

Технологии **QNX**® для создания
высоконадежных систем автоматизации

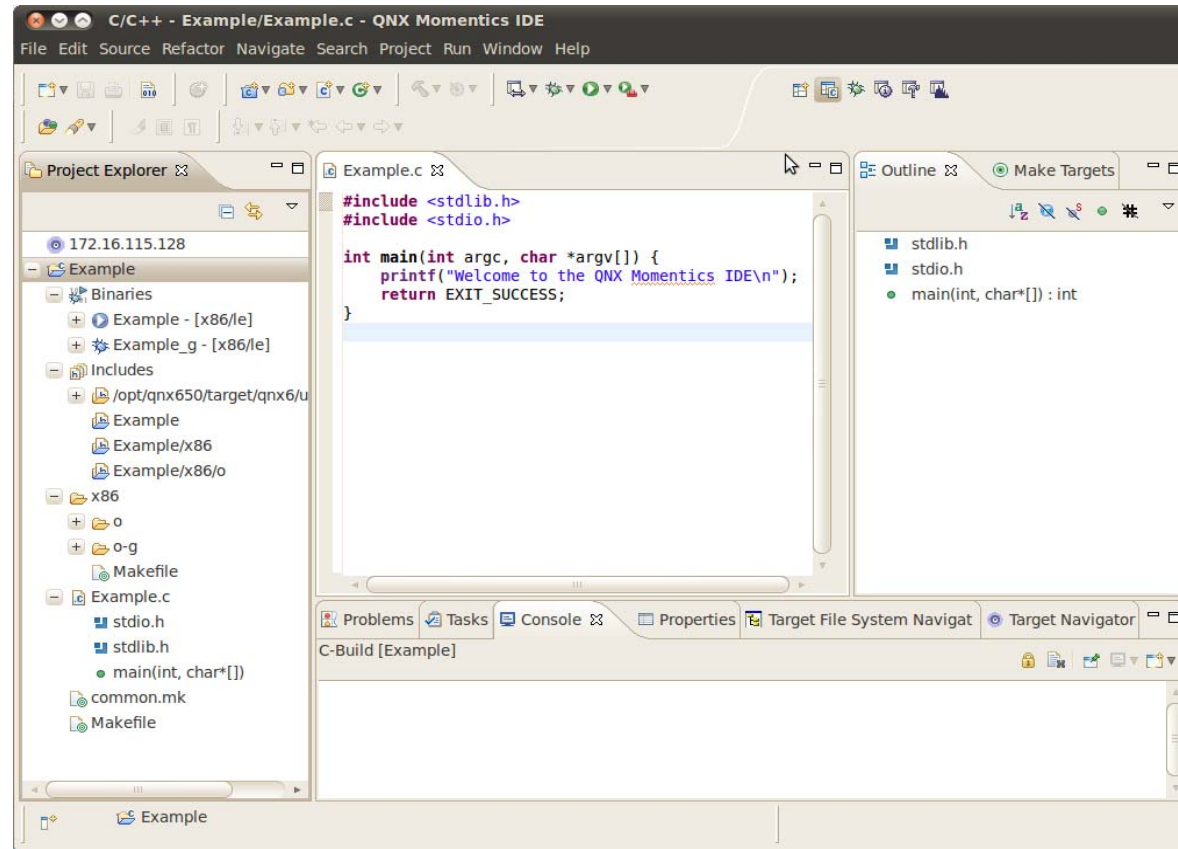


Глеб Крылов, SWD Software
Особенности разработки, отладки и анализа
приложений QNX.



QNX Momentics IDE

QNX Momentics IDE — среда разработки, которая входит в состав комплекта разработчика QNX SDP.



Проекты (Projects)— являются контейнерами для рабочих файлов в среде разработки.

Содержат:

- файлы исходных кодов C/C++;
- заголовочные файлы;
- Makefile;
- и др. файлы в зависимости от типа проекта.

Мастер создания проекта — вызывается через главное меню (File > New > QNX C Project).

Необходимо задать:

- ✓ Имя проекта.
- ✓ Тип проекта: **приложение** / статическая / динамическая библиотека.
- ✓ Аппаратную платформу: **x86** / ARM / MIPS / PPC / SH
- ✓ Дополнительные опции по необходимости.

Компиляция и компоновка

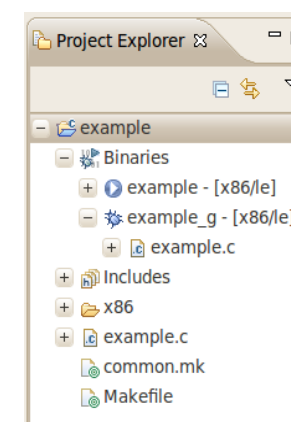
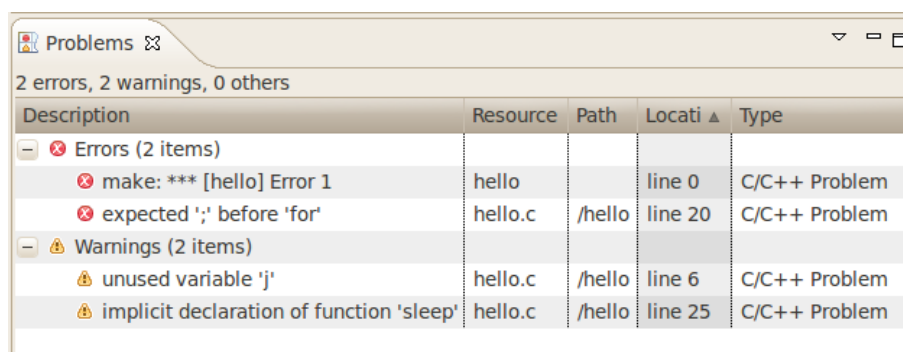
Компиляция и компоновка — выполняется с помощью пункта меню **Build Project**.

В результате получаем:

перечень ошибок и предупреждений
во вкладке **Problems**.

или

исполняемые файлы
в каталоге проекта
Binaries.

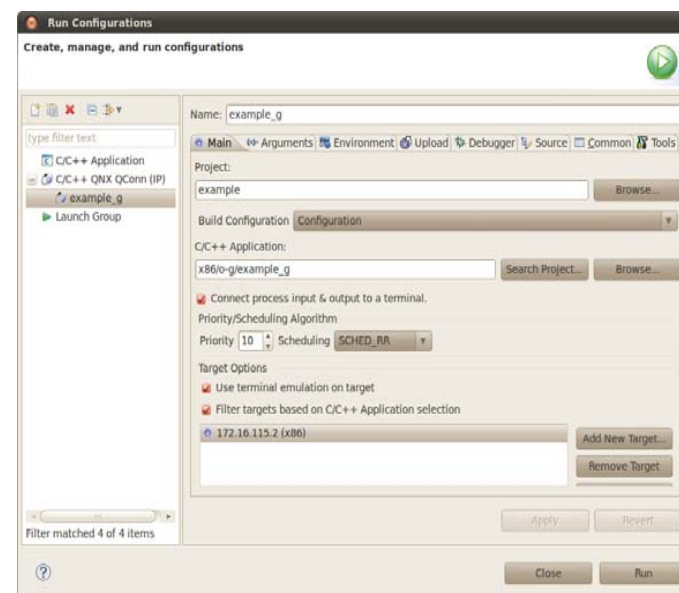


Запуск

Запуск приложений на целевой системе из IDE, выполняется с помощью **мастера конфигурации запуска (Run Configuration)**.

Необходимо задать:

- ✓ Тип конфигурации:
C/C++ QNX Qconn (IP)
- ✓ Имя конфигурации.
- ✓ Проект и исполняемый файл.
- ✓ Целевую систему.
- ✓ Дополнительные параметры запуска по необходимости.



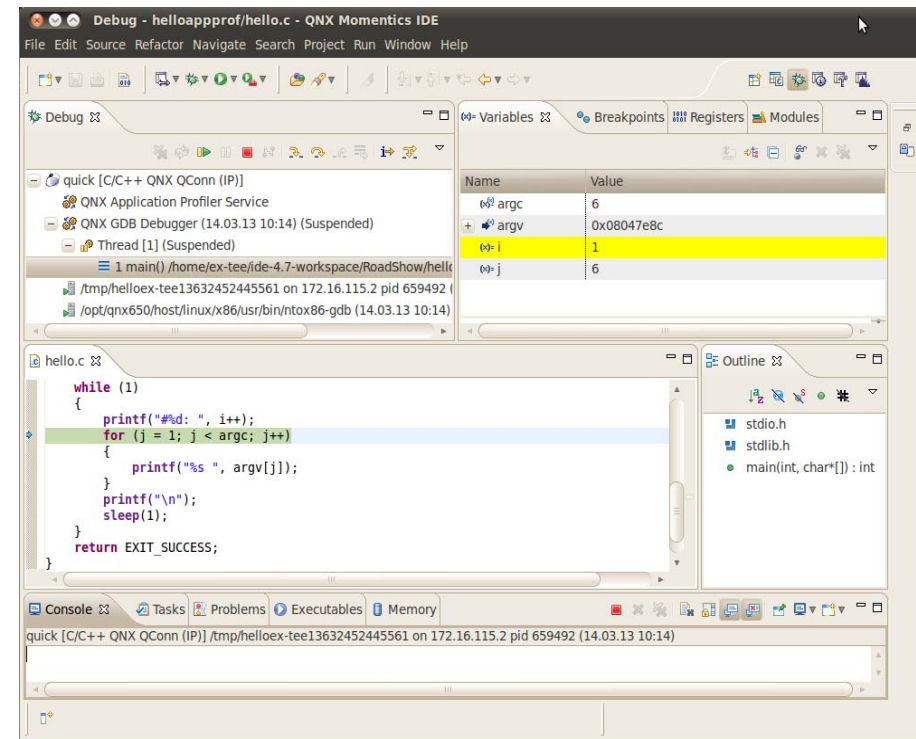
Настройки будут сохранены, для последующих запусков.

Debug perspective — предоставляет удобный набор инструментов для проведения отладки.

Позволяет:

Отслеживать состояние переменных.

- ✓ Пошаговое исполнять код.
- ✓ Просматривать стек вызовов функций.
- ✓ Локализовать ошибки.



Возможна отладка:

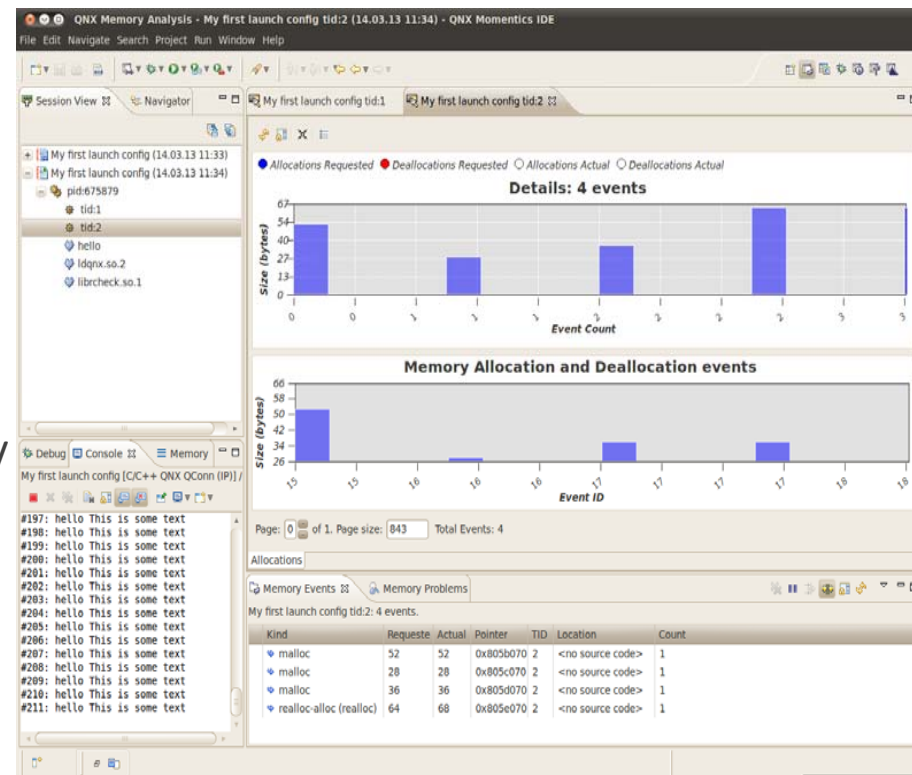
- однопоточных / многопоточных приложений;
- несколько взаимодействующих приложений.

Анализ памяти

QNX Memory Analysis perspective — предоставляет удобный набор инструментов для анализа использования памяти.

Позволяет выявить:

- ✓ Утечки памяти.
- ✓ Неэффективное использование.
- ✓ Неверное обращение с указателями на адреса областей памяти.
- ✓ Некорректное выделение / освобождение памяти.



Профилирование

QNX Application profiling perspective — предоставляет удобный набор инструментов для анализа производительности приложения.

Позволяет:

- ✓ Исследовать общую производительность.
- ✓ Обнаружить неэффективные области кода.

The screenshot displays the QNX Application Profiler interface. The top window shows the 'Profiler Sessions' list with 'appprofile_singlethread' selected. The 'Execution Time' window shows a 'Threads Tree' for 'ApProf Configuration #20'. The tree lists threads and functions with their deep and shallow times. The 'appprofile_singlethread.c' window shows the source code with execution statistics for various functions.

Name	Deep Time	Shallow Time
[100.00%] Thread 1	15,045 s	15,045 s
[<0.01%] <filtered>	1,000 ms	1,000 ms
[51.65%] func1	7,771 s	7,771 s
func1:78 (self)	0,173 s	0,173 s
func1:78 (self)	0,173 s	0,173 s
func1:78 (self)	0,209 s	0,209 s
func1:78 (self)	25,000 ms	25,000 ms
func1:78 (self)	0,252 s	0,252 s
func1:78 (self)	0,258 s	0,258 s
func1:78 (self)	0,292 s	0,292 s
func1:78 (self)	30,000 ms	30,000 ms

```
func1 (count);
func2 (count/2);
}
uint64_t
func1 (uint64_t n) /* 33.3% of
51.65% 7,771 s {
    volatile uint64_t a, b, c;
    if (optv)
        printf ("%s: Now in func1()\n", progname);
    for (a = 0; a < 2*n; a++) {} /* 50% of fi
157.49% 2,331 s
137.11% 1,972 s
    for (b = 0; b < n; b++) {} /* 25% of fi
    for (c = 0; c < n; c++) {} /* 25% of fi
    return 1;
}
uint64_t
func2 (uint64_t n) /* 66.7% of
48.34% 7,273 s {
```

Полученная информация помогает оптимизировать и повысить эффективность приложения.

Покрытие кода

QNX Code Coverage perspective — предоставляет удобный набор инструментов для анализа покрытия кода.

Позволяет выявить:

- ✓ Общий процент покрытия для файла исходного текста.
- ✓ Строки кода, которые исполняются полностью, частично или не исполняются, при работе приложения.

The screenshot displays the QNX Code Coverage perspective in the QNX Momentics IDE. The interface is divided into several panes:

- Code Coverage Sessions:** Shows a tree view of coverage data for 'example1_g [GCC Code Coverage] (18.03.13 17:10)'. Underneath, 'example1' is listed with an overall coverage of 83,33%. The 'appprofile_looper.c' file is expanded, showing coverage for various functions: 'app_prof_hdr' [0%], 'func1' [100%], 'func2' [100%], 'other_thread' [83,33%], 'dofuncs' [100%], 'main' [83,33%], and 'options' [77,78%].
- Code Editor:** Displays the source code for 'appprofile_looper.c'. The code is color-coded to indicate coverage: green for fully covered lines, yellow for partially covered lines, and red for lines not covered. The code includes a `main` function that prints the program name, sets up a signal handler for `SIGTERM`, sets priority to 5, and creates a thread.
- Properties:** A table showing coverage statistics for the selected file:

Property	Value
Coverage Info	
Lines Fully Covered	83,33% (40 lines)
Lines Not Covered	16,67% (8 lines)
Lines Partially Covered	0% (0 lines)
Total Block Coverage	83,78% (31/37 code blocks)
Total Line Coverage	83,33% (40/48 lines)
Info	

The status bar at the bottom indicates the current file path as `/home/ex-t...4/example1` and shows a button for 'Collection coverage data for example1'.

Заключение

Совокупность мощного инструментария визуальной среды разработки

QNX Momentics IDE и удаленной работы с целевой системой через протокол **Qconn**, дают широкие возможности по созданию:

- качественного,
- выверенного,
- надежного программного обеспечения на платформе **QNX Neutrino**.

Спасибо за внимание!

Подробная документация на сайте производителя:

www.qnx.com

Техническая поддержка компании SWD Software:

support.qnx@swd.ru



Глеб Крылов, инженер отдела сервисов

Тел: (812) 611-07-51, 611-07-59

www.swd.ru