

Software:

1. Setup camera

a. `sudo raspi-config`

i. Enable I2C

b. `sudo vim /boot/firmware/config.txt`

```
# Automatically load overlays for detected cameras
camera_auto_detect=0
```

i.

```
[all]
enable_uart=1
dtoverlay=imx708
```

ii.

iii. Reboot for changes to take effect

c. `rpicam-still --list-cameras`

```
~ $ rpicam-still --list-cameras
```

```
Available cameras
```

```
-----
0 : imx708 [4608x2592 10-bit RGGB] (/base/soc/i2c0mux/i2c@1/imx708@1a)
    Modes: 'SRGGB10_CSI2P' : 1536x864 [120.13 fps - (768, 432)/3072x1728 crop]
    2304x1296 [56.03 fps - (0, 0)/4608x2592 crop]
    4608x2592 [14.35 fps - (0, 0)/4608x2592 crop]
```

d. `mkdir Photos` for script later

```
~ $ mkdir Photos
```

```
~ $ ls
```

```
Desktop  Documents  Downloads  Music  Photos  Pictures  Public  Templates  Videos
```

2. Set up printer

a. `scp printer-driver-sn_3.13.16_deb.deb`

```
((base) ~ $ scp printer-driver-sn_3.13.16_deb.deb ~/Downloads
```

```
printer-driver-sn_3.13.16_deb 100% 966KB 6.7MB/s 00:00
```

b. `sudo dpkg -i printer-driver-sn_3.13.16_deb.deb`

```
~/Downloads $ sudo dpkg -i printer-driver-sn_3.13.16_deb.deb
```

```
Selecting previously unselected package printer-driver-sn.
(Reading database ... 144018 files and directories currently installed.)
Preparing to unpack printer-driver-sn_3.13.16_deb.deb ...
Unpacking printer-driver-sn (3.13.16) ...
Setting up printer-driver-sn (3.13.16) ...
CPU_BITS: 64
SYS_NAME: Linux
CPU_ARCH: aarch64
Configure USB device 0x2D37 ...
Configure USB device 0x2D84 ...
/var/log/prnlog is created.
no crontab for root
*/1 * * * * /opt/sn/printer-driver-sn/bin/mvimg.sh
Driver success installed!
Please navigate to http://localhost:631/ for manage your printers.
```

c. `sudo apt install cups enscript -y`

- d. lsusb (verify printer is seen)
- e. lpstat -p

```

~ $ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 003: ID 2d37:847c Zhuhai Poskey Technology Co.,Ltd DB402
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
~ $ lpstat -p
printer DB402 is idle.  enabled since Wed 11 Mar 2026 04:40:33 PDT
~ $

```

3. If you only want to set up a basic script with no Pi-day edits that just takes normal photos and prints them out, follow the steps here and skip #4. Otherwise, jump to #4.

- a. scp ledcam.py
- b. Run ledcam.py with python3.

4. If you would like to set up the advanced script with the Pi-day edits, follow these next steps. This script is more complicated since the Pi-mode uses the rembg library to remove the background from the photos we take and then performs image masking to create the person-in-a-pie effect we're going for. We're going to create a virtual environment to make this happen.

- a. lpstat -p #get name of printer
- b. lpadmin -d DB402 #set the printer as default
- c. mkdir pi_booth && cd pi_booth
- d. sudo apt update && sudo apt install libcap-dev -y
- e. python3 -m venv --system-site-packages ~/pi_booth/env
- f. source ~/pi_booth/env/bin/activate
- g. pip install rembg opencv-python pillow gpiozero
- h. pip install picamera2

```

~/pi_booth $ python3 -m venv --system-site-packages ~/pi_booth/env
~/pi_booth $ source ~/pi_booth/env/bin/activate

```

- i. pip install "rembg[cpu]"
- j. Whenever the pi_booth environment is activated, you will be able to run the script.

Try getting a basic version of the booth running before moving on to the subsequent steps. It's going to be much harder to debug if your electronics are in the box!