ISEE IGEPv2 BOARD

IGEPv2 BOARD SDK USER MANUAL

MAN-PR-IGEP.0020-002.06

(Revision 2.06 - 11/01/2011)

ISEE (Integration Software & Electronics Engineering)
Crta. De Martorell 95, Local 7 – Terrassa
(08224) – Barcelona – SPAIN.
+34.93.789.12.71
support@iseebcn.com
www.iseebcn.com

CONTENTS

0 COPYRIGHT NOTICE

VERSION CONTROL

REVISION	DATE	DESCRIPTION
1.00	19/03/2009	Initial version
1.01	17/07/2009	Revision
1.02	21/12/2009	Upgrade to igep0020b + sgx demo
1.03	29/01/2010	See changelog
1.04	13/05/2010	See changelog
2.05	14/12/2010	Remake. Move HOWTOs manuals to http://labs.igep.es

1 PREFACE

1.1 VERY QUICK START GUIDE

Run IGEPv2 board

1. Connect peripherals (monitor and USB keyboard or serial debug link) to IGEPv2 board and plug power jack.

A) Monitor and USB keyboard mouse plugged in IGEPv2 board

2. Wait 1 minute to see some “poky” graphical screen and later graphical desktop with application icons.

B) RS232 Serial debug link from IGEPv2 to your computer

3. Plug serial cable in your computer from IGEPv2 UART3 serial debug
   a. Plug in ID9 to DB9 serial cable on IGEPv2 J990 connector, plus cable DB9-DB9 RS232 to your computer RS232 serial COM port
   b. Plug in ID9 to DB9 serial cable on IGEPv2 J990 connector, plus cable DB9-DB9 RS232, plus RS232 to USB converter to your computer USB Host port
4. Open serial terminal program (Minicom, HyperTerminal, …) in your computer
5. When you power up your IGEPv2, you will see some “BIOS text” lines. Wait 1 minute to boot and prompt login on your serial link (UART3 serial debug J990 connector +cable IDC10-DB9 + serial DB9-DB9 cable + serial DB9-USB converter + serial terminal program)
6. Enter default user and password:

USERNAME: **root**
PASSWORD: **letmein**

C) Ethernet link from IGEPv2 to your computer

7. Plug Ethernet LAN cable from IGEPv2 to your local network (LAN router/switch/port)
8. Open secure shell console from your computer to IGEPv2 default IP address:

192.168.254.254

1.2 ORGANIZATION OF THE MANUAL

This manual is divided into 5 parts.

- The first part introduces basic concepts and useful links about IGEPv2 board environment and this manual.
- The second part is a short description about IGEPv2 user hardware interface. If you want to know about IGEPv2 hardware board details you should read "IGEPv2 BOARD HARDWARE MANUAL" document (public download from www.iseebcn.com IGEP website).
- The third part is the main content of this document. It explains all details about IGEPv2 SDK (Installation process, Virtual Machine format, contents, toolchain, rootfs, packages, helpers …)
- In the fourth part, there are several tricks and useful information about IGEPv2 SDK and board interconnection
- The fifth part is about IGEPv2 support

1.3 IGEP ECOSYSTEM

All the information is located on IGEP website www.iseebcn.com or http://www.igep.es

If you don't have an account at http://www.igep.es, please first establish an account.

1.4 USEFUL WEB LINKS AND EMAILS

ISEE IGEP platform web site: www.iseebcn.com or http://www.igep.es

ISEE IGEP support wiki: http://labs.igep.es

ISEE IGEP support public forum: http://www.igep.es/forum/

ISEE shop: http://shop.igep.es

ISEE Software Repositories: http://git.igep.es

Other interesting links:

http://www.vmware.com/products/player/

http://www.7-zip.org/

http://www.codeblocks.org/

http://www.putty.org/

http://www.kernel.org

http://www.pokylinux.org/

1.5 USER REGISTRATION

Users who want to have full access into IGEP platform web services, have to be registered (free registration).

Go to ISEE website: http://www.igep.es and register yourself.

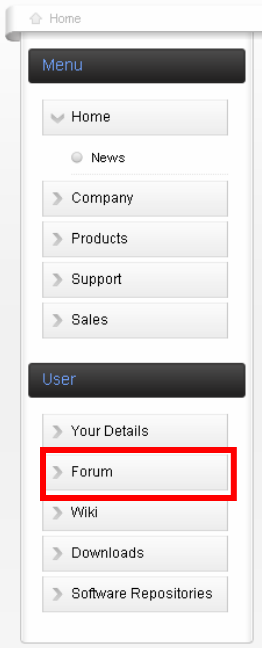
After that, you will receive the password access in a short time.

1.5.1 FREE CUSTOMER SUPPORT

There is a public forum where users can learn and contribute about the IGEP platform.

Users who want to use this service, have to be registered at www.igep.es website.

If you are registered, just go to User Menu → Forum



2 INTRODUCING IGEPV2 BOARD

2.1 IGEPV2 BOARD SDK FEATURES

ISEE provides customers with SDK built-in virtual machine box. Every piece of software is included, installed and configured to play and have fun with our ISEE IGEPv2 BOARD. These easy deploy form speeds up getting started with ISEE IGEPv2 BOARD environment.

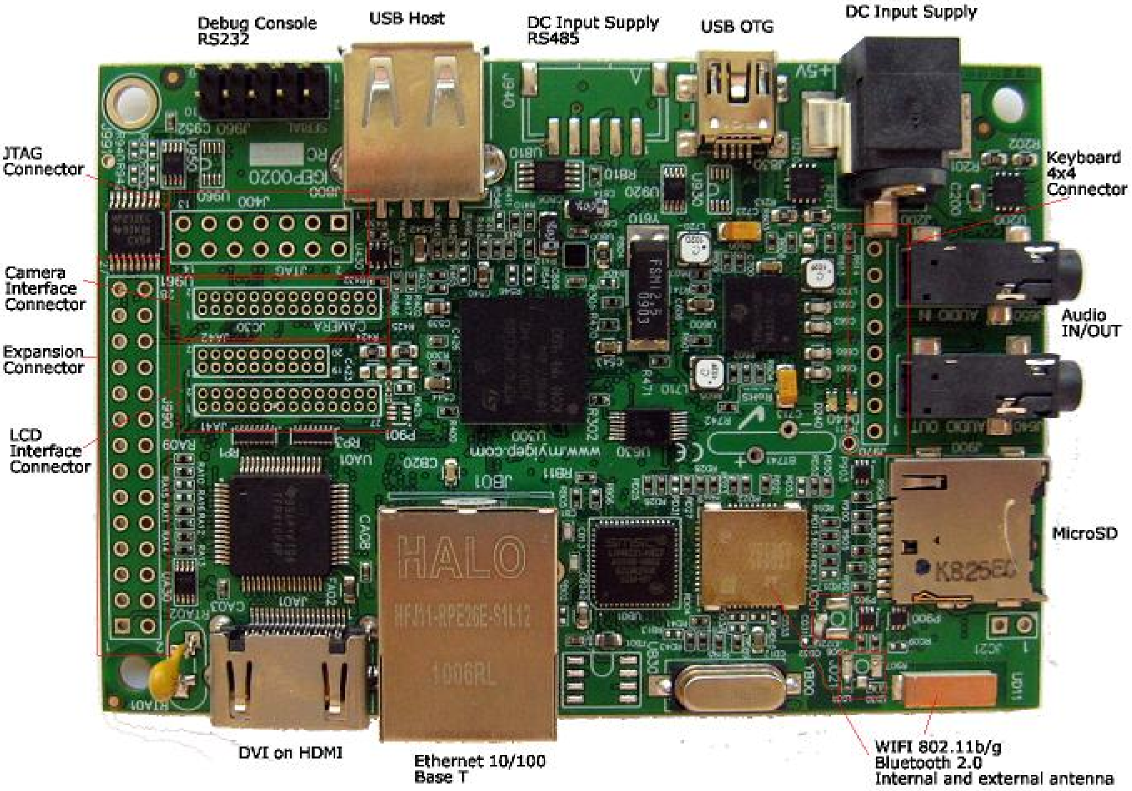


Figure 1 ISEE IGEPv2 board features

NOTE:

You should read first "IGEPv2 BOARD HARDWARE MANUAL" document.

Download it from www.igep.es

2.2 GET YOUR IGEPV2 BOARD POWER ON

1. Connect the DVI cable connector from the IGEPv2 DVI connector to the TFT DVI-D connector.

The Digital Visual Interface (DVI) is a video standard interface designed to provide very high visual quality on digital display devices such as flat panel LCD computer displays and digital projectors. It is partially compatible with the High-Definition Multimedia Interface (HDMI) standard in digital mode (DVI-D), and VGA in analog mode (DVI-A).

Users will need a cable with male DVI-D connector for the TFT, and male HDMI connector from IGEPv2 Board.

Figure 2 DVI cable

Figure 4 TFT monitor example

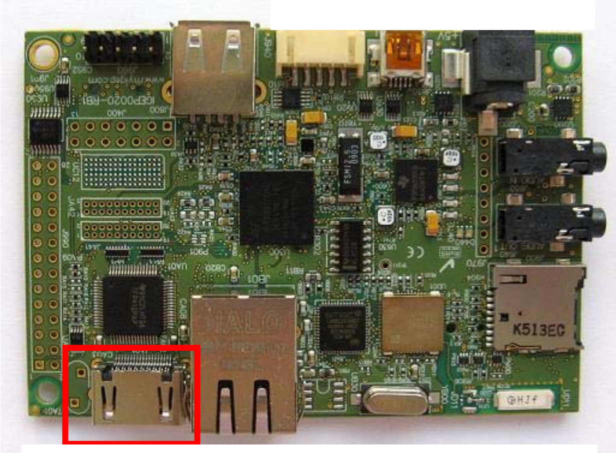


Figure 3 IGEP2 Board DVI connector

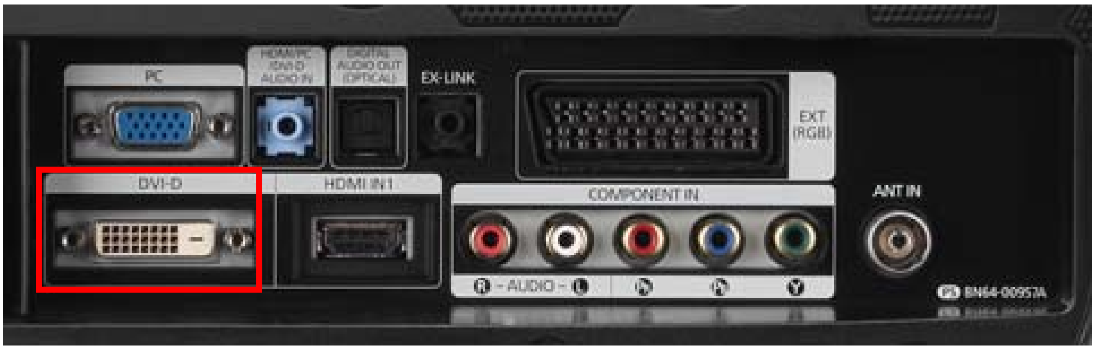


Figure 5 DVI-D Connector TFT detail example

2. Connect the stereo audio output from the IGEPv2 board to the TV Audio input.

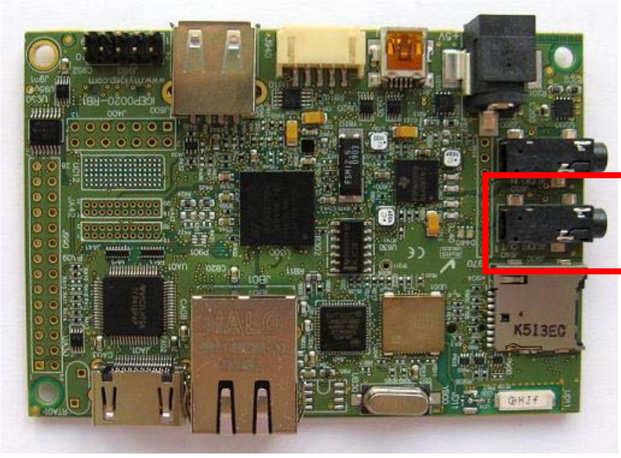


Figure 7 Stereo audio cable

Figure 6 IGEPV2 Board stereo audio output connector

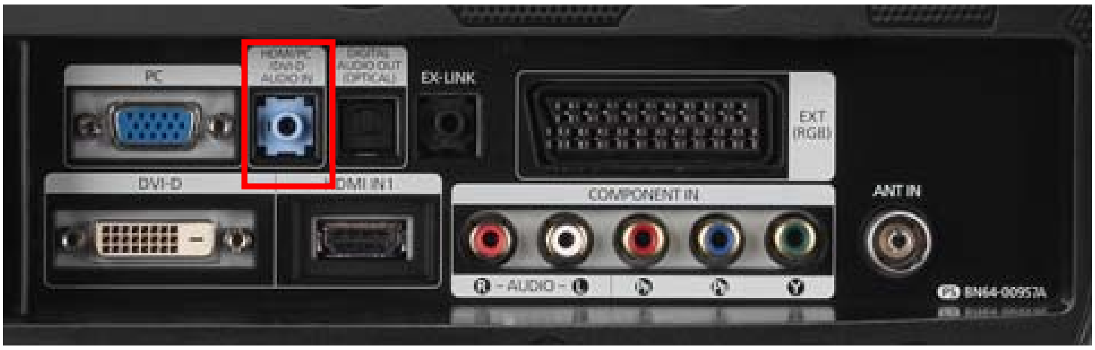


Figure 8 Stereo input Connector TFT detail example

3. Connect the Ethernet cable

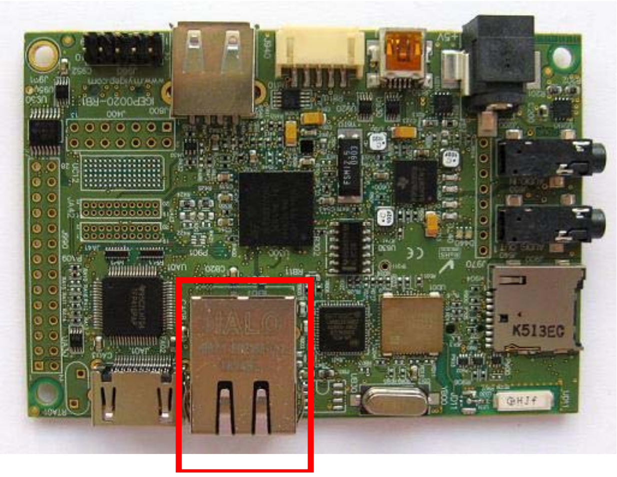


Figure 10 LAN cable

Figure 9 IGEPv2 Board Ethernet connector

4. Connect the USB keyboard and USB Mouse.

Figure 12 USB keyboard

Figure 11 USB mouse

Note: Only USB 2.0 devices work on IGEPV2 USB host connector.

a. Option1: using the IGEPV2 USB host connector and a USB hub 2.0.

Figure 13 USB hub

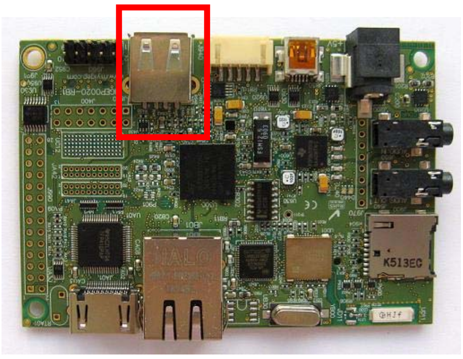
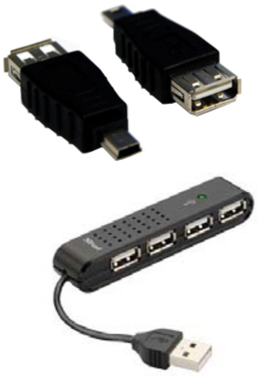


Figure 14 IGEPv2 Board USB host

Note: Do not connect a USB mouse 1.0 into the USB host connector without using a USB hub 2.0, because it will not work!!!

b. Option2: using the IGEPv2 USB mini OTG connector, a miniOTG to USB adapter and a USB hub 1.0 or 2.0.



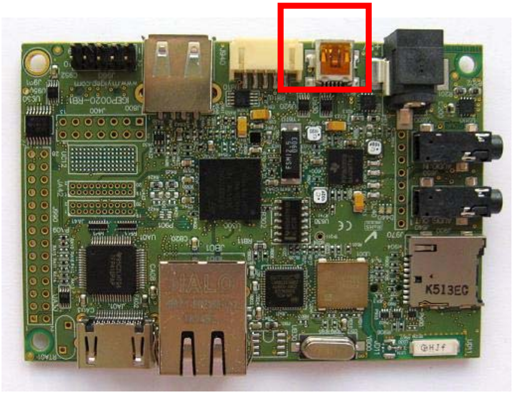


Figure 15 IGEPv2 board USB OTG

5. Power the board

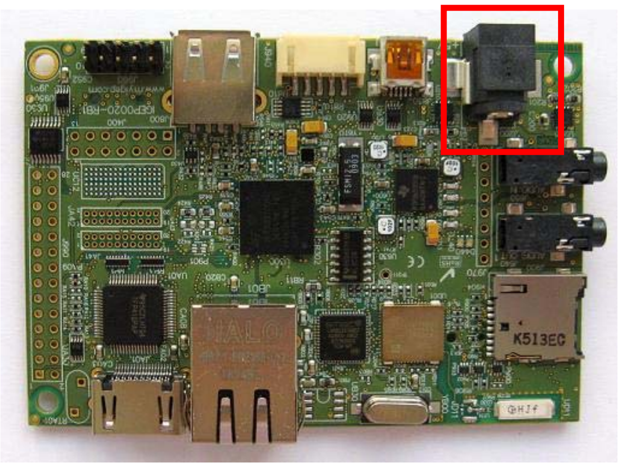
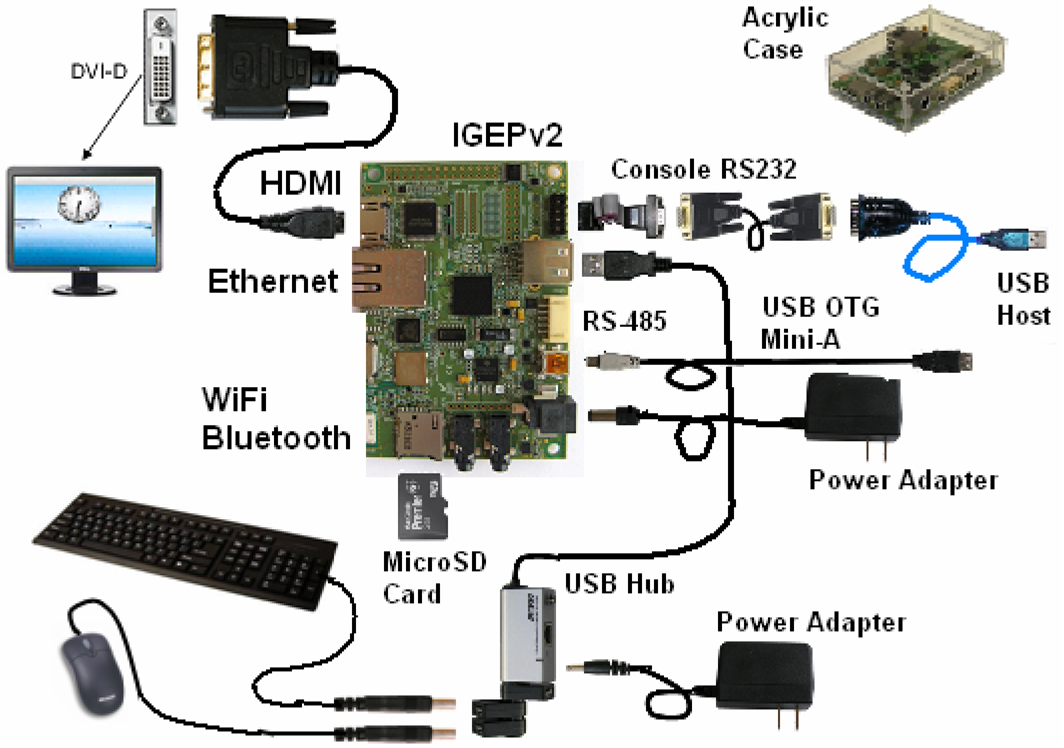


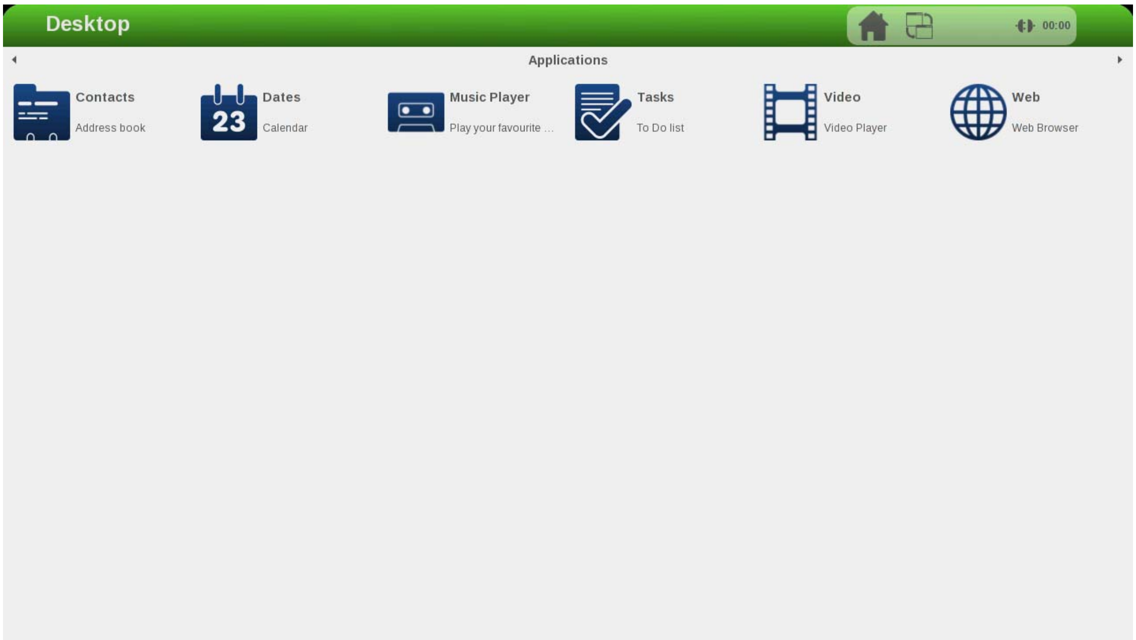
Figure 17 IGEPv2 board power jack

Figure 16 Power supply AC/DC adaptor

6. Resume



7. IGEPv2 DVI display output screenshot example:



3 IGEP V2 SDK VIRTUAL MACHINE

3.1 GETTING STARTED

3.1.1 WHY SDK IN A VIRTUAL MACHINE?

We use a virtual machine to make it easy to install IGEP SDK and operating system independent, and machine independent. It is based on VMware Player (http://www.vmware.com/products/player/). It is not open source but it is freeware. Nowadays, you could use other alternative virtual machine infrastructure (kvm, qemu, VirtualBox…).

VMware Player runs virtual machines on your Windows or Linux PC. This free desktop virtualization software application makes it easy to operate any virtual machine created by VMware Workstation, VMware Fusion, VMware Server or VMware ESX, as well as Microsoft Virtual Server virtual machines or Microsoft Virtual PC virtual machines. You can also use Player to evaluate one of the many virtual appliances available from the VMware Virtual Appliance Marketplace.

- Run multiple operating systems simultaneously on a single PC
- Experience the benefits of preconfigured products without any installation or configuration hassles
- Share data between host computer and virtual machine

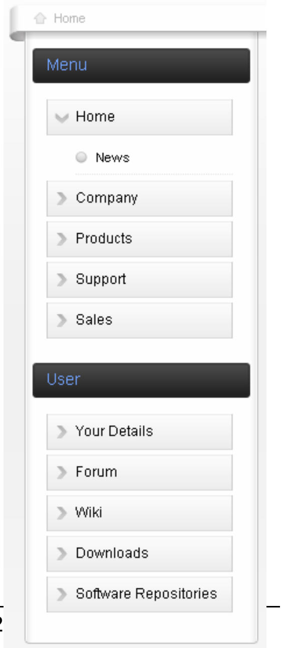
3.1.2 DOWNLOAD SDK VIRTUAL MACHINE

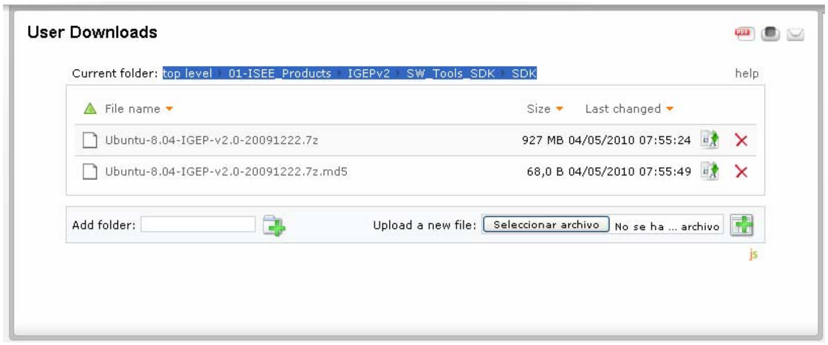
To download the SDK virtual machine, first you have to be registered on IGEP web site: www.igep.es (see chapter “1.5 USER REGISTRATION”).

If so, then go to ISEE IGEP web site: http://www.igep.es, and login with your username and password.

Press on User Menu → Download button (see figure)

Select directory: 01-ISEE_Products → IGEPv2 → SW_Tools_SDK → SDK





User Downloads

Current folder: top level › 01-ISEE_Products › IGEPv2 › SW_Tools_SDK › SDK

help

File name	Size	Last changed
Ubuntu-8.04-IGEP-v2.0-20091222.7z	927 MB	04/05/2010 07:55:24
Ubuntu-8.04-IGEP-v2.0-20091222.7z.md5	68,0 B	04/05/2010 07:55:49

Add folder:

Upload a new file: Seleccionar archivo No se ha ... archivo

Select the (latest) virtual machine and download it to your PC or workstation.

Be aware that all IGEP virtual machines are compressed with 7-zip compressor. We have used 7-zip compressor because of its higher rate compression.

7-Zip is open source software. Most of the source code is under the GNU LGPL license. You can download a free version of 7z from: http://www.7-zip.org/

Uncompress the IGEP virtual machine files.

3.1.3 SDK DETAILS BUILT-IN VIRTUAL MACHINE IMAGE

Ubuntu distribution based:

Release: Ubuntu Hardy Heron

Version: 8.04

Edition: Desktop, x86

URL: http://releases.ubuntu.com/hardy/

Account information

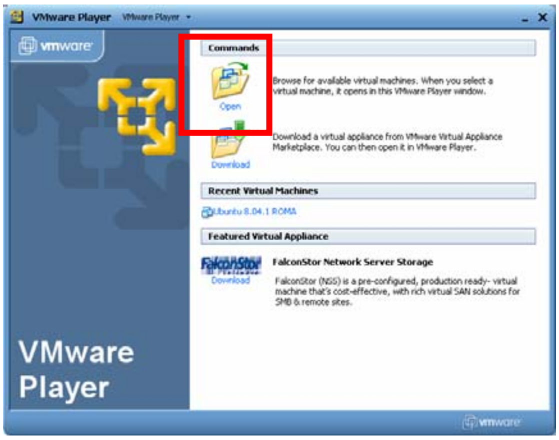
Username: **jdoe**

Password: **letmein**

Default IP address: 192.168.254.10

3.1.4 PLAY IGEPV2 SDK VIRTUAL MACHINE

Execute your vmplayer, press “Open” button and select the *.vmx file that you will find in the uncompressed downloaded directory.



Browse for available virtual machines. When you select a virtual machine, it opens in this VMware Player window.

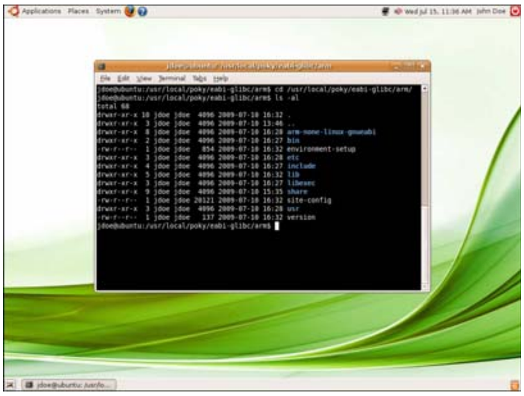
Download a virtual appliance from VMware Virtual Appliance Marketplace. You can then open it in VMware Player.

FalconStor (NSS) is a pre-configured, production ready- virtual machine that's cost-effective, with rich virtual SAN solutions for SMB & remote sites.

Figure 18 Virtual machine player from Vmware

If everything goes ok, you have to see a Desktop like this one:

Figure 19 Ubuntu 8.04 IGEPv2



```
jdoe@ubuntu:/usr/local/poky/eabi-glibc/arm$ cd /usr/local/poky/eabi-glibc/arm/
jdoe@ubuntu:/usr/local/poky/eabi-glibc/arm$ ls -al
total 68
drwxr-xr-x 10 jdoe jdoe  4096 2009-07-10 16:32 .
drwxr-xr-x  3 jdoe jdoe  4096 2009-07-10 13:46 ..
drwxr-xr-x  8 jdoe jdoe  4096 2009-07-10 16:28 arm-none-linux-gnueabi
drwxr-xr-x  2 jdoe jdoe  4096 2009-07-10 16:27 bin
-rw-r--r--  1 jdoe jdoe   854 2009-07-10 16:32 environment-setup
drwxr-xr-x  3 jdoe jdoe  4096 2009-07-10 16:28 etc
drwxr-xr-x  4 jdoe jdoe  4096 2009-07-10 16:27 include
drwxr-xr-x  5 jdoe jdoe  4096 2009-07-10 16:32 lib
drwxr-xr-x  3 jdoe jdoe  4096 2009-07-10 16:27 libexec
drwxr-xr-x  9 jdoe jdoe  4096 2009-07-10 15:35 share
-rw-r--r--  1 jdoe jdoe 28121 2009-07-10 16:32 site-config
drwxr-xr-x  3 jdoe jdoe  4096 2009-07-10 16:28 usr
-rw-r--r--  1 jdoe jdoe   137 2009-07-10 16:32 version
jdoe@ubuntu:/usr/local/poky/eabi-glibc/arm$
```

Figure 20 Ubuntu 8.04 IGEPV2 Console window

3.2 IMPROVE IGEPV2 SDK VIRTUAL MACHINE

If you need to improve, update or install more software in built-in ubuntu distribution, it's very easy.

3.2.1 GRAPHICAL PROCEDURE

In this case, you just go to the "Synaptic Package Manager" in "System\Administration" menu.

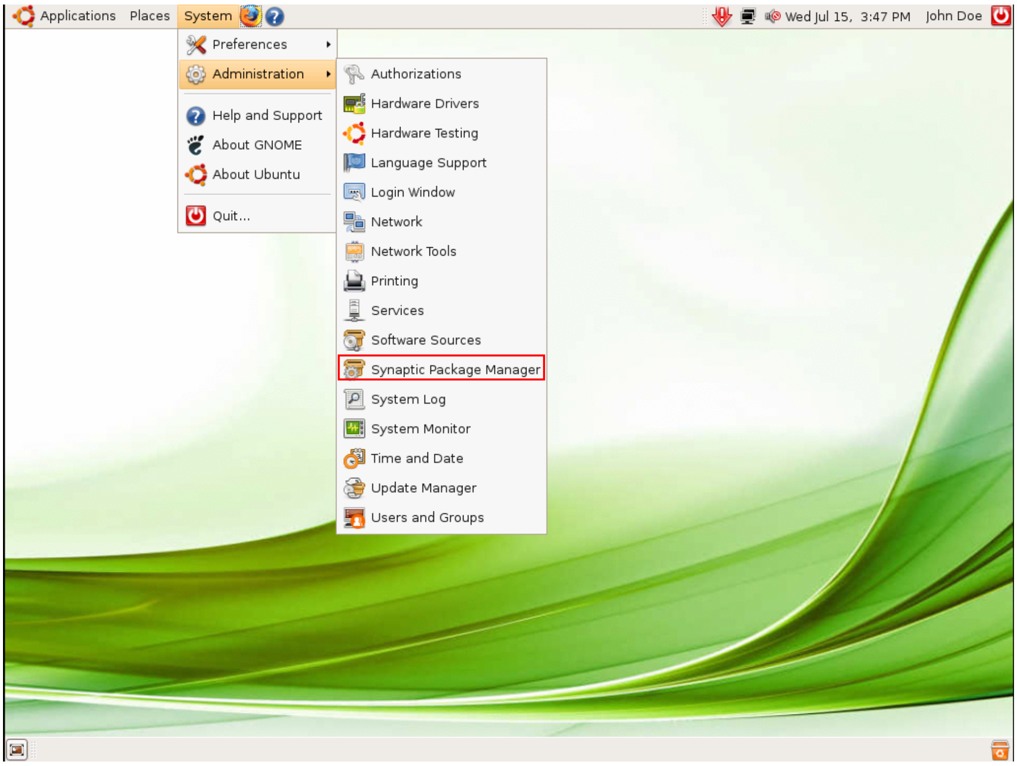


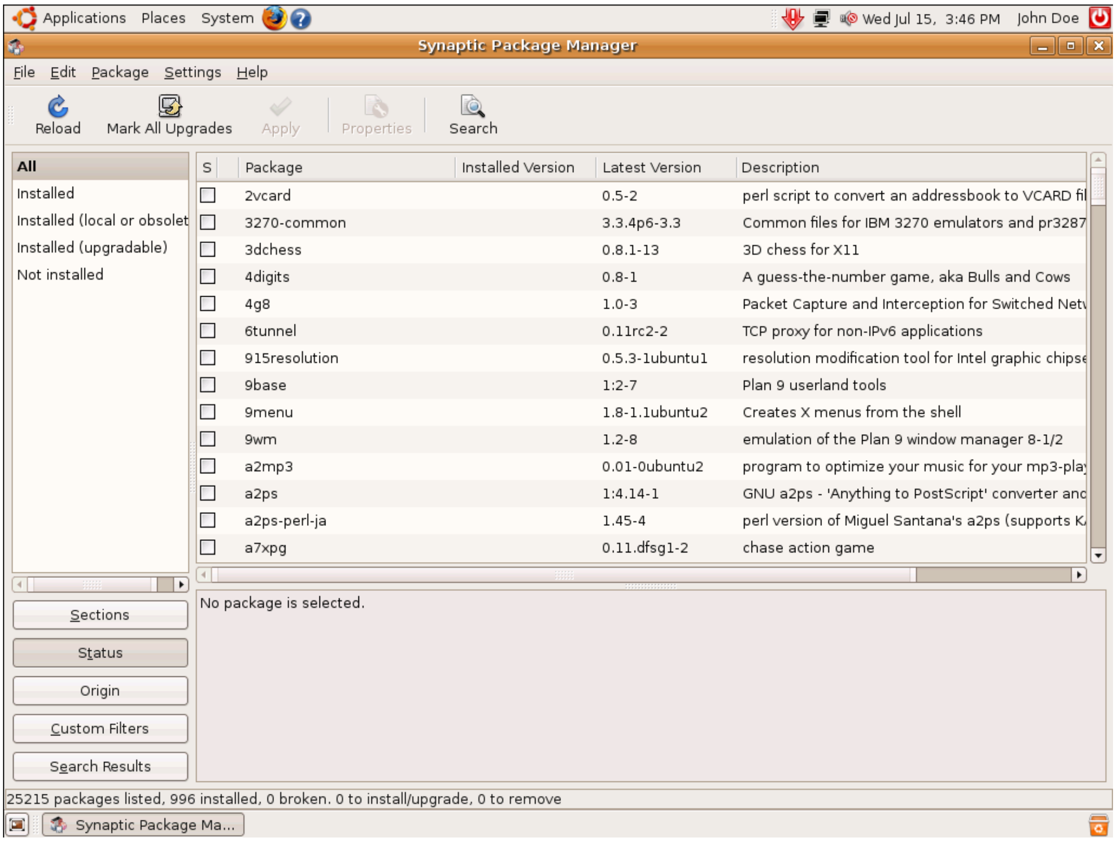
Figure 21 Synaptic Package Manager Menu

Synaptic application is a graphical package management program for apt. It provides the same features as the apt-get command line utility with a GUI front-end based on Gtk+.

Features:

- Install, remove and upgrade any packages or whole system.
- Manage package repositories (sources, list).
- Find packages by name, description and several other attributes.
- Browse all available online documentation related to a package.

- Download the latest changelog of a package.
- Lock packages to the current version.
- Force the installation of a specific package version.
- Undo/Redo of selections.
- Built-in terminal emulator for the package manager.
- And select the packages you wish to install.



S	Package	Installed Version	Latest Version	Description
□	2vcard		0.5-2	perl script to convert an addressbook to VCARD fil
□	3270-common		3.3.4p6-3.3	Common files for IBM 3270 emulators and pr3287
□	3dchess		0.8.1-13	3D chess for X11
□	4digits		0.8-1	A guess-the-number game, aka Bulls and Cows
□	4g8		1.0-3	Packet Capture and Interception for Switched Netv
□	6tunnel		0.11rc2-2	TCP proxy for non-IPv6 applications
□	915resolution		0.5.3-1ubuntu1	resolution modification tool for Intel graphic chipse
□	9base		1:2-7	Plan 9 userland tools
□	9menu		1.8-1.1ubuntu2	Creates X menus from the shell
□	9wm		1.2-8	emulation of the Plan 9 window manager 8-1/2
□	a2mp3		0.01-0ubuntu2	program to optimize your music for your mp3-play
□	a2ps		1:4.14-1	GNU a2ps - 'Anything to PostScript' converter anc
□	a2ps-perl-ja		1.45-4	perl version of Miguel Santana's a2ps (supports K/
□	a7xpg		0.11.dfsg1-2	chase action game

No package is selected.

25215 packages listed, 996 installed, 0 broken. 0 to install/upgrade, 0 to remove

Figure 22 Synaptic Package Manager program

3.2.2 COMMAND LINE TEXT PROCEDURE

You can also use the “apt-get” command line program to install and remove software packages from your system (one at a time or many at once), as well as update your system's internal listing of what packages are available from your currently selected list or repositories.

Open bash command line terminal…

To install a “*NameOfPackage*” package:

```
# apt-get  install NameOfPackage
```

To reinstall a package

```
# apt-get install --reinstall NameOfPackage
```

To remove a package:

```
# apt-get remove NameOfPackage
```

to remove a package and configuration file:

```
# apt-get --purge remove NameOfPackage
```

To search for a package:

```
# apt-cache search NameOfPackage
```

To update the repository (list of available .deb's):

```
# apt-get update
```

To upgrade your system (can be useful in maintaining an up-to-date system):

```
# apt-get upgrade
```

To see a short list of common commands:

```
# apt-get --help
```

3.3 CROSS-DEVELOPMENT AND SDK

Cross-development in general, refers to the overall software development process that eventually produces a single application or a complete system running on an platform that is different from the development platform. This is accomplished using a cross-compiler toolchain and cross-compiled libraries.

Peter Seebach defines cross-compilation as follows: "*Cross compilation occurs when a compiler running on one system produces executables for another system -- this is an important concept when the target system doesn't have a native set of compilation tools, or when the host system is faster or has greater resources.*"

Cross-development usually involves two different platforms, the host platform where actual development work takes place, and the target platform where the final application is tested and run.

The IGEPv2 SDK Virtual Machine provides a Linux based desktop computer properly configured for applications development with a binary cross-compiler toolchain. Development host is setup with:

- A TFTP server to get kernel image for the target
- A NFS server to use network files system for the root filesystem for the target.
- A software development kit (SDK), a set of development tools that allows a software engineer to create applications for a certain software package, software framework or hardware platform.
- Several root filesystem (rootfs) binary images for the ISEE targets
- A kernel image for the ISEE targets
- A free graphical C++ IDE (Code::blocks)

3.3.1 CROSS-DEVELOPMENT AND SDK GENERAL SPECIFICATION

POKY, embedded Linux platform builder

The Poky Linux SDK (http://www.pokylinux.org/) is the main development infrastructure used by ISEE IGEPv2 team. It is improved SDK based on OpenEmbedded project (http://www.openembedded.org).

The Poky Linux SDK will sit neatly on top of any device using the GNOME Mobile software stack, providing a well defined user experience. Poky Linux has a growing open source community backed up by the principal developer and maintainer of Poky, OpenedHand? Ltd. The user interface environment used by Poky Linux is

Sato; it is designed to work well with screens at very high DPI and restricted size. It is coded with focus on efficiency and speed so that it works smoothly on hand-held and other embedded hardware.

GCC

GNU Compiler Collection (http://gcc.gnu.org), "*arm-none-linux-gnueabi*" architecture.

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, and Ada, as well as libraries for these languages (libstdc++, libgcj...).

GCC development is a part of the GNU Project, aiming to improve the compiler used in the GNU system including the GNU/Linux variant. The GCC development effort uses an open development environment and supports many other platforms in order to foster a world-class optimizing compiler, to attract a larger team of developers, to ensure that GCC and the GNU system work on multiple architectures and diverse environments, and to more thoroughly test and extend the features of GCC.

Libc

GNU libc library (http://www.gnu.org/software/libc/libc.html)

Any Unix-like operating system needs a C library: the library which defines the "system calls'' and other basic facilities such as open, malloc, printf, exit...

The GNU C library is used as the C library in the GNU system and most systems with the Linux kernel.

Linux

ISEE SDK runs Linux kernel (http://www.kernel.org) and their drivers for IGEPv2 BOARD hardware (http://git.igep.es).

Linux is a clone of the operating system UNIX, written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX and Single UNIX Specification compliance.

It has all the features you would expect in a modern fully-fledged UNIX, including true multitasking, virtual memory, shared libraries, demand loading, shared copy-on-write executables, proper memory management, and multistack networking including IPv4 and IPv6.

Although originally developed first for 32-bit x86-based PCs (386 or higher), today Linux also runs on (at least) the Alpha AXP, Sun SPARC, Motorola 68000, PowerPC, ARM, Hitachi SuperH, IBM S/390, MIPS, HP PA-RISC, Intel IA-64, AMD x86-64, AXIS CRIS, Renesas M32R, Atmel AVR32, Renesas H8/300, NEC V850, Tensilica

Xtensa, and Analog Devices Blackfin architectures; for many of these architectures in both 32- and 64-bit variants.

Linux is easily portable to most general-purpose 32- or 64-bit architectures as long as they have a paged memory management unit (PMMU) and a port of the GNU C compiler (gcc) (part of The GNU Compiler Collection, GCC). Linux has also been ported to a number of architectures without a PMMU, although functionality is then obviously somewhat limited. See the µClinux project for more info.

Root File System (rootfs / user space binary applications)

ISEE provides several binary demo rootfs images. They are based on Poky Linux (http://www.pokylinux.org/) or Open Embedded (http://www.openembedded.org/).

They could be built from scratch using Poky Linux infrastructure. It allows developers to create their own complete Linux Distribution for their embedded systems.

Some Poky Linux advantages:

- support for many hardware architectures
- multiple releases for those architectures
- tools for speeding up the process of recreating the base after changes have been made
- easy to customize
- runs on any Linux distribution
- cross-compiles 1000's of packages including GTK+, Xwindows, Mono, Java, and about anything else you might ever need

3.4 FIRST STEPS WITH IGEPV2 SDK VIRTUAL MACHINE

Virtual Machine emphasizes "simple" and does not require you to spend many hours trying to set up a development environment.

During development, the target system cans NFS-mount its root filesystem from your file server to provide a complete diskless Linux system. The target system will attempt to mount its root filesystem from the server as /srv/nfs/<distro>/<project>/<machine>, where

- <distro> is poky
- <project> is poky-image-sato
- <machine> is igep0020

You could change the U-Boot environment using serial link (or also from userspace)

NOTE: You will find more updated and detailed manuals at http://labs.igep.es or https://labs.igep.es/index.php/How_to_modify_the_uboot_environment_from_userspace

IGEPv2 SDK Virtual Machine provides a demo root filesystem located into /srv/nfs/poky/poky-image-sato/igep0020 which can be mounted via NFS. Follow next steps to run the demo.

Next, power up your board and stop u-boot's autoboot by pressing a key on your **serial link** console.

IMPORTANT NOTE: It is typical you try to stop uboot by pressing a key on the USB keyboard connected to IGEPv2 board. Do not try this. It doesn't work. You need to use serial link (serial cable connected from J990 IGEPv2 connecter to your computer serial port)

Now, it's time to change the U-Boot environment.

```
U-Boot> setenv serverip <your VM ip>

U-Boot> setenv distro poky

U-Boot> setenv machine igep0020
```

Setup project selection:

```
U-Boot> setenv project poky-image-sato   // poky sato
```

or

```
U-Boot> setenv project poky-image-minimal   // poky minimal
```

Then run “nfs-boot” u-boot command to boot from virtual machine NFS exported demo:

```
U-Boot> setenv bootcmd 'run nfs-boot'
```

If you want to set as default boot option, you can also save your new environment

```
U-Boot> saveenv
```

The last step is run the boot command.

```
U-Boot> run bootcmd
```

If all is ok, something like this should appear in the DVI monitor connected to IGEPv2 board.

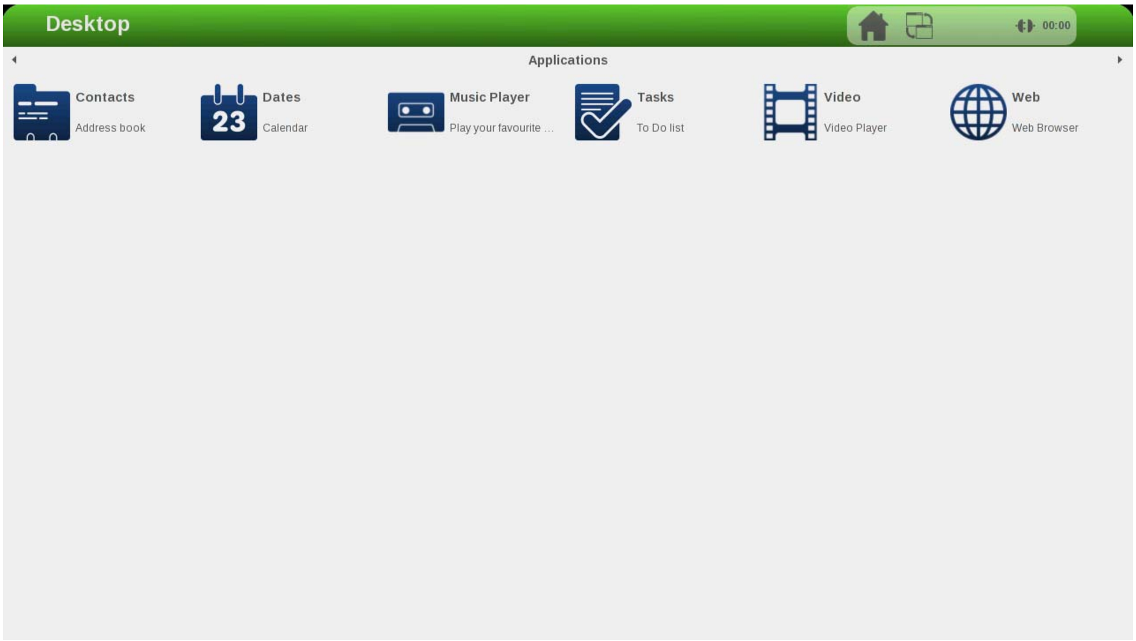


Figure 23 Poky-sato Desktop

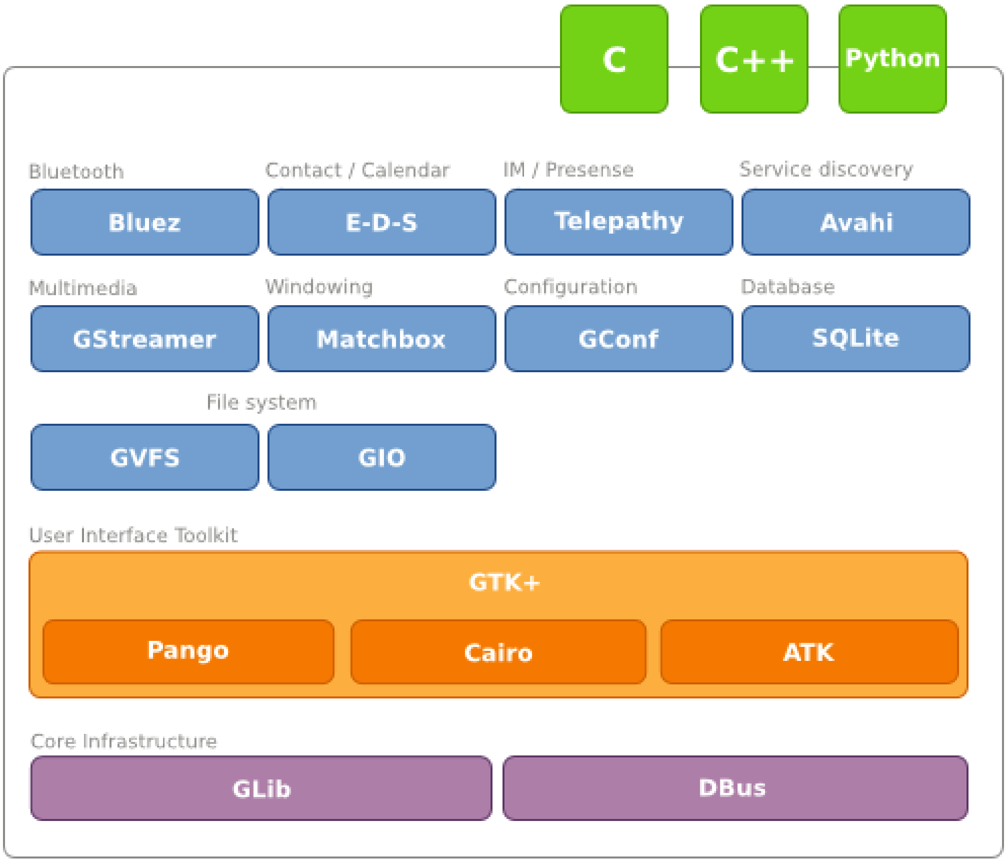
NOTE: You will find more updated and detailed manuals at http://labs.igep.es

3.5 POKY LINUX SOFTWARE DEVELOPMENT KIT

IGEPv2 SDK Virtual Machine provides a Poky Linux SDK. The Poky Linux is primarily a small linux distribution based on open source software. Poky SDK is located into the /usr/local/poky directory and contain a setup script, which can be sourced to initialize a suitable environment. After sourcing this, the compiler, a special version of pkgconfig and other useful utilities are added to the PATH. Variables to assist pkgconfig and autotools are also set.

3.5.1 ARCHITECTURE

The Poky Linux SDK will sit neatly on top of any device using the GNOME Mobile software stack, providing a well defined user experience. Poky Linux has a growing open source community backed up by the principal developer and maintainer of Poky, OpenedHand? Ltd. The user interface environment used by Poky Linux is Sato, it is designed to work well with screens at very high DPI and restricted size. It is coded with focus on efficiency and speed so that it works smoothly on hand-held and other embedded hardware.



The GNOME Mobile is a diverse stack of open source, mobile application development technologies that includes the GTK toolkit for interface construction, the GConf application configuration service which leverages XML for data persistence, the extensible GnomeVFS file access abstraction layer which provides support for network transparent file manipulation, the highly flexible GStreamer multimedia framework which supports dynamic media editing as well as playback, the powerful D-Bus interprocess communication system, the BlueZ Bluetooth stack, the nascent Telepathy instant messaging and presence management framework, and the Avahi service for Zeroconf service discovery. Please refer to http://www.gnome.org/mobile for a more detailed description.

3.5.2 BUILD SOFTWARE USING THE POKY LINUX SDK

First of all you need to initialize a suitable environment in a bash console inside your virtual machine. You can do this sourcing once the environment-setup script.

```
$ source /usr/local/poky/eabi-glibc/arm/environment-setup
```

Create a single .c file (hello-world.c), using your preferred editor

```
#include <stdio.h>

int main (int argc, char **argv)

{

  printf("Hello world !\n");

  return 0;

}
```

3.5.2.1 BUILD YOUR PROGRAM USING BUILT-IN CROSS TOOLCHAIN

Cross toolchain tools are available into the built-in virtual machine Poky SDK.

You only need open bash terminal prompt and write commands:

```
$ arm-none-linux-gnuabi-gcc -o hello-world hello-world.c

$ file hello-world

hello-world: ELF 32-bit LSB executable, ARM, version 1 (SYSV), for
GNU/Linux 2.6.14, dynamically linked (uses shared libs), not stripped
```

Copy the binary result to your NFS-mounted root filesystem on your local virtual machine path

```
$ sudo cp hello-world /srv/nfs/poky/poky-image-sato/igep0020a/home/root
[sudo] password for jdoe: letmein
```

And run the program in your target device (serial console or ssh network terminal)

```
# cd /home/root

# ./hello-world

Hello World !
```

NOTE: This HOW TO could be deprecated. Please refer to http://labs.igep.es to the last up to date revision.

3.5.2.2 BUILD YOUR PROGRAM USING AUTOTOOL

Using the SDK with autotool enabled packages is straightforward; just pass the appropriate host option to configure

```
$ ./configure --target=arm-none-linux-gnueabi --host=arm-none-linux-gnueabi
```

For example, you can download the GNU Hello program and build for your device

```
$ wget http://ftp.gnu.org/gnu/hello/hello-2.4.tar.gz

$ tar xzf hello-2.4.tar.gz

$ cd hello-2.4

$ ./configure --target=arm-none-linux-gnueabi --host=arm-none-linux-gnueabi

$ make

$ file src/hello

src/hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), for GNU/Linux 2.6.14, dynamically linked (uses shared libs), not stripped
```

And run the program in your target device (serial console or ssh network terminal)

```
# cd /home/root

# ./hello-world

Hello World !
```

NOTE: This HOW TO could be deprecated. Please refer to http://labs.igep.es to the last up to date revision.

3.5.2.3 INSTALL IPK SOFTWARE PACKAGE

You could get a binary ipk package with already compiled software. In this case, you will download some “*NameOfPackage*”.ipk package inside IGEPv2 board. Then, you could install ipk package with “opkg” tool like any distribution with package manager system (like “dpkg” in Debian/Ubuntu, “rpm” in Fedora/CentOS...)

1. copy inside IGEPv2 board a ipk software package from some repository or your own built ipk package

```
# scp NameOfPackage.ipk .
```

2. Install new packages from inside IGEPv2 board (ssh terminal)

```
# opkg install <NameOfPackage.ipk>
```

3. **That’s all!!!**

Other useful opkg commands are:

- To view the list of available packages:

```
# opkg list
```

- or view the list of installed packages:

```
# opkg list_installed
```

- Held for more available options:

```
# opkg --help
```

If you also need some software that is typically not found in the repository, either because it's too new or too specific, you can build and then copy in your root filesystem image (**see chapter “3.5.2 BUILD SOFTWARE USING THE POKY LINUX SDK”**).

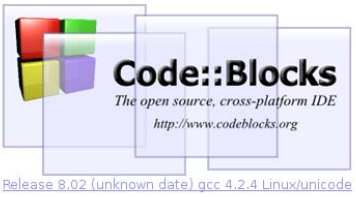
For example, with an autotooled package you can do this

```
$ ./configure --target=arm-none-linux-gnueabi --host=arm-none-linux-gnueabi --prefix=/srv/nfs/poky/myrootfs/igep0020

$ make

$ sudo make install
```

NOTE: This HOW TO could be deprecated. Please refer to http://labs.igep.es to the last up to date revision.

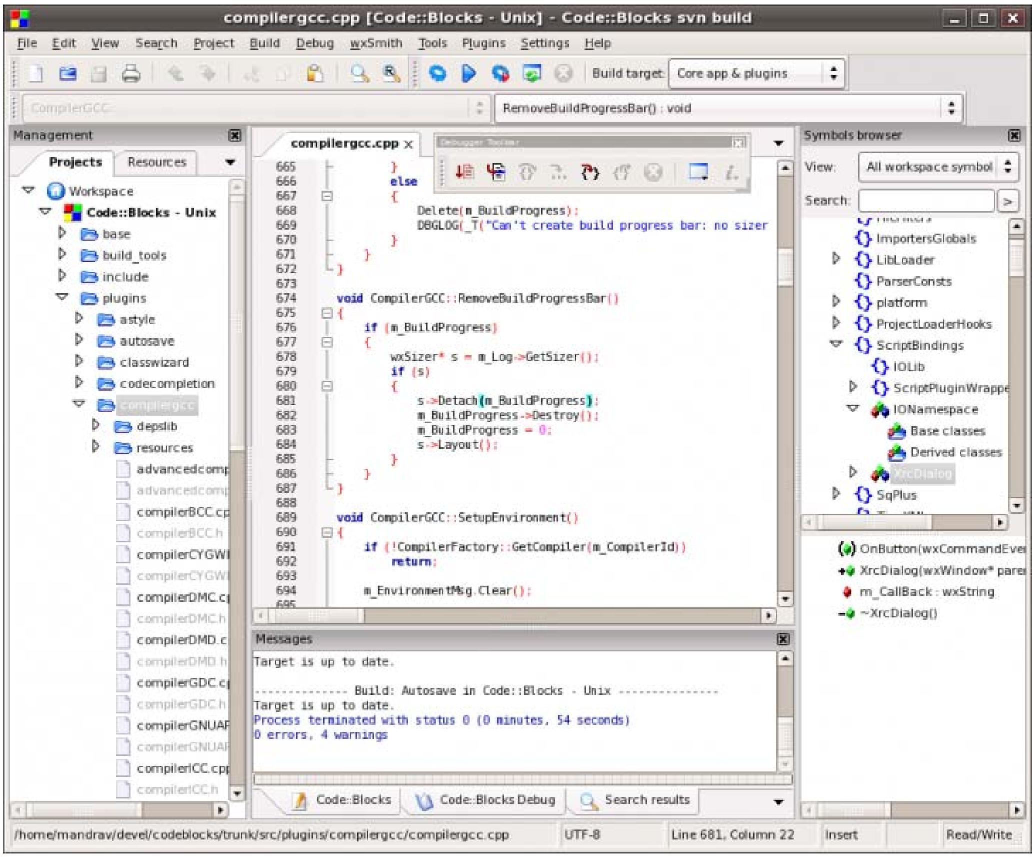
3.5.3 BUILD SOFTWARE USING CODE::BLOCKS IDE

Code::Blocks is a free C++ IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.


Code::Blocks
The open source, cross-platform IDE
http://www.codeblocks.org
Release 8.02 (unknown date) gcc 4.2.4 Linux/unicode

An IDE with all the features user needs, having a consistent look, feel and operation across platforms.

Built around a plugin framework, Code::Blocks can be extended with plugins. Any kind of functionality can be added by installing/coding a plugin. For instance, compiling and debugging functionality is already provided by plugins!

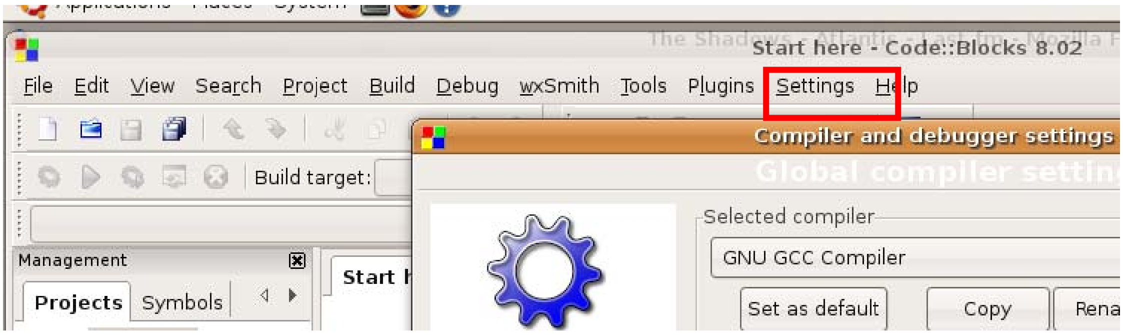


```
            }
            else
            {
                Delete(m_BuildProgress);
                DBGLOG(_T("Can't create build progress bar: no sizer
            }
        }
}

void CompilerGCC::RemoveBuildProgressBar()
{
    if (m_BuildProgress)
    {
        wxSizer* s = m_Log->GetSizer();
        if (s)
        {
            s->Detach(m_BuildProgress);
            m_BuildProgress->Destroy();
            m_BuildProgress = 0;
            s->Layout();
        }
    }
}

void CompilerGCC::SetupEnvironment()
{
    if (!CompilerFactory::GetCompiler(m_CompilerId))
        return;

    m_EnvironmentMsg.Clear();
```

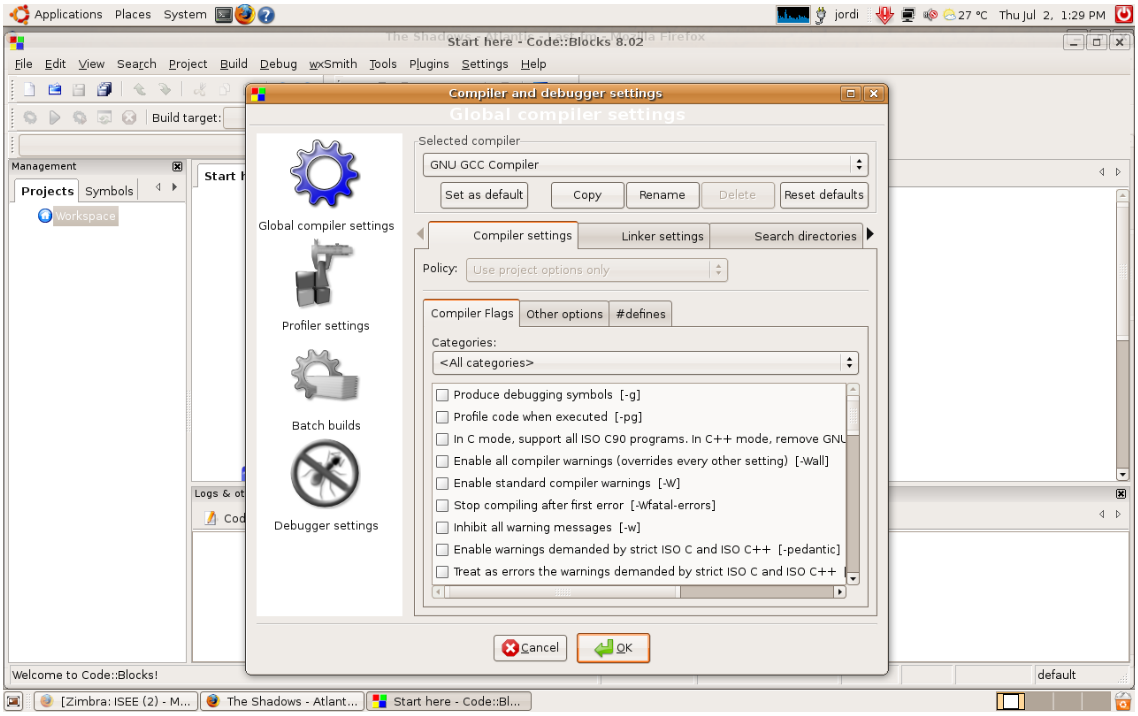
```
Target is up to date.

-------------- Build: Autosave in Code::Blocks - Unix ---------------
Target is up to date.
Process terminated with status 0 (0 minutes, 54 seconds)
0 errors, 4 warnings
```

Code::Blocks configuration

Go to Compiler and debugger Setting menu: Settings--> Compiler and debugger


Start here - Code::Blocks 8.02
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
Build target:
Management
Projects Symbols
Start h
Compiler and debugger settings
Global compiler settin
Selected compiler
GNU GCC Compiler
Set as default
Copy
Rena

The compiler and debugger settings window will appear:



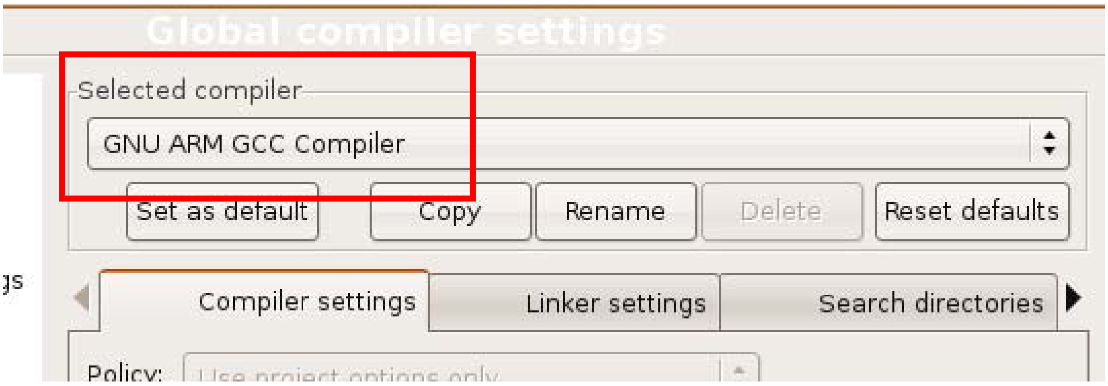
Selected compiler: GNU GCC Compiler

Policy: Use project options only

Categories: <All categories>

- ☐ Produce debugging symbols [-g]
- ☐ Profile code when executed [-pg]
- ☐ In C mode, support all ISO C90 programs. In C++ mode, remove GNU
- ☐ Enable all compiler warnings (overrides every other setting) [-Wall]
- ☐ Enable standard compiler warnings [-W]
- ☐ Stop compiling after first error [-Wfatal-errors]
- ☐ Inhibit all warning messages [-w]
- ☐ Enable warnings demanded by strict ISO C and ISO C++ [-pedantic]
- ☐ Treat as errors the warnings demanded by strict ISO C and ISO C++

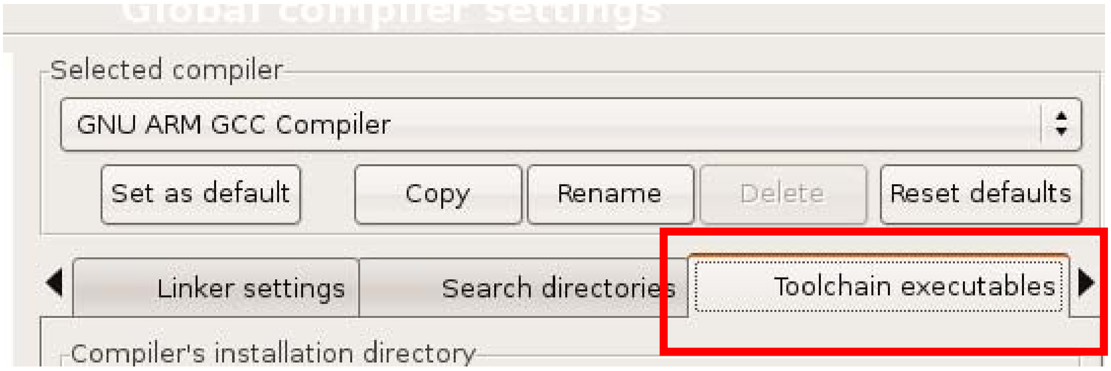
Choose GNU ARM GCC Compiler from Selected compiler window.



Selected compiler: GNU ARM GCC Compiler

Policy: Use project options only

Go to Toolchain executables window.



Compiler’s installation directory: */usr/local/poky/eabi-glibc/arm/*

Program Files

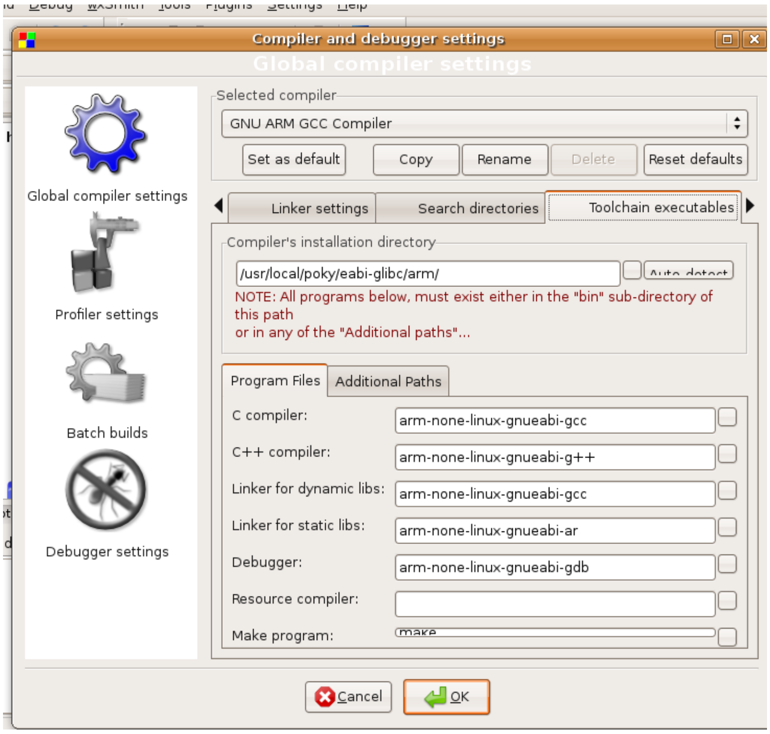
C Compiler: *arm-none-linux-gnueabi-gcc*

C++ Compiler: *arm-none-linux-gnueabi-g++*

Linker for dynamic libs: *arm-none-linux-gnueabi-gcc*

C Linker for static libs: *arm-none-linux-gnueabi-ar*

Debugger: *arm-none-linux-gnueabi-gdb*



Selected compiler: GNU ARM GCC Compiler

Compiler's installation directory: /usr/local/poky/eabi-glibc/arm/

NOTE: All programs below, must exist either in the "bin" sub-directory of this path
or in any of the "Additional paths"...

C compiler:	arm-none-linux-gnueabi-gcc
C++ compiler:	arm-none-linux-gnueabi-g++
Linker for dynamic libs:	arm-none-linux-gnueabi-gcc
Linker for static libs:	arm-none-linux-gnueabi-ar
Debugger:	arm-none-linux-gnueabi-gdb
Resource compiler:	
Make program:	make

3.5.4 INSTALL ADDITIONAL LIBRARIES INTO THE POKY LINUX SDK

Sooner or later you will want to compile an application that has dependencies which can't be found inside the software development kit. Like install new packages into your rootfs. The “opkg-target” tool manages and installs ipk packages into the SDK installed inside your virtual machine (or workstation).

```
$ source /usr/local/poky/eabi-glibc/arm/environment-setup

$ opkg-target install <NameOfPackage.ipk>
```

To get a list of all available packages:

```
$ opkg-target list
```

To view the list of already installed packages:

```
$ opkg-target list_installed
```

NOTE: This HOW TO could be deprecated. Please refer to http://labs.igep.es to the last up to date revision.

4 IGEPV2 SDK TRICKS

4.1 HOW TO CONNECT CABLE TO IGEPV2 SERIAL DEBUG PORT

Support for RS232 is provided by a 10 pin header on the IGEPV2 Board for access to an onboard RS232 transceiver.

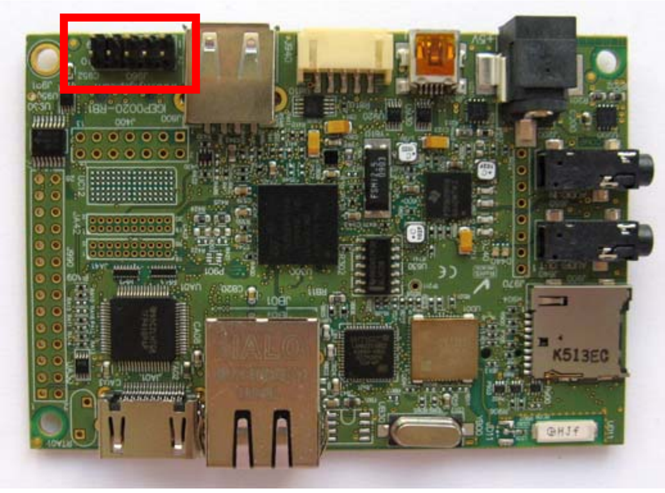


Figure 24 IDC10 header

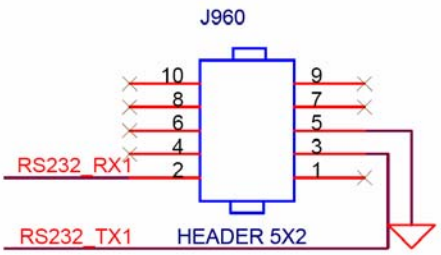


Figure 25 J960 schematic

NOTE:

Look out for pin 1, it is a polarity connector.

Wrong use could damage your board!!!

User will require an IDC10 to DB9 flat cable like the one shown below:

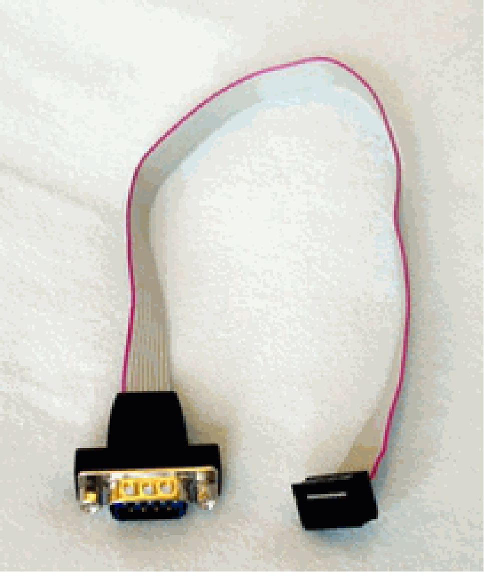
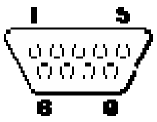
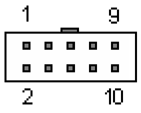


Figure 26 IDC10 to DB9 cable

Connectors are numbered:

DB9	IDC10
	
Pin 1	Pin 1
Pin 2	Pin 2
Pin 3	Pin 3
Pin 4	Pin 4
Pin 5	Pin 5
Pin 6	Pin 6
Pin 7	Pin 7
Pin 9	Pin 9
NC	Pin 10

4.2 HOW TO OPEN A LINUX CONSOLE TO IGEPV2 BOARD

You can user Serial debug port interface or Ethernet.

4.2.1 SERIAL CONSOLE FROM LINUX

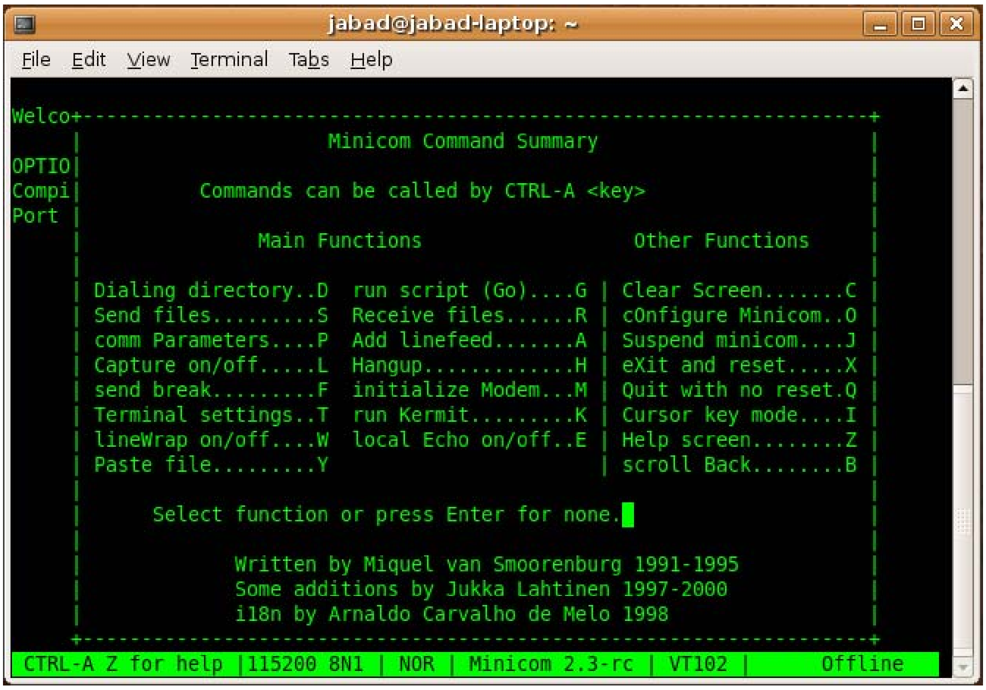
Minicom is a text-based modem control and terminal emulation program for Unix-like operating systems. Minicom includes a dialing directory, ANSI and VT100 emulation, an (external) scripting language, and other features. Minicom is a menu-driven communications program. It also has an auto zmodem download.

Minicom is installed on IGEPV2 VM SDK.

Execute minicom.

```
$ minicom
```

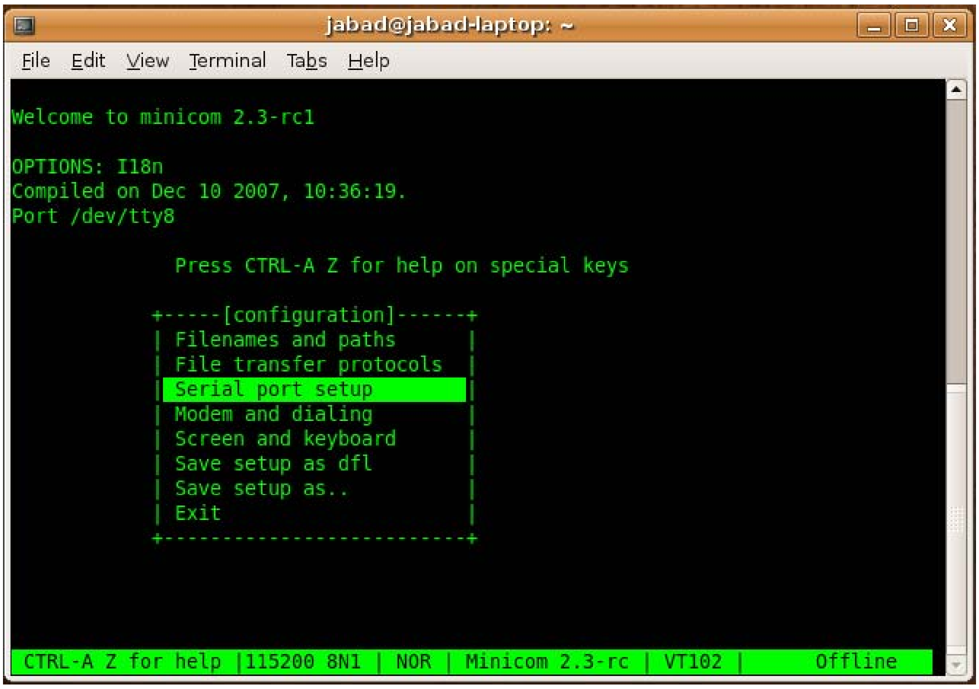
Go to the Minicom Command Summary: Ctrl-A Z.



```
Welco+-------------------------------------------------------------------+
     |                    Minicom Command Summary                        |
OPTIO|                                                                   |
Compi|          Commands can be called by CTRL-A <key>                   |
Port |                                                                   |
     |               Main Functions                  Other Functions     |
     |                                                                   |
     | Dialing directory..D  run script (Go)....G | Clear Screen.......C |
     | Send files.........S  Receive files......R | cOnfigure Minicom..O |
     | comm Parameters....P  Add linefeed.......A | Suspend minicom....J |
     | Capture on/off.....L  Hangup.............H | eXit and reset.....X |
     | send break.........F  initialize Modem...M | Quit with no reset.Q |
     | Terminal settings..T  run Kermit.........K | Cursor key mode....I |
     | lineWrap on/off....W  local Echo on/off..E | Help screen........Z |
     | Paste file.........Y                       | scroll Back........B |
     |                                                                   |
     |         Select function or press Enter for none.                  |
     |                                                                   |
     |              Written by Miquel van Smoorenburg 1991-1995          |
     |              Some additions by Jukka Lahtinen 1997-2000           |
     |              i18n by Arnaldo Carvalho de Melo 1998                |
     +-------------------------------------------------------------------+
 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3-rc | VT102 |      Offline
```

Figure 27 Minicom Command Summary

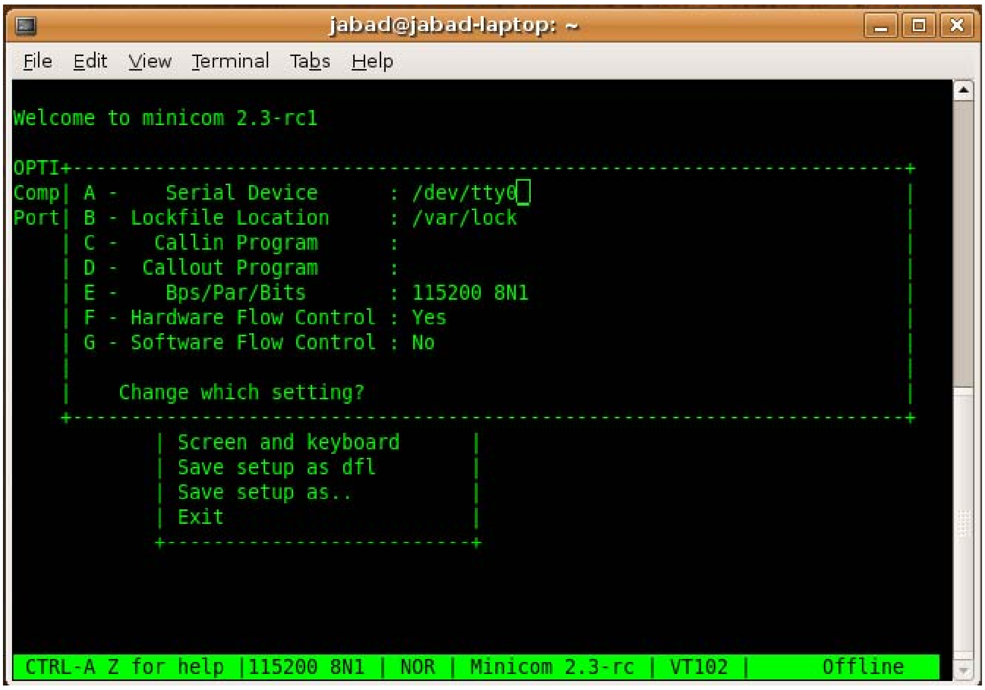
Press O to Configure Minicom, and Select Serial port setup option.



```
Welcome to minicom 2.3-rc1

OPTIONS: I18n
Compiled on Dec 10 2007, 10:36:19.
Port /dev/tty8

              Press CTRL-A Z for help on special keys

            +-----[configuration]------+
            | Filenames and paths      |
            | File transfer protocols  |
            | Serial port setup        |
            | Modem and dialing        |
            | Screen and keyboard      |
            | Save setup as dfl        |
            | Save setup as..          |
            | Exit                     |
            +--------------------------+



 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3-rc | VT102 |      Offline
```

Figure 28 Minicom Configure

Chose Serial port setup option, and configure your Serial Device port, and Bps/Parity/Bits: 115200 8N1



```
Welcome to minicom 2.3-rc1

OPTI+-----------------------------------------------------------------------+
Comp| A -    Serial Device      : /dev/tty0                                  |
Port| B - Lockfile Location     : /var/lock                                  |
    | C -   Callin Program      :                                            |
    | D -  Callout Program      :                                            |
    | E -    Bps/Par/Bits       : 115200 8N1                                 |
    | F - Hardware Flow Control : Yes                                        |
    | G - Software Flow Control : No                                         |
    |                                                                        |
    |    Change which setting?                                               |
    +-----------------------------------------------------------------------+
            | Screen and keyboard      |
            | Save setup as dfl        |
            | Save setup as..          |
            | Exit                     |
            +--------------------------+



 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3-rc | VT102 |      Offline
```

Figure 29 Minicom Serial port Setup

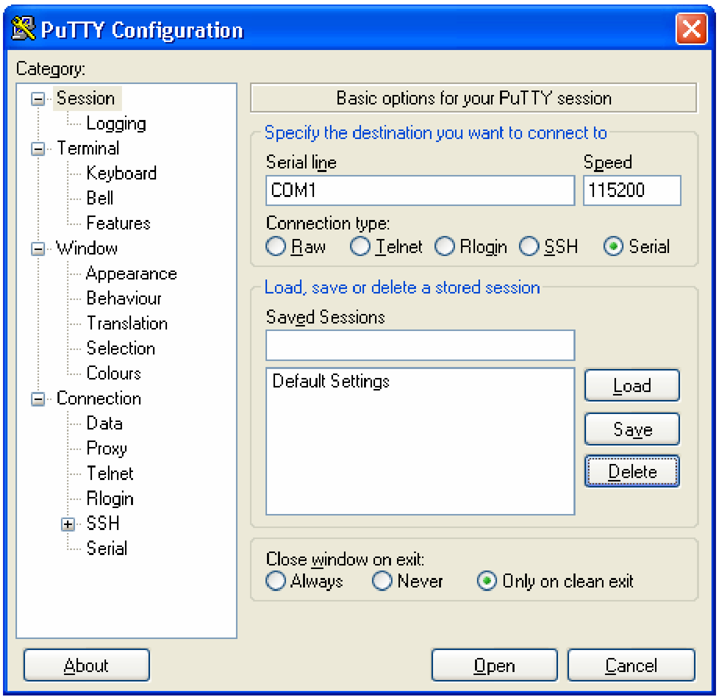
4.2.2 SERIAL CONSOLE FROM WINDOWS

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download it from:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

You can use *putty* to open a console using the serial debug port.

Open putty. Choose Serial line. Configure Speed to 115200. Select Serial Connection type. Press on Open button.



Serial line: COM1
Speed: 115200
Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 30 Putty main window

4.2.3 REMOTE TCP/IP SECURE SHELL CONSOLE FROM LINUX

Open a Linux console via Ethernet or WiFi interface link:

```
# by default IGEPV2 board comes configured with IP 192.168.254.254

$ ssh root@192.168.254.254

password: letmein
```

If you are not in the range 192.168.254.xxx, you can configure an alias to your Ethernet interface into your IGEPv2 SDK virtual machine:

```
$ sudo ifconfig eth0:alias0 192.168.254.xxx
```

If you want to change the IP address:

```
$ sudo ifconfig eth0 192.168.xxx.xxx
```

4.2.4 REMOTE TCP/IP SECURE SHELL CONSOLE FROM WINDOWS

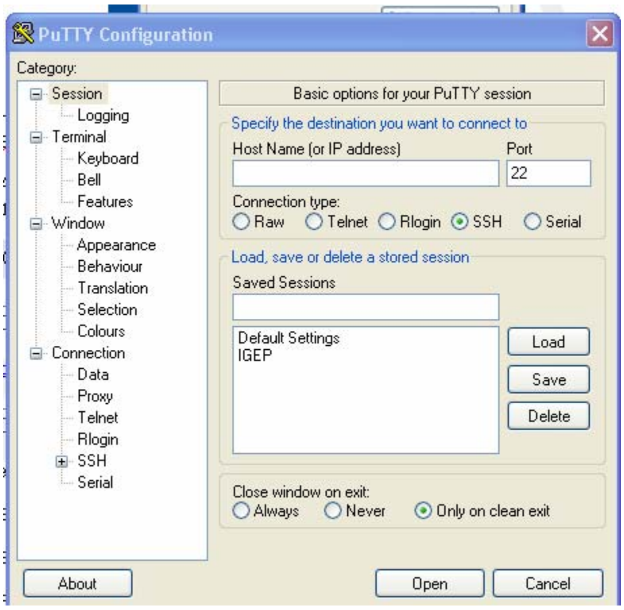
Open putty.

Choose SSH Connection type.

Write the IP Address

Select port 22

Press on button Open.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
IGEP

Close window on exit:
○ Always ○ Never ● Only on clean exit

4.3 IGEPV2 BOARD SERIAL DEBUG LINK BOOTLOG EXAMPLE

Just for user information, here you have the boot up traces…

These traces have been captured through the serial debug port.

```
60
Texas Instruments X-Loader 1.4.4-0 (Oct 21 2010 - 21:21:22)
Loading u-boot.bin from onenand


U-Boot 2010.06-0 (Nov 02 2010 - 10:07:55)

OMAP3630/3730-GP ES2.0, CPU-OPP2, L3-165MHz
IGEP v2 board + LPDDR/ONENAND
I2C:   ready
DRAM:  512 MiB
Muxed OneNAND(DDP) 512MB 1.8V 16-bit (0x58)
OneNAND version = 0x0031
Chip support all block unlock
Chip has 2 plane
block = 2048, wp status = 0x2
Scanning device for bad blocks
Bad eraseblock 176 at 0x01600000
Bad eraseblock 2648 at 0x14b00000
Bad eraseblock 2649 at 0x14b20000
OneNAND: 512 MiB
OneNAND: Read environment from 0x00200000
In:    serial
Out:   serial
Err:   serial
Die ID #4ee800011ff0000001592f350102302a
Net:   smc911x-0
Hit any key to stop autoboot:  0
No MMC card found

** Unable to use mmc 0:1 for fatload **

** Unable to use mmc 0:1 for fatload **
smc911x: detected LAN9221 controller
smc911x: autonegotiation timed out
smc911x: MAC ac:de:48:00:02:54
Using smc911x-0 device
ping failed; host 192.168.254.10 is not alive

OneNAND read: offset 0x280000, size 0x300000
 3145728 bytes read: OK
## Booting kernel from Legacy Image at 80000000...
   Image Name:   OpenedHand Linux (Poky)/2.6.35.7
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2881420 Bytes = 2.7 MiB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
```

```
OK

Starting kernel...

Uncompressing Linux... done, booting the kernel.
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 2.6.35.7 (eballetbo@jupiter) (gcc version 4.3.3 (GCC) ) #1 Tue Nov 2
09:52:51 CET 2010
[    0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c53c7f
[    0.000000] CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
[    0.000000] Machine: IGEP v2 board
[    0.000000] Reserving 14680064 bytes SDRAM for VRAM
[    0.000000] Memory policy: ECC disabled, Data cache writeback
[    0.000000] OMAP3630 ES1.2 (l2cache iva sgx neon isp 192mhz_clk )
[    0.000000] SRAM: Mapped pa 0x40200000 to va 0xfe400000 size: 0x100000
[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 130048
[    0.000000] Kernel command line: console=ttyS2,115200n8 console=tty0
omapfb.mode=dvi:1024x768MR-16@60 root=/dev/mtdblock4 rootfstype=jffs2
ip=192.168.254.254:192.168.254.10:192.168.254.10:255.255.255.0::eth0:
[    0.000000] PID hash table entries: 2048 (order: 1, 8192 bytes)
[    0.000000] Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
[    0.000000] Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
[    0.000000] Memory: 256MB 256MB = 512MB total
[    0.000000] Memory: 499460k/499460k available, 24828k reserved, 0K highmem
[    0.000000] Virtual kernel memory layout:
[    0.000000]     vector  : 0xffff0000 - 0xffff1000   (   4 kB)
[    0.000000]     fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
[    0.000000]     DMA     : 0xffc00000 - 0xffe00000   (   2 MB)
[    0.000000]     vmalloc : 0xe0800000 - 0xf8000000   ( 376 MB)
[    0.000000]     lowmem  : 0xc0000000 - 0xe0000000   ( 512 MB)
[    0.000000]     modules : 0xbf000000 - 0xc0000000   (  16 MB)
[    0.000000]       .init : 0xc0008000 - 0xc0037000   ( 188 kB)
[    0.000000]       .text : 0xc0037000 - 0xc0542000   (5164 kB)
[    0.000000]       .data : 0xc055e000 - 0xc05977a0   ( 230 kB)
[    0.000000] Hierarchical RCU implementation.
[    0.000000]  RCU-based detection of stalled CPUs is disabled.
[    0.000000]  Verbose stalled-CPUs detection is disabled.
[    0.000000] NR_IRQS:402
[    0.000000] Clocking rate (Crystal/Core/MPU): 26.0/200/800 MHz
[    0.000000] Reprogramming SDRC clock to 200000000 Hz
[    0.000000] dpll3_m2_clk rate change failed: -22
[    0.000000] GPMC revision 5.0
[    0.000000] IRQ: Found an INTC at 0xfa200000 (revision 4.0) with 96 interrupts
[    0.000000] Total of 96 interrupts on 1 active controller
[    0.000000] OMAP GPIO hardware version 2.5
[    0.000000] OMAP clockevent source: GPTIMER1 at 32768 Hz
[    0.000000] Console: colour dummy device 80x30
[    0.000000] console [tty0] enabled
[    0.000000] Calibrating delay loop... 799.29 BogoMIPS (lpj=3121152)
[    0.000000] pid_max: default: 32768 minimum: 301
[    0.000000] Mount-cache hash table entries: 512
[    0.000000] CPU: Testing write buffer coherency: ok
[    0.000000] regulator: core version 0.5
[    0.000000] regulator: dummy:
[    0.000000] NET: Registered protocol family 16
[    0.000000] IGEP2: Hardware Revision C (B-NON compatible)
[    0.000183] OMAP DMA hardware revision 5.0
[    0.012115] bio: create slab <bio-0> at 0
[    0.013824] SCSI subsystem initialized
[    0.015625] usbcore: registered new interface driver usbfs
[    0.015808] usbcore: registered new interface driver hub
[    0.015991] usbcore: registered new device driver usb
[    0.016418] i2c_omap i2c_omap.1: bus 1 rev4.0 at 2600 kHz
[    0.019531] twl4030: PIH (irq 7) chaining IRQs 368...375
[    0.019561] twl4030: power (irq 373) chaining IRQs 376..383
[    0.020019] twl4030: gpio (irq 368) chaining IRQs 384..401
[    0.022735] regulator: VUSB1V5: 1500 mV normal standby
[    0.023193] regulator: VUSB1V8: 1800 mV normal standby
[    0.023620] regulator: VUSB3V1: 3100 mV normal standby
[    0.024780] twl4030_usb twl4030_usb: Initialized TWL4030 USB module
[    0.025939] regulator: VMMC1: 1850 <--> 3150 mV at 3000 mV normal standby
[    0.026489] regulator: VDVI: 1800 mV normal standby
[    0.026672] i2c_omap i2c_omap.3: bus 3 rev4.0 at 100 kHz
[    0.028076] Advanced Linux Sound Architecture Driver Version 1.0.23.
[    0.028686] Switching to clocksource 32k_counter
[    0.042022] musb_hdrc: version 6.0, musb-dma, otg (peripheral+host), debug=0
[    0.044281] musb_hdrc musb_hdrc: USB OTG mode controller at fa0ab000 using DMA, IRQ 92
[    0.044647] NET: Registered protocol family 2
[    0.044799] IP route cache hash table entries: 4096 (order: 2, 16384 bytes)
[    0.045074] TCP established hash table entries: 16384 (order: 5, 131072 bytes)
[    0.045379] TCP bind hash table entries: 16384 (order: 4, 65536 bytes)
[    0.045562] TCP: Hash tables configured (established 16384 bind 16384)
[    0.045562] TCP reno registered
[    0.045593] UDP hash table entries: 256 (order: 0, 4096 bytes)
[    0.045623] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[    0.045776] NET: Registered protocol family 1
[    0.046051] RPC: Registered udp transport module.
[    0.046081] RPC: Registered tcp transport module.
[    0.046112] RPC: Registered tcp NFSv4.1 backchannel transport module.
[    0.048217] VFS: Disk quotas dquot_6.5.2
[    0.048278] Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[    0.048675] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[    0.049041] JFFS2 version 2.2. (NAND) Â© 2001-2006 Red Hat, Inc.
[    0.049652] Btrfs loaded
[    0.049682] msgmni has been set to 975
[    0.052093] alg: No test for stdrng (krng)
[    0.052154] io scheduler noop registered
[    0.052185] io scheduler deadline registered (default)
[    0.114990] OMAP DSS rev 2.0
[    0.115081] OMAP DISPC rev 3.0
[    0.115112] OMAP DSI rev 1.0
[    0.285766] Serial: 8250/16550 driver, 4 ports, IRQ sharing enabled
[    0.379364] serial8250.0: ttyS0 at MMIO 0x4806a000 (irq = 72) is a ST16654
[    0.567108] serial8250.1: ttyS1 at MMIO 0x4806c000 (irq = 73) is a ST16654
[    0.753631] serial8250.2: ttyS2 at MMIO 0x49020000 (irq = 74) is a ST16654
[    1.357238] console [ttyS2] enabled
[    1.453399] serial8250.3: ttyS3 at MMIO 0x49042000 (irq = 80) is a ST16654
[    1.556457] brd: module loaded
[    1.563262] loop: module loaded
[    1.568359] OneNAND driver initializing
[    1.572326] omap2-onenand omap2-onenand: initializing on CS0, phys base 0x20000000, virtual
base e0900000
[    1.582092] Muxed OneNAND(DDP) 512MB 1.8V 16-bit (0x58)
[    1.587402] OneNAND version = 0x0031
[    1.593353] Scanning device for bad blocks
[    1.605163] Bad eraseblock 176 at 0x01600000
[    1.716247] onenand_bbt_wait: ecc error = 0x2222, controller error 0x2400
[    1.723114] Bad eraseblock 2648 at 0x14b00000
[    1.727539] onenand_bbt_wait: ecc error = 0x2222, controller error 0x2400
[    1.734375] Bad eraseblock 2649 at 0x14b20000
[    1.800140] Creating 5 MTD partitions on "omap2-onenand":
[    1.805603] 0x000000000000-0x000000080000 : "X-Loader"
[    1.811950] 0x000000080000-0x000000200000 : "U-Boot"
[    1.817993] 0x000000200000-0x000000280000 : "Environment"
[    1.824432] 0x000000280000-0x000000580000 : "Kernel"
[    1.830474] 0x000000580000-0x000020000000 : "File System"
[    1.838439] vcan: Virtual CAN interface driver
[    1.843017] CAN device driver interface
[    1.846954] smsc911x: Driver version 2008-10-21.
[    1.852752] smsc911x-mdio: probed
[    1.856170] eth0: attached PHY driver [Generic PHY] (mii_bus:phy_addr=0:01, irq=-1)
[    1.864135] net eth0: MAC Address: ea:83:1c:35:39:94
[    1.869323] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.876037] ehci-omap.0 supply hsusb0 not found, using dummy regulator
[    1.882781] ehci-omap ehci-omap.0: OMAP-EHCI Host Controller
[    1.888549] ehci-omap ehci-omap.0: new USB bus registered, assigned bus number 1
[    1.896118] ehci-omap ehci-omap.0: irq 77, io mem 0x48064800
[    1.911651] ehci-omap ehci-omap.0: USB 2.0 started, EHCI 1.00
[    1.917510] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
[    1.924377] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    1.931671] usb usb1: Product: OMAP-EHCI Host Controller
[    1.937042] usb usb1: Manufacturer: Linux 2.6.35.7 ehci_hcd
[    1.942657] usb usb1: SerialNumber: ehci-omap.0
[    1.947875] hub 1-0:1.0: USB hub found
[    1.951721] hub 1-0:1.0: 3 ports detected
[    1.982177] g_ether gadget: using random self ethernet address
[    1.988067] g_ether gadget: using random host ethernet address
[    1.994415] usb0: MAC 56:2d:23:e1:4c:99
[    1.998321] usb0: HOST MAC 3e:f8:38:e4:16:00
[    2.002685] g_ether gadget: Ethernet Gadget, version: Memorial Day 2008
[    2.009399] g_ether gadget: g_ether ready
[    2.013458] musb_hdrc musb_hdrc: MUSB HDRC host driver
[    2.018646] musb_hdrc musb_hdrc: new USB bus registered, assigned bus number 2
[    2.026000] usb usb2: New USB device found, idVendor=1d6b, idProduct=0002
[    2.032867] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    2.040161] usb usb2: Product: MUSB HDRC host driver
[    2.045166] usb usb2: Manufacturer: Linux 2.6.35.7 musb-hcd
[    2.050781] usb usb2: SerialNumber: musb_hdrc
[    2.055847] hub 2-0:1.0: USB hub found
[    2.059661] hub 2-0:1.0: 1 port detected
[    2.064208] mice: PS/2 mouse device common for all mice
[    2.070526] input: twl4030_pwrbutton as /devices/platform/i2c_omap.1/i2c-1/1-
0049/twl4030_pwrbutton/input/input0
[    2.081268] i2c /dev entries driver
[    2.086212] mmci-omap-hs.0 supply vmmc_aux not found, using dummy regulator
[    2.094299] mmci-omap-hs.1 supply vmmc not found, using dummy regulator
[    2.101074] mmci-omap-hs.1 supply vmmc_aux not found, using dummy regulator
[    2.110656] usbcore: registered new interface driver hiddev
[    2.116516] usbcore: registered new interface driver usbhid
[    2.122131] usbhid: USB HID core driver
[    2.202880] No device for DAI omap-mcbsp-dai-0
[    2.207397] No device for DAI omap-mcbsp-dai-1
[    2.211944] No device for DAI omap-mcbsp-dai-2
[    2.216461] No device for DAI omap-mcbsp-dai-3
[    2.220947] No device for DAI omap-mcbsp-dai-4
[    2.225463] IGEP v2 SoC init
[    2.229125] asoc: twl4030 <-> omap-mcbsp-dai-0 mapping ok
[    2.239929] ALSA device list:
[    2.242950]   #0: igep2 (twl4030)
[    2.246520] TCP cubic registered
[    2.250000] NET: Registered protocol family 10
[    2.254882] lo: Disabled Privacy Extensions
[    2.259429] NET: Registered protocol family 17
[    2.263977] can: controller area network core (rev 20090105 abi 8)
[    2.270294] NET: Registered protocol family 29
[    2.274810] ThumbEE CPU extension supported.
[    2.280426] mmc1: new SDIO card at address 0001
[    2.285308] Power Management for TI OMAP3.
[    2.290954] VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 3
[    2.300231] fbcvt: 1024x768@60: CVT Name - .786M3-R
[    2.322662] Console: switching to colour frame buffer device 128x48
[    2.856964] net eth0: SMSC911x/921x identified at 0xe08d2000, IRQ: 336
[    3.865051] IP-Config: Complete:
[    3.868194]      device=eth0, addr=192.168.254.254, mask=255.255.255.0, gw=192.168.254.10,
[    3.876647]      host=192.168.254.254, domain=, nis-domain=(none),
[    3.882995]      bootserver=192.168.254.10, rootserver=192.168.254.10, rootpath=
[   11.991394] VFS: Mounted root (jffs2 filesystem) on device 31:4.
[   11.997680] Freeing init memory: 188K
[   18.267456] OMAP Watchdog Timer Rev 0x31: initial timeout 60 sec
[   21.321289] usbcore: registered new interface driver usbserial
[   21.411437] USB Serial support registered for generic
[   21.467712] usbcore: registered new interface driver usbserial_generic
[   21.474365] usbserial: USB Serial Driver core
[   21.591827] twl4030_wdt twl4030_wdt: Failed to register misc device
[   21.622680] twl4030_wdt: probe of twl4030_wdt failed with error -16
[   21.913604] USB Serial support registered for GSM modem (1-port)
[   21.967926] usbcore: registered new interface driver option
[   21.973541] option: v0.7.2:USB Driver for GSM modems
[   21.998626] twl_rtc twl_rtc: rtc core: registered twl_rtc as rtc0
[   22.032012] twl_rtc twl_rtc: Power up reset detected.
[   22.068847] twl_rtc twl_rtc: Enabling TWL-RTC.
[   22.496124] lib80211: common routines for IEEE802.11 drivers
[   22.904571] cfg80211: Calling CRDA to update world regulatory domain
[   23.398406] libertas_sdio: Libertas SDIO driver
[   23.402984] libertas_sdio: Copyright Pierre Ossman
[   24.187438] libertas: 5c:da:d4:39:00:15, fw 9.70.3p36, cap 0x00000303
[   24.193939] libertas: unidentified region code; using the default (USA)
[   24.205047] libertas: wlan0: Marvell WLAN 802.11 adapter
[   43.637023] Bluetooth: Core ver 2.15
[   43.642852] NET: Registered protocol family 31
[   43.647399] Bluetooth: HCI device and connection manager initialized
[   43.653778] Bluetooth: HCI socket layer initialized
[   43.730865] Bluetooth: L2CAP ver 2.14
[   43.734619] Bluetooth: L2CAP socket layer initialized
[   43.795715] Bluetooth: HIDP (Human Interface Emulation) ver 1.2
[   43.938812] Bluetooth: RFCOMM TTY layer initialized
[   43.943878] Bluetooth: RFCOMM socket layer initialized
[   43.949035] Bluetooth: RFCOMM ver 1.11

OpenedHand Linux (Poky) 3.3.1 igep0020 ttyS2

igep0020 login:
```

5 HOW TO..

5.1 SEE HTTP://LABS.IGEP.ES

NOTE: You will find more detailed updated HOWTOs at http://labs.igep.es

- http://labs.igep.es/index.php/Category:How_to_forge
- http://labs.igep.es/index.php/Ubuntu_8.04_IGEP_v2.0_SDK_Virtual_Machine

6 FAQS

6.1 SEE HTTP://LABS.IGEP.ES

NOTE: You will find more detailed updated FAQs at http://labs.igep.es

http://labs.igep.es/index.php/Frequently_Asked_Questions_and_Their_Answers

7 CHANGELOG

Revision 1.00

- Initial internal and partners version

Revision 1.01

- Initial public version

Revision 1.02

- General modifications to update manual to igep0020b.
- New Chapter 3.7.3 POKY-APPLICATION-SGX DEMOS
- Revision Chapter 3.8 Creating a custom root file system
- New Chapter 6. FAQS

Revision 1.03

- New Chapter HOW TO...
- Corrections on chapter 3.8

Revision 1.04

- Updated links

Revision 2.05

- Remake document
- Move often updated chapters to http://labs.igep.es public wiki site (FAQs, HOWTOs...)