

Table of Contents

Scansystem with Canon CanoScan LiDE 110	3
Software	3
Installation	3
Action Script	4
Network Scanner	6
Links	6

Scansystem with Canon CanoScan LiDE 110

Unfortunately, there are still letters, some of them are useful for the tax return or just useful to have in digital form because of searching is much easier. So I want an system which is always ready to use. I bought a scanner and connected it to my Server. But how to use the buttoms on the scanner? So hear the setup for an [Canon CanoScan LiDE 110](#) scanner. Most configuration work out of the box. Important is *don't use scanbuttond*, it's no longer really maintained and complicated when you want to use it with scanners that are not supported by default additionally it seems to have some usb issues on Arch Linux because theoretical, it would detect my scanner after patching sources.

Software

The software chosen for this is [scanbd](#) which work with [sane](#). scanbd and the included socket will made the scanner accessible and read the buttoms pressed. Sane itself is important for its config for all the scanner and for local scanning.

I've made these with:

OS	Arch Linux x64
sane Version	1.0.24-2
scanbd Version	1.3-2

Installation

The documentation in the Arch Wiki for this is very good so I will limit myself to the most important. Install the software form [AUR](#). I used aurget but you also can do it manually or with other tools.

```
pacman -S sane
aurget -S scanbd
```

copy the sane configs for using with scanbd

```
cp /etc/sane.d/* /etc/scanbd/sane.d/
```

tell the local sane only to use network scanner so it won't block the scanner. We will make the scanner accessible via a socket for the local sane.

[/etc/sane.d/dll.conf](#)

```
net
```

[/etc/sane.d/net.conf](#)

```
connect_timeout = 3
```

```
localhost # scanbm is listening on localhost
```

configure scanbd's sane part to **not** use network

[/etc/scanbd/sane.d/dll.conf](#)

```
#net
genesys
#... whatever other scanner backend needed ...
```

When I installed scanbd inside the default config was a full debugging loglevel enabled. Change this before you start scanbd to save disk space on your system.

[/etc/scanbd/scanbd.conf](#)

```
...
    debug-level = 2
...
```

Make adjustments to use the "PDF" button on the [LIDE 110](#) and insert this after the copy action.

[/etc/scanbd/scanbd.conf](#)

```
    action pdf {
        filter = "^file$"
        numerical-trigger {
            from-value = 1
            to-value   = 0
        }
        desc    = "Scan to PDF"
        script = "test.script"
    }
```

enable and start services and the socket

```
systemctl enable scanbd.service
systemctl start scanbd.service
systemctl start scanbm.socket
```

The default test script will only work with logger which is shipped with syslog-ng!

Now you should see an action to button pressed in your log.

Action Script

But you need an useful action script to do something when one button is pressed.

Here is my:

[/etc/scanbd/action.script](#)

```
#!/bin/bash

SCAN_DIR=/data/user/Scanner
DATE_T=`date +%d-%m-%Y_%H%M%S`
SEND_TO=user@example.org
PDF_DIR=$SCAN_DIR/PDF
MAIL_DIR=$SCAN_DIR/Mail
SCAN_IN_DIR=$SCAN_DIR/scan
PRINT_DIR=$SCAN_DIR/printed
TMP_DIR=$SCAN_DIR/tmp
PRINTER_DEF="SamsungML-1660"

case $SCANBD_ACTION in
pdf)
logger -t "scanbd_action: $0" "$SCANBD_ACTION - scan and convert to pdf"
scanimage -d $SCANBD_DEVICE --mode Gray --resolution 300 --format=tiff > $TMP_DIR/tempscan_${DATE_T}.tiff
logger -t "scanbd: $0" "$SCANBD_ACTION - convert tiff to PDF"
convert -quality 5 -compress jpeg $TMP_DIR/tempscan_${DATE_T}.tiff $PDF_DIR/scan_${DATE_T}.pdf
rm $TMP_DIR/tempscan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - finished"
;;
scan)
logger -t "scanbd_action: $0" "$SCANBD_ACTION - scan as Tiff"
scanimage -d $SCANBD_DEVICE --resolution 300 --mode Color --format=tiff > $SCAN_IN_DIR/scan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - finished"
;;
copy)
logger -t "scanbd_action: $0" "$SCANBD_ACTION - create an copy"
scanimage -d $SCANBD_DEVICE --resolution 300 --format=tiff --mode=Linear > $PRINT_DIR/scan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - print on $PRINTER_DEF"
lpr -P $PRINTER_DEF $PRINT_DIR/scan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - finished"
;;
email)
logger -t "scanbd_action: $0" "$SCANBD_ACTION - scan as PDF and send via mail"
scanimage -d $SCANBD_DEVICE --mode Gray --resolution 300 --format=tiff > $TMP_DIR/tempscan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - convert tiff to PDF"
```

```
convert -quality 5 -compress jpeg $TMP_DIR/tempscan_${DATE_T}.tiff
$MAIL_DIR/scan_${DATE_T}.pdf
rm $TMP_DIR/tempscan_${DATE_T}.tiff
logger -t "scanbd_action: $0" "$SCANBD_ACTION - send PDF as mail"
SIZE=$(ls -lah $MAIL_DIR/scan_${DATE_T}.pdf | awk '{print $5}')
logger -t "scanbd_action: $0" "Größe $SIZE"
echo -ne "Document scanned on Canon LiDE 110\n\nFile:
scan_${DATE_T}.pdf" | mailx -s "scanned document" -a
$MAIL_DIR/scan_${DATE_T}.pdf $SEND_TO
logger -t "scanbd_action: $0" "$SCANBD_ACTION - finished"
;;
*)
logger -t "scanbd: $0" "unknown Action"
;;
esac

exit 0
```

Network Scanner

With sane you can use your scanner via network and XSane or scanimage on an desktop installation.

1. edit /etc/scanbd/sane.d/saned.conf and allow your ip range, no service restart is needed because saned is loaded when anyone connect to the socket
2. edit /etc/sane.d/net.conf on your client and set the right server
3. search for scanners on the client scanimage -L

Links

https://wiki.archlinux.org/index.php/Scanner_Button_Daemon

http://en.wikipedia.org/wiki/Scanner_Access_Now_Easy

From:

<https://www.eanderalx.org/> - **EanderAlx.org**

Permanent link:

https://www.eanderalx.org/linux/scanner_lide110

Last update: **28.10.2013 16:58**

