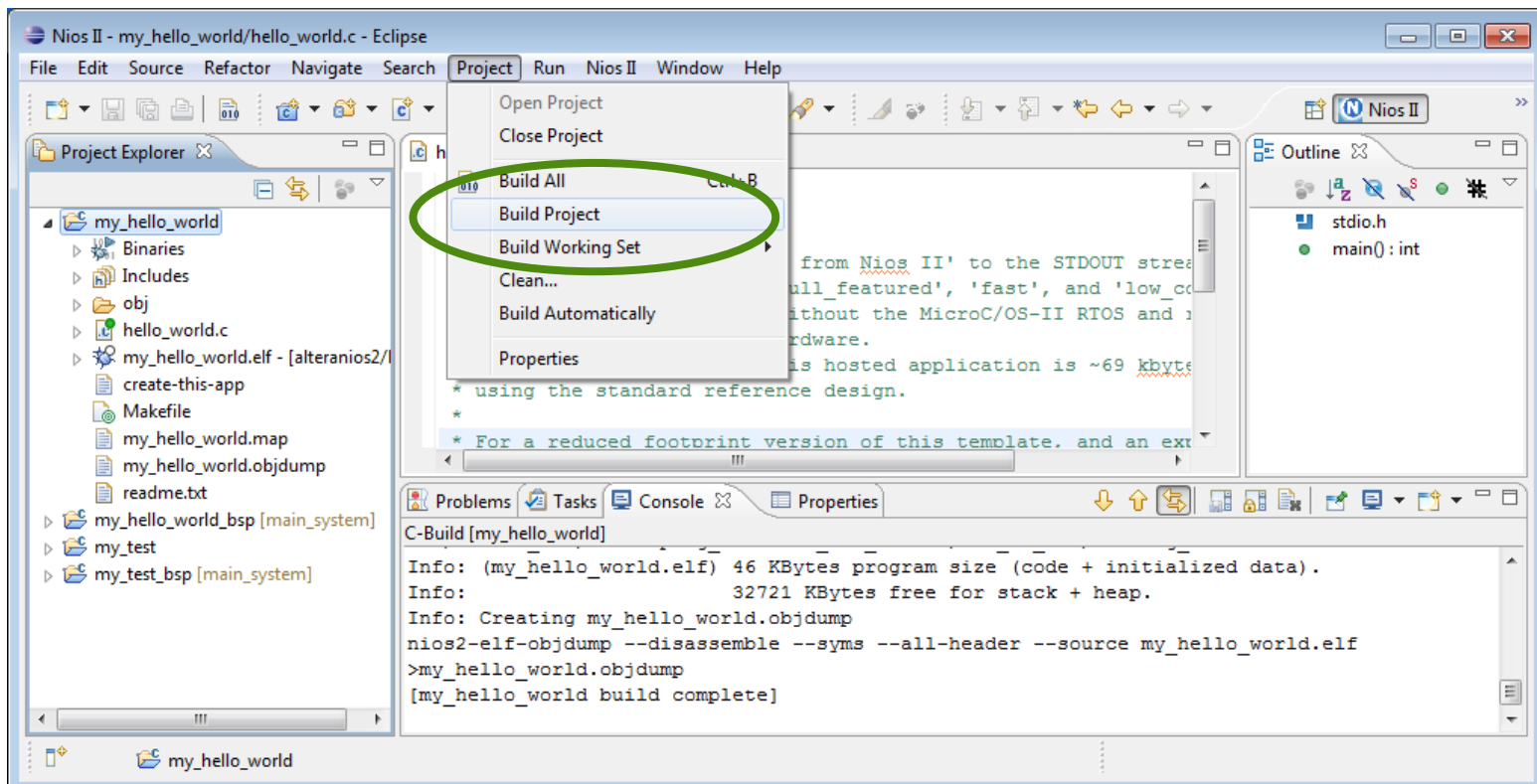


# Запуск редактора BSP из меню Nios II



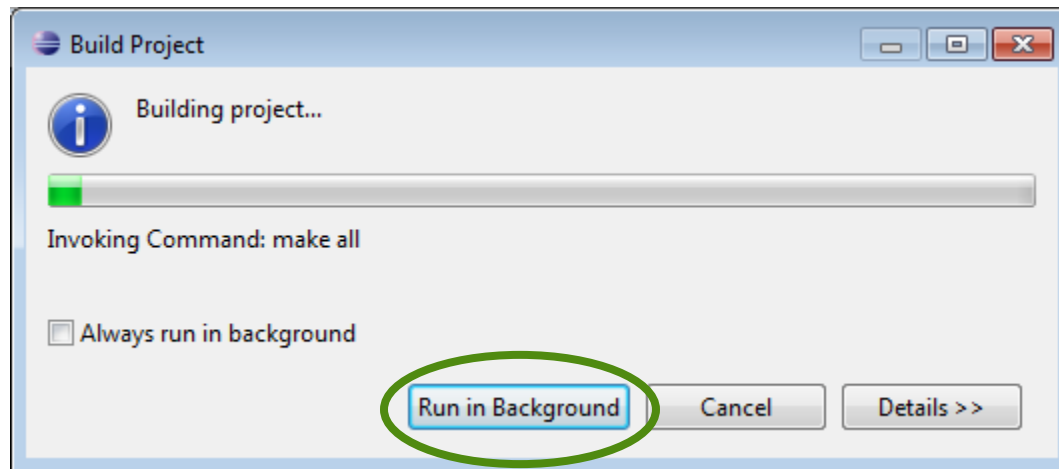
# Сборка ПО (Build)

- Для сборки проекта
  - Выберите его и вызовите меню Project > Build Project
  - Вызовите контекстное меню на проекте и выберите Build Project
- Автоматически компилирует проект приложения и BSP



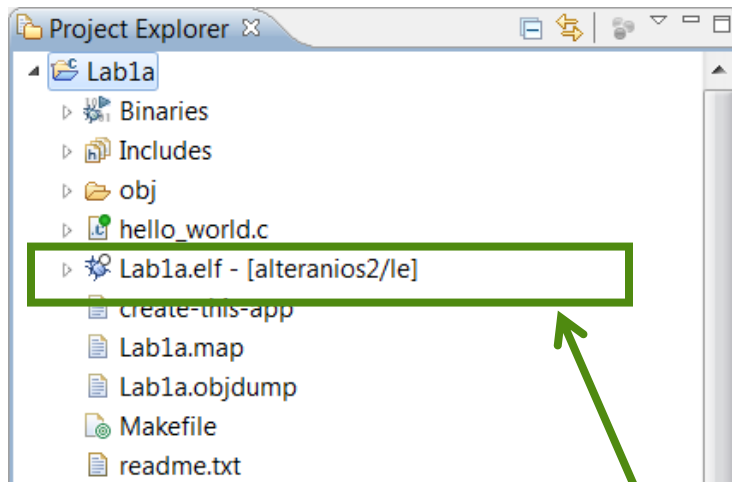
# Сборка в фоновом режиме

- Позволяет выполнять другие действия во время сборки проекта

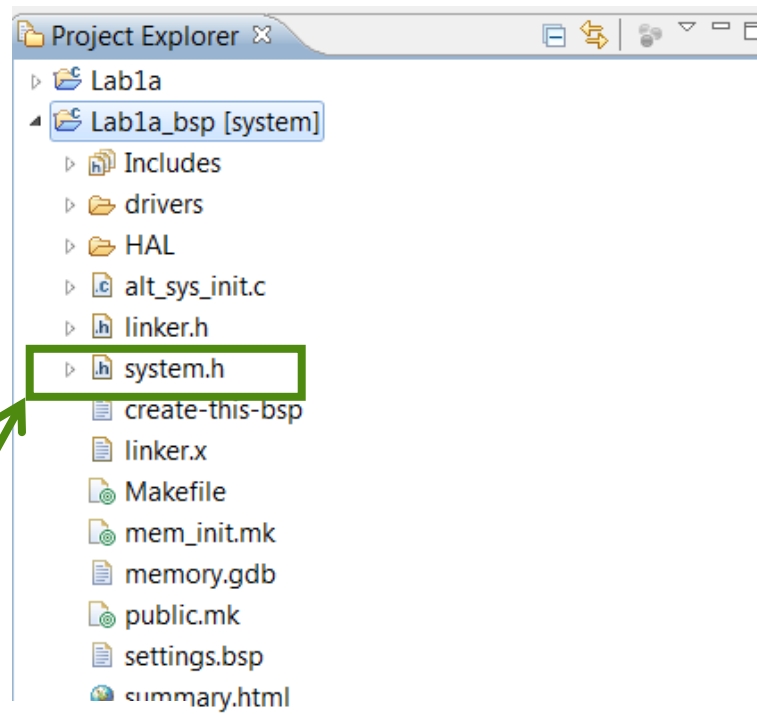


# Структура папок проекта приложения и BSP

## Application Project



## BSP Project



**Важные файлы**

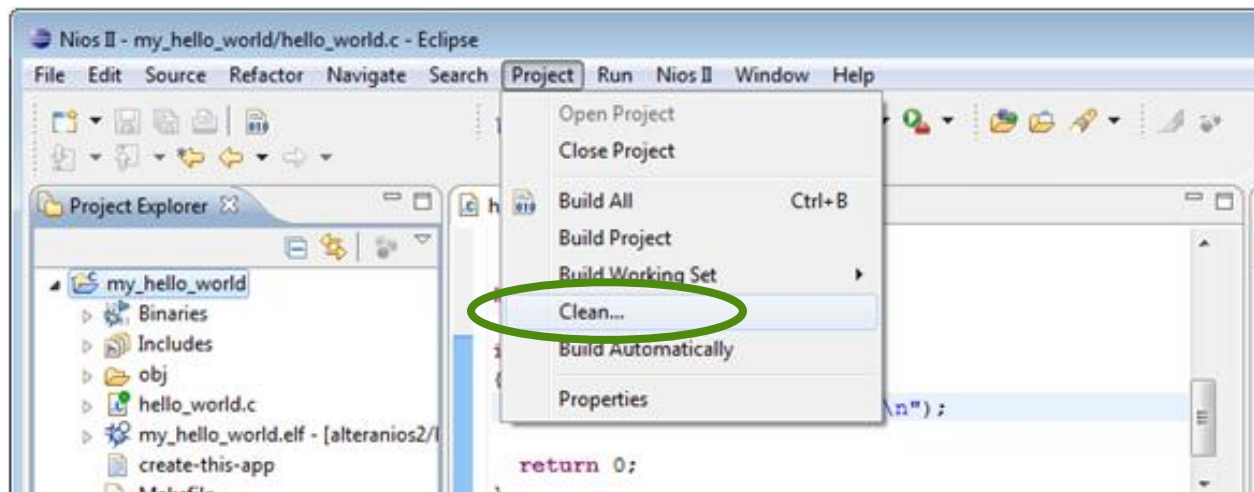
# Важные файлы

- Файл `.elf` *(в папке проекта приложения)*
  - Исполняемый формат, содержит откомпилированный исходный код
  - Может быть загружен в систему
- Файл `system.h` *(в папке BSP)*
  - Заголовочный файл с описанием системы
  - Содержит описание всех периферийных модулей (имя, базовый адрес, номер запроса прерывания и т.д.)

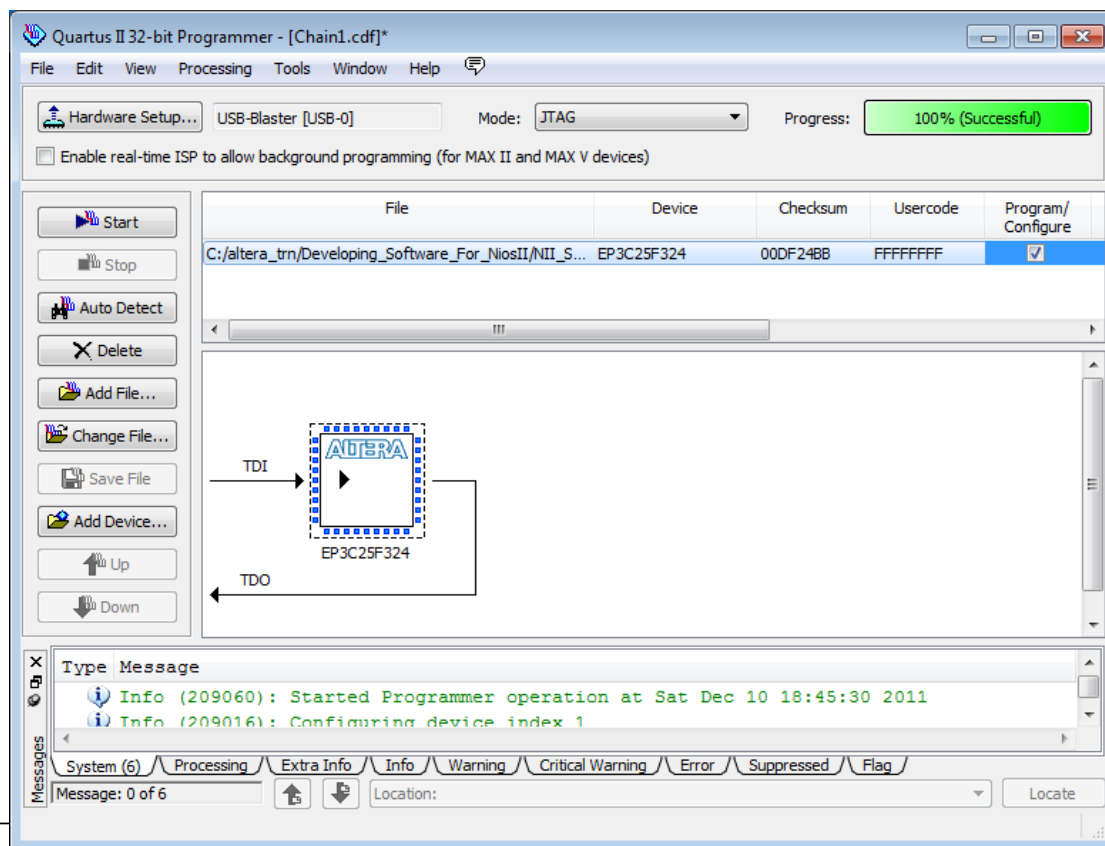
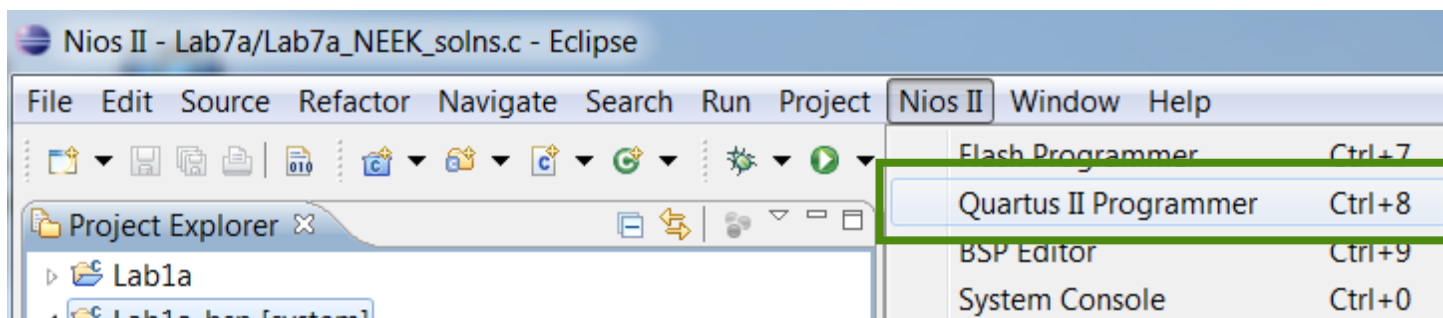


# Очистка проекта

- Удаляет все файлы, сгенерированные при компиляции проекта
  - Выберите **Clean Project** в контекстном меню проекта или вызовите **Clean** в меню **Project**
- При сборке будет осуществлена полная перекомпиляция
- Иногда позволяет убрать ошибки, возникающие при компиляции



# Загрузка конфигурации ПЛИС

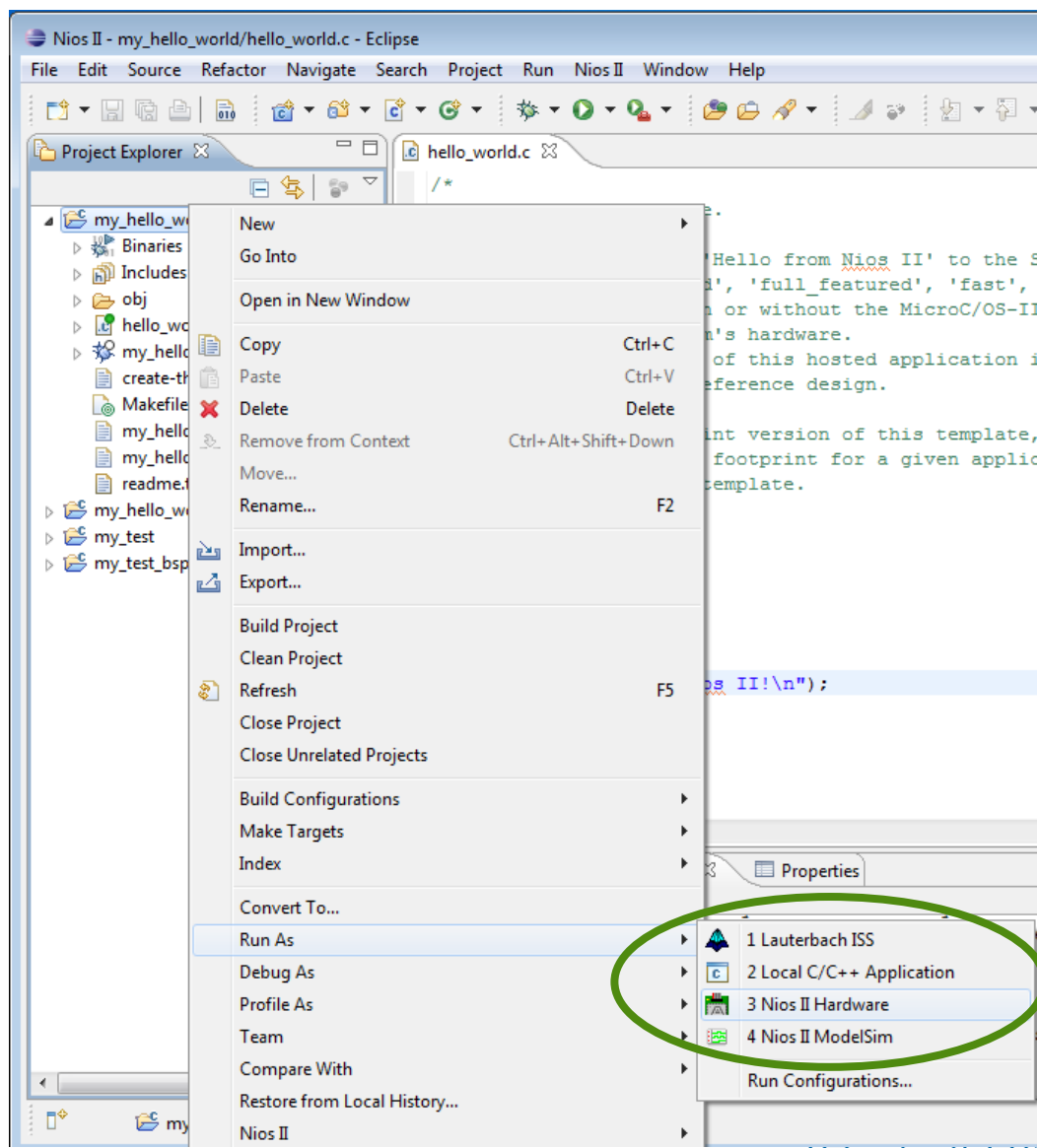


# Запуск кода на целевой платформе

## ■ Выбор платформы

- Плата
- ModelSim
- Симулятор

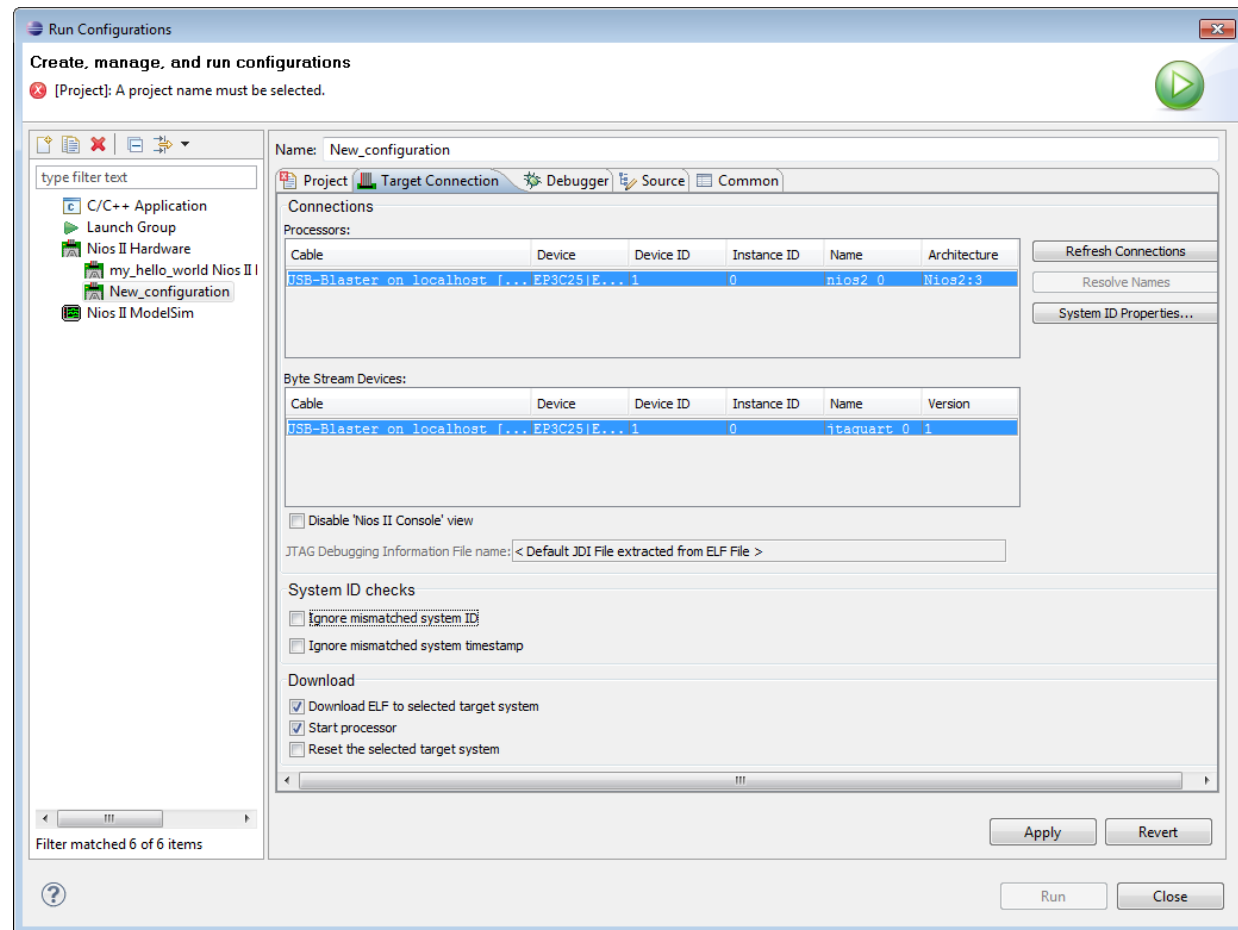
## ■ Вызовите контекстное меню и выберите **Run As** или используйте меню **Run > Run As**



# Настройка конфигураций запуска

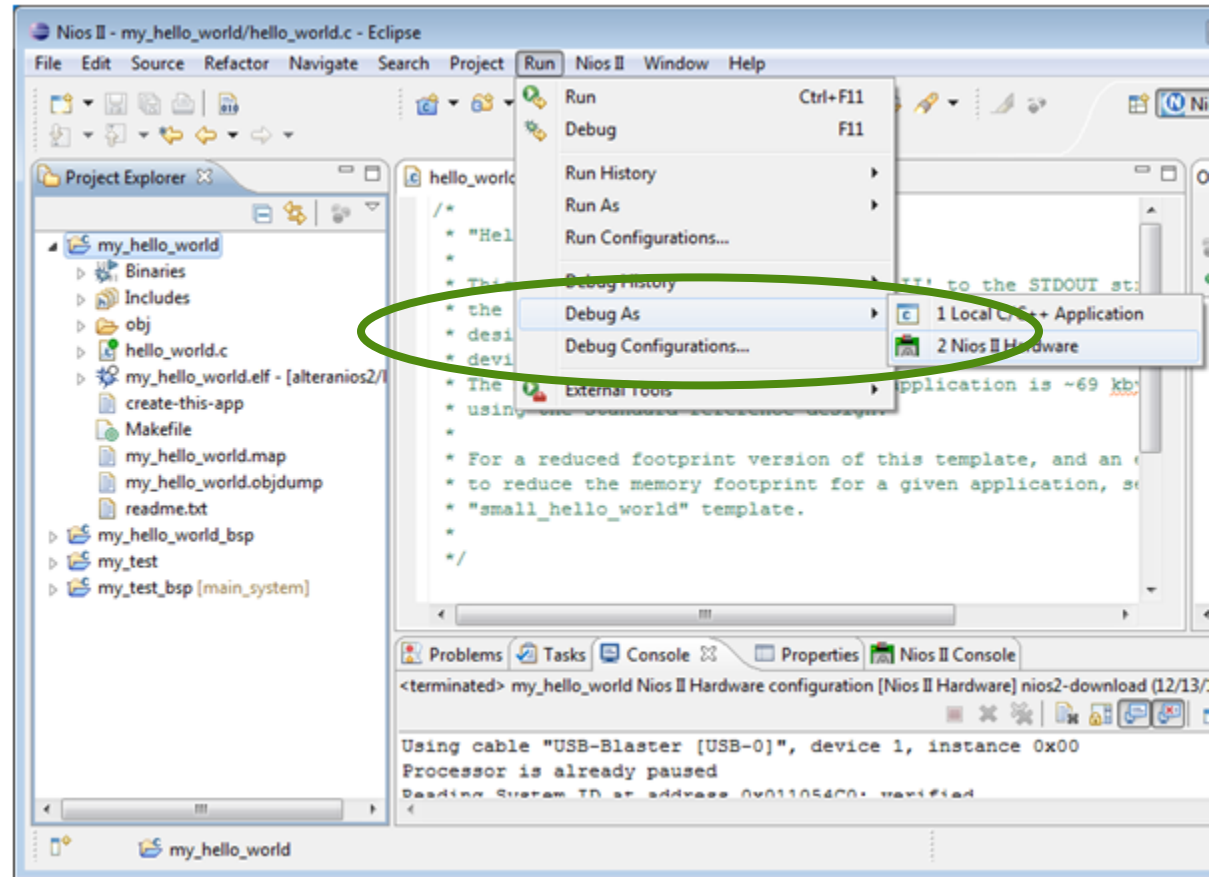
## Меню Run > Run Configurations...

- Выбор проекта приложения
- Настройка соединения с ПЛИС
- Проверка System ID
- Настройка загрузки кода и режима запуска



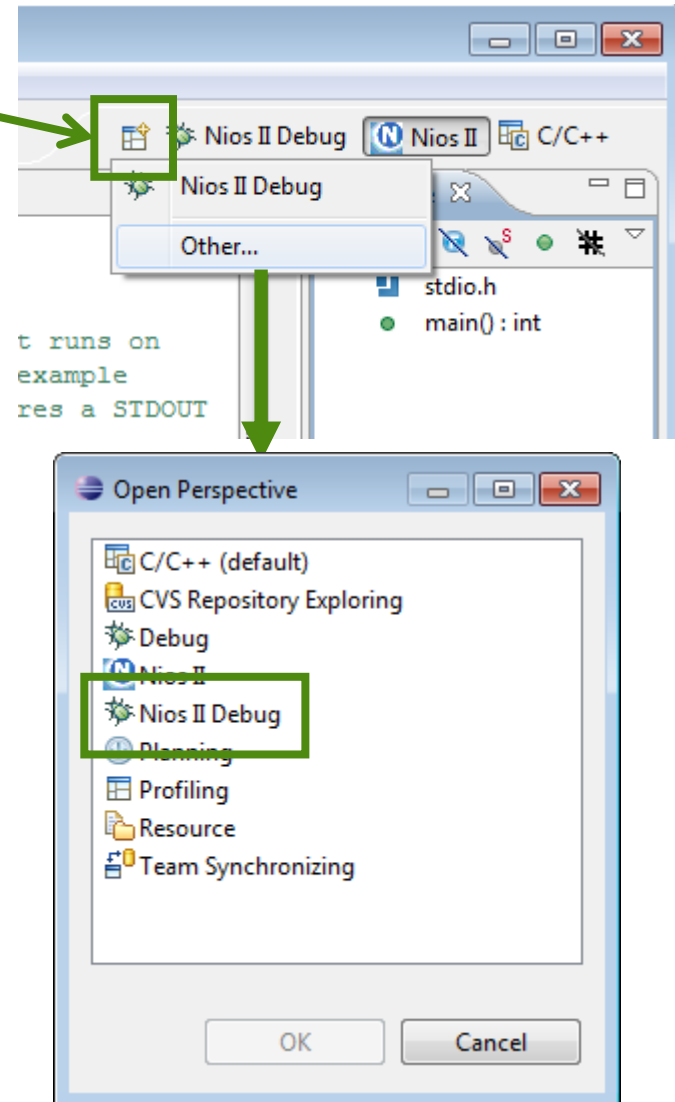
# Отладчик Nios II Debugger

- Вызовите контекстное меню и выберите **Debug As** или используйте меню **Run > Debug As**
- В конфигурации ядра процессора должно быть включено JTAG Debug Core



# Переключение в перспективу отладки

- Переключение вручную
- Смена перспективы предлагается автоматически при запуске отладчика
- Различные перспективы предоставляют удобный набор открытых окон
  - Nios II – разработка кода
  - Debug – отладка
  - Profiling – профилирование



# Nios II Debug Perspective

The screenshot displays the Nios II Debug Perspective, which is divided into several panes. The top-left pane shows the 'Debug' console with a list of threads, including 'Thread [1] (Suspended: Breakpoint hit.)' and '3 main() Lab1b.c:49 0x08000380'. The top-right pane shows the 'Variables' pane with a table of variables: 'rx\_char' with value '1' and 'line' with value '0x09ffff'. The bottom-left pane shows the 'Lab1b.c' source code with a breakpoint set at line 49. The bottom-right pane shows the 'Console' window with the text 'Lab1b Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jtaguart\_0' and a prompt 'Type Value Now (followed by Enter):' with the input '1'.

**Запуск, пауза и остановка выполнения**

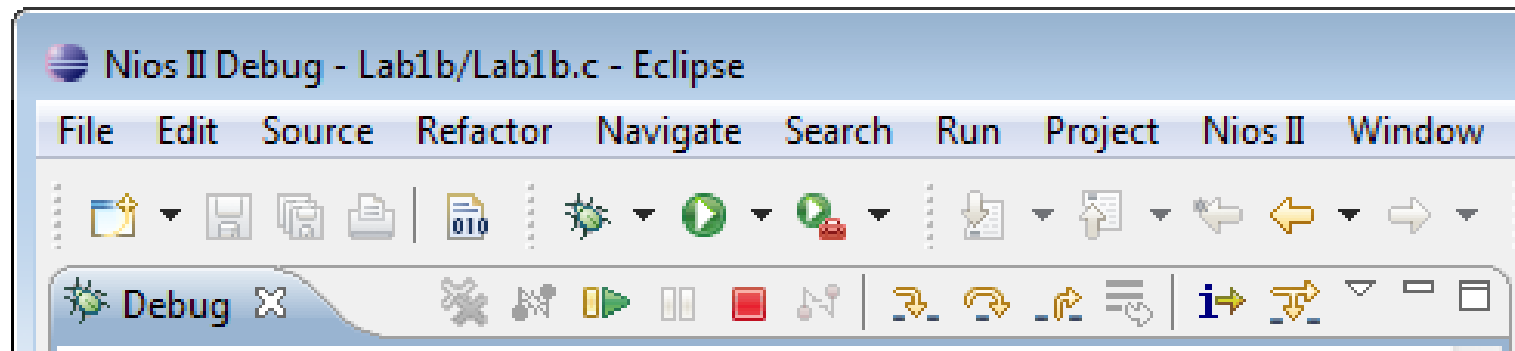
**Пошаговое выполнение**

**Просмотр и редактирование регистров, памяти и т.д.**

**Двойной щелчок создает точку останова**

**Консоль**

# Управление отладкой



**Resume**

**Suspend**

**Terminate**

**Step Into**

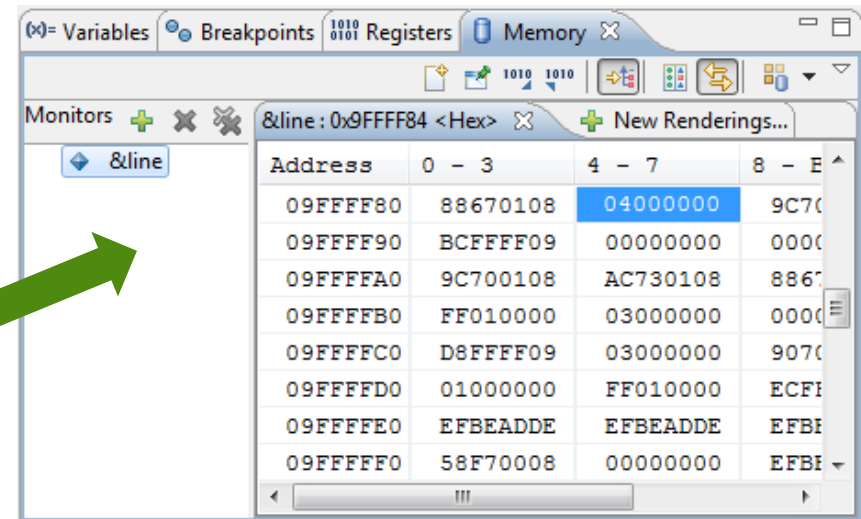
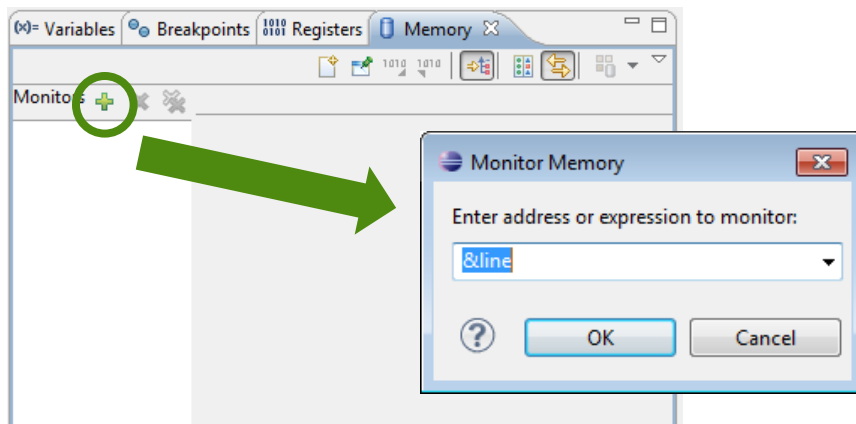
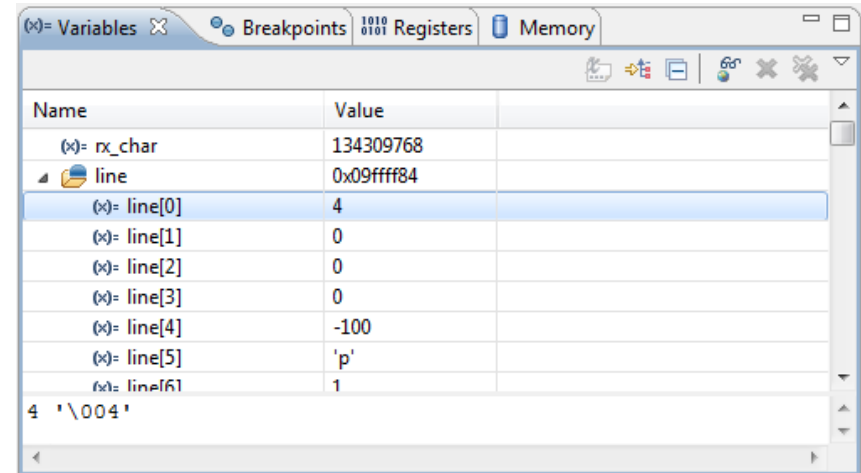
**Step Over**

**Step Return**

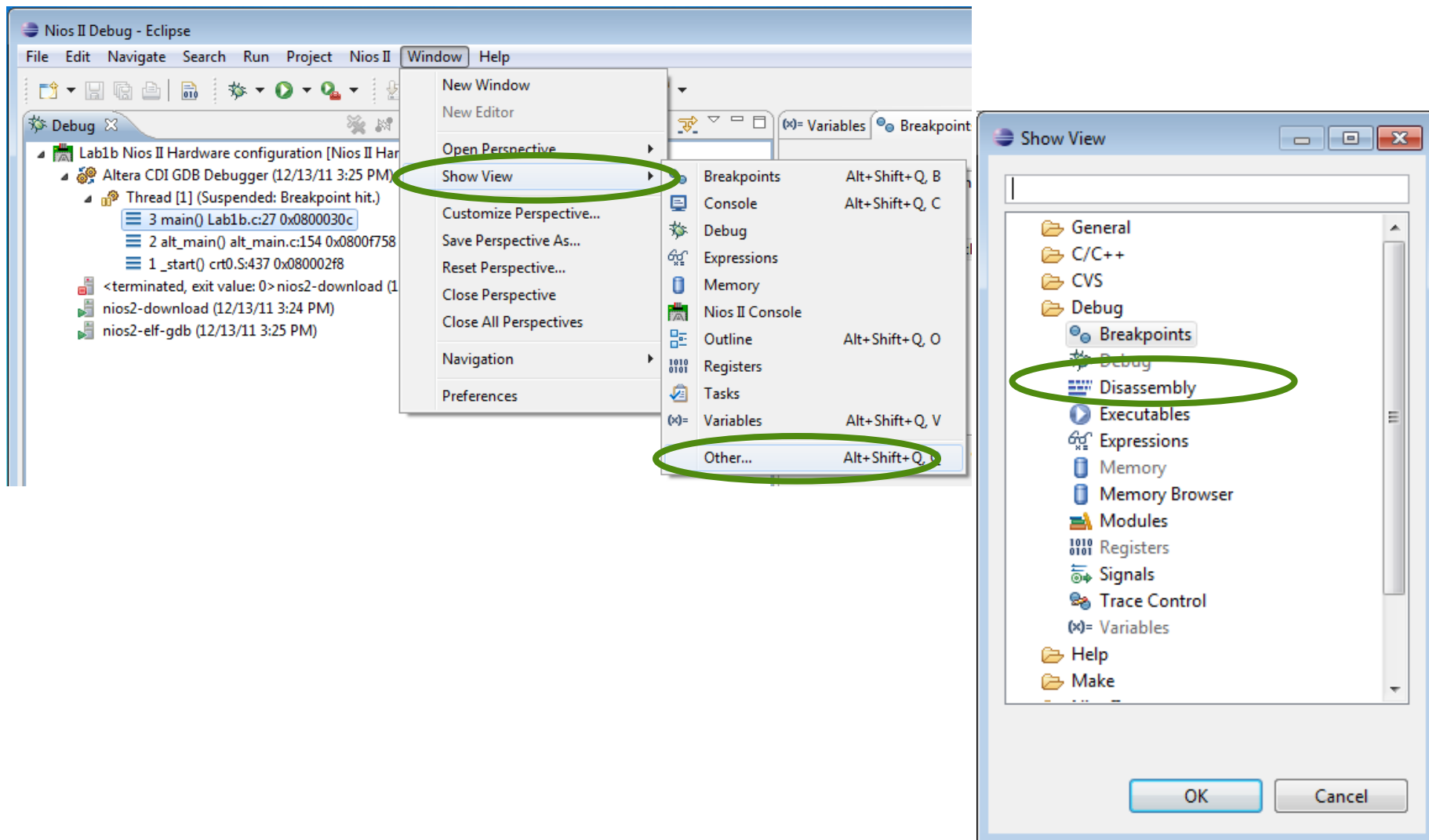


# Окна отладчика

- Стандартные окна
  - Variables - переменные
  - Memory - память
  - Registers – регистры процессора
  - Breakpoints – точки останова

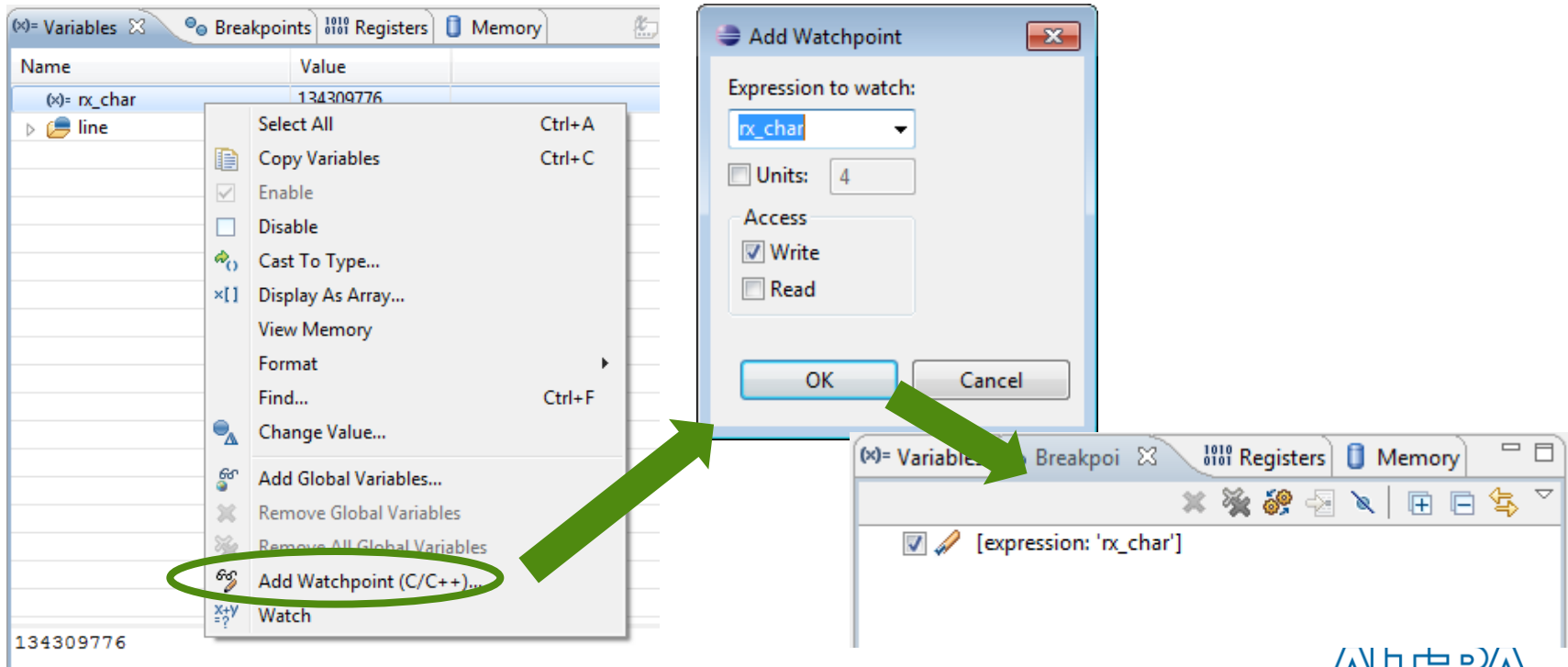


# Дополнительные окна отладчика



# Breakpoints и Watchpoints

- Breakpoint – точка останова. Срабатывает при выполнении инструкции по определенному адресу
- Watchpoint – контрольная точка. Отслеживается запись или чтения по определенному адресу
- Выводятся в окне **Breakpoints**



# Симулятор Nios II

- Осуществляет программное моделирование набора инструкций
- Используется симулятор фирмы Lauterbach (скачивается и устанавливается отдельно)
- Требуется указать .elf файл для отладки

