GCC & Cross Compilation

Isfahan Linux User Group

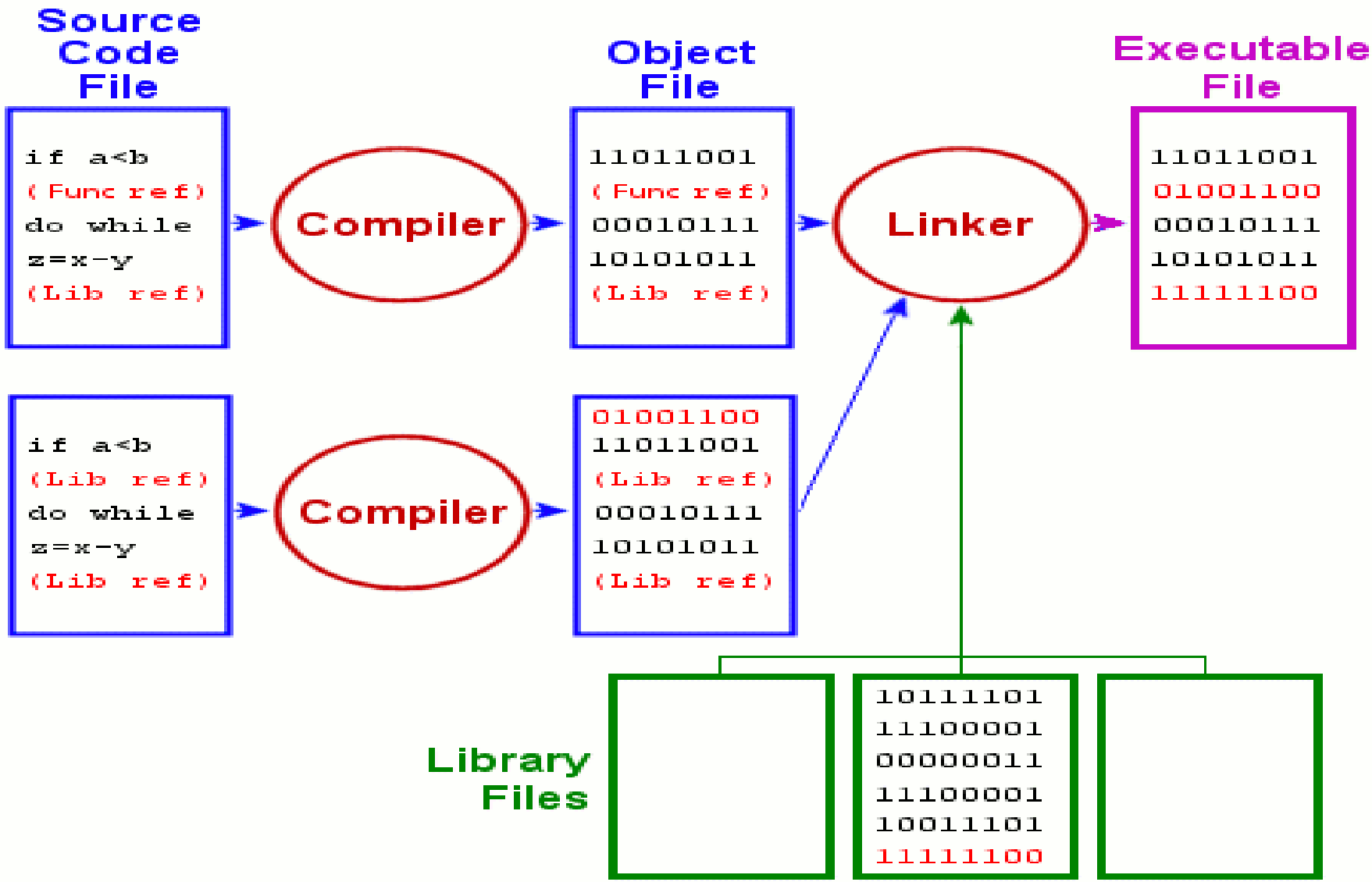
GCC & Cross Compilation

- GNU Compiler Collection (GCC)
- Compile
- Cross Compile
- Avr-gcc (WinAVR – eclipse)
- GNU ARM

GNU Compiler Collection (GCC)

- Microsoft Windows, Linux, BSD, Mac
- C, C++, Objective-C, Objective-C++, Fortran, Java, Ada, Go
- Multi-architecture support ۶۰ (AVR, ARM, PowerPC, ...)
- Optimization (O۰~۳, OS)
- IDE (Eclipse)
- -std=C۹۹ - gnu۹۹- C۱۱-gnu۱۱
- -nostartfiles

Compile


Source Code File
if a<b
(Func ref)
do while
z=x-y
(Lib ref)
Compiler
Object File
11011001
(Func ref)
00010111
10101011
(Lib ref)
Linker
Executable File
11011001
01001100
00010111
10101011
11111100
if a<b
(Lib ref)
do while
z=x-y
(Lib ref)
Compiler
01001100
11011001
(Lib ref)
00010111
10101011
(Lib ref)
Library Files
10111101
11100001
00000011
11100001
10011101
11111100

Cross Compile

- Architecture.
  - ٨٠x٨۶ (uP , CISC)
  - Arm (uC, RISC)
  - Avr (uC, RISC)
  - Power PC (uC, RISC)
- Toolchain
  - Avr-gcc
  - GNU ARM
- Uses of cross compilers
  - Embedded computers (ex. Windows Mobile, Android)
  - Compiling for multiple machines

Programmers Notepad 2 - [snap.c]

File Edit View Tools Window Help

C / C++ | dx | Find

index.php | snap.h | snap.c

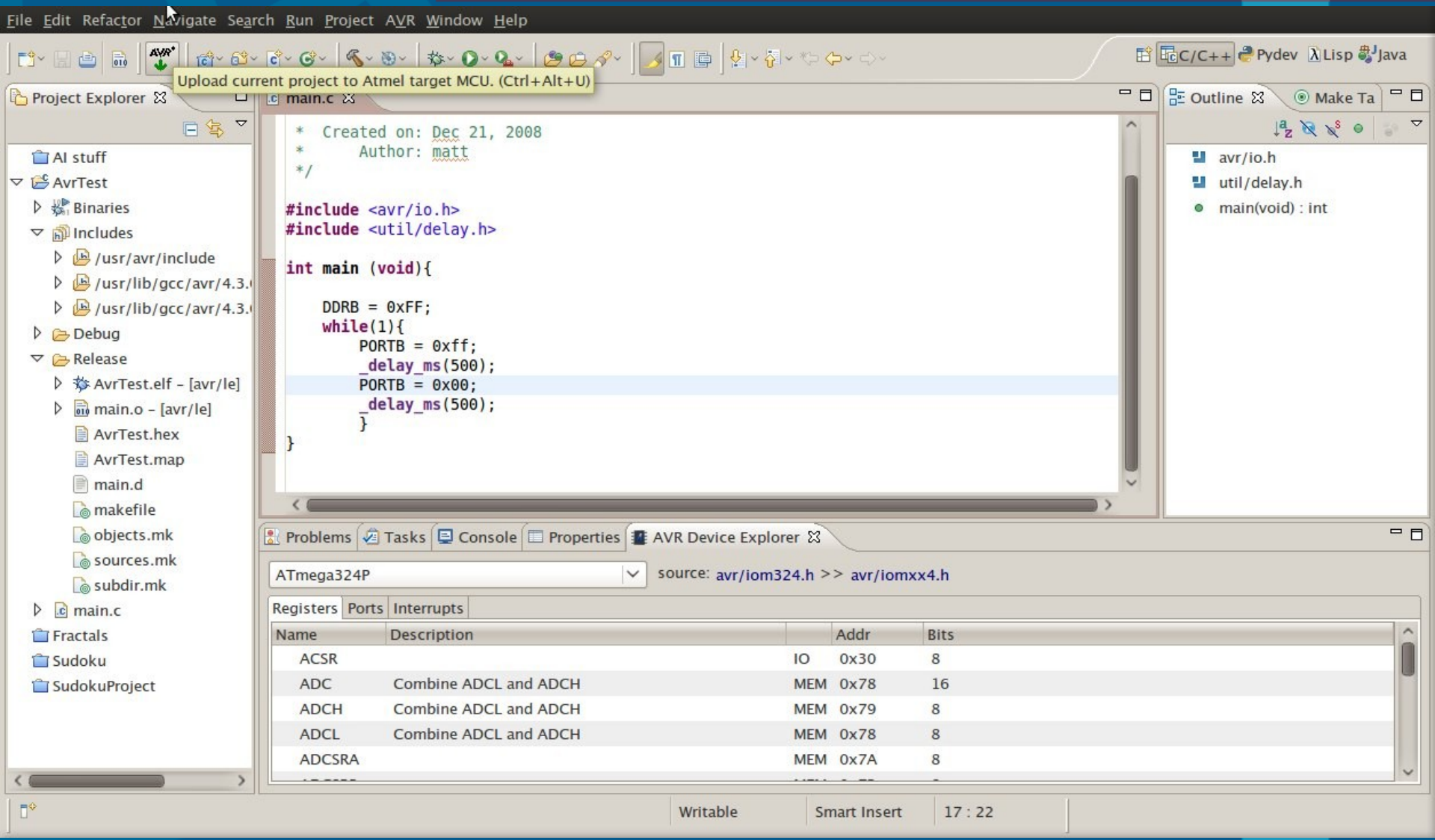
```
49
50  // ------------------------------------------------------------------------
51  // Setzt den Status zurück
52
53  void snap_reset(tSNAPStatus *status)
54  {
55      status->status = S_SYNCBYTE;
56      status->_pos = 0;
57      status->_p = NULL;
58  }
59
60  // ------------------------------------------------------------------------
61  // Wertet das naechste Byte des Datenstroms aus
62
63  uint8_t decode_snap_stream(tSNAPStatus *snap, uint8_t data)
64  {
65      uint8_t status = snap->status;
66
67      switch (status) {
68          case S_SYNCBYTE:
69              if (data == SYNC) {
70                  snap->status = S_HDB2;
71              }
72              break;
73
74          case S_HDB2:
75              snap->msg->hdb2 = data;
76              snap->status = S_HDB1;
```

Output

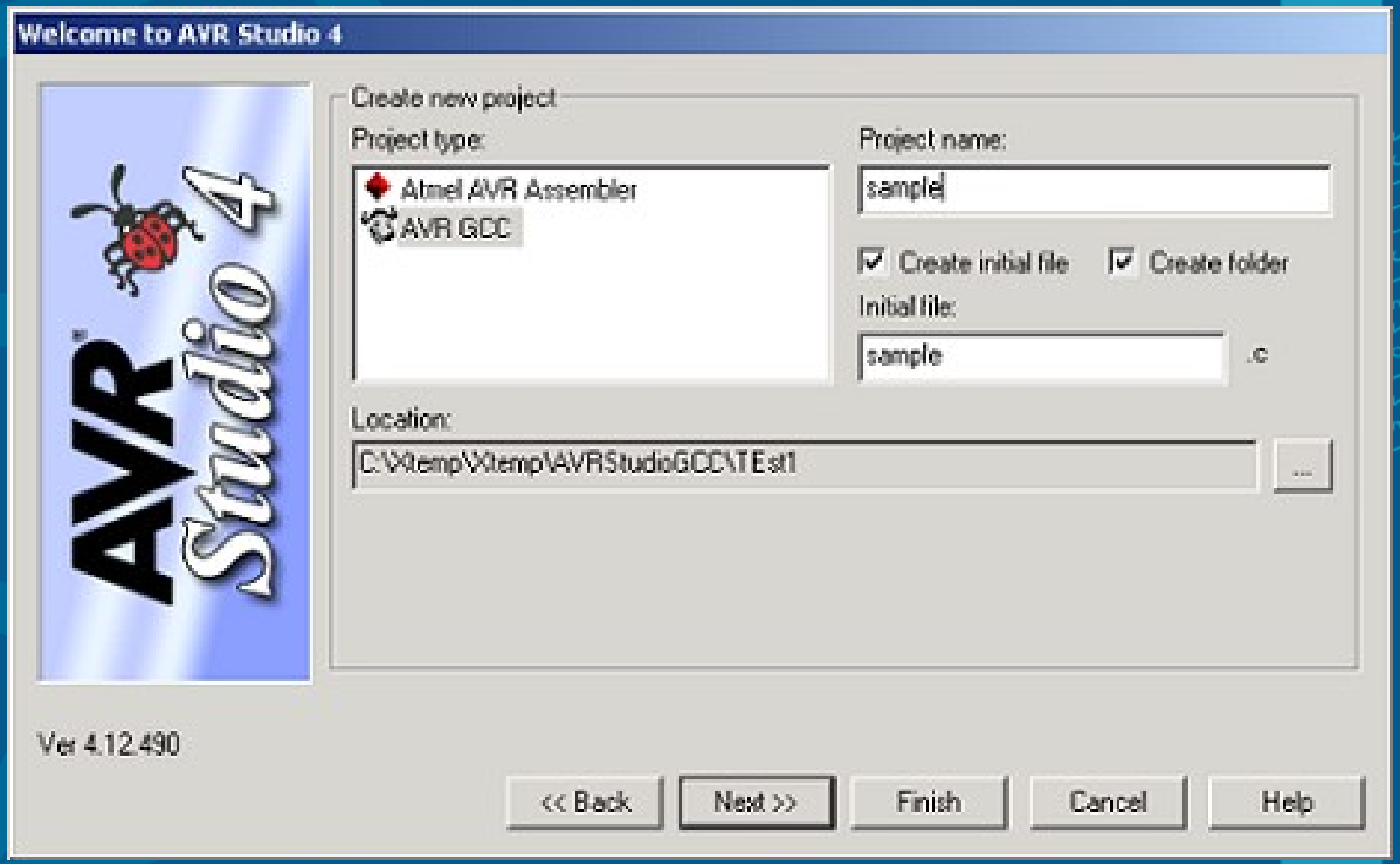
```
Compiling C: snap.c
avr-gcc -c -mmcu=atmega168 -I. -gdwarf-2 -DF_CPU=8000000UL -Os -funsigned-char -funsigned-bitfields -fpack-struct -fshor

Linking: snap.elf
avr-gcc -mmcu=atmega168 -I. -gdwarf-2 -DF_CPU=8000000UL -Os -funsigned-char -funsigned-bitfields -fpack-struct -fshort-e

Creating load file for Flash: snap.hex
avr-objcopy -O ihex -R .eeprom snap.elf snap.hex
```

[62:1] : 154 | ANSI | CR+LF | INS | Ready

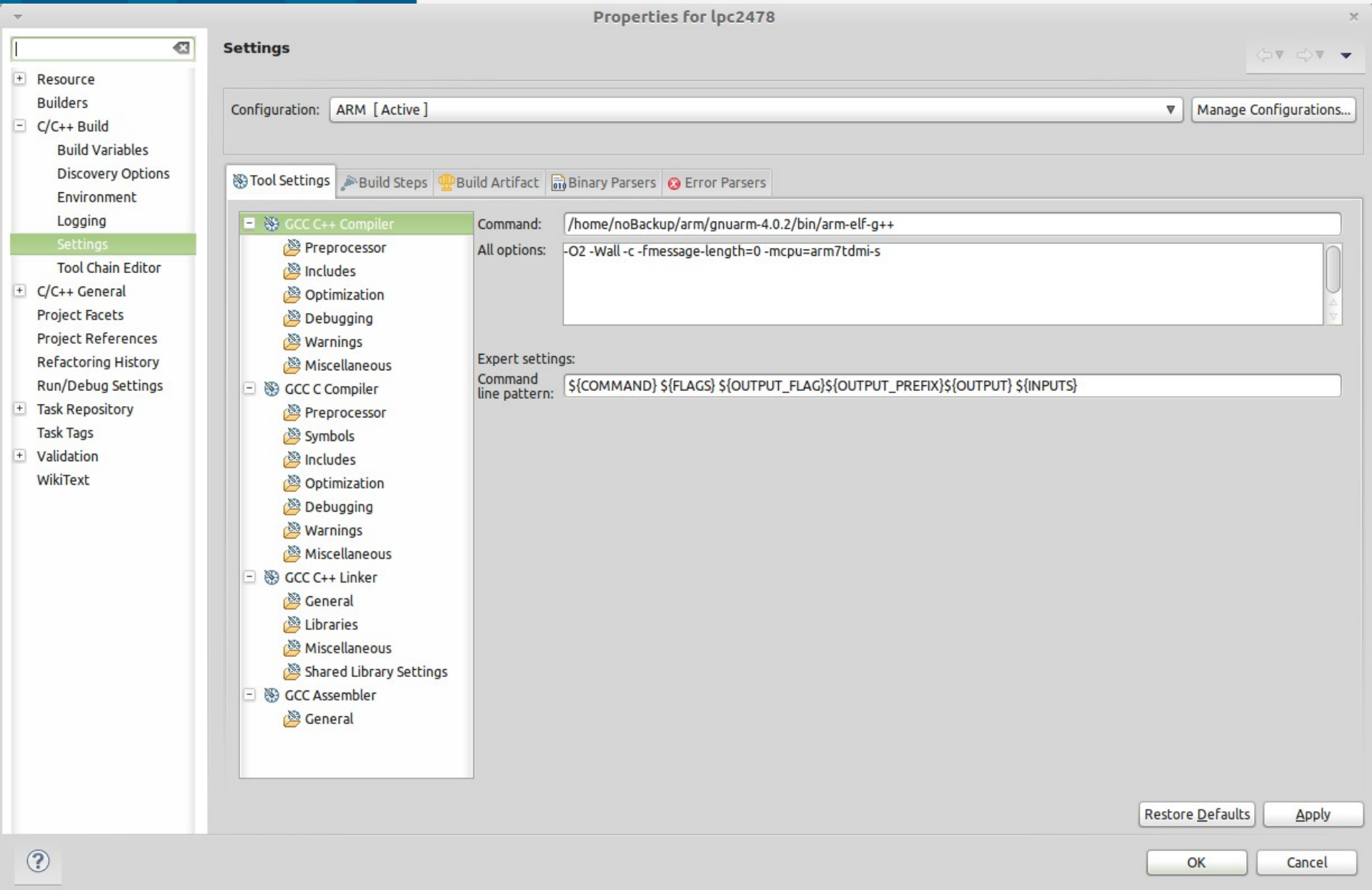
Avr-Eclipse


File Edit Refactor Navigate Search Run Project AVR Window Help
Upload current project to Atmel target MCU. (Ctrl+Alt+U)
C/C++ Pydev Lisp Java
Project Explorer
main.c
Outline
Make Ta
AI stuff
AvrTest
Binaries
Includes
/usr/avr/include
/usr/lib/gcc/avr/4.3.
/usr/lib/gcc/avr/4.3.
Debug
Release
AvrTest.elf - [avr/le]
main.o - [avr/le]
AvrTest.hex
AvrTest.map
main.d
makefile
objects.mk
sources.mk
subdir.mk
main.c
Fractals
Sudoku
SudokuProject
* Created on: Dec 21, 2008
* Author: matt
*/
#include <avr/io.h>
#include <util/delay.h>
int main (void){
DDRB = 0xFF;
while(1){
PORTB = 0xff;
_delay_ms(500);
PORTB = 0x00;
_delay_ms(500);
}
}
avr/io.h
util/delay.h
main(void) : int
Problems Tasks Console Properties AVR Device Explorer
ATmega324P
source: avr/iom324.h >> avr/iomxx4.h
Registers Ports Interrupts
Name Description Addr Bits
ACSR IO 0x30 8
ADC Combine ADCL and ADCH MEM 0x78 16
ADCH Combine ADCL and ADCH MEM 0x79 8
ADCL Combine ADCL and ADCH MEM 0x78 8
ADCSRA MEM 0x7A 8
Writable
Smart Insert
17 : 22

Avr Stdio


Welcome to AVR Studio 4
AVR Studio 4
Create new project
Project type:
Atmel AVR Assembler
AVR GCC
Project name:
sample
Create initial file
Create folder
Initial file:
sample
.c
Location:
C:\Xtemp\Xtemp\AVRStudioGCC\TEst1
...
Ver 4.12.490
<< Back
Next >>
Finish
Cancel
Help

GNU ARM


Properties for lpc2478
Settings
Resource
Builders
C/C++ Build
Build Variables
Discovery Options
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
Project Facets
Project References
Refactoring History
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText
Configuration: ARM [Active]
Manage Configurations...
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
GCC C++ Compiler
Preprocessor
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C Compiler
Preprocessor
Symbols
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C++ Linker
General
Libraries
Miscellaneous
Shared Library Settings
GCC Assembler
General
Command:
/home/noBackup/arm/gnuarm-4.0.2/bin/arm-elf-g++
All options:
-O2 -Wall -c -fmessage-length=0 -mcpu=arm7tdmi-s
Expert settings:
Command line pattern:
${COMMAND} ${FLAGS} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT} ${INPUTS}
Restore Defaults
Apply
OK
Cancel