GadgetPC & MINI-MAX/ARM9

Single Board Computers

C Programming Guide for Linux

Document Revision: 1.02

Date: 3 February, 2010

BiPOM Electronics, Inc.

16301 Blue Ridge Road, Missouri City, Texas 77489
Telephone: 1-713-283-9970   Fax: 1-281-416-2806
E-mail: info@bipom.com
Web: www.bipom.com

1 Overview

BiPOM's ARM Development System (GNUARM) consists of:

- Micro-IDE – A Windows IDE for microcontroller development
- GCC (GNUARM) C Compiler, Linker
- Downloaders for various boards
- Project examples for various platforms

BiPOM's version of GNUARM supports the following ARM microcontrollers:

- AT91SAM7 series from ATMEL
- AT91SAM9 series from ATMEL (these are used on GadgetPC and MINI-MAX/ARM9 series)
- LPC2000 series from NXP
- LPC2300 series from NXP

Micro-IDE is a versatile Integrated Development System (IDE) that has project manager, syntax-colored multi-window editor, terminal, tools, utilities and many other features.

Various downloaders and toolkits plug into Micro-IDE to simplify microcontroller program development in various languages such as Assembly, C and BASIC.

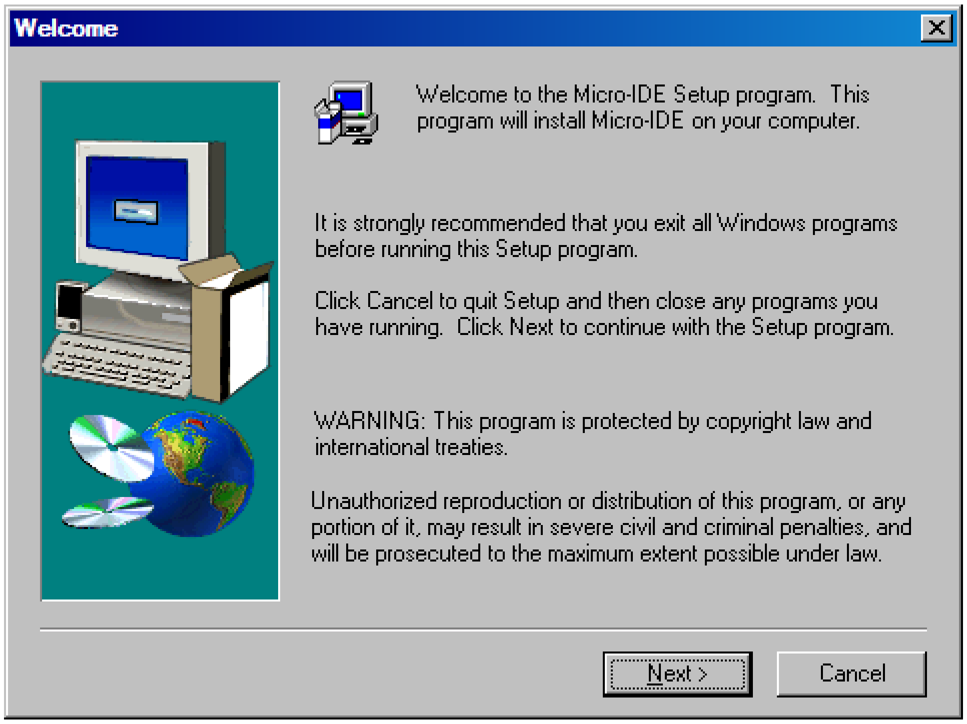
Software Installation

Download ARM Development System from:

http://www.bipom.com/armdev_down.php

Open the zip file armdev.zip and install by running setup.exe.

A Welcome screen will appear:



Welcome to the Micro-IDE Setup program. This program will install Micro-IDE on your computer.

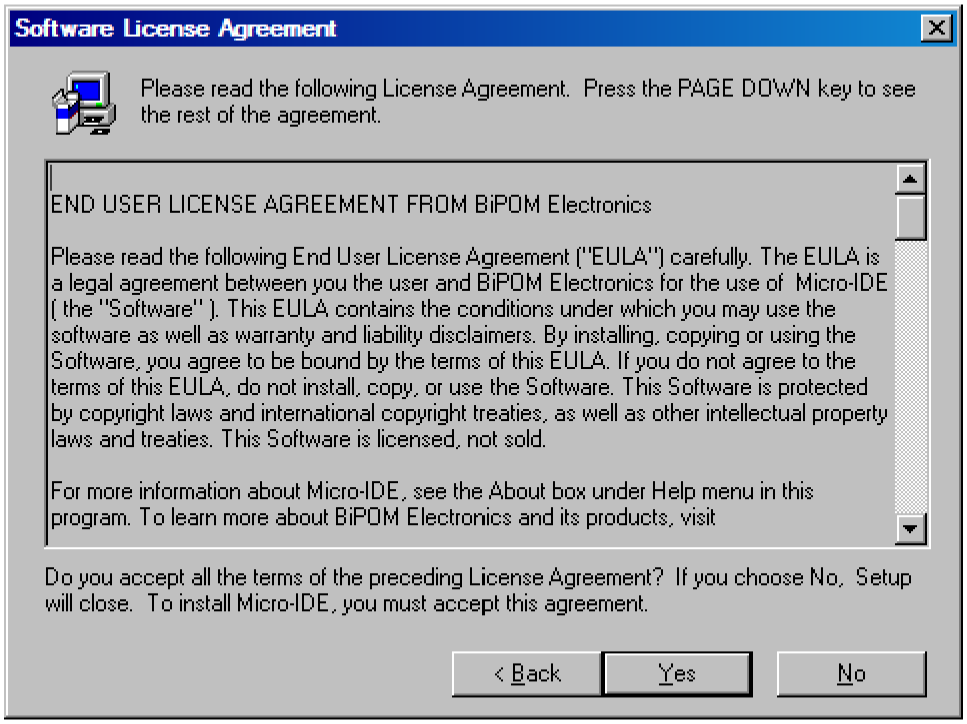
It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Click on Next. End User Agreement will appear:



Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

END USER LICENSE AGREEMENT FROM BiPOM Electronics

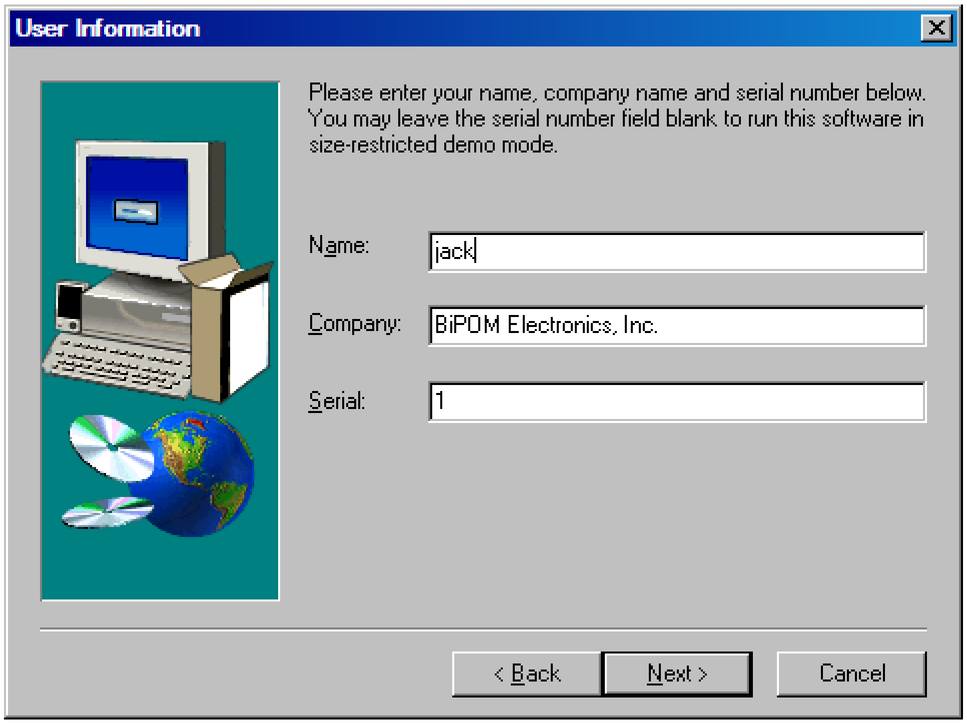
Please read the following End User License Agreement ("EULA") carefully. The EULA is a legal agreement between you the user and BiPOM Electronics for the use of Micro-IDE (the "Software"). This EULA contains the conditions under which you may use the software as well as warranty and liability disclaimers. By installing, copying or using the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install, copy, or use the Software. This Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. This Software is licensed, not sold.

For more information about Micro-IDE, see the About box under Help menu in this program. To learn more about BiPOM Electronics and its products, visit

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Micro-IDE, you must accept this agreement.

Please read the agreement and click Yes if you wish to continue with installation.

Enter your name, company (if applicable) and serial number:



Please enter your name, company name and serial number below. You may leave the serial number field blank to run this software in size-restricted demo mode.

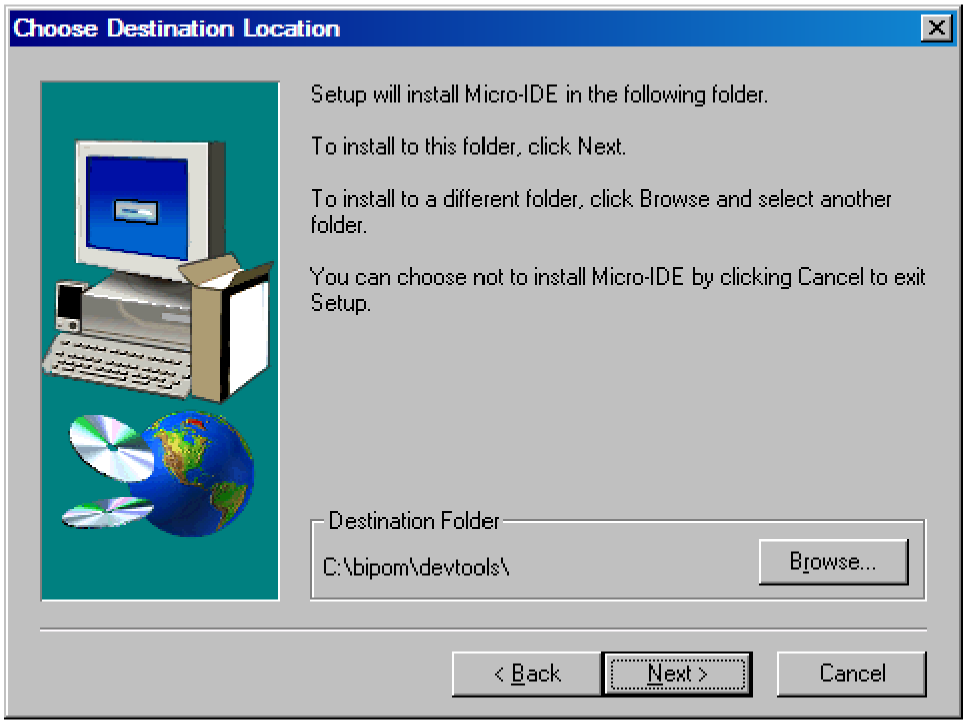
Name: jack

Company: BiPOM Electronics, Inc.

Serial: 1

You can enter any Serial Number here in the Serial field. Click Next to continue.

Select the disk location where the software will be installed. Using the default location of c:\bipom\devtools is recommended:



Setup will install Micro-IDE in the following folder.

To install to this folder, click Next.

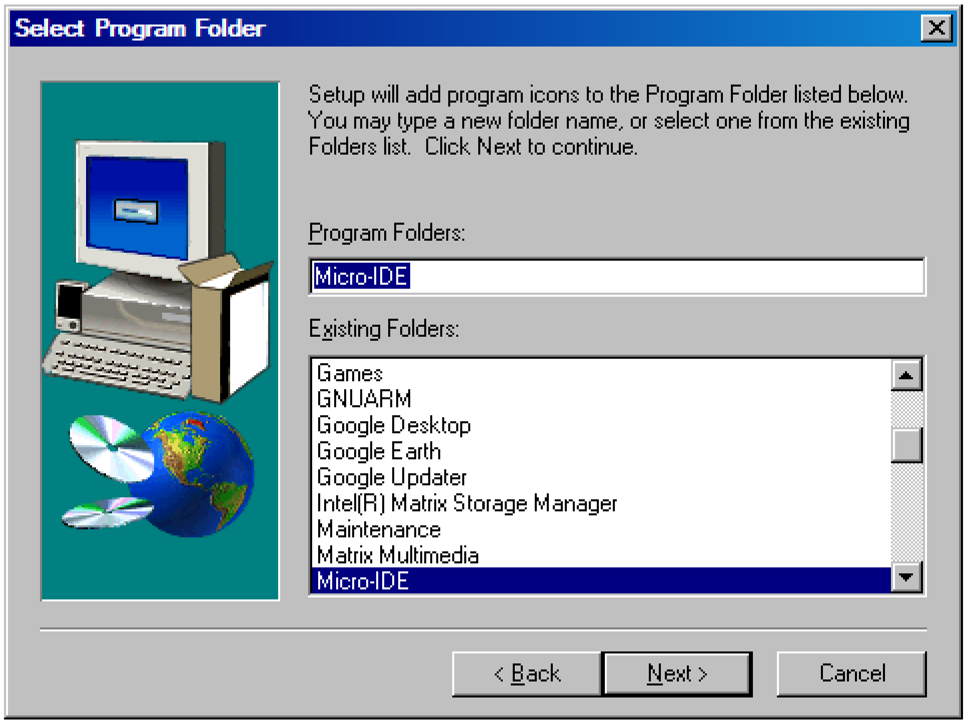
To install to a different folder, click Browse and select another folder.

You can choose not to install Micro-IDE by clicking Cancel to exit Setup.

Destination Folder

C:\bipom\devtools\

Click Next. Select the Program Folder where the icons for Micro-IDE will be installed.
Default selection is **Micro-IDE** folder.



Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

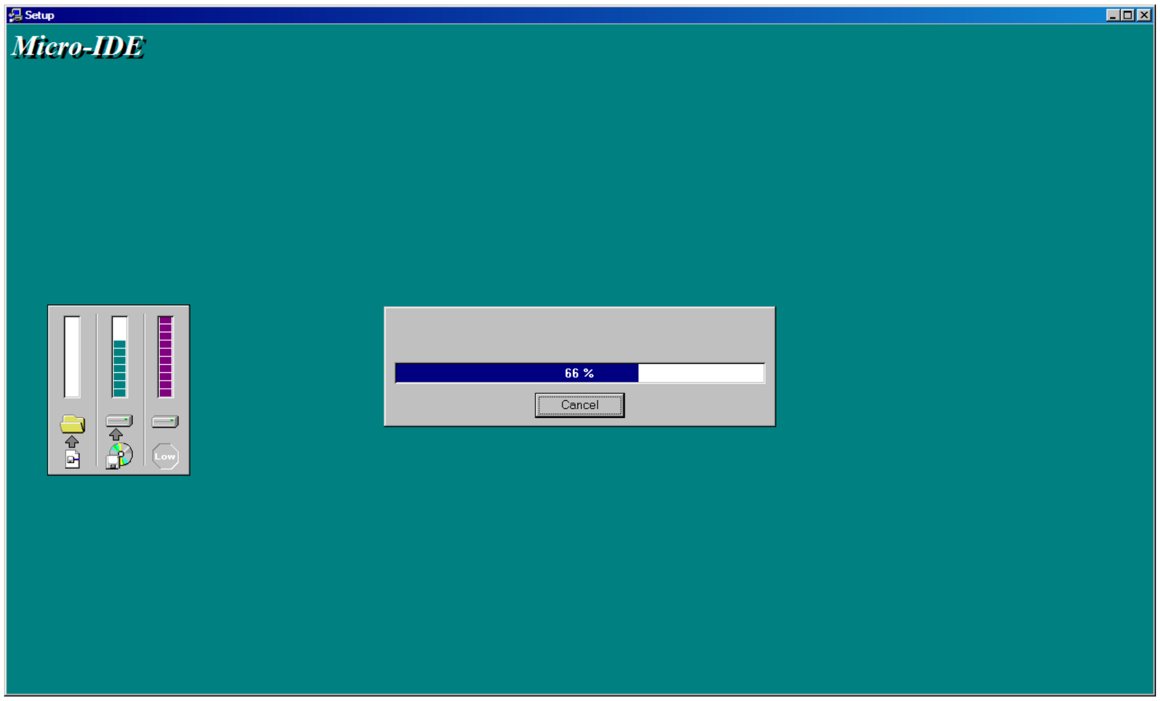
Program Folders:

Micro-IDE

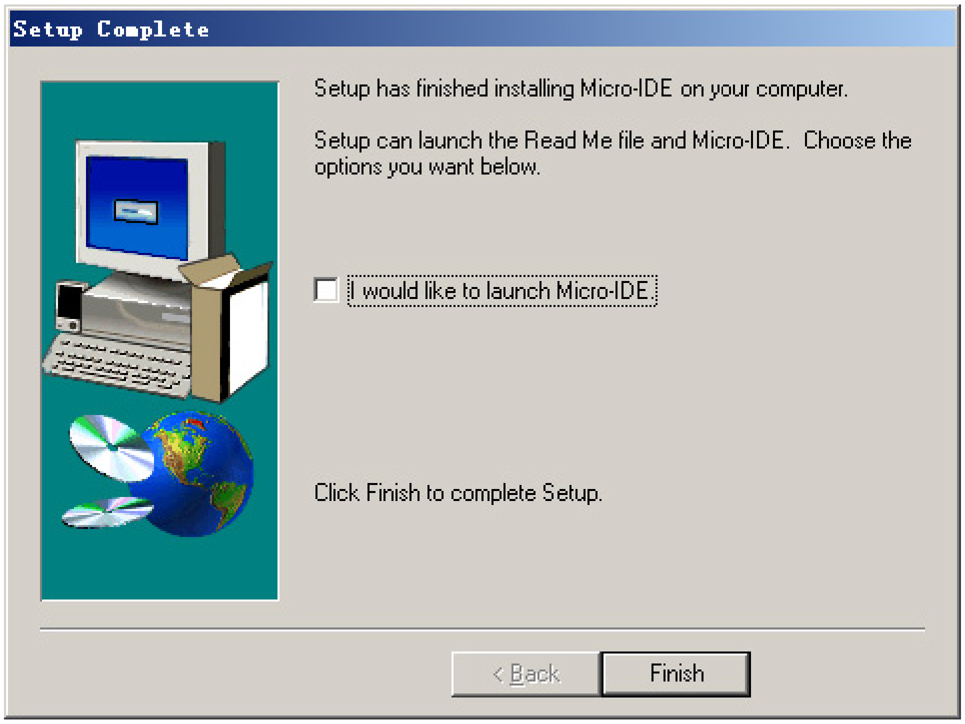
Existing Folders:

Games
GNUARM
Google Desktop
Google Earth
Google Updater
Intel(R) Matrix Storage Manager
Maintenance
Matrix Multimedia
Micro-IDE

Click Next. Micro-IDE will be installed and you will see the progress:



When the installation is complete, you will be given an option to start Micro-IDE now:



Setup has finished installing Micro-IDE on your computer.

Setup can launch the Read Me file and Micro-IDE. Choose the options you want below.

☐ I would like to launch Micro-IDE.

Click Finish to complete Setup.

Uncheck the option and click Finish button to finish the installation.

1.1.1 Installing the GNU ARM C Compiler

For creating programs that will run under Linux on GadgetPC or MINI-MAX/ARM9 series boards, Code Sourcery G++ Lite should be used.

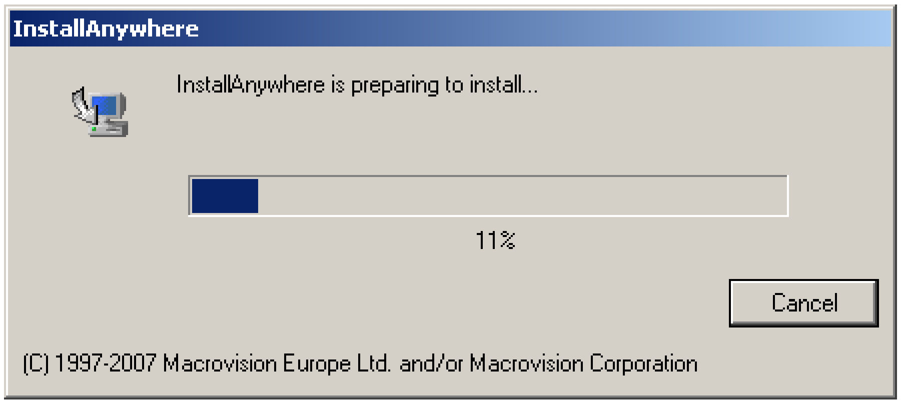
Download the latest version of **IA32 Windows Installer** for **Sourcery G++ Lite 2009q3-67 for ARM GNU/Linux** from:

www.codesourcery.com

Typically, this package has a long name such as:

arm-2009q3-67-arm-none-linux-gnueabi.exe

After the download is complete, run this package. You will see a progress bar that shows the progress of file extraction:

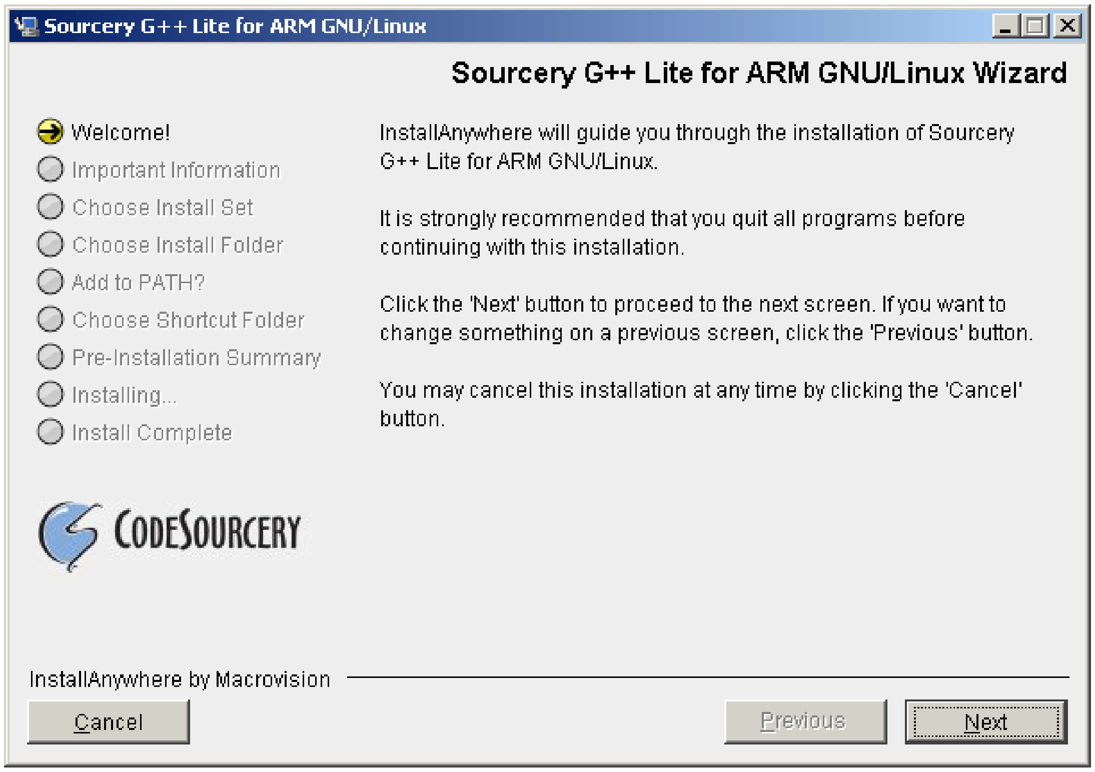


InstallAnywhere is preparing to install...

11%

(C) 1997-2007 Macrovision Europe Ltd. and/or Macrovision Corporation

After the file extraction is complete, a Welcome dialog will appear:


Sourcery G++ Lite for ARM GNU/Linux

Sourcery G++ Lite for ARM GNU/Linux Wizard

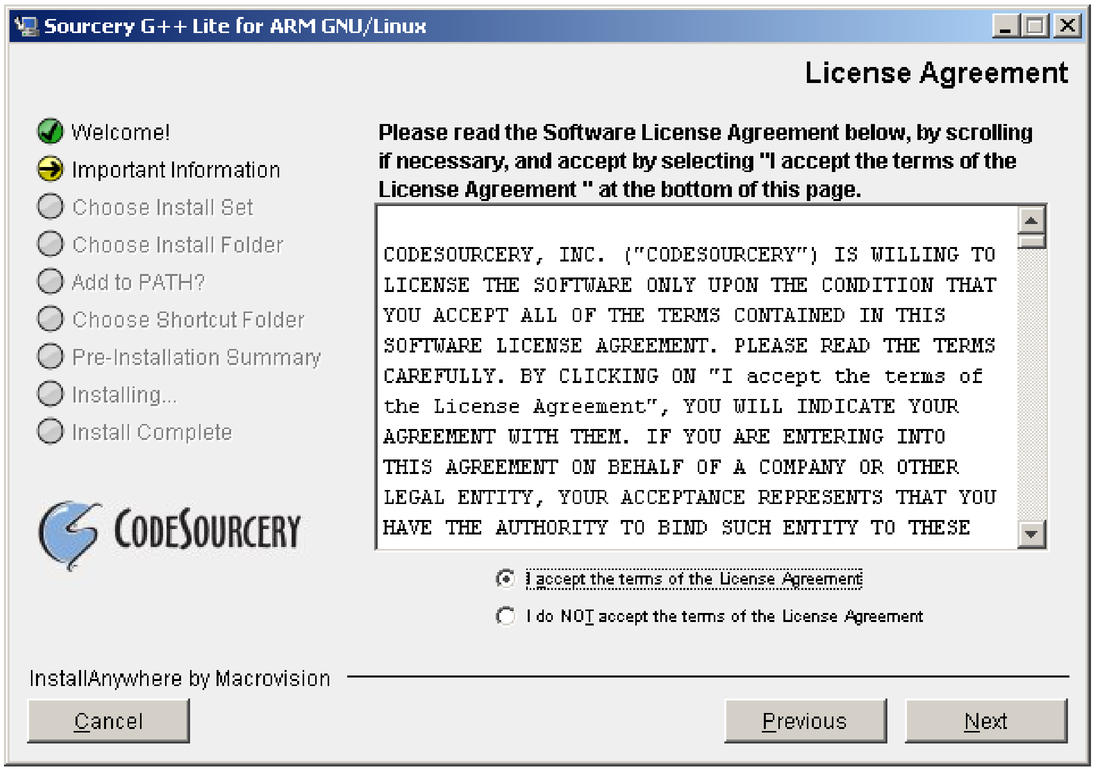
InstallAnywhere will guide you through the installation of Sourcery G++ Lite for ARM GNU/Linux.

It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

Click Next. A License Agreement will appear:


Sourcery G++ Lite for ARM GNU/Linux

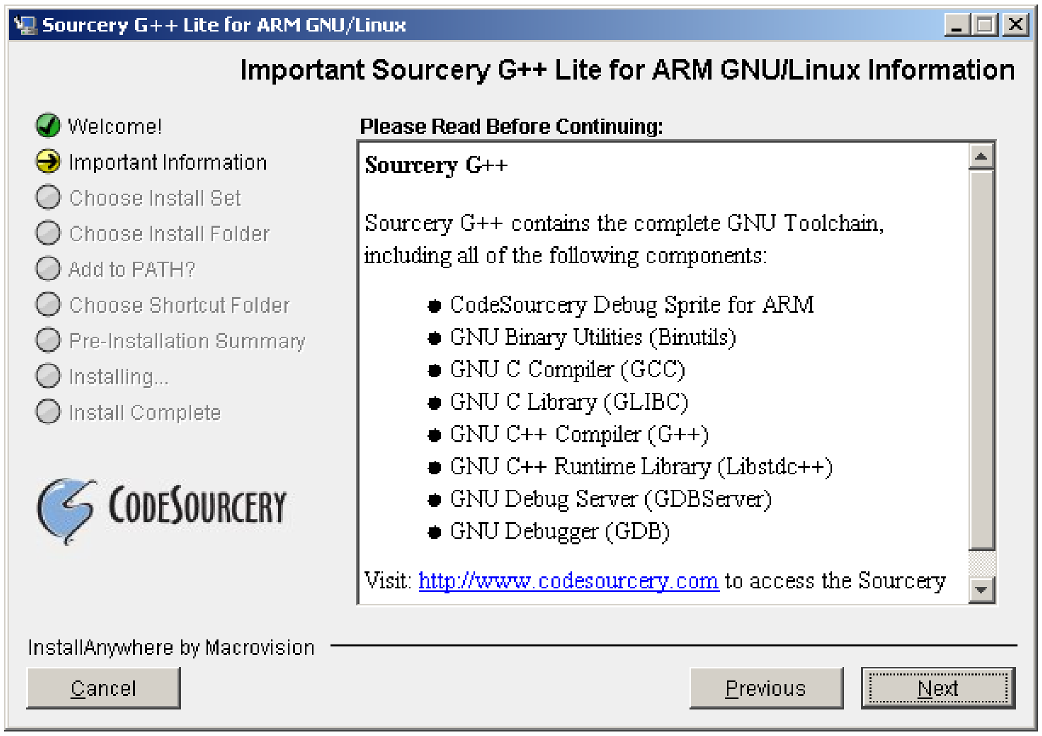
License Agreement

Please read the Software License Agreement below, by scrolling if necessary, and accept by selecting "I accept the terms of the License Agreement " at the bottom of this page.

```
CODESOURCERY, INC. ("CODESOURCERY") IS WILLING TO
LICENSE THE SOFTWARE ONLY UPON THE CONDITION THAT
YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS
SOFTWARE LICENSE AGREEMENT. PLEASE READ THE TERMS
CAREFULLY. BY CLICKING ON "I accept the terms of
the License Agreement", YOU WILL INDICATE YOUR
AGREEMENT WITH THEM. IF YOU ARE ENTERING INTO
THIS AGREEMENT ON BEHALF OF A COMPANY OR OTHER
LEGAL ENTITY, YOUR ACCEPTANCE REPRESENTS THAT YOU
HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE
```

[✓ I accept the terms of the License Agreement] I do NOT accept the terms of the License Agreement

Check the “I accept the terms of the License Agreement” option and click Next. Important Information window will appear:



Important Sourcery G++ Lite for ARM GNU/Linux Information

Please Read Before Continuing:

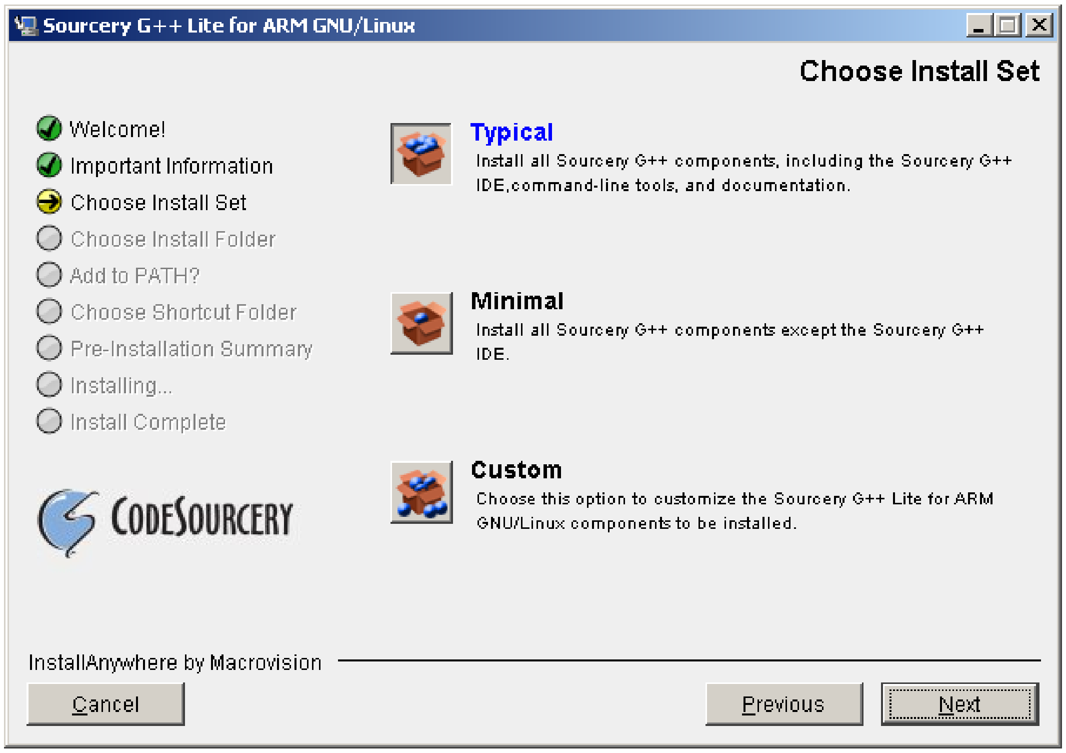
Sourcery G++

Sourcery G++ contains the complete GNU Toolchain, including all of the following components:

- CodeSourcery Debug Sprite for ARM
- GNU Binary Utilities (Binutils)
- GNU C Compiler (GCC)
- GNU C Library (GLIBC)
- GNU C++ Compiler (G++)
- GNU C++ Runtime Library (Libstdc++)
- GNU Debug Server (GDBServer)
- GNU Debugger (GDB)

Visit: http://www.codesourcery.com to access the Sourcery

Click Next. Choose Install Set as Typical:



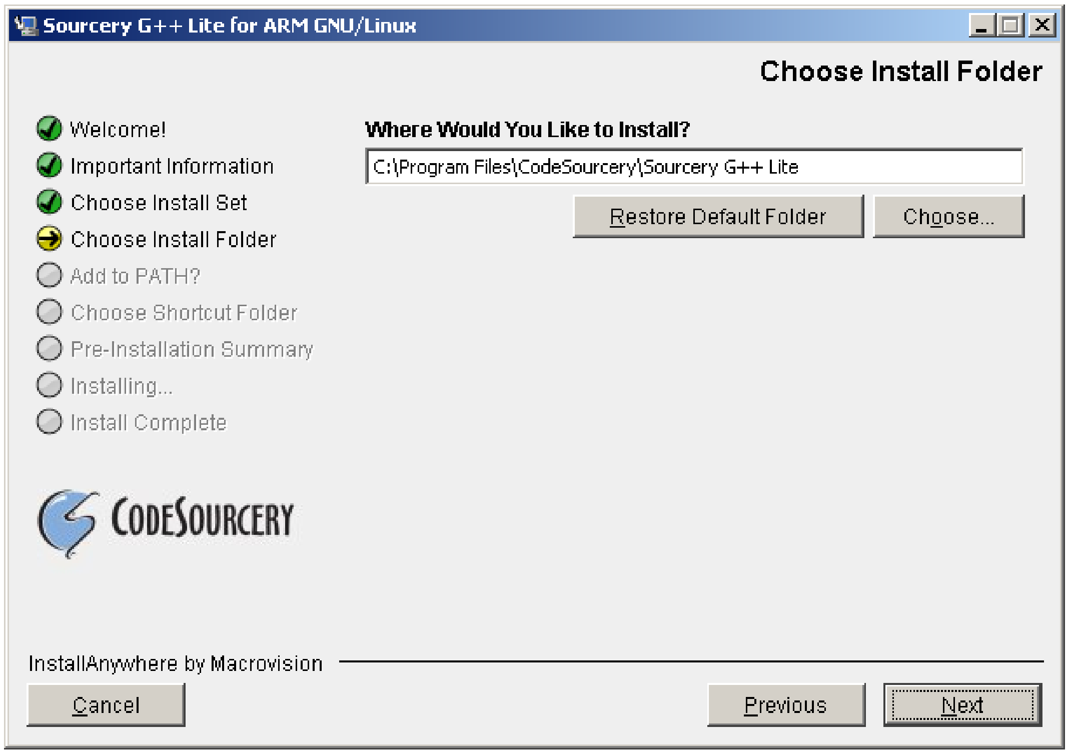
Choose Install Set

Typical
Install all Sourcery G++ components, including the Sourcery G++ IDE,command-line tools, and documentation.

Minimal
Install all Sourcery G++ components except the Sourcery G++ IDE.

Custom
Choose this option to customize the Sourcery G++ Lite for ARM GNU/Linux components to be installed.

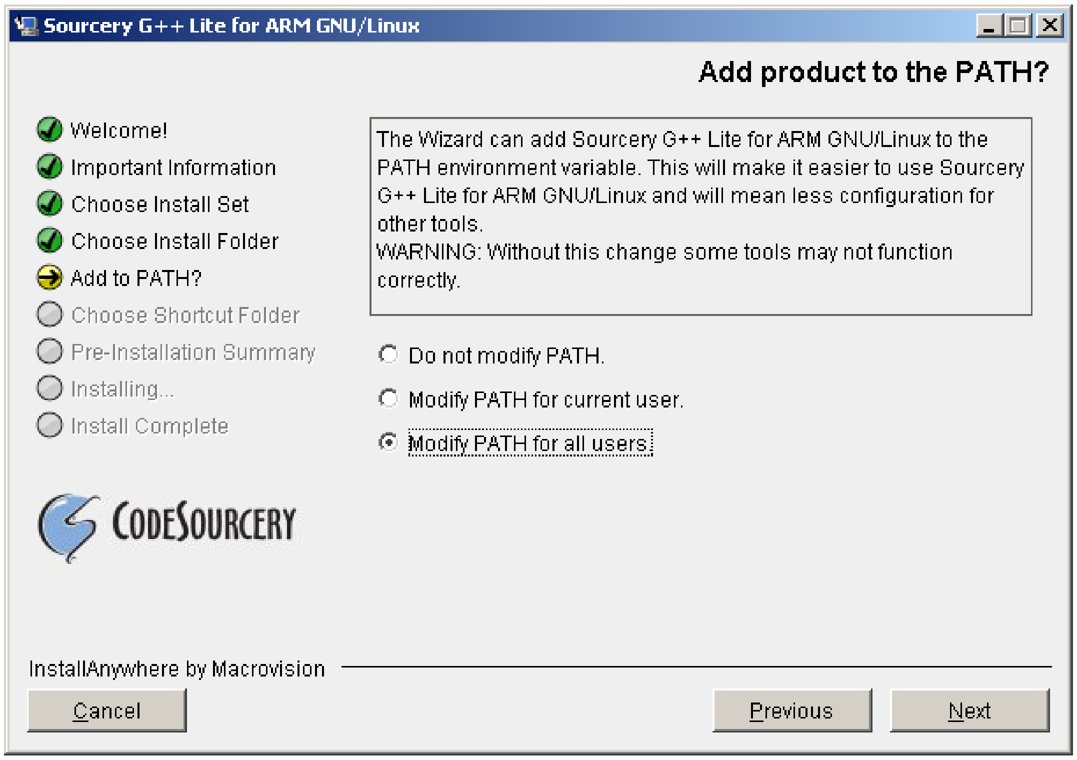
Click Next. Install Folder window will appear. Use the default location:



Where Would You Like to Install?

C:\Program Files\CodeSourcery\Sourcery G++ Lite

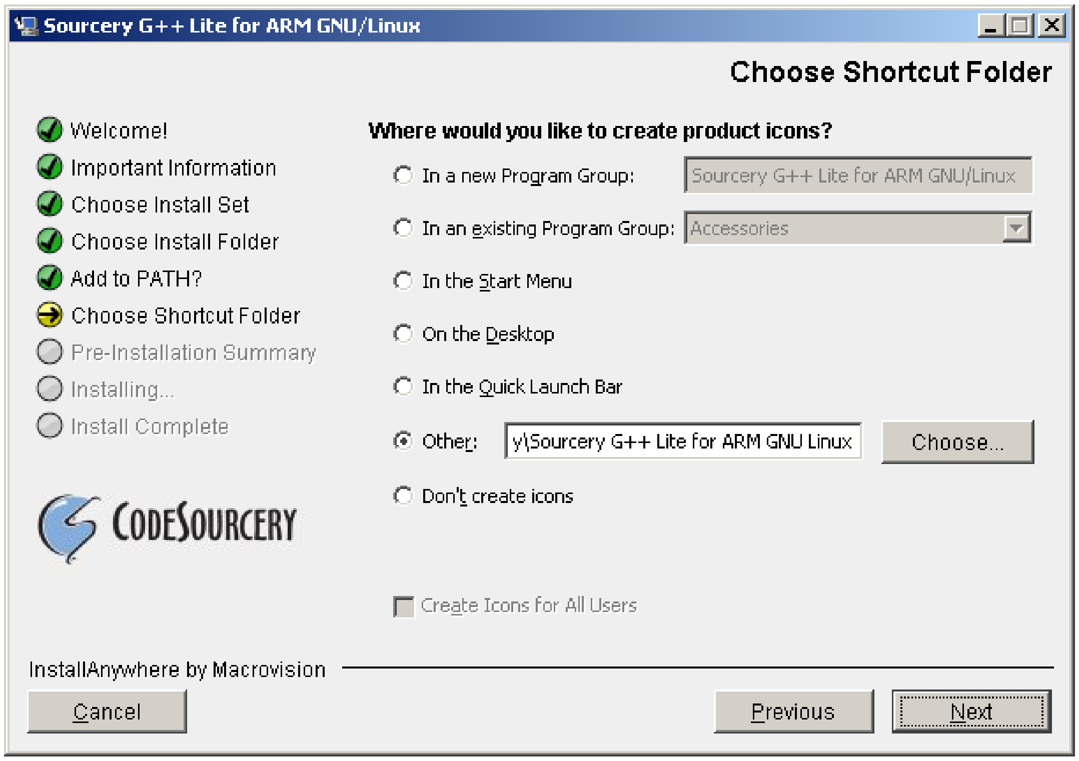
Click Next. Path window will appear. Select the option: Modify PATH for all users:



The Wizard can add Sourcery G++ Lite for ARM GNU/Linux to the PATH environment variable. This will make it easier to use Sourcery G++ Lite for ARM GNU/Linux and will mean less configuration for other tools.
WARNING: Without this change some tools may not function correctly.

Do not modify PATH. Modify PATH for current user. [✓ Modify PATH for all users.]

Click Next. Shortcut Folder window will appear:


Sourcery G++ Lite for ARM GNU/Linux

Choose Shortcut Folder

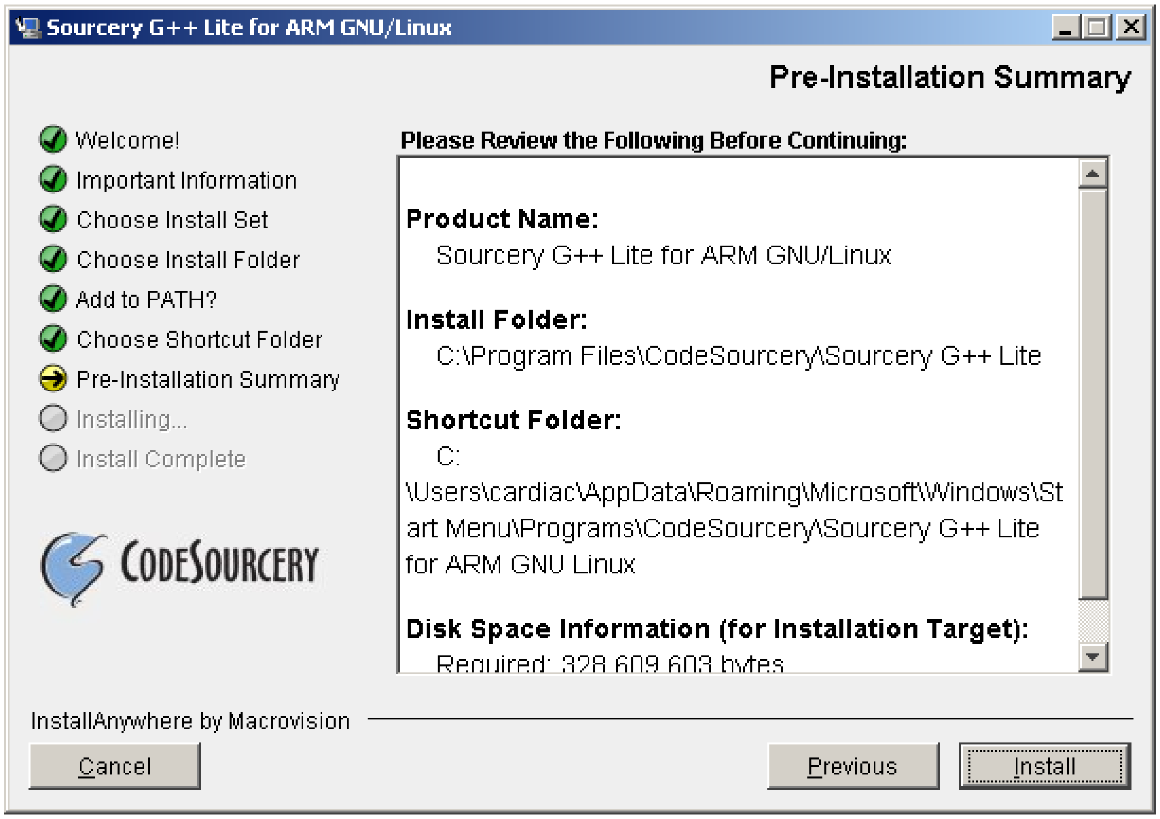
- Welcome!
- Important Information
- Choose Install Set
- Choose Install Folder
- Add to PATH?
- Choose Shortcut Folder
- Pre-Installation Summary
- Installing...
- Install Complete

Where would you like to create product icons?

- In a new Program Group: Sourcery G++ Lite for ARM GNU/Linux
- In an existing Program Group: Accessories
- In the Start Menu
- On the Desktop
- In the Quick Launch Bar
- [✓ Other:] y\Sourcery G++ Lite for ARM GNU Linux
- Don't create icons

Create Icons for All Users

Leave all settings on this window as is. Click Next. Pre-Installation Summary window will appear:


Sourcery G++ Lite for ARM GNU/Linux

Pre-Installation Summary

- Welcome!
- Important Information
- Choose Install Set
- Choose Install Folder
- Add to PATH?
- Choose Shortcut Folder
- Pre-Installation Summary
- Installing...
- Install Complete

Please Review the Following Before Continuing:

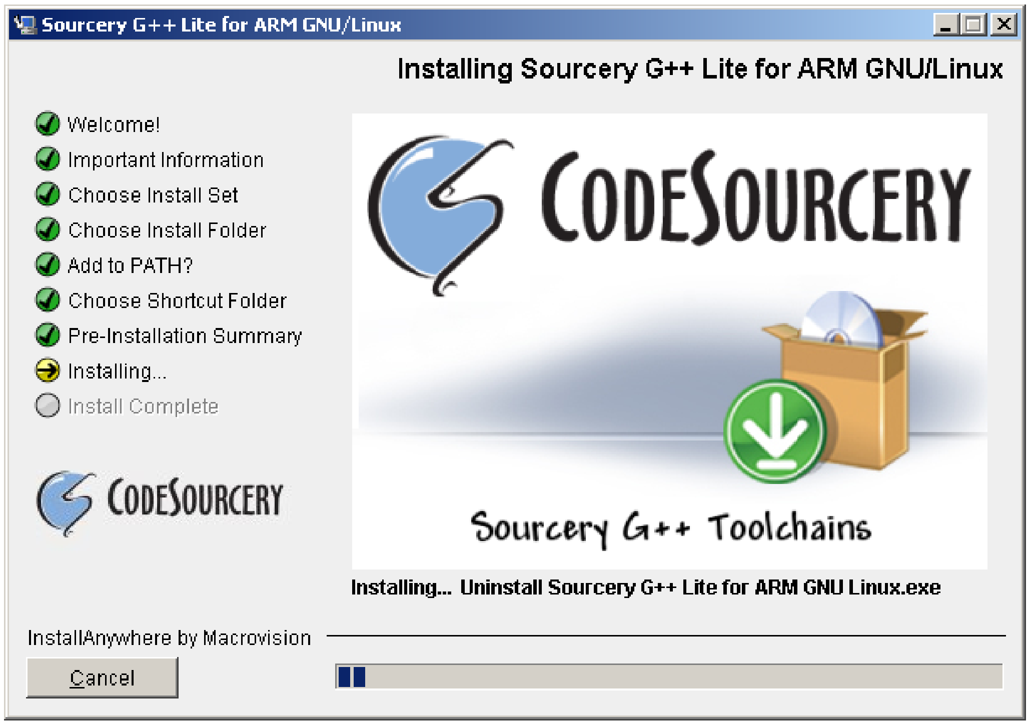
Product Name:
Sourcery G++ Lite for ARM GNU/Linux

Install Folder:
C:\Program Files\CodeSourcery\Sourcery G++ Lite

Shortcut Folder:
C:
\Users\cardiac\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\CodeSourcery\Sourcery G++ Lite for ARM GNU Linux

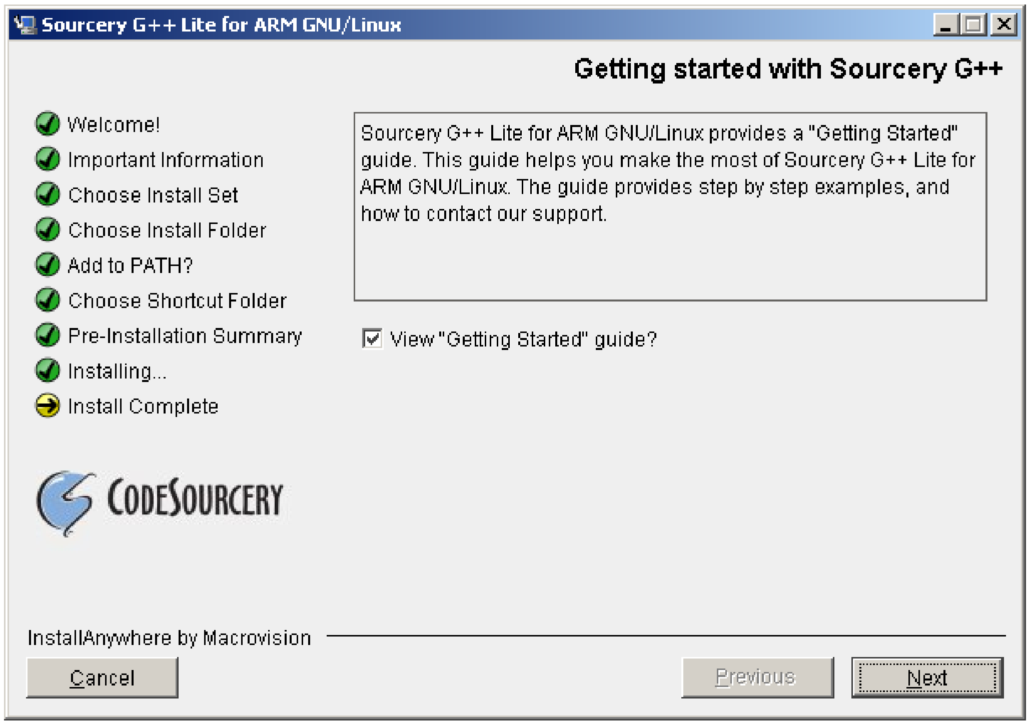
Disk Space Information (for Installation Target):
Required: 328,609,603 bytes

Click Install. Installation will start and you will see the progress:



Installing... Uninstall Sourcery G++ Lite for ARM GNU Linux.exe

When the installation is complete, following page will appear:



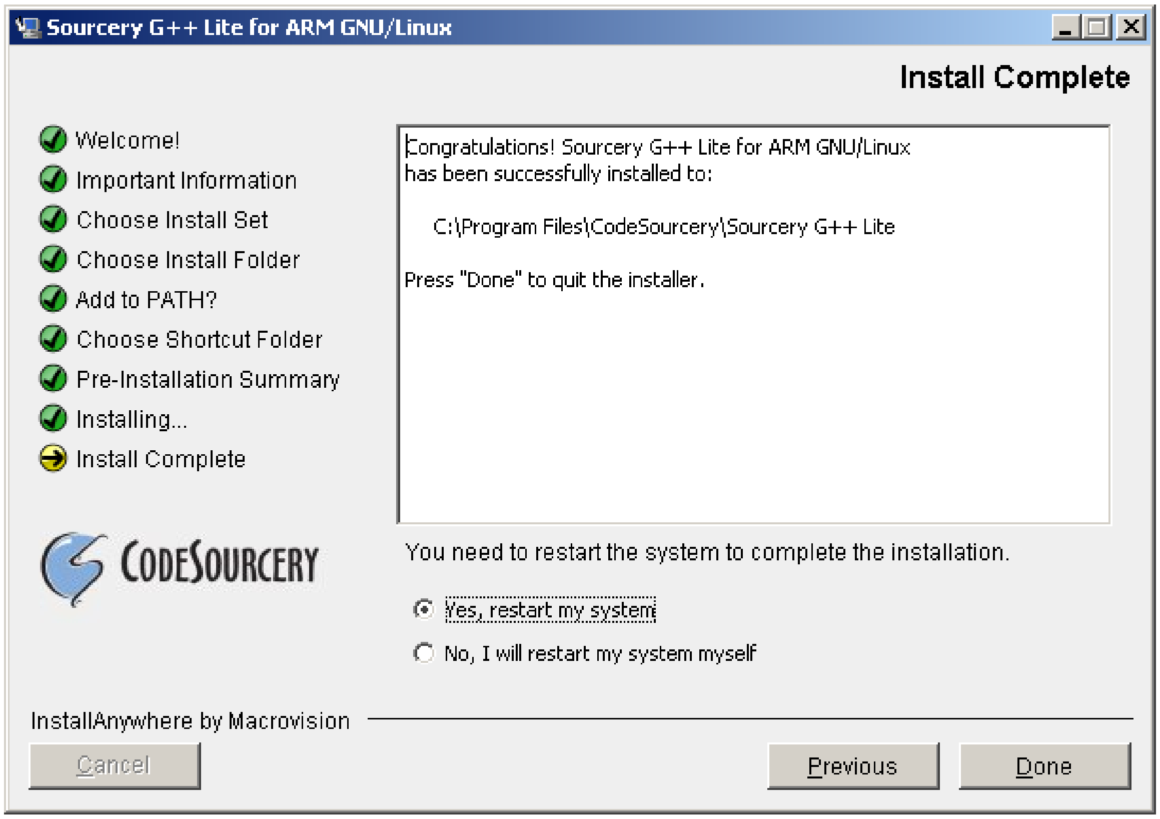
Sourcery G++ Lite for ARM GNU/Linux provides a "Getting Started" guide. This guide helps you make the most of Sourcery G++ Lite for ARM GNU/Linux. The guide provides step by step examples, and how to contact our support.

☑ View "Getting Started" guide?

Keep the “View Getting Start Guide” option checked. Click Next.

Getting Started Guide will launch. You can review this to learn more about Code Sourcery and ARM GNU/Linux tools.

You will also see the final window of the installation:



Congratulations! Sourcery G++ Lite for ARM GNU/Linux has been successfully installed to:

C:\Program Files\CodeSourcery\Sourcery G++ Lite

Press "Done" to quit the installer.

You need to restart the system to complete the installation.

[✓ Yes, restart my system]

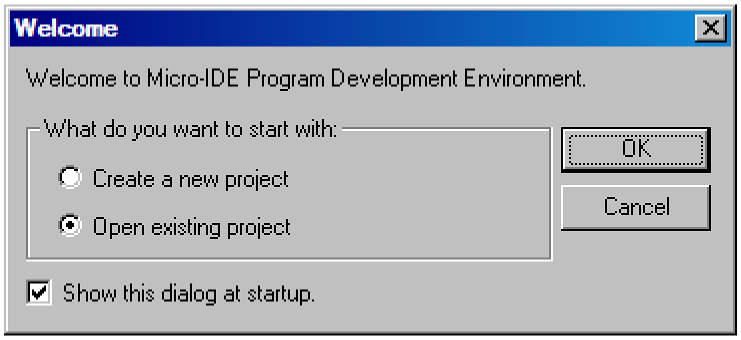
No, I will restart my system myself

Make sure that all open applications are closed and data is saved on your computer. Keep "Yes, restart my system" option checked and click Done. This will restart your computer.

1.1.2 Building and Running Example Programs

Now that the development environment is installed and ready, you can build programs and run programs on your GadgetPC or MINI-MAX/ARM9 series board. Follow the steps below:

1. Run Micro-IDE from Windows Start menu. When Micro-IDE starts, the Project Selection window appears:


Welcome

Welcome to Micro-IDE Program Development Environment.

What do you want to start with:

- Create a new project
- [✓ Open existing project]

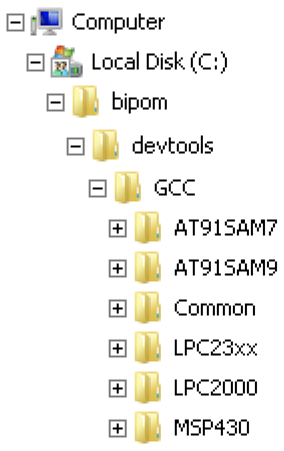
[✓] Show this dialog at startup.

Click OK to select an existing example project.

ARM Development System is distributed with several example programs that illustrate how to program the ARM9 micro-controller under Linux. Example projects are located under the Examples folder under following folders:

\bipom\devtools\GCC\AT91SAM9\Examples\Linux

The main folder for the GNU toolchain is “GCC”. A folder is created under “GCC” for each microcontroller family. There is also a “common” folder that contains the common files.



Under each microcontroller family directory, there is further classification of files:

cstartup: C start up file for that microcontroller family
examples: project examples for that microcontroller family
include: C language include files for that microcontroller family
lib: precompiled library files for that microcontroller family
src: C language source files for that microcontroller family

Project examples are typically designed to run on BiPOM boards.

Library files under lib folder typically have the naming convention:

lib<library name>.a

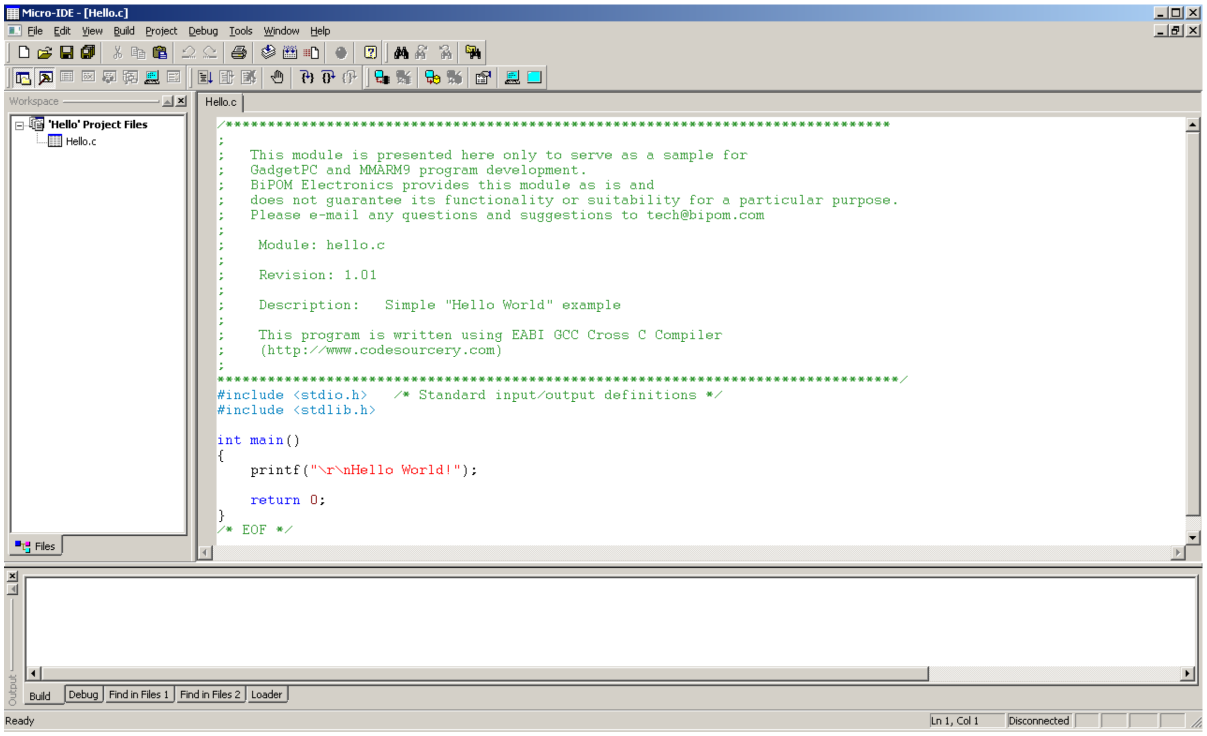
where <library name> is the name of the library. This follows the UNIX library naming convention. For example, BiPOM graphics library is called “gl” so the name of the library file for “gl” is

libgl.a

Examples under AT91SAM9 folder are compatible with BiPOM’s GadgetPC and MINI-MAX/ARM9 series boards. Examples in the **Linux** folder under the examples for AT91SAM9 are specifically designed to run under ARM9 Linux.

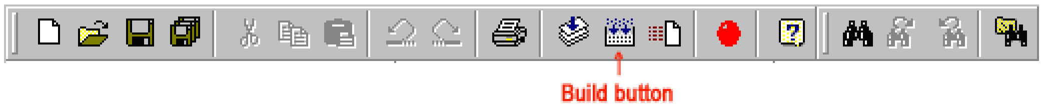
2. Open the example project Hello.prj from

\bipom\devtools\GCC\AT91SAM9\Examples\Linux\Hello

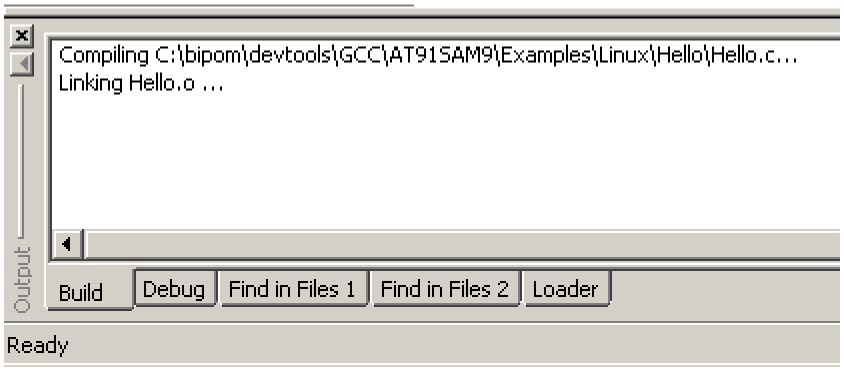


```
/******************************************************************************
;
;   This module is presented here only to serve as a sample for
;   GadgetPC and MMARM9 program development.
;   BiPOM Electronics provides this module as is and
;   does not guarantee its functionality or suitability for a particular purpose.
;   Please e-mail any questions and suggestions to tech@bipom.com
;
;    Module: hello.c
;
;    Revision: 1.01
;
;    Description:   Simple "Hello World" example
;
;    This program is written using EABI GCC Cross C Compiler
;    (http://www.codesourcery.com)
;
*******************************************************************************/
#include <stdio.h>   /* Standard input/output definitions */
#include <stdlib.h>

int main()
{
    printf("\r\nHello World!");

    return 0;
}
/* EOF */
```

3. Click the Build button on the main toolbar. This will build the Hello project:

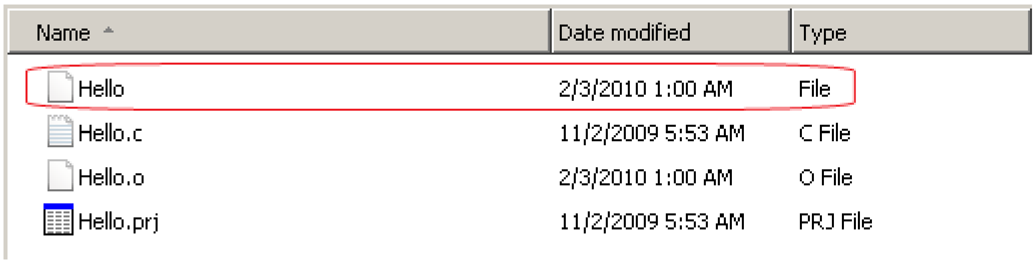


If the project builds successfully, you should see a message indicating no errors on the Output Window:



```
Compiling C:\bipom\devtools\GCC\AT91SAM9\Examples\Linux\Hello\Hello.c...
Linking Hello.o ...
```

If the build completed without errors, the executable (binary) file called **hello** will be created in the **Hello** folder. You can see this in Windows Explorer:

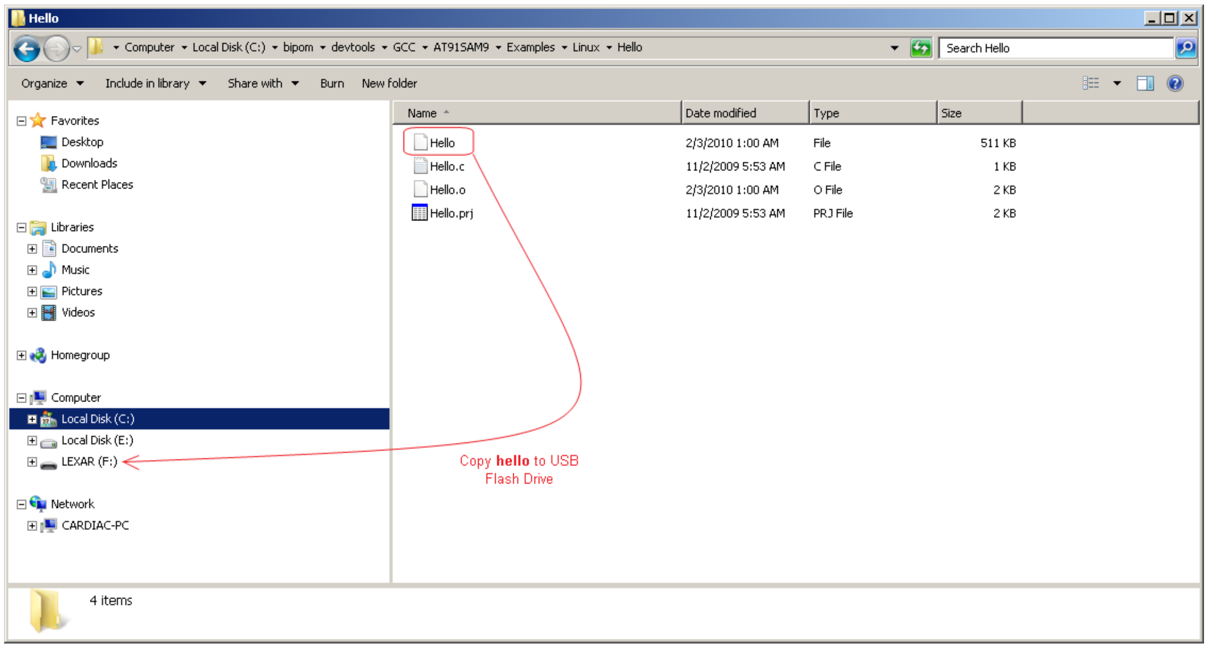


Name	Date modified	Type
Hello	2/3/2010 1:00 AM	File
Hello.c	11/2/2009 5:53 AM	C File
Hello.o	2/3/2010 1:00 AM	O File
Hello.prj	11/2/2009 5:53 AM	PRJ File

You can now copy this file **hello** to GadgetPC or MINI-MAX/ARM9 series board and run the program under Linux.

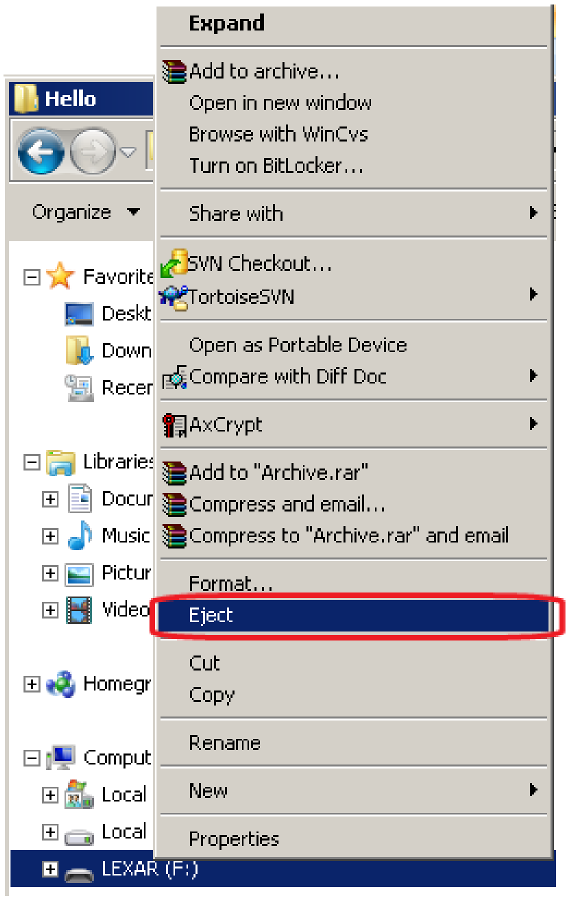
You can use either FTP or SSH to copy the file over the network or you can copy the file to the USB Flash Drive (the USB drive that is used for booting Linux) and run it from the USB Flash drive on the board.

To copy using the USB Flash drive, plug the USB Flash Drive to your PC’s USB port:

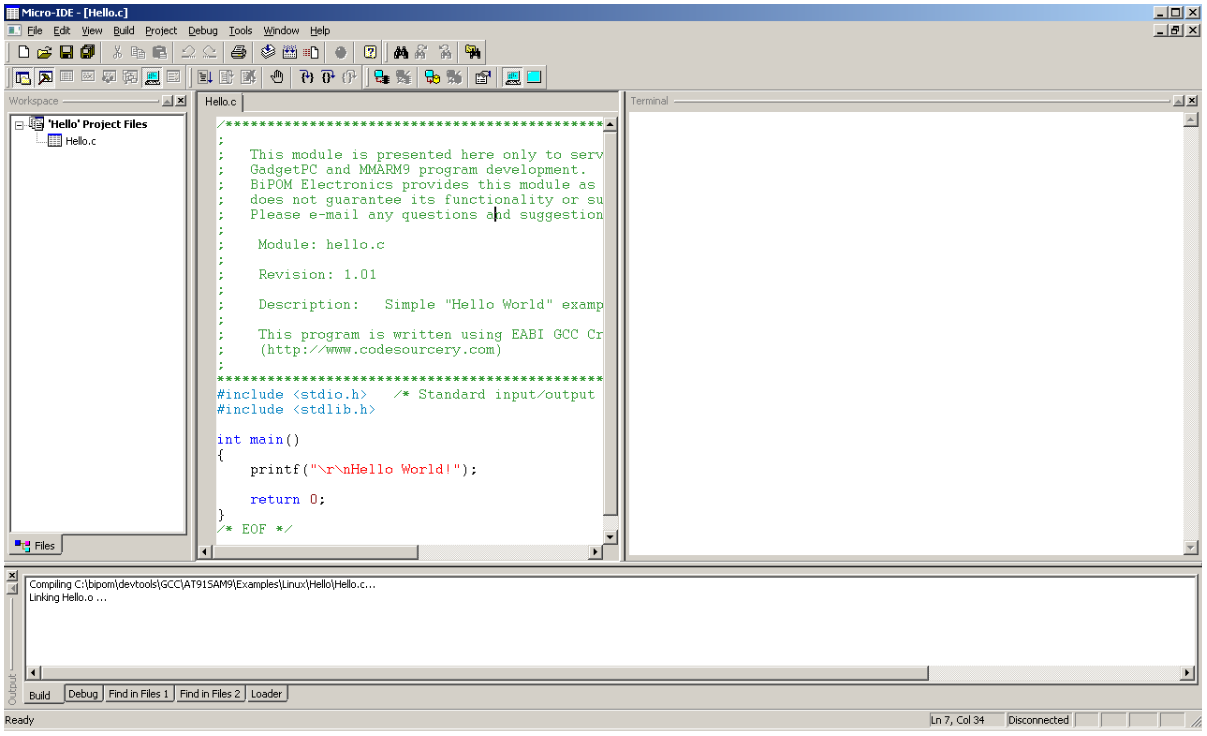


Name	Date modified	Type	Size
Hello	2/3/2010 1:00 AM	File	511 KB
Hello.c	11/2/2009 5:53 AM	C File	1 KB
Hello.o	2/3/2010 1:00 AM	O File	2 KB
Hello.prj	11/2/2009 5:53 AM	PRJ File	2 KB

After you copy the program, safely eject the USB Flash Drive from your PC.



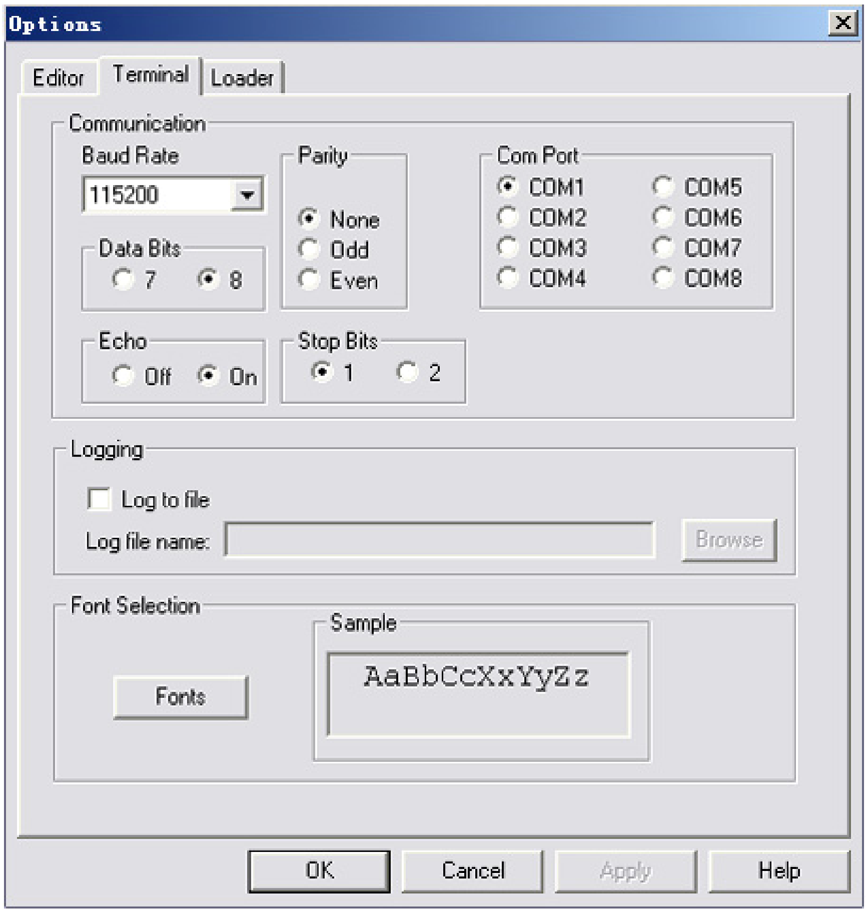
Select View->Terminal in Micro-IDE. A terminal window will appear:



```
/*********************************************
;
;   This module is presented here only to serv
;   GadgetPC and MMARM9 program development.
;   BiPOM Electronics provides this module as
;   does not guarantee its functionality or su
;   Please e-mail any questions and suggestion
;
;    Module: hello.c
;
;    Revision: 1.01
;
;    Description:   Simple "Hello World" examp
;
;    This program is written using EABI GCC Cr
;    (http://www.codesourcery.com)
;
**********************************************
#include <stdio.h>    /* Standard input/output
#include <stdlib.h>

int main()
{
    printf("\r\nHello World!");

    return 0;
}
/* EOF */
```

```
Compiling C:\bipom\devtools\GCC\AT91SAM9\Examples\Linux\Hello\Hello.c...
Linking Hello.o ...
```

Select Tools->Options->Terminal and set the COM port to your PC's serial port that is connected to the board (for example COM1):



Communication

Baud Rate: 115200

Data Bits: 7 [✓ 8]

Parity: [✓ None] Odd Even

Com Port: [✓ COM1] COM2 COM3 COM4 COM5 COM6 COM7 COM8

Echo: Off [✓ On]

Stop Bits: [✓ 1] 2

Logging

☐ Log to file

Log file name:

Font Selection

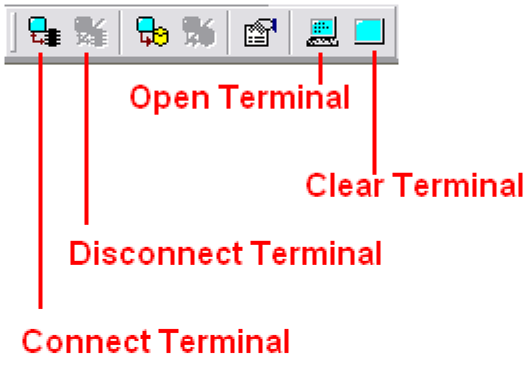
Sample: AaBbCcXxYyZz

Set baud rate to 115200. Other parameters are

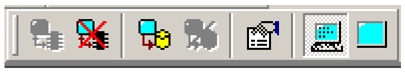
Parity: None
Data Bits: 8
Stop Bit: 1
Echo: Off

Click the OK button to save the setting and close the Options dialog.

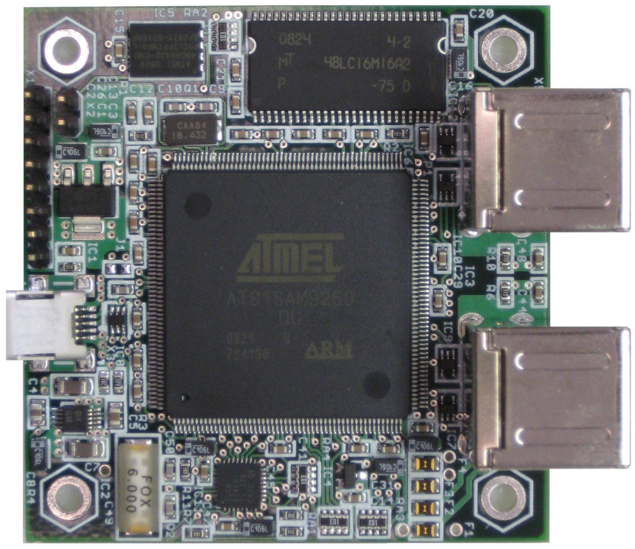
Using the Terminal Toolbar, connect the terminal (which opens the COM port). To do this, click the Connect Terminal button on the Terminal Toolbar:



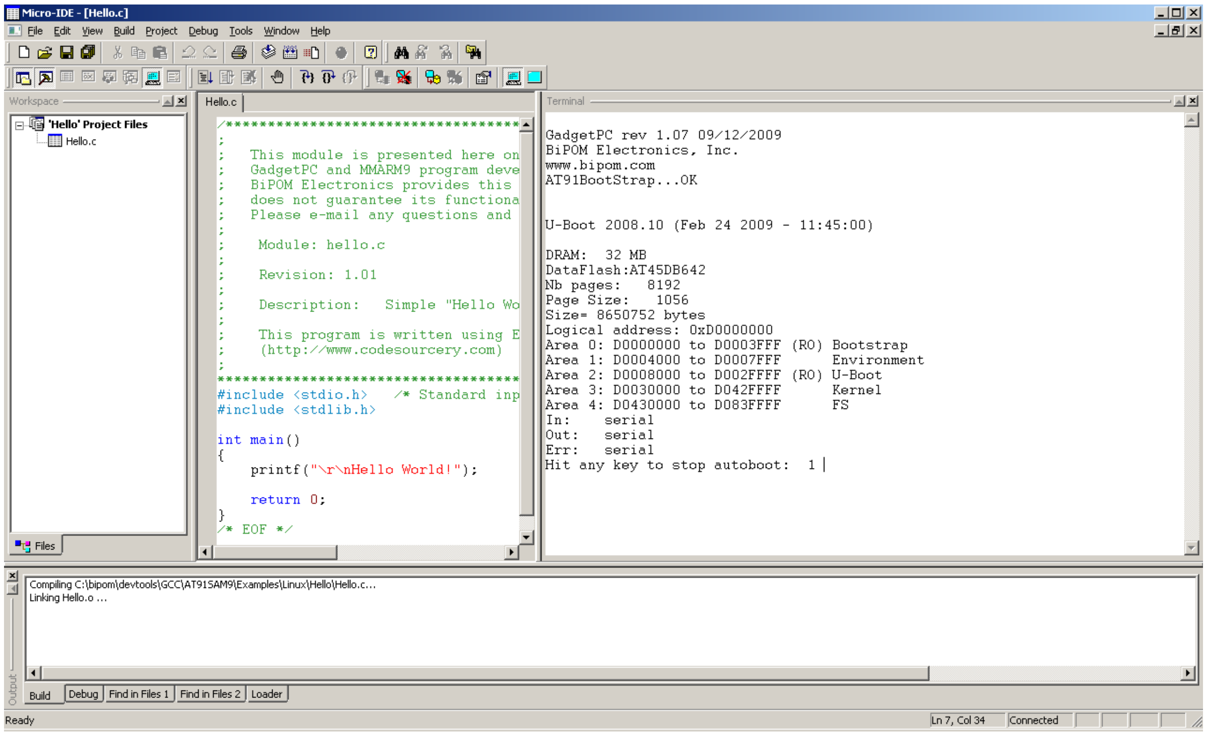
The Terminal Toolbar should now look like this:



Plug the USB Flash Drive to your board and power the board:



As Linux boots, you will see the boot messages:

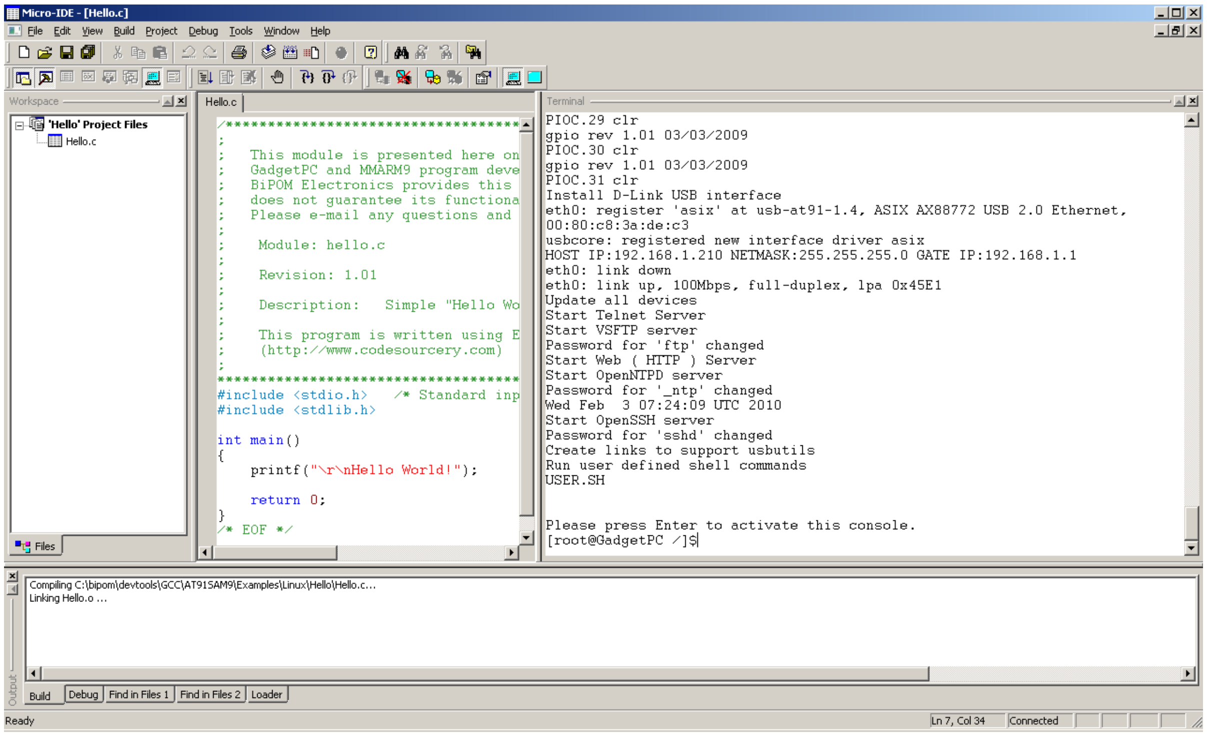


```
/************************************
;
;   This module is presented here on
;   GadgetPC and MMARM9 program deve
;   BiPOM Electronics provides this
;   does not guarantee its functiona
;   Please e-mail any questions and
;
;    Module: hello.c
;
;    Revision: 1.01
;
;    Description:   Simple "Hello Wo
;
;    This program is written using E
;    (http://www.codesourcery.com)
;
*************************************
#include <stdio.h>    /* Standard inp
#include <stdlib.h>

int main()
{
    printf("\r\nHello World!");

    return 0;
}
/* EOF */
```

```
GadgetPC rev 1.07 09/12/2009
BiPOM Electronics, Inc.
www.bipom.com
AT91BootStrap...OK


U-Boot 2008.10 (Feb 24 2009 - 11:45:00)

DRAM:  32 MB
DataFlash:AT45DB642
Nb pages:   8192
Page Size:   1056
Size= 8650752 bytes
Logical address: 0xD0000000
Area 0: D0000000 to D0003FFF (RO) Bootstrap
Area 1: D0004000 to D0007FFF      Environment
Area 2: D0008000 to D002FFFF (RO) U-Boot
Area 3: D0030000 to D042FFFF      Kernel
Area 4: D0430000 to D083FFFF      FS
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  1
```

```
Compiling C:\bipom\devtools\GCC\AT91SAM9\Examples\Linux\Hello\Hello.c...
Linking Hello.o ...
```

When Linux completes booting, press Enter and you will be at the console prompt:



```
/************************************
;
;   This module is presented here on
;   GadgetPC and MMARM9 program deve
;   BiPOM Electronics provides this
;   does not guarantee its functiona
;   Please e-mail any questions and
;
;    Module: hello.c
;
;    Revision: 1.01
;
;    Description:   Simple "Hello Wo
;
;    This program is written using E
;    (http://www.codesourcery.com)
;
*************************************
#include <stdio.h>    /* Standard inp
#include <stdlib.h>

int main()
{
    printf("\r\nHello World!");

    return 0;
}
/* EOF */
```

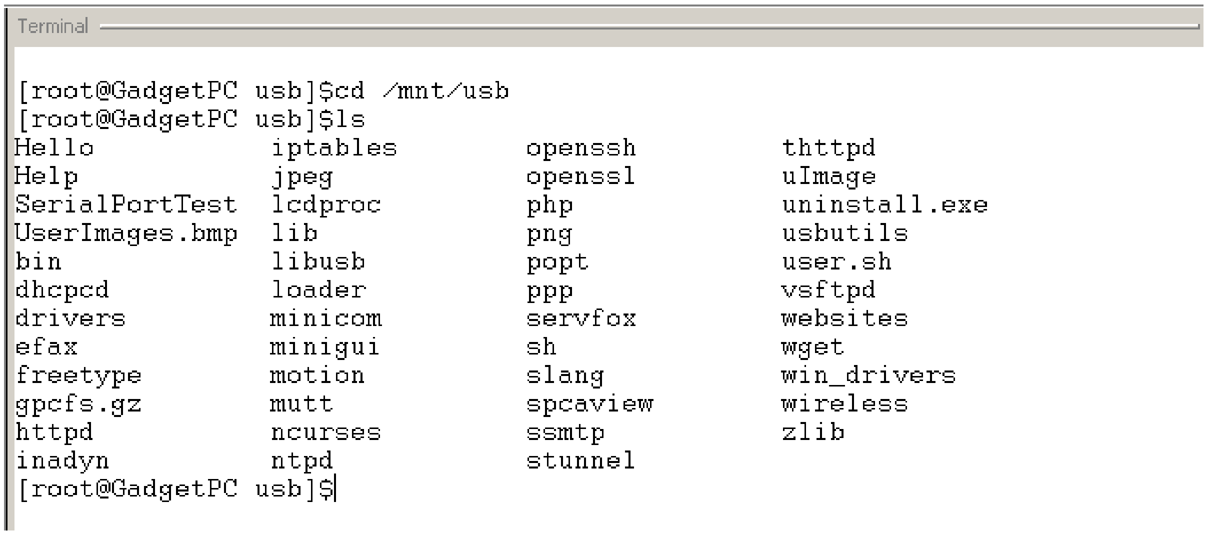
```
PIOC.29 clr
gpio rev 1.01 03/03/2009
PIOC.30 clr
gpio rev 1.01 03/03/2009
PIOC.31 clr
Install D-Link USB interface
eth0: register 'asix' at usb-at91-1.4, ASIX AX88772 USB 2.0 Ethernet,
00:80:c8:3a:de:c3
usbcore: registered new interface driver asix
HOST IP:192.168.1.210 NETMASK:255.255.255.0 GATE IP:192.168.1.1
eth0: link down
eth0: link up, 100Mbps, full-duplex, lpa 0x45E1
Update all devices
Start Telnet Server
Start VSFTP server
Password for 'ftp' changed
Start Web ( HTTP ) Server
Start OpenNTPD server
Password for '_ntp' changed
Wed Feb  3 07:24:09 UTC 2010
Start OpenSSH server
Password for 'sshd' changed
Create links to support usbutils
Run user defined shell commands
USER.SH


Please press Enter to activate this console.
[root@GadgetPC /]$
```

```
Compiling C:\bipom\devtools\GCC\AT91SAM9\Examples\Linux\Hello\Hello.c...
Linking Hello.o ...
```

Type **cd /mnt/usb** to change directory. Press Enter.

Type **ls** for directory listing. Press Enter.

You will see the contents of directory **/mnt/usb**:

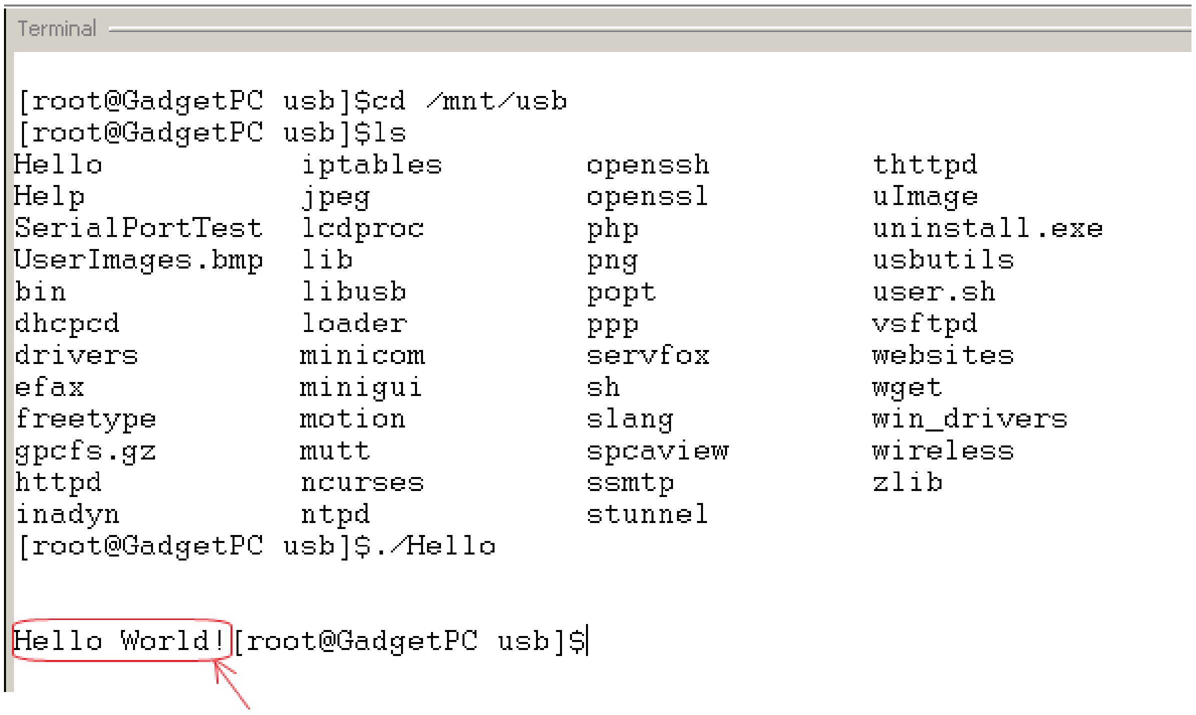


```
[root@GadgetPC usb]$cd /mnt/usb
[root@GadgetPC usb]$ls
Hello           iptables        openssh         thttpd
Help            jpeg            openssl         uImage
SerialPortTest  lcdproc         php             uninstall.exe
UserImages.bmp  lib             png             usbutils
bin             libusb          popt            user.sh
dhcpcd          loader          ppp             vsftpd
drivers         minicom         servfox         websites
efax            minigui         sh              wget
freetype        motion          slang           win_drivers
gpcfs.gz        mutt            spcaview        wireless
httpd           ncurses         ssmtp           zlib
inadyn          ntpd            stunnel
[root@GadgetPC usb]$
```

Please note the Hello example that has been copied. To run the Hello example, type:

./Hello

Hello program will run and print a message to console:

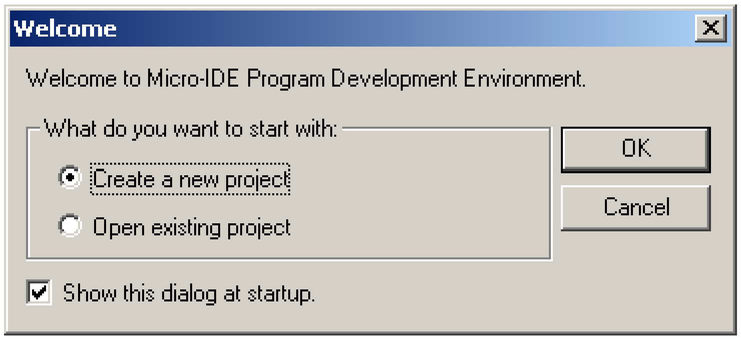


```
[root@GadgetPC usb]$cd /mnt/usb
[root@GadgetPC usb]$ls
Hello           iptables        openssh         thttpd
Help            jpeg            openssl         uImage
SerialPortTest  lcdproc         php             uninstall.exe
UserImages.bmp  lib             png             usbutils
bin             libusb          popt            user.sh
dhcpcd          loader          ppp             vsftpd
drivers         minicom         servfox         websites
efax            minigui         sh              wget
freetype        motion          slang           win_drivers
gpcfs.gz        mutt            spcaview        wireless
httpd           ncurses         ssmtp           zlib
inadyn          ntpd            stunnel
[root@GadgetPC usb]$./Hello

Hello World![root@GadgetPC usb]$
```

1.1.3 Writing Your Own Programs

1.1.3.1 Creating Projects

To create your own project, start Micro-IDE. It will ask you to Open a Project or Create a Project:



Welcome to Micro-IDE Program Development Environment.

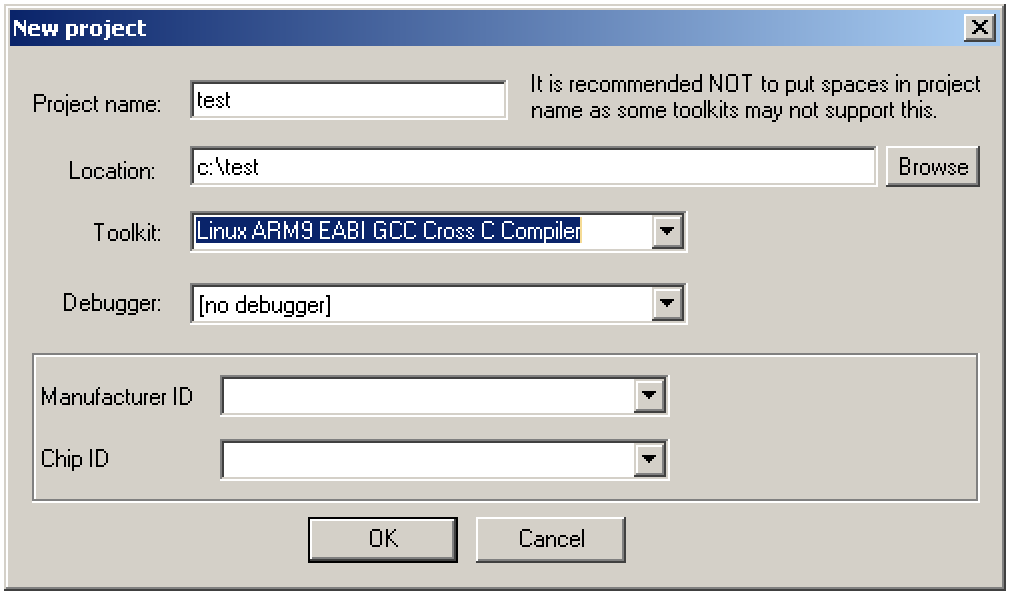
What do you want to start with:

[✓ Create a new project] Open existing project

[✓] Show this dialog at startup.

Select Create a new project and click OK.

This will display the New Project dialog:



Project name: test

It is recommended NOT to put spaces in project name as some toolkits may not support this.

Location: c:\test

Toolkit: Linux ARM9 EABI GCC Cross C Compiler

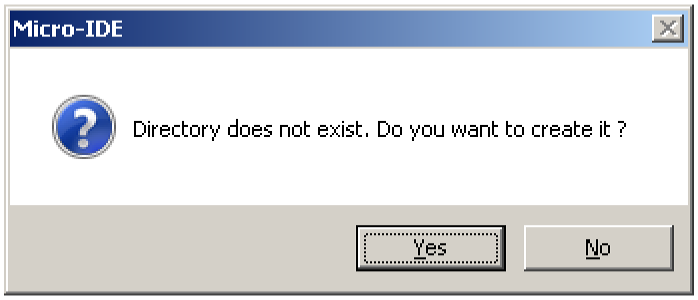
Debugger: [no debugger]

Manufacturer ID

Chip ID

Enter the name of the new project and its location (this example uses ***test*** as the project name and c:\test as the project location). Select **Linux ARM9 EABI GCC Cross C Compiler** as the Toolkit. Click OK; the new project with the name of test under c:\test will be created.

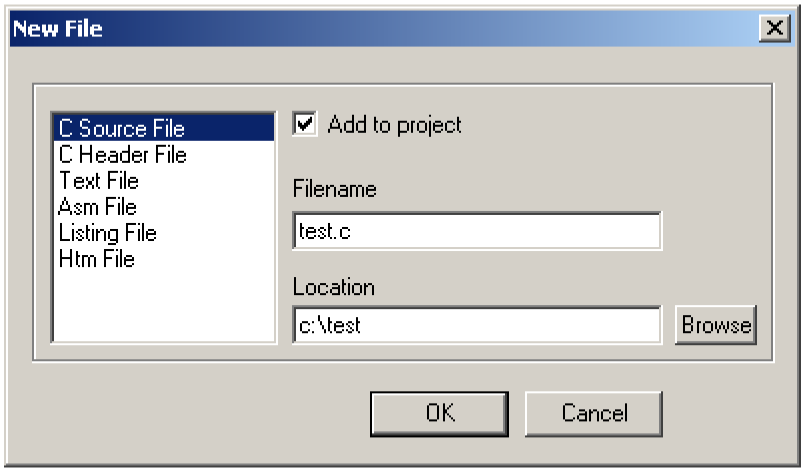
If the directory c:\test does not exist, Micro-IDE will ask you if the directory should be created:


Micro-IDE

Directory does not exist. Do you want to create it ?

Click Yes.

Create a new C file for the project. Select File menu and click New File. The New File dialog displays:


New File

C Source File
C Header File
Text File
Asm File
Listing File
Htm File

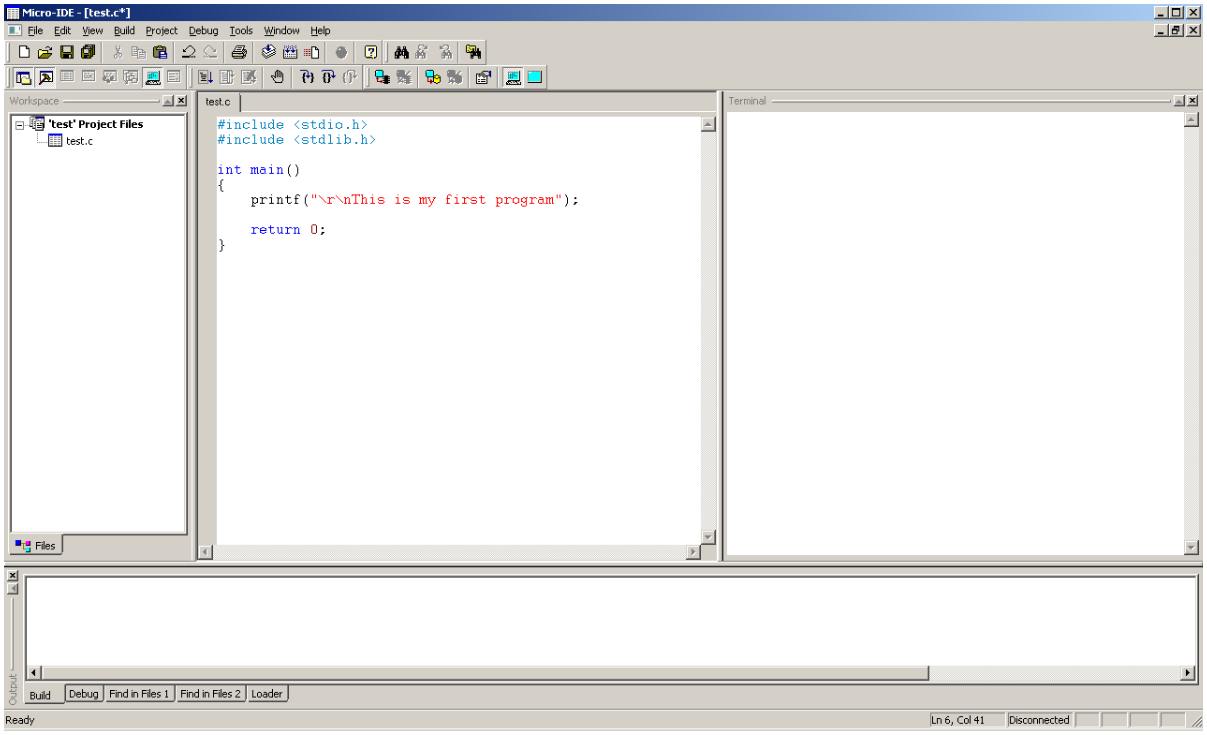
✓ Add to project

Filename: test.c

Location: c:\test

Check Add to project option and enter the Filename “test”. A blank C file (test.c) will be created and added to the test project.

Type the C Language sentences into test.c as shown below. The program should now look like this:



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

int main()
{
    printf("\r\nThis is my first program");

    return 0;
}
```

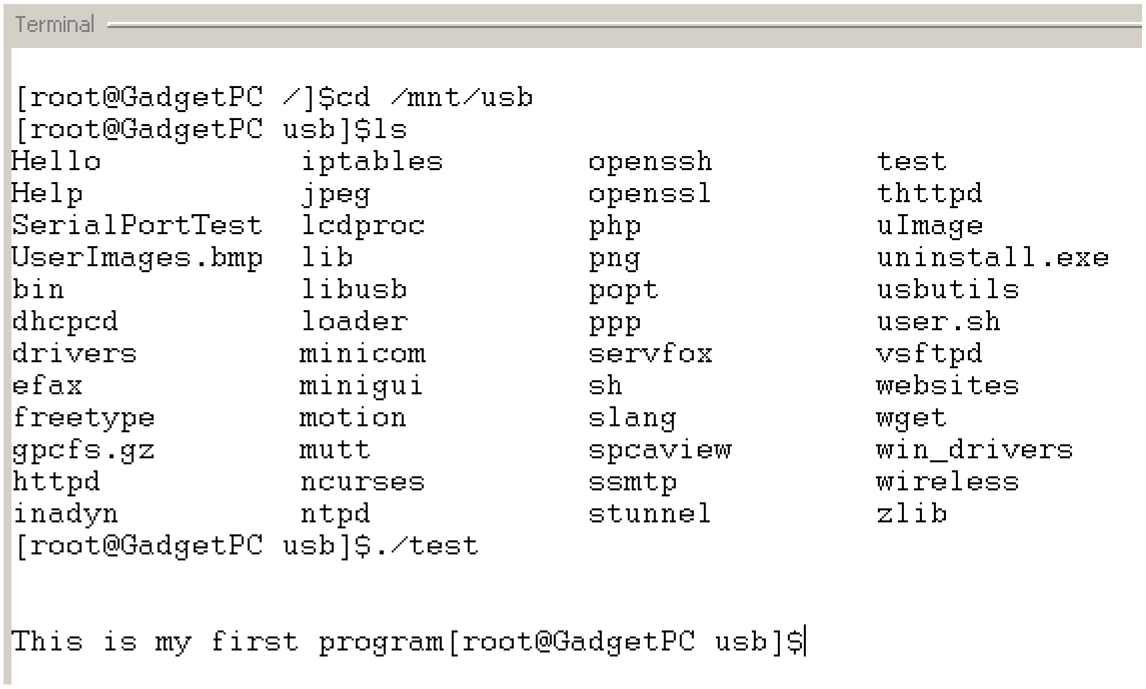
Similar to the Hello example, this program send a message to the serial console.

Build the program by clicking the Build All button (or by selecting Build->Build test from the menu).

If the program builds successfully, you will see the following messages on the Output Window:

Compiling c:\test\test.c…
Linking test.o …

Copy the executable file **test** to the board and run it under Linux (similar to how you did with **hello** example:


```
[root@GadgetPC /]$cd /mnt/usb
[root@GadgetPC usb]$ls
Hello           iptables        openssh         test
Help            jpeg            openssl         thttpd
SerialPortTest  lcdproc         php             uImage
UserImages.bmp  lib             png             uninstall.exe
bin             libusb          popt            usbutils
dhcpcd          loader          ppp             user.sh
drivers         minicom         servfox         vsftpd
efax            minigui         sh              websites
freetype        motion          slang           wget
gpcfs.gz        mutt            spcaview        win_drivers
httpd           ncurses         ssmtp           wireless
inadyn          ntpd            stunnel         zlib
[root@GadgetPC usb]$./test


This is my first program[root@GadgetPC usb]$
```

Congratulations!!! You have built and executed your first Linux C program on the GadgetPC or MINI-MAX/ARM9 series board. J