Use ALL the devices!

Converting Android tablets to

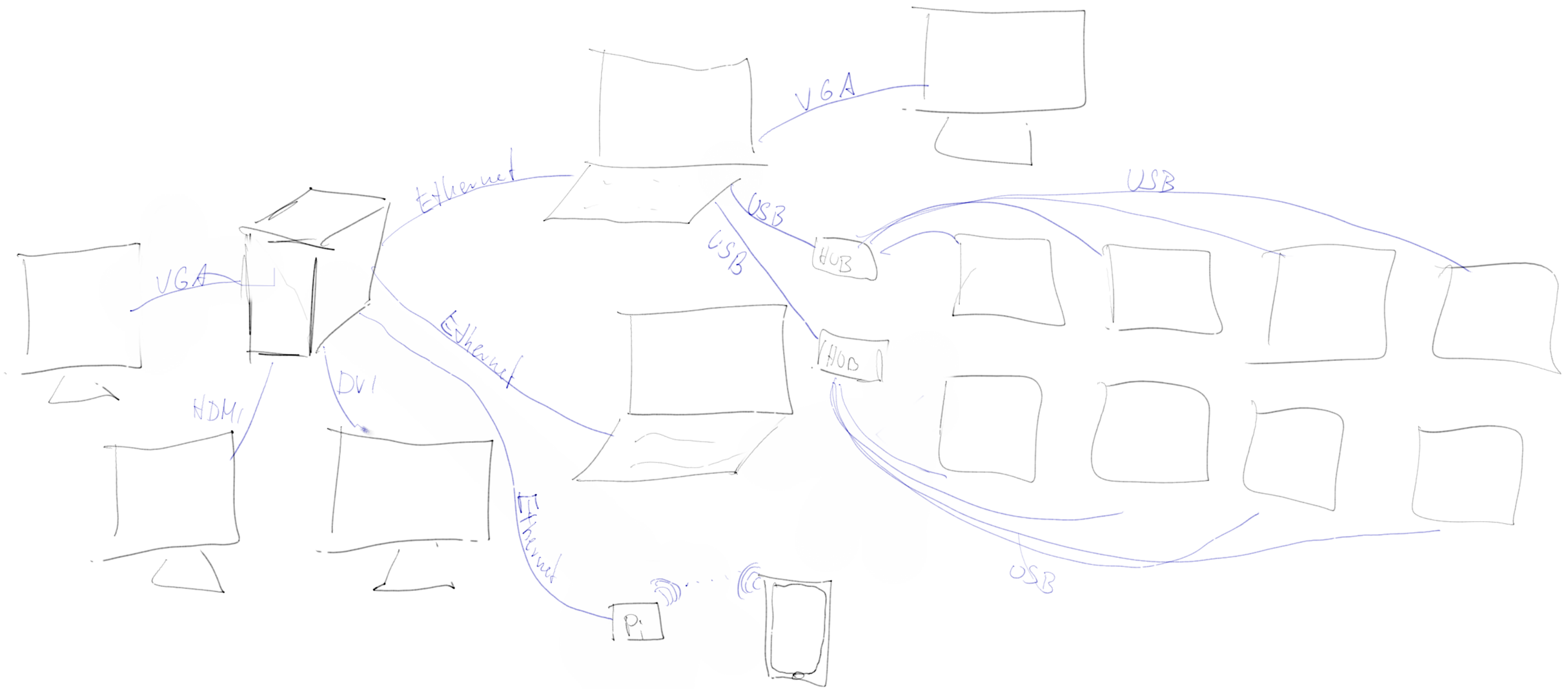
- → move screens
- → move speakers
- → move drawing tablets

<Drahflow>

2021-05-14

Stratum 0

Move Screens



```
#!/bin/zsh

set -eu

action="${1:-on}"
subcmd="${2:-}"
pw="$(cat $HOME/.displays.pw)"

conf() {
  cat <<CONF
Lifetab4:1280x800:COL0:none:pw
Nova:800x600:COL0:none:pw
Laptop2:1366x768:COL1:none:none
Laptop1:1366x768:COL1:none:none
SamsungWhite:1024x600:COL1:none:pw
SamsungBlack2:1024x600:COL1:none:pw
SamsungBlack1:1024x600:COL1:none:pw
ViewSonic:1280x1024:COL2:none:none
Belinea:1280x1024:COL2:DVI-D-0:pw
Lifetab2:1280x800:COL2:none:pw
Write:2560x1600:COL2:none:pw:multicol
Samsung:1360x768:COL3:HDMI-0:pw
Lenovo:1280x800:COL3:none:pw
Medion:1280x1024:COL4:VGA-0:pw
Lifetab3:1280x800:COL4:none:pw
Phone:800x480:COL4:none:pw
*:COL0:shift:2036:pw
*:COL1:shift:700:pw
*:COL2:shift:0:pw
*:COL3:shift:700:pw
*:COL4:shift:900:pw
CONF
}

parseName() {
  cut -d: -f1
}

parseWidth() {
  cut -d: -f2 | cut -dx -f1
"~/bin/displays" 407L, 10121C                                 34,0-1        Top
```

this one is a bit big

y-Axis corrections

```
  sum=0
  conf | grep -B999 "^*:${1}" | grep "^*" | grep -v "$1" | cut -d: -f2 | while read col; do
    sum="$(($sum + $(columnWidth $col)))"
  done

  echo "$sum"
}

framebufferWidth() {
  sum=0
  conf | grep "^*:COL" | cut -d: -f2 | while read col; do
    sum="$(($sum + $(columnWidth $col)))"
  done

  echo "$sum"
}

framebufferHeight() {
  max=0
  conf | grep "^*:COL" | cut -d: -f2 | while read col; do
    if [ "$max" -lt "$(columnHeight $col)" ]; then
      max="$(columnHeight $col)"
    fi
  done

  echo "$max"
}

size() {
  if [ "$1" = "framebuffer" ]; then
    echo "$(framebufferWidth)x$(framebufferHeight)"
  else
    conf | grep "^$1:" | parseSize
  fi
}

monsize() {
  size "$1" | sed -e 's/\(.*\)x\(.*\)/\1\/100x\2\/100/'
}

```

numeric expansion

6566 × 4448

123,0-1 25%

```
  ssh "$1" 'DISPLAY=:0 xset dpms force off'
  ssh "$1" 'DISPLAY=:0 screen -dmS i3lock i3lock'
}

setupBelina() { }
setupSamsung() { }
setupMedion() { }

setupLaptop1() {
  uxterm -T displays -e "ssh -R $(port Laptop1):localhost:$(port Laptop1) drahflow@192.168.3.109 'export DISPLAY=:0; ~/i3.install/bin/i3-msg workspace 1; ~/i3.install/bin/i3-msg move workspace to output eDP1; vncviewer FullScreen=0 ViewOnly=1 UseLocalCursor=0 geometry=1366x768+0+0 localhost:$(port Laptop1)'" &

  sleep 2

  ssh drahflow@192.168.3.109 'DISPLAY=:0 ~/i3.install/bin/i3-msg floating enable'
  ssh drahflow@192.168.3.109 'DISPLAY=:0 ~/i3.install/bin/i3-msg resize shrink up 10px'
  ssh drahflow@192.168.3.109 'DISPLAY=:0 ~/i3.install/bin/i3-msg resize set 1368 770'
  ssh drahflow@192.168.3.109 'DISPLAY=:0 xdotool mousemove 0 767'
}

setupViewSonic() {
  sleep 10

  uxterm -T displays -e "ssh -R $(port ViewSonic):localhost:$(port ViewSonic) root@192.168.3.8 'export DISPLAY=:0; i3-msg workspace 2; i3-msg move workspace to output HDMI-1; vncviewer FullScreen=0 ViewOnly=1 UseLocalCursor=0 geometry=1280x1024+0+0 localhost:$(port ViewSonic)'" &

  sleep 2

  ssh root@192.168.3.8 'DISPLAY=:0 i3-msg floating enable'
}

setupLaptop2() {
  unlock root@192.168.3.8 || true

  uxterm -T displays -e "ssh -R $(port Laptop2):localhost:$(port Laptop2) root@192.168.3.8 'export DISPLAY=:0; i3-msg workspace 1; i3-msg move workspace to output eDP1; vncviewer FullScreen=0 ViewOnly=1 UseLocalCursor=0 geometry=1366x768+0+0 localhost:$(port Laptop2)'" &
```

automatic port assignment

each device has some custom setup code

exactly fullscreen

add and map mon

```
28
27        ssh "${acc}" adb -s "${device}" shell su -c ${debian}/start.sh
26        sleep 5
25
24        if [ "$extraEscape" = "escape" ]; then
23          ssh "${acc}" adb -s "${device}" shell su -c ${debian}/enter.sh\\\\ rm\\\\ -f\\\\ /tmp/.X0-lock
22          uxterm -T displays -e "ssh -tt ${acc} adb -s ${device} shell su -c ${debian}/enter.sh\\\\\\\\
/root/X.sh" &
21          sleep 15
20        else
19          ssh "${acc}" adb -s "${device}" shell su -c ${debian}/enter.sh rm -f /tmp/.X0-lock
18          uxterm -T displays -e "ssh -tt ${acc} adb -s ${device} shell su -c ${debian}/enter.sh /root/X.
sh" &
17          sleep 15
16        fi
15      fi
14    fi
13
12    if [ "$action" != "off" -a "$subcmd" != "off" ]; then
11      if [ "$extraEscape" = "escape" ]; then
10        uxterm -T displays -e "ssh -R ${port}:localhost:${port} -tt ${acc} socat TCP-CONNECT:127.0.0.1:$
{port} SYSTEM:\\'adb -s ${device} shell -e none su -c ${debian}/enter.sh\\\\\\\\\\\\\\\\ /socat.sh\\'" &
9         sleep 10
8         uxterm -T displays -e "ssh -tt ${acc} adb -s ${device} shell su -c ${debian}/enter.sh\\\\\\\\ /r
oot/vnc.sh" &
7       else
6         uxterm -T displays -e "ssh -R ${port}:localhost:${port} -tt ${acc} socat TCP-CONNECT:127.0.0.1:$
{port} SYSTEM:\\'adb -s ${device} shell -e none su -c ${debian}/enter.sh /socat.sh\\'" &
5         sleep 10
4         uxterm -T displays -e "ssh -tt ${acc} adb -s ${device} shell su -c ${debian}/enter.sh /root/vnc.
sh" &
3       fi
2     fi
1   }
0
1   setupSamsungWhite() {
2     android "$(port SamsungWhite)" 30045b283451c100 /system/debian no
3   }
4
```

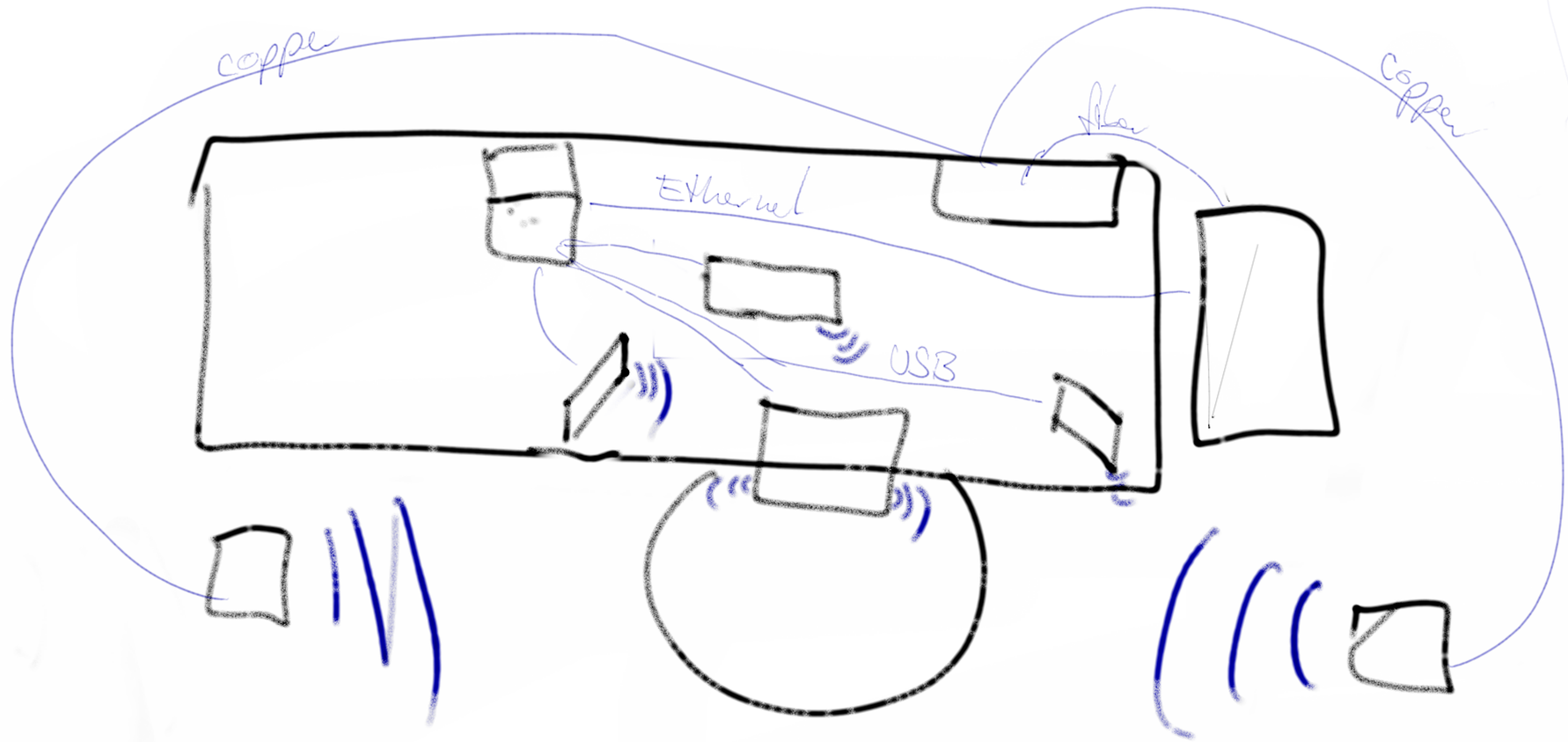
~/bin/tablet-brightness

```
#!/bin/zsh

acc=root@192.168.3.8

for device in 30045b283451c100 3004575aaf74c100 3004924d2e1b7100; do (
  if [ "$1" = "on" ]; then
    echo 'echo '128' > /sys/class/backlight/panel/brightness; exit' | ssh "$acc" adb -s $device shell su
  else
    echo 'echo '0' > /sys/class/backlight/panel/brightness; exit' | ssh "$acc" adb -s $device shell su
  fi
) & done

for device in DEA0017625 DEA0025691 DEA0001695 DEA0009624; do (
  if [ "$1" = "on" ]; then
    echo 'echo '1' > /sys/class/backlight/rk28_bl/brightness; exit' | ssh "$acc" adb -s $device shell su
  else
    echo 'echo '0' > /sys/class/backlight/rk28_bl/brightness; exit' | ssh "$acc" adb -s $device shell su
  fi
) & done

for device in FA49RWJRZLT8FY7H 0123456789ABCDEF; do (
  if [ "$1" = "on" ]; then
    echo 'echo '64' > /sys/class/leds/lcd-backlight/brightness; exit' | ssh "$acc" adb -s $device shell su
  else
    echo 'echo '0' > /sys/class/leds/lcd-backlight/brightness; exit' | ssh "$acc" adb -s $device shell su
  fi
) & done

for device in 2n21i811800000cvj0o0; do (
  if [ "$1" = "on" ]; then
    echo 'echo '96' > /sys/class/backlight/pwm-backlight/brightness; exit' | ssh "$acc" adb -s $device shell su
  else
```

:set wrap 1,1 Top

```
37
36 # lighting controls
35 bindsym Mod4+Delete exec "ssh pi@192.168.3.3 ./off.py"
34 bindsym Mod4+Shift+Delete exec "ssh pi@192.168.3.3 ./color.py"
33 bindsym Mod4+Ctrl+Delete exec "ssh pi@192.168.3.3 ./blueish.py"
32 bindsym Mod4+Ctrl+Shift+Delete exec "ssh pi@192.168.3.3 ./on.py"
31
30 bindsym Mod4+End exec "ssh root@192.168.3.2 'echo off >> /var/run/gpiod/17'"
29 bindsym Mod4+Shift+End exec "ssh root@192.168.3.2 'echo on >> /var/run/gpiod/17'"
28
27 bindsym Mod4+Next exec "tablet-brightness off"
26 bindsym Mod4+Shift+Next exec "tablet-brightness on"
25
24 bindsym Mod4+Prior exec "tablet-paper off"
23 bindsym Mod4+Shift+Prior exec "tablet-paper on"
22
21 # displays setup
20 bindsym Mod4+Shift+Scroll_Lock exec "desktop-setup"
19 assign [title="^displays$"] workspace 1
18 assign [title="^remoteplay$"] workspace 4
17
16 # intercom controls
15 bindsym Mod4+Insert exec "mute-flur"
14 bindsym Mod4+Shift+Insert exec "unmute-flur"
13
12 bindsym Mod4+Home exec "mute-jens unten"
11 bindsym Mod4+Shift+Home exec "unmute-jens unten"
10 bindsym Mod4+Ctrl+Home exec "unmute-jens"
 9
 8 # Redo
 7 bindsym Mod4+Tab exec "redo redo"
 6
 5 # Start i3bar to display a workspace bar (plus the system information i3status
 4 # finds out, if available)
 3 bar {
 2         status_command /home/drahflow/bin/status
 1 }
 0
 1 new_window pixel 1
                                                                  226,0-1       Bot
```

Move Speakers



```
5
4  void streamStateChanged(pa_stream *IGN(stream), void *IGN(userdata)) {
3    pa_stream_state_t state = pa_context_get_state(ctx);
2    fprintf(stderr, "pulseaudio stream state changed: %d\n", state);
1  }
0
1  void dataAvailable(pa_stream *IGN(stream), size_t IGN(bytes), void *IGN(userdata)) {
2    size_t available;
3    const void *data;
4
5    if(pa_stream_peek(stream, &data, &available)) {
6      fprintf(stderr, "Failed to read data from stream: %s\n", pa_strerror(pa_context_errno(ctx)));
7      return;
8    }
9
10   struct timespec t;
11   if(clock_gettime(CLOCK_REALTIME, &t)) {
12     fprintf(stderr, "Failed to get current time: %s\n", strerror(errno));
13     return;
14   }
15
16   dataPacket packet;
17   packet.length = available + 3 * sizeof(uint64_t);
18   packet.position = position;
19   packet.time = (uint64_t)(t.tv_sec) * 1000000000 + t.tv_nsec;
20   memcpy(packet.data, data, available);
21
22   write(1, &packet, sizeof(packet) - sizeof(packet.data) + available);
23
24   position += available;
25   // fprintf(stderr, "Data transmitted. Position now at: %llu\n", (long long unsigned int)position);
26
27   if(pa_stream_drop(stream)) {
28     fprintf(stderr, "Failed to acknowledge stream data: %s\n", pa_strerror(pa_context_errno(ctx)));
29     return;
30   }
31 }
32
33 void contextStateChanged(pa_context *IGN(ctx), void *IGN(userdata)) {
                                                    33,0-1        25%
```

position in recording stream

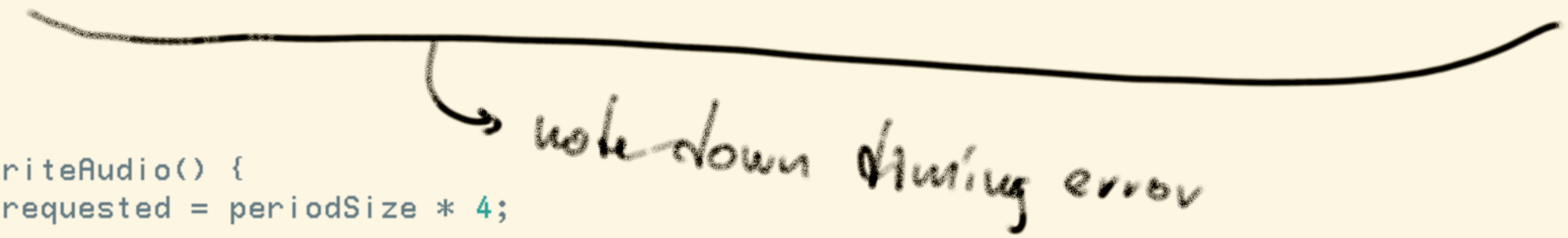
local realtime

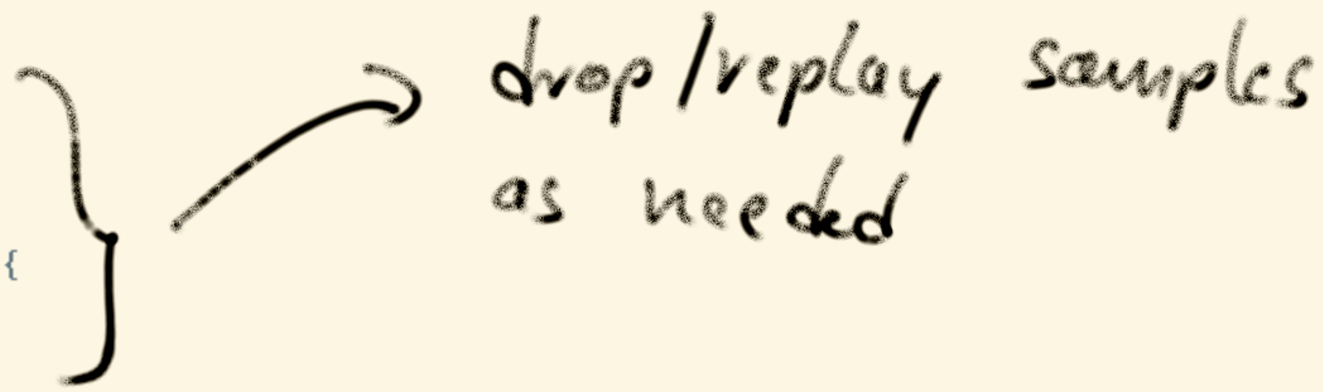
raw waveform

```
    dataPacket *packet = (dataPacket *)receiveBuffer;
    if(receivePos < sizeof(packet->length) || receivePos < packet->length) continue;

    struct timespec t;
    if(clock_gettime(CLOCK_REALTIME, &t)) {
      fprintf(stderr, "Failed to get current time: %s\n", strerror(errno));
    }

    uint64_t now = (uint64_t)(t.tv_sec) * 1000000000 + t.tv_nsec;
    double packetToPlayIn = (packet->time + targetLatency * 1000000000 - now) / 1000000000;

    int dataLen = packet->length - sizeof(*packet) + sizeof(packet->data);
    int64_t localPosition = packet->position - senderOffset;
    int64_t desiredLocalPosition = 4 * sampleRate * targetLatency;

    if(packetToPlayIn < 0) {
      fprintf(stderr, "Packet arrived too late.\n");
    } else if(localPosition < 0) {
      fprintf(stderr, "Playback is too far ahead.\n");

      failureSound(audioBuffer, sizeof(audioBuffer));
      senderOffset = packet->position - frameAlign(desiredLocalPosition);
      localPositionAvg = localPosition = packet->position - senderOffset;
    } else if(localPosition + dataLen > (int)sizeof(audioBuffer)) {
      fprintf(stderr, "Playback is too far behind.\n");

      failureSound(audioBuffer, sizeof(audioBuffer));
      senderOffset = packet->position - frameAlign(desiredLocalPosition);
      localPositionAvg = localPosition = packet->position - senderOffset;
    } else {
      memcpy(audioBuffer + localPosition, packet->data, dataLen);

      localPositionAvg = (1 - localPositionBlend) * localPositionAvg + localPositionBlend * localPosit
ion;
    }

    if(++debugCounter > debugRate) {
      fprintf(stderr, "Packet for: +%lfs, buf pos: %lld, avg %f, delta %d\n", packetToPlayIn, (long lo
ng int)localPosition, localPositionAvg, samplesTooMuch);
                                                                  244,0-1       69%
```

insert into output by byte counter

```
29      }
28
27      uint64_t shift = packet->length;
26      memmove(receiveBuffer, receiveBuffer + shift, receivePos - shift);
25      receivePos -= shift;
24
23      if(localPositionAvg > desiredLocalPosition + maximumDrift) {
22        samplesTooMuch = frameAlign(localPositionAvg - desiredLocalPosition);
21        localPositionAvg = (0.1 * desiredLocalPosition + 0.9 * localPositionAvg);
20      } else if(localPositionAvg < desiredLocalPosition - maximumDrift) {
19        samplesTooMuch = frameAlign(localPositionAvg - desiredLocalPosition);
18        localPositionAvg = (0.1 * desiredLocalPosition + 0.9 * localPositionAvg);
17      }
16    }
15  }
14
13  void writeAudio() {
12    int requested = periodSize * 4;
11
10    int err = snd_pcm_writei(handle, audioBuffer, periodSize);
 9    if(err == -EAGAIN) return;
 8    if(err < 0) {
 7        fprintf(stderr, "Err: %s\n", snd_strerror(err));
 6        if(xrun_recovery(handle, err) < 0) {
 5            printf("Write error: %s\n", snd_strerror(err));
 4            exit(EXIT_FAILURE);
 3        }
 2        return;
 1    }
 0
 1    if(samplesTooMuch > 1) {
 2      requested += samplesTooMuch;
 3      samplesTooMuch = 0;
 4    } else if(samplesTooMuch < -1) {
 5      requested += samplesTooMuch;
 6      samplesTooMuch = 0;
 7    }
 8
 9    if(requested < 0) requested = 0;
```





```
28    fi
27  }
26
25  play() {
24    name="$1"
23    device="$2"
22    latency="$3"
21
20    if [ "$action" != "stop" ]; then
19      uxterm -T remoteplay -e "( echo '/root/remoteplay/play.sh $latency; exit'; sleep 1; ./pulse-sender
        '$name' ) | ssh -e none root@192.168.3.8 adb -s $device shell -e none su -c /data/debian-jessie/enter
        .sh" &
18      sleep 2
17      uxterm -T remoteplay -e "echo 'snice -20 alsa-receiver; exit' | ssh -e none root@192.168.3.8 adb -
        s $device shell -e none su -c /data/debian-jessie/enter.sh" &
16    else
15      uxterm -T remoteplay -e "echo 'killall alsa-receiver; exit' | ssh -e none root@192.168.3.8 adb -s
        $device shell -e none su -c /data/debian-jessie/enter.sh" &
14    fi
13  }
12
11  if [ "$1" = "Write" -o "$1" = "all" ]; then
10    sync Write "2n21i811800000cvj0o0"
9     play Write "2n21i811800000cvj0o0" 0.122
8     # play Write "DEA0017625" 0.150
7   fi
6
5   # if [ "$1" = "Lifetab1" -o "$1" = "all" ]; then
4   #   sync Lifetab1 "DEA0017625"
3   #   play Lifetab1 "DEA0017625" 0.122
2   #   # play Lifetab1 "DEA0017625" 0.150
1   # fi
0
1   if [ "$1" = "Lifetab2" -o "$1" = "all" ]; then
2     sync Lifetab2 "DEA0025691"
3     play Lifetab2 "DEA0025691" 0.120
4   fi
5
6   if [ "$1" = "Lifetab3" -o "$1" = "all" ]; then
                                                    51,0-1          68%
```

start receiver

send data

sound > display

122 ms delay

~ 60cm farther

Playback | **Recording** | Output Devices | Input Devices | Configuration

Remoteplay to Write: forwarding *from* — Monitor of Null Output

Front Left — 141% (8.86 dB)
Front Right — 141% (8.86 dB)
Silence — 100% (0 dB)

Remoteplay to Lifetab2: forwarding *from* — Monitor of Null Output

Front Left — 130% (6.78 dB)
Front Right — 130% (6.78 dB)
Silence — 100% (0 dB)

Remoteplay to Lifetab3: forwarding *from* — Monitor of Null Output

Front Left — 31% (-30.32 dB)
Front Right — 132% (7.22 dB)
Silence — 100% (0 dB)

Remoteplay to Lifetab4: forwarding *from* — Monitor of Null Output

Front Left — 137% (8.20 dB)
Front Right — 30% (-31.25 dB)
Silence — 100% (0 dB)

Remoteplay to Main: forwarding *from* — Monitor of Null Output

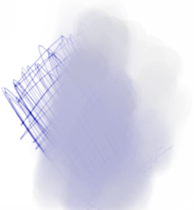
Front Left — 88% (-3.22 dB)
Front Right — 88% (-3.22 dB)
Silence — 100% (0 dB)

Show: Applications

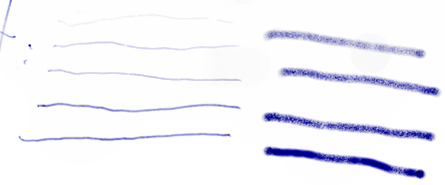
Move Drawing Tablets

On the Toshiba Excite Write:

- /dev/input/event1 → touchscreen
  - multitouch
    - touch area
      - finger — erase
      - generic stylus — blur



- /dev/input/event2 → digitizer
  - 16k² + 1k pressure — line weight:



  - passive pen
  - 1 button → color switch

+ white

    - → marker

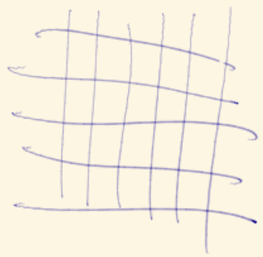
```
8
7       SDL_Rect *upd = updateRects + updates++;
6       upd->x = beginX;
5       upd->y = beginY;
4       upd->w = endX - beginX;
3       upd->h = endY - beginY;
2     }
1   }
0
1   void process(struct input *dev) {
2     static const uint32_t bufferSize = sizeof(struct input_event) * 4096;
3     static struct input_event *buffer = NULL;
4     if(!buffer) buffer = malloc(bufferSize);
5
6     int ret = read(dev->fd, buffer, bufferSize);
7     if(ret == -1) return;
8
9     struct input_event *end = (struct input_event *)((char *)buffer + ret);
10    for(struct input_event *i = buffer; i != end; ++i) {
11      if(dev->buttons & BTN_STYLUS) {
12        ++dev->stylusTime;
13      }
14
15    switch(i->type) {
16      case EV_ABS:
17        switch(i->code) {
18          // Digitizer
19          case ABS_X: dev->x = i->value; break;
20          case ABS_Y: dev->y = i->value; break;
21          case ABS_PRESSURE: dev->pressure = i->value; break;
22          // Touchscreen
23          case ABS_MT_POSITION_X: dev->x = i->value; break;
24          case ABS_MT_POSITION_Y: dev->y = i->value; break;
25          case ABS_MT_TOUCH_MAJOR: dev->width = i->value; break;
26          case ABS_MT_WIDTH_MAJOR: break; // identical to ABS_MT_TOUCH_MAJOR
27          case ABS_MT_TRACKING_ID: break; // TODO
28          default:
29            SDL_Log("Unknown EV_ABS code: %d", i->code);
30        }
31        break;
```

just read(2) from raw dev.

save per data

```
20
19 void background(struct rgba *p, int x, int y) {
18   if(backgroundImage && x > 0 && x < backgroundImage->w && y > 0 && y < backgroundImage->h) {
17     if(backgroundImage->format->BitsPerPixel == 24) {
16       struct rgb *bg = (struct rgb *)((char *)backgroundImage->pixels + x * sizeof(struct rgb) + y * b
15       p->r = bg->r;
14       p->g = bg->g;
13       p->b = bg->b;
12     } else {
11       struct rgba *bg = (struct rgba *)((char *)backgroundImage->pixels + x * sizeof(struct rgba) + y
10       p->r = bg->b;
 9       p->g = bg->g;
 8       p->b = bg->r;
 7     }
 6   } else {
 5     p->r = (y % 50 && x % 50)? 255: 200;
 4     p->g = (y % 50 && x % 50)? 252: 190;
 3     p->b = (y % 50 && x % 50)? 248: 220;
 2   }
 1 }
```

me = very lazy with pixel formats



```
 0
 1 void digitizerCoordinates(int x, int y, float *screenX, float *screenY) {
 2   *screenX = (x - 91.0) * 2560 / (21658 + 91);
 3   *screenY = (y + 10.0) * 1600 / (13536);
 4
 5   // float yFixup = 0;
 6   // if(*screenY > 50 && *screenY < 80) {
 7   //   yFixup = (*screenY - 50) / 30.0;
 8   // } else if(*screenY >= 70 && *screenY < 210) {
 9   //   yFixup = 1;
10   // } else if(*screenY >= 200 && *screenY < 230) {
11   //   yFixup = (230 - *screenY) / 30.0;
12   // }
13   // *screenY = *screenY + 11 * yFixup;
14 }
15
16 void touchscreenCoordinates(int x, int y, float *screenX, float *screenY) {
17   *screenX = x * 2560 / 4096;
18   *screenY = y * 1600 / 4096;
19 }
```

manual calibration

not perfectly linear, but this made it worse

touch screen was calibrated already

```
20
19  touchscreen.fd = open("/dev/input/event1", O_RDONLY | O_NONBLOCK);
18  digitizer.fd = open("/dev/input/event2", O_RDONLY | O_NONBLOCK);
17
16  int updatesSinceSave = 0;
15  int waitForSave = 0;
14  while(1) {
13    updates = 0;
12
11    SDL_LockSurface(surface);
10    process(&touchscreen);
 9    process(&digitizer);
 8    SDL_UnlockSurface(surface);
 7
 6    if(updates) {
 5      SDL_UpdateWindowSurfaceRects(window, updateRects, updates);
 4
 3      updatesSinceSave = 1;
 2      waitForSave = 1500;
 1    }
0
 1    struct timespec s = {
 2      .tv_sec = 0,
 3      .tv_nsec = 1000000,
 4    };
 5    nanosleep(&s, &s);
 6
 7    if(updatesSinceSave && backgroundPath) {
 8      if(waitForSave) {
 9        --waitForSave;
10      } else {
11        savePng(backgroundPath);
12        SDL_Log("Saved.");
13        updatesSinceSave = 0;
14      }
15    }
16  }
17
18  SDL_DestroyWindow(window);
19
```

non-blocking reads in here

me= very lazy with event loops

auto-save on inactivity

500,0-1 98%

```
#!/bin/zsh

acc=root@192.168.3.8
device=2n21i811800000cvj0o0

(
  if [ "${1:-on}" = "on" ]; then
    echo 'killall xtightvncviewer; killall -9 write-paper; DISPLAY=:0 /root/write-paper/write-paper; e
  elif [ "$1" = "edit" ]; then
    convert "$2" -resize 2560x1600 ~/tmp/paper.png
    scp ~/tmp/paper.png root@192.168.3.8:/tmp/paper.png
    ssh "$acc" adb -s $device push /tmp/paper.png /data/debian-jessie/tmp/paper.png
    sleep 1
    echo 'killall xtightvncviewer; killall -9 write-paper; DISPLAY=:0 /root/write-paper/write-paper /t
  else
    echo 'killall -9 write-paper; exit' | ssh "$acc" adb -s $device shell su -c /data/debian-jessie/en
    displays Write fix
  fi
) &

sleep 2

echo "Press enter to copy back."
read

ssh "$acc" adb -s $device pull /data/debian-jessie/tmp/paper.png /tmp/paper.png
sleep 1
scp root@192.168.3.8:/tmp/paper.png ~/tmp/paper.png
convert "$2" -resize 2560x1600 ~/tmp/paper.png -flatten ~/tmp/paper.new.png
cp ~/tmp/paper.new.png "$2"

echo "Result:"
echo "$2"
```

default background

upload + edit

re-enable VNC display

download + crop

"~/bin/tablet-paper" 33L, 1166C 1,1 All

Get it at:

gitli.stratum0.org/Drahflow/remoteplay/write-paper

github.com/Drahflow/bin

<Happy Hacking>

<Drahflow>
<drahflow.name/contact.html>

Stratum 0 e. V. Braunschweig
`https://stratum0.org/`