NAME

sane-hp5590 - SANE backend for Hewlett-Packard 4500C/4570C/5500C/5550C/5590/7650 Workgroup/Document scanners

DESCRIPTION

The **sane-hp5590** library implements a SANE (Scanner Access Now Easy) backend that provides access to the following Hewlett-Packard Workgroup/Document scanners:

- ScanJet 4500C

- ScanJet 4570C

- ScanJet 5500C

- ScanJet 5550C

- ScanJet 5590

- ScanJet 7650

If you own a scanner other than the ones listed above that works with this backend, please let us know this by sending the scanner's exact model name and the USB vendor and device ids (e.g. from */sys/bus/usb/devices*, *sane-find-scanner* or syslog) to us. Even if the scanner's name is only slightly different from the models mentioned above, please let us know.

OPTIONS

The options the backend supports can either be selected through command line options to programs like scanimage or through GUI elements in *xscanimage* or *xsane*. Valid command line options and their syntax can be listed by using:

```
scanimage --help -d hp5590:interface:device
```

where *interface* and *device* specify the device in question, as in the configuration file. Add *--all-options* to also list the hardware read-out options. The -d parameter and its argument can be omitted to obtain information on the first scanner identified.

Use the command:

```
scanimage -L
```

to list all devices recognized by your SANE installation.

DEVICE SPECIFIC OPTIONS

-l *n*   Top-left X position of scan area in **mm**. Allowed range: 0 .. 215.889.

-t *n*   Top-left Y position of scan area in **mm**. Allowed range: 0 .. 297.699.

-x *n*   X width of scan-area in **mm**. Allowed range: 0 .. 215.889.

-y *n*   Y height of scan-area in **mm**. Allowed range: 0 .. 297.699.

By default, the maximum size will be scanned.

--mode *mode*

Select color mode. *mode* must be one of: "Color", "Color (48 bits)", "Gray", "Lineart".

- "Color" - Scanning is done with 3 * 8 bit RGB color values per pixel.

- "Color (48 bits)" - Scanning is done with 3 * 16 bit RGB color values per pixel.

- "Gray" - Scanning is done with 1 * 8 bit gray value per pixel.

- "Lineart" - Scanning is done with 1 bit black and white value per pixel.

--source *source*

Select the source for scanning. *source* must be one of: "Flatbed", "ADF", "ADF Duplex", "TMA Slides", "TMA Negatives".

- "Flatbed" - Scan document on the flat document glass.

- "ADF" - Scan frontsides of documents with automatic document feeder.

- "ADF Duplex" - Scan front- and backsides of documents with automatic document feeder. Note, the backside images must be rotated in a separate post process step.

- "TMA Slides" - Slide scanning with transparent media adapter. (Not fully supported by hp5590 backend).

- "TMA Negatives" - Negative film scanning with transparent media adapter. (Not fully supported by hp5590 backend).

--resolution *res*

Set the resolution of the scanned image in **dpi**. *res* must be one of: 100, 200, 300, 600, 1200, 2400.

Default settings: Lineart, Flatbed, 100dpi.

--extend-lamp-timeout[=yes|no]

Extend lamp timeout period. no = 15 minutes, yes = 1 hour. (Default: no)

--wait-for-button[=yes|no]

Wait for button press before scanning starts. (Default: no)

--preview[=yes|no]

Request a preview-quality scan. (Default: no)

--hide-eop-pixel[=yes|no]

Hide end-of-page indicator pixels and overwrite with color of next neighbor pixels. (Default: yes)
The scanner uses the last pixel in every scan line for storing the end-of-page status. This is needed to detect the end of the document sheet when the automatic document feeder (ADF) is used. Unfortunately the end-of-page pixels are also generated in flatbed scans. It is recommended to hide these pixels.

--trailing-lines-mode *mode*

Filling mode of trailing lines after end of page when automatic document feeder (ADF) is used. *mode* must be one of: "last", "raw", "raster", "white", "black", "color". (Default: "last")

- "last" = repeat the last scan line (recommended),

- "raw" = read raw scan data (not recommended),

- "raster" = generate black and white pixel pattern,

- "white" = white pixels,

- "black" = black pixels,

- "color" = RGB or gray colored pixels (see next option).

--trailing-lines-color *n*

Set color value for filling trailing scan lines in trailing lines mode "color" (see previous option). (Default color: violet)
The RGB color value must be specified and calculated as 65536 * r + 256 * g + b, with r, g, b being values in the range of 0 .. 255.

READ OUT OPTIONS

The following options allow reading out the button state, counter value, color setting, and the state of document in ADF. This can be used to programmatically control corresponding scanner options like switching between *flatbed* and *ADF* mode, or triggering prost processing tasks after scanning.

--button-pressed

Get the id of the last button pressed. Id is one of "none", "power", "scan", "collect", "file", "email", "copy", "up", "down", "mode", "cancel".
The scanner stores the id of the last button pressed until it is read. After read out, the state is reset and subsequent readings will return "none".

--color-led
> Get the state of the color LED indicators. The state is either "color" or "black_white".

--counter-value
> Get the counter value as shown on LCD. The value is in the range of 1 .. 99.

--doc-in-adf
> Get the state of the document-available indicator of the automatic document feeder (ADF). The state is either "yes" or "no".

HINTS FOR USERS OF SCANBD

Scanbd is a scanner button daemon, which can read scanner buttons and trigger scan actions.

Do not use the old *scanbuttond* interface with hp5590. It is outdated and shall not be used any more. Scanbd's regular interface is fully supported by the current version of the *hp5590* backend.

This example shows a minimum configuration file and the corresponding script file for scanbd to be included in *scanbd.conf*.

- **hp5590.conf**

```
device hp5590 {
    # Device matching
    filter = "^hp5590.*"
    desc = "HP5590 Scanner Family"

    # Read out counter value and store in environment variable.
    function function_lcd_counter {
        filter = "^counter-value.*"
        desc  = "hp5590: LCD counter"
        env   = "SCANBD_FUNCTION_LCD_COUNTER"
    }

    # Run scan script when button is pressed.
    action do-scan {
        filter = "^button-pressed.*"
        desc  = "hp5590: Scan button pressed"
        script = "scan_action.script"
        string-trigger {
            from-value  = "none"
            to-value    = "scan"
        }
    }
}
```

- **scan_action.script**

```
#!/bin/bash
echo device = $SCANBD_DEVICE
echo action = $SCANBD_ACTION
echo counter = $SCANBD_FUNCTION_LCD_COUNTER
scanfile="$HOME/tmp/scans/scan-$(date +%s).pnm"
case $SCANBD_ACTION in
do-scan)
    scanimage -d "$SCANBD_DEVICE" > "$scanfile"
    ;;
*)
    echo Warning: Unknown scanbd action: "$SCANBD_ACTION"
    ;;
esac
```

FILES

@LIBDIR@/libsane-hp5590.a
>The static library implementing this backend.

@LIBDIR@/libsane-hp5590.so
>The shared library implementing this backend (present on systems that support dynamic loading).

ENVIRONMENT

>If the library was compiled with debug support enabled, this environment variable controls the debug level
>for this backend.

SANE_DEBUG_HP5590

>Higher debug levels increase the verbosity of the output:

```
10 - generic processing
20 - verbose backend messages
40 - HP5590 high-level commands
50 - HP5590 low-level (USB-in-USB) commands
```

Example:
>export SANE_DEBUG_HP5590=50

SEE ALSO

>**sane**(7), **sane-usb**(5)

AUTHORS

>Ilia Sotnikov <hostcc@gmail.com>.