Related Products	MOPSlcdGX1 (PGX1)
Subject	Using Linux on MOPSlcdGX1
Document Name	PGX1Linux_E111.doc
Usage	Common

1. REVISION HISTORY

Date	Document Name	Subjects added, changed, deleted	Changed by
06-Dec-02	PGX1Linux_E110.DOC	Initial release of Application Note	H. Bruhn
14-Jan-03	PGX1Linux_E111.DOC	Corrected layout problem with table 1	H. Bruhn

2. TABLE OF CONTENTS

1. REVISION HISTORY 1

2. TABLE OF CONTENTS 2

3. INTRODUCTION 3

4. GRAPHIC MODES 3

4.1. Frame buffer device instead of console 3

4.1.1. Frame buffer device already implemented in the kernel 3

4.1.2. Kernel without frame buffer device 4

4.2. Configuration of Xfree86 5

3. INTRODUCTION

The MOPSlcdGX1 equipped with the Geode GX1processor and the companion chip CS5530A (both of Natinal Semiconductor) requires special adjustments when using Linux as operating system. Herein we will describe the usage of a frame buffer device instead of the charset console and the usage of Xfree86.
This description is based on the distribution package SuSE 7.2. When using other distribution packages some different actions may be necessary.

4. GRAPHIC MODES

The built in graphic controller of the National Geode offers no certain X-Server. So the easiest way to use graphics under Linux is to use the frame buffer, which may be already built in the distribution package.

4.1. Frame buffer device instead of console

The character set console sometimes shows problems with the word-wrap and/or shows characters on cleared lines/screen. To avoid this behavior the easiest way is to use the frame buffer device instead of the console.

4.1.1. Frame buffer device already implemented in the kernel

If you are using SuSE, newer then Version 6.4, or Redhat, newer then Version 6.1 there is no need to change the kernel because this function is already integrated.
The only upgrade needed is to change the /etc/lilo.conf and rerun – lilo.

In the following example there are 3 labels:

- fblinux: uses the frame buffer device with 1024x768 with 16 bit
- ask: asks which mode the frame buffer device should use (see table 1)
- linux: use the standard text console

```
# LILO configuration file
# Start LILO global Section
boot=/dev/hda
vga = normal
read-only
prompt
timeout=100
# End LILO global Section
image = /boot/vmlinuz
  root = /dev/hda3
  label = fblinux
  VGA = 791
image = /boot/vmlinuz
  root = /dev/hda3
  label = ask
  VGA = ASK
image = /boot/vmlinuz
  root = /dev/hda3
  label = linux
```

Colors	640x480	800x600	1024x768	12280x1024
265 (8bit)	769	771	773	775
32000 (15bit)	784	787	790	793
65000 (16bit)	785	788	791	794
16.7 Mill. (24bit)	786	789	792	795

Table 1

If you change your /etc/lilo.conf, you have to rerun lilo:

lilo

Then you can reboot and during the display of "lilo" you can key the TAB key until a list of the valid labels in etc/lilo.conf appears. Pick the one you want to select.

4.1.2. Kernel without frame buffer device

If you have an older distribution package of SuSE or Redhat, or a distribution package with a Linux kernel without frame buffer support, you have to compile a new kernel.
The complete “HOW TO” can be found here:
http://www.linuxdoc.org/HOWTO/Framebuffer-HOWTO-10.html

Here are the steps in short form:

1. Make sure you have Kernel sources later than 2.2.0.

2. Log in as root and change to the directory of the kernel sources (usually /usr/src/linux)

3. Configure the kernel

   make menuconfig

   Select "Code maturity level options" and set "Prompt for development and/or incomplete code/drivers".
   Select "Console drivers" set "Support for frame buffer devices" to built-in (even if it says EXPERIMENTAL). Then configure the driver. Use "VESA VGA graphics console.
   Finally enable "Advanced low level driver options" and make sure that 16 and 32 bpp packed pixel support are enabled.

4. Compile the kernel
   First do:
   make dep
   Then:
   make clean
   And:
   make bzImage

Copy the Kernel from /(depending on the distribution package)/bzImage to /boot/linux.fb

5. Edit /etc/lilo.conf
   See 3.1.1, but use your created image name, eg.:

```
image = /boot/linux.fb
  root = /dev/hda3
  label = fblinux
  VGA = 791
```

   And then you must rerun-lilo:

   lilo

4.2. Configuration of Xfree86

For SuSE it's quite easy to implement because the tool SAX can be used to configure the X-Server automatically. For Xfree 3.6.6 just use:

Sax –s fbdev

If you have Xfree 4.x.x you should use:

Sax2 –m fbdev

This starts the X-Server configuration and allows you to configure the necessary settings (mouse, keyboard, monitor, resolution, etc.)
It may be necessary to install the packet "xfbdev" from the series "xsrv" before doing the above mentioned.

For RedHat the configuration can't be done with sax. Here it is necessary to make a body of the /etc/XF86Config with the command

Xconfigurator

Don't worry about the video card selection when you do this and try to avoid letting Xconfigurator probe or test your board. All you want is a skeleton file to edit and all that is of interest is the "Device" and "Screen" sections. Then change your "Device" and "Screen" sections to something like in the following example based on the configuration of Xfree86 3.6.6 for usage with the MOPSlcdGX1.

```
# SaX autogenerated XF86Config file
# This file was generated from the SaX
# Version: 2.8 - sax@suse.de
# Date: Mon Oct  1 04:08:20 CEST 2001
# Xserver:FBDEV
# MouseVendor:MICROSOFT
# MouseName:STANDARD MOUSE
# RamDac:60
# Dac8:60
# Dac16:00
# Dac24:
# Dac32:

Section "Files"
  RgbPath  "/usr/X11R6/lib/X11/rgb"
  FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
  FontPath "/usr/X11R6/lib/X11/fonts/local"
  FontPath "/usr/X11R6/lib/X11/fonts/misc:unscaled"
  FontPath "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
  FontPath "/usr/X11R6/lib/X11/fonts/Type1"
  FontPath "/usr/X11R6/lib/X11/fonts/misc"
  FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
  FontPath "/usr/X11R6/lib/X11/fonts/100dpi"
EndSection

Section "ServerFlags"
  AllowMouseOpenFail
EndSection

Section "Module"
EndSection

Section "Keyboard"
  Protocol "Standard"
  XkbRules "xfree86"
  XkbModel "pc104"
  XkbLayout     "de"
  XkbVariant    "nodeadkeys"
```

```
EndSection

Section "Pointer"
  Protocol      "Microsoft"
  Device        "/dev/ttyS1"
  SampleRate         60
  BaudRate      1200
EndSection

Section "Monitor"
  Identifier    "Primary-Monitor"
  VendorName    "!!! VESA !!!"
  ModelName     "1024X768@75HZ"
  HorizSync     31-69
  VertRefresh   50-85
  Modeline "1400x1050" 59.93 1400 1416 1704 1816 1050 1050 1055 1097
  Modeline "1280x960" 59.90 1280 1296 1552 1664 960 960 965 1003
  Modeline "1600x1000" 59.91 1600 1616 1968 2080 1000 1000 1004 1044
  Modeline "1024x864" 59.89 1024 1040 1216 1328 864 864 870 902
  Modeline "800x600" 55.33 800 816 928 1040 600 600 608 626
  Modeline "1152x864" 59.99 1152 1168 1384 1496 864 864 870 902
  Modeline "1280x1024" 59.90 1280 1296 1552 1664 1024 1024 1029 1070
  Modeline "640x480" 35.36 640 656 720 832 480 480 486 501
  Modeline "1024x768" 59.89 1024 1040 1216 1328 768 768 774 802
  Modeline "1600x1200" 59.91 1600 1616 1968 2080 1200 1200 1204 1253
EndSection

Section "Device"
  Identifier    "Primary-Card"
  VendorName    "!!! GENERIC SERVER SELECTION !!!"
  BoardName          "FRAMEBUFFER"
  Chipset       "mediagx"
EndSection

Section "Screen"
    Driver "FBDev"
    Device "Primary-Card"
    Monitor     "Primary-Monitor"
    DefaultColorDepth     16
     Subsection "Display"
        Depth   32
      Modes      "default"
     EndSubsection
     Subsection "Display"
        Depth   24
      Modes      "default"
     EndSubsection
     Subsection "Display"
        Depth   16
      Modes      "default"
     EndSubsection
Subsection "Display"
        Depth   8
      Modes      "default"
     EndSubsection
EndSection
```

If you are using a distribution package of RedHat it might also be necessary to link the frame buffer X server to be your default X server:

cd /etc/X11
rm –f X
ln –s ../../usr./X11R6/bin/XF86_FBDev X

When all prerequisites are fulfilled (configuration of window manager, etc.) you can start the X Server with the command

startx