



Инновационная компания Promwad

Применение среды Eclipse для  
разработки приложений на  
платформе uClinux/AD Blackfin

## Возможности Eclipse

---

- ➔ Свободно распространяемая
- ➔ С открытым исходным кодом
- ➔ Платформо-независимая
- ➔ Служит платформой для разработки модулей расширений (plug-in)
- ➔ Plug-in поддерживающие языки программирования C/C++, PHP, Python, Perl, и других
- ➔ Plug-in поддерживающие инструментарий сторонних производителей



# Окно Eclipse IDE

C/C++ - whetstone/whetstone.c - Eclipse

File Edit Source Refactor Navigate Search Run Project Window Help

```
whetstone.c
```

```
K = L * K - (L-J) * K;  
L = (L-K) * (K+J);  
E1[L-1] = J + K + L;  
E1[K-1] = J * K * L;  
}  
  
#ifdef PRINTOUT  
IF (JJ==II)POUT (N6,J,K,E1[1],E1[2],E1[3],E1[4]);  
#endif  
  
/*  
C  
C Module 7: Trigonometric functions  
C  
C */  
X = 0.5;  
Y = 0.5;  
  
for (I = 1; I <= N7; I++) {  
X = T * DATAN(T2*DSIN(X)*DCOS(X)/(DCOS(X+Y)+DCOS(X-Y)-1.0));  
Y = T * DATAN(T2*DSIN(Y)*DCOS(Y)/(DCOS(X+Y)+DCOS(X-Y)-1.0));  
}  
  
#ifdef PRINTOUT  
IF (JJ==II)POUT (N7,J,K,X,X,Y,Y);  
#endif  
  
/*  
C  
C Module 8: Procedure calls  
C  
C */  
X = 1.0;  
Y = 1.0;  
Z = 1.0;  
  
for (T = 1; T <= N9; T++)
```

Outline

- stdlib.h
- stdio.h
- string.h
- math.h
- time.h
- # DSIN
- # DCOS
- # DATAN
- # DLOG
- # DEXP
- # DSQRT
- # IF
- POUT(long, long, long, double, double, double, double) : void
- PA(double[]) : void
- P0(void) : void
- P3(double, double, double\*) : void
- # USAGE
- T : double
- T1 : double
- T2 : double
- E1 : double[]
- J : int
- K : int
- L : int
- main(int, char\*[]) : int
- PA(double[]) : void
- P0(void) : void
- P3(double, double, double\*) : void
- POUT(long, long, long, double, double, double, double) : void

Console

<terminated> whetstone [C/C++ Application] whetstone

Writable Smart Insert 272 : 1

## Необходимые компоненты

---

- ➔ Eclipse IDE for C/C++ Developers
- ➔ CDT C/C++ Development Tools
- ➔ Java Runtime Environment
- ➔ Blackfin toolchain
- ➔ Blackfin plug-in



# Быстрый просмотр функций и дерево вызовов

The screenshot displays the Eclipse IDE interface for a C/C++ project named 'mount/mount.c'. The main editor window shows the source code for the 'mount' utility, with the 'lock\_mtab()' function highlighted. The right-hand side of the IDE features a 'Call Hierarchy' view, which shows a tree of function calls starting from 'main(int, char \*\*)' in the file '/mount/mount.c'. The 'lock\_mtab()' function is currently selected and expanded, showing its internal calls to 'handler(int)', 'die(int, char \*)', 'setlkw\_timeout(int)', 'lstat(const char \*, stat \*)', '\_\_errno\_location()', 'open(const char \*, int)', 'vfprintf(FILE \*, const char \*, char \* (\*)\*)', 'fprintf(FILE \*, const char \*)', 'unlock\_mtab()', 'exit(int)', 'strerror(int)', 'alarm(unsigned int)', 'fcntl(int, int)', and 'close(int)'. The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, and the cursor is at line 516, column 8.

# Скрытие и автозавершение кода, навигация по коду

The screenshot displays the Eclipse IDE interface with the following components:

- Editor (Top Left):** Shows the source code for `mount.c`. The function `fstype(const char *device)` is selected. The code includes comments and function definitions like `swapped`, `tested`, `procfs`, `procclose`, `procopen`, `procnex`, and `is_in_proc`.
- Editor (Bottom Left):** Shows a different part of the code, a `switch` statement. A tooltip is visible over the `argc` variable, listing its type and other related variables like `argv`, `argc0`, and `argv0`. A message at the bottom of the tooltip says "Press 'Ctrl+Space' to show Template Proposals".
- Outline (Right):** A list of symbols from the file, including functions like `clear_string_opts`, `parse_string_opt`, `mount_quiet`, `print_one`, `print_all`, `parse_opt`, `parse_opts`, `fix_opts_string`, `swapped`, `magic_known`, `tested`, `fstype`, `procfs`, `procclose`, `procopen`, `procnex`, `is_in_proc`, `already`, `create_mtab`, `mountcount`, `mount5`, `try_mount5`, `try_mount_one`, `set_proc_name`, `mount_one`, `mounted`, `DISKMAJOR`, `mount_all`, `version`, `longopts`, `usage_string`, `usage`, and `main`.

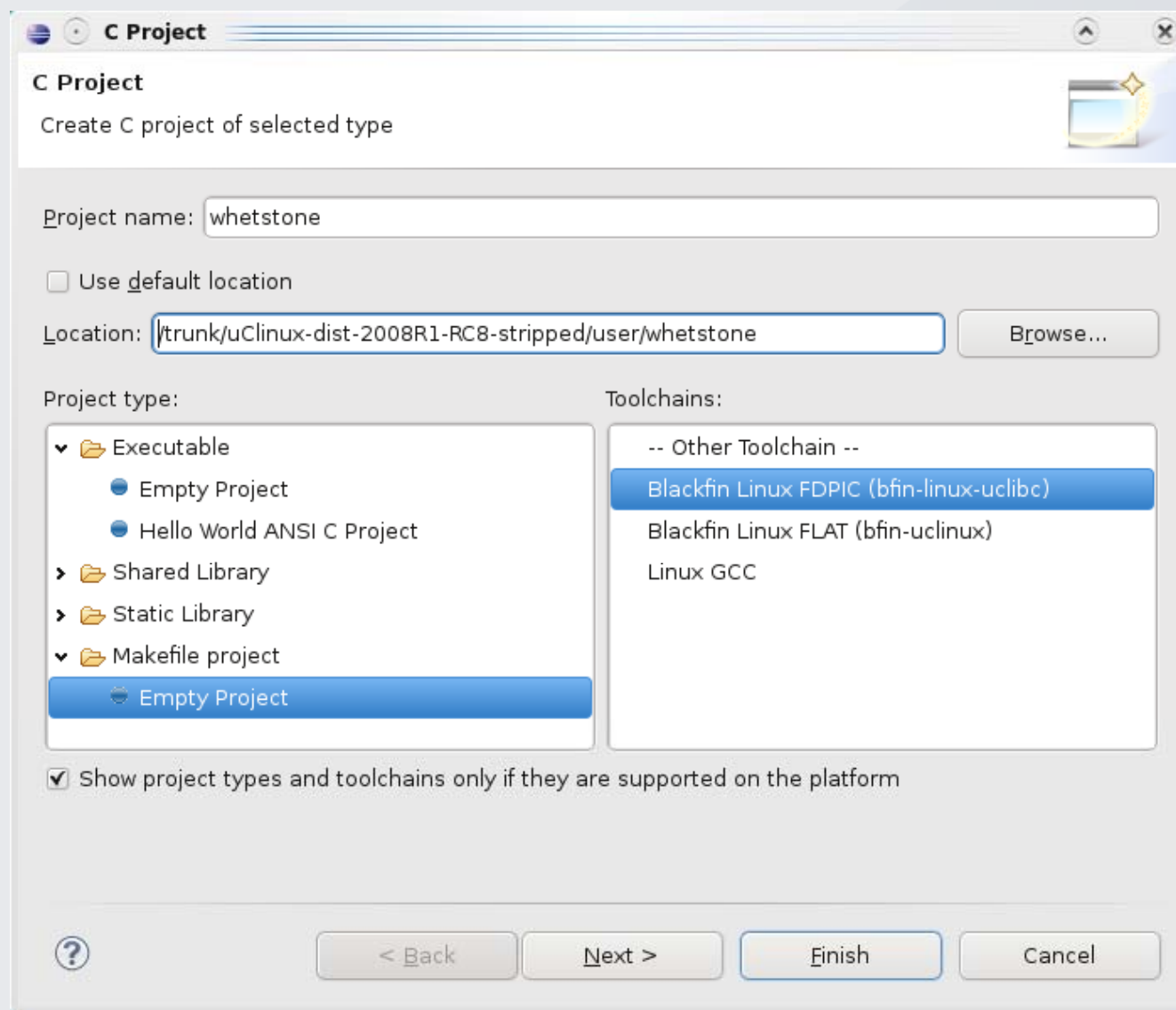
## Необходимые компоненты

---

- ➔ Eclipse IDE for C/C++ Developers
- ➔ CDT C/C++ Development Tools
- ➔ Java Runtime Environment
- ➔ Blackfin toolchain
- ➔ Blackfin plug-in



# Создания проекта



# Настройка проекта

## ➔ Директория сборки проекта

Build location

Build directory:

Workspace...

File system...

Variables...

## ➔ Опции сборки и очистки

Workbench Build Behavior

Workbench build type:

Make build target:

Build on resource save (Auto build)

Note: See Workbench automatic build preference

Build (Incremental build)

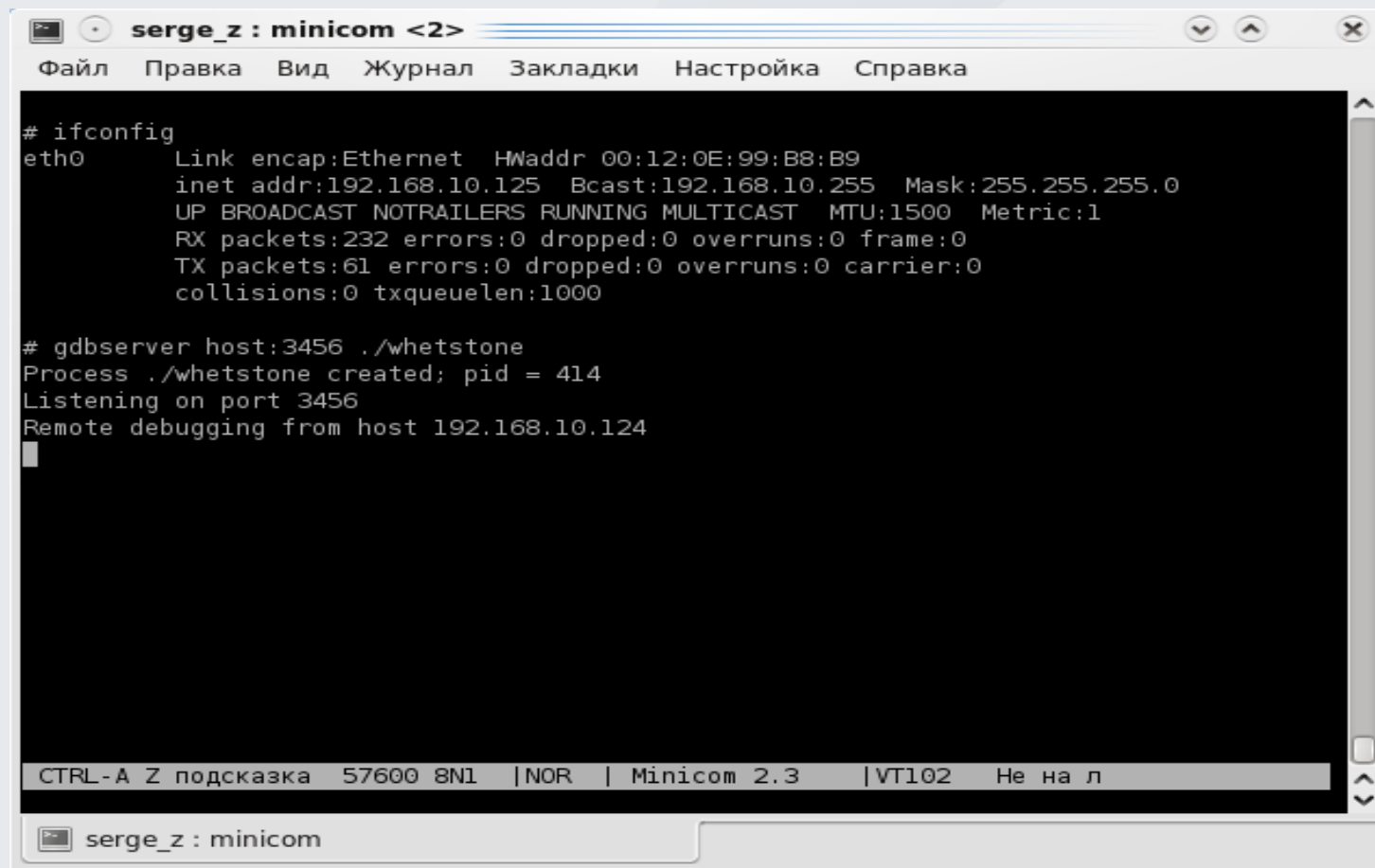
Clean

## Удаленная отладка с помощью gdbserver

- ➔ Добавление gdbserver в сборку uClinux-dist

```
[ ] gdbreplay (old)
[*] gdbserver (old)
[ ] grep
```

- ➔ Запуск gdbserver на целевой системе



```
serge_z : minicom <2>
Файл  Правка  Вид  Журнал  Закладки  Настройка  Справка

# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:12:0E:99:B8:B9
          inet addr:192.168.10.125  Bcast:192.168.10.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:232 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000

# gdbserver host:3456 ./whetstone
Process ./whetstone created; pid = 414
Listening on port 3456
Remote debugging from host 192.168.10.124
█

CTRL-A Z подсказка  57600 8N1  |NOR  | Minicom 2.3  |VT102  Не на л
serge_z : minicom
```

# Создание отладочной конфигурации

## ➔ Создание новой отладочной конфигурации

The screenshot shows the 'New Debug Configuration' dialog in an IDE. On the left, a tree view shows the configuration hierarchy: Blackfin Debug Targets > C/C++ Application > whetstone. The 'whetstone' configuration is selected. Below the tree, it says 'Filter matched 7 of 19 items'. On the right, the configuration details are shown. The 'Name' field contains 'whetstone'. The 'Project' field also contains 'whetstone' with a 'Browse...' button. The 'C/C++ Application' field contains 'whetstone' with a 'Search Project...' button and a 'Browse...' button. At the bottom, it says 'Using GDB (DSF) Remote System Process Launcher - [Select other...](#)' with 'Apply' and 'Revert' buttons.

## ➔ Выбор типа запуска GDB

This dialog allows you to specify which launcher to use when multiple launchers are available for a configuration and launch mode.

Use configuration specific settings [Change Workspace Settings...](#)

Launchers:

Standard Create Process Launcher  
GDB (DSF) Remote System Process Launcher  
GDB (DSF) Create Process Launcher

# Настройка отладочной конфигурации

Name: whetstone

type filter text

- Blackfin Debug Targets
- ▼ C/C++ Application
  - whetstone
  - C/C++ Attach to Application
  - C/C++ Postmortem Debugger
  - GDB Hardware Debugging
  - Launch Group

Filter matched 7 of 19 items

Main Debugger Source Common

Debugger: gdbserver Debugger

Stop on startup at: main

**Debugger Options**

Main Shared Libraries Connection

GDB debugger: bfin-linux-uclibc-gdb Browse...

GDB command file: Browse...

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

- Non-stop mode (Note: Requires non-stop GDB)
- Enable Reverse Debugging at startup (Note: Requires Reverse GDB)

Using GDB (DSF) Remote System Process Launcher - [Select other...](#)

Apply Revert

# Настройка отладочной конфигурации

type filter text

- Blackfin Debug Targets
  - C/C++ Application
    - whetstone**
    - C/C++ Attach to Application
    - C/C++ Postmortem Debugger
    - GDB Hardware Debugging
    - Launch Group

Filter matched 7 of 19 items

Name: whetstone

Main Debugger Source Common

Debugger: gdbserver Debugger

Stop on startup at: main

**Debugger Options**

Main Shared Libraries Connection

Type: TCP

Host name or IP address: 192.168.10.125

Port number: 3456

Using GDB (DSF) Remote System Process Launcher - [Select other...](#)

Apply Revert

# Окно Eclipse при отладке приложения

The screenshot displays the Eclipse IDE interface during a debug session. The main editor window shows the source code of `whetstone.c`, with a `for` loop highlighted. The `Variables` view on the right shows the current values of `X`, `Y`, and `I`. The `Registers` view at the bottom right shows the state of the CPU registers. The `Debug` console at the bottom left shows the execution flow, and the `Breakpoints` view shows three breakpoints set in `whetstone.c`.

```
whetstone.c
C
C  Module 7: Trigonometric functions
C
C  */
X = 0.5;
Y = 0.5;
for (I = 1; I <= N7; I++) {
    X = T * DATAN(T2*DSIN(X)*DCOS(X)/(DCOS(X+Y)+DCOS(X-Y)-1.0));
    Y = T * DATAN(T2*DSIN(Y)*DCOS(Y)/(DCOS(X+Y)+DCOS(X-Y)-1.0));
}
#ifdef PRINTOUT
    IF (JJ==II) POUT (N7,J,K,X,X,Y,Y);
#endif
/*
C
C  Module 8: Procedure calls
C
C  */
X = 1.0;
Y = 1.0;
Z = 1.0;
for (I = 1; I <= N8; I++)
    P3(X,Y,&Z);
#ifdef PRINTOUT
    IF (JJ==II) POUT (N8,J,K,X,Y,Z,Z);
#endif
```

Variables

```
X = 0.49991743156697077
Y = 0.49990334966669581
I = 3
```

Registers

Name	Value	Description
General Registers		General Purpose ar
r0	-1635865336	
r1	1071644266	
r2	0	
r3	0	
r4	1072692947	

Debug Console

```
whetstone [C/C++ Application]
whetstone
  Thread [1] (Suspended : Step)
    main() at /home/serge_z/projects/mediabox/trunk/uClinux-dist-2008R1-RC8-stripped/user/whetstone.c:168
    gdb
    whetstone
```

Breakpoints

- whetstone.c [line: 168]
- whetstone.c [line: 217]
- whetstone.c [line: 266]

# Окно Eclipse при отладке приложения

The screenshot displays the Eclipse IDE interface during a debug session. The main editor shows the source code of 'whetstone.c'. The code includes a for loop that is currently being executed, with the following lines highlighted:

```
for (I = 1; I <= N7; I++) {  
    X = T * DATAN(T2*DSIN(X)*DCOS(X)/(DCOS(X+Y)+DCOS(X-Y)-1.0));  
    Y = T * DATAN(T2*DSIN(Y)*DCOS(Y)/(DCOS(X+Y)+DCOS(X-Y)-1.0));  
}
```

The Variables view on the right shows the following data:

Name	Type	Value
argc	int	1
argv	char **	0x131fea4
I	long int	1
N1	long int	0
N2	long int	12000
N3	long int	14000
N4	long int	345000
N6	long int	210000
N7	long int	32000
N8	long int	899000

The Registers view at the bottom shows assembly instructions and register values, including:

```
00679532: main+2390 R0 = 0x0 (X); /* R0=  
00679534: main+2392 R1 = 0x1ff (X); /* R1=  
00679538: main+2396 R1 <= 0x15;  
0067953a: main+2398 [FP -0x28] = R0;  
0067953c: main+2400 [FP -0x24] = R1;  
268 for (I = 1; I <= N7; I++) {  
0067953e: main+2402 R0 = 0x1 (X); /* R0=  
00679540: main+2404 [FP -0x7c] = R0;  
00679542: main+2406 JUMP.S 0x0679742 <main+2918>;  
269 X = T * DATAN(T2*DSIN(X)*DCOS(X)/(DCO  
00679544: main+2408 R0 = [FP -0x30];  
00679546: main+2410 R1 = [FP -0x2c];  
00679548: main+2412 P3 = [FP + -0x8c];  
0067954c: main+2416 CALL 0x067889e <_init+158>;  
00679550: main+2420 R3 = R0;  
00679552: main+2422 R4 = R1;  
00679554: main+2424 P2 = [FP + -0x8c];  
00679558: main+2428 R0 = [P2 + 0x3c];  
0067955c: main+2432 P2 = R0;  
0067955e: main+2434 R0 = [P2];  
00679560: main+2436 R1 = [P2 + 0x4];  
00679562: main+2438 R2 = R1;  
00679564: main+2440 [SP + 0xc] = R2;
```

# Ваши вопросы?

<http://www.promwad.com>

<http://eclipse.org>

<http://blackfin.uclinux.org>

<http://www.analog.com>

