## NAME

sane−umax_pp − SANE backend for Umax Astra parallel port flatbed scanners

## DESCRIPTION

The **sane−umax_pp** library implements a SANE (Scanner Access Now Easy) backend that provides access to Umax parallel port flatbed scanners.  The following scanners work with this backend:

```
Model:
−−−−−−−−−−−−−−−−−−−−
Astra 610P
Astra 1220P
HP3200C
Astra 1600P
Astra 2000P
Genius ColorPage-Life Pro
```

This backend handles 75x75, 150x150, 300x300, 600x600 and 600x1200 for 1220P/1600P/2000P dpi scan resolutions, and 75x75, 150x150, 300x300 and 300x600 for 610P. In color and gray levels, there is a software lineart mode.

The new generation models share a newer version of the 610P ASIC embedded in an EPAT chip.  Only parts such as CCD and ADC change from one to another. They even all reports being UMAX Astra 1220P via IEEE1284.  There isn't software way to recognize them properly. Under windows, model is set by the driver installed, regardless of the hardware.

### EPP/ECP MODES ONLY

The current version of the backend uses only EPP or ECP mode to communicate with the scanner. PS/2 mode isn't implemented. The 610P only use SPP. It is recommended that you set your parallel port to EPP in BIOS with the current version of this backend. You can leave it to ECP or ECP+EPP, but in this case you may not use ppdev but only direct hardware access if you have to use ECP.  ECPEPP will only work if you use a 2.4 or 2.6 kernel with ppdev character device support.

This backend does support parport sharing only *if you have a kernel with ppdev support.*

Note that if you don't use the ppdev character device, the backend needs to run as root. To allow user access to the scanner run the backend through the network interface (See saned(8) and sane−net(5)).  A more relaxed solution (security wise) is to add suid bit to the frontend (See chmod(1)).  The backend drop root privileges as soon as it can, right after gaining direct access to IO ports, which lessen risks when being root.

## DEVICE NAMES

This backend expects device names of the form:

*port value*

Where **value** is :

auto       autodetect all parallel ports and probe them for scanner

safe−auto
           autodetect all parallel ports and probe them for scanner, but does not try direct hardware access

/dev/ppi0
           uses *BSD ppi device, depending on the number of available parallel port, you have to use /dev/ppi1, /dev/ppi2, ...

/dev/parport0
           uses linux ppdev device, depending on the number of available parallel port, you have to use /dev/parport1, /dev/parport2, ...

0x378    does direct hardware access on the given address. Usual values are 0x378, 0x278, 0x3BC
In this case, you have to run the scanner as root (*BSD and linux), or with 'IOPL=yes' on
OS/2

**NOTE:** in all cases, you must have sufficient privileges to get access to the chosen device or address. Depending on the security settings, devices may not be available for all users.  You have to change permissions on the dev/ppi* or /dev/parport* devices.

You can rename any device using the

> *name devname*
> *model model*
> *vendor vendor*

options. These options apply to the last port option.

## CONFIGURATION

Please make sure to edit umax_pp.conf **before** you use the backend.

The contents of the **umax_pp.conf** file is a list of options and device names that correspond to Umax scanners.  Empty lines and lines starting with a hash mark (#) are ignored.

The eight options supported are **red−gain**, **green−gain**, **blue−gain**, **red−offset**, **green−offset**, **blue−offset**, **astra**, and **buffer**.

Options **red−gain** , **green−gain** and **blue−gain** allow you to adjust the sensitivity of your scanner for the given color. Values range from 0 (lowest gain) to 15 (highest). If the advanced option "Gain" isn't checked in the frontend, the backend does automatic gain calibration, and do not use user provided values.

Options **red−offset** , **green−offset** and **blue−offset** allow you to adjust the offset of your scanner for the given color. Values range from 0 (lowest offset) to 15 (highest).

Option **astra** allows you to change the model of your scanner. Current auto detection is based on side effects on scanning when using 1220P command set on other models, so it may fail on unknown hardware combination. Valid values are 610, 1220, 1600 and 2000. It is useful only when autodetection fails to detect properly your scanner model. If your scanner work properly but is reported wrongly, let it be that way.  The only valid case to change the model is when your scanner produces "black" or "inverted" scans. In this case you can put the model. Be aware that it will prevent scanner model autodetection.

Option **buffer** allows you to change the size of the scan buffer. The size must be specified in bytes. The default value is 2 megabytes. Decreasing this value will improve the smoothness of progress bar in the frontend, but will stall the scan more often.

## FILES

*@CONFIGDIR@/umax_pp.conf*
> The backend configuration file (see also description of **SANE_CONFIG_DIR** below).

*@LIBDIR@/libsane−umax_pp.a*
> The static library implementing this backend.

*@LIBDIR@/libsane−umax_pp.so*
> The shared library implementing this backend (present on systems that support dynamic loading).

## ENVIRONMENT
### SANE_CONFIG_DIR
> This environment variable specifies the list of directories that may contain the configuration file.
> Under UNIX, the directories are separated by a colon (':'), under OS/2, they are separated by a
> semi-colon (';').  If this variable is not set, the configuration file is searched in two default directories: first, the current working directory (".") and then in @CONFIGDIR@.  If the value of the

environment variable ends with the directory separator character, then the default directories are searched after the explicitly specified directories. For example, setting **SANE_CONFIG_DIR** to "/tmp/config:" would result in directories "tmp/config", ".", and "@CONFIGDIR@" being searched (in this order).

**SANE_DEBUG_UMAX_PP**

If the library was compiled with debug support enabled, this environment variable controls the debug level for this backend. E.g., a value of 128 requests all debug output to be printed. Smaller levels reduce verbosity.

```
level   debug output
------- -----------------------------
 0       nothing
 1       errors
 2       warnings & minor errors
 3       additional information
 4       debug information
 5       code flow (not supported yet)
 6       special debug information
```

**SANE_DEBUG_UMAX_PP_LOW**

This variable sets the debug level for the SANE interface for the Umax ASIC. Note that enabling this will spam your terminal with some million lines of debug output.

```
level   debug output
------- -----------------------------
 0       nothing
 1       errors
 8       command blocks
 16      detailed code flow
 32      dump datafiles
 255     everything
```

## SEE ALSO

sane(7), sane−net(5), saned(8)

For latest bug fixes and information see
*http://umax1220p.sourceforge.net/*

## AUTHOR

Stéphane Voltz <stef.dev@free.fr>

## CREDITS

Support for the 610P has been made possible thank to an hardware donation by William Stuart.

## BUG REPORTS

If something doesn't work, please contact me. But I need some information about your scanner to be able to help you...

*SANE version*

run "scanimage −V" to determine this

*the backend version and your scanner hardware*
>     run "SANE_DEBUG_UMAX_PP=255 scanimage −L 2>log" as root. If you don't get any output from the umax_pp backend, make sure a line "umax_pp" is included into your @CONFIGDIR@/dll.conf.  If your scanner isn't detected, make sure you've defined the right port address, or the correct device in your umax_pp.conf.

*the name of your scanner/vendor*
>     also a worthy information. Please also include the optical resolution and lamp type of your scanner, both can be found in the manual of your scanner.

*any further comments*
>     if you have comments about the documentation (what could be done better), or you think I should know something, please include it.