Open Waveforms for Sustainable, Low-Power E-Paper Displays

Alex Wenger

FOSDEM'26 | Brussels Jan 31 & Feb 1 2026

Why Displays Matter for Energy

Displays are everywhere – and their energy adds up.

Rapid growth of digital displays:

- **- Public signage**
- **- Smart meters**
- **- Industrial HMIs**
- **- IoT & infrastructure dashboards**

Sustainability challenge:

- **- We should not _remove_ displays**
- **- We must _design and drive them intelligently**

E-Paper: Not “Energy-Free” by Default

- Zero power in static state

- Very slow updates
  - seconds to minutes
  - Depends on temperature & color depth

- Limited color & grayscale compared to emissive displays
- Real energy cost occurs during updates
- Update energy depends strongly on driving strategy

E-Paper Display Fundamentals

Pixel consists of several of microcapsules, each contains:

- Electrically charged pigment particles
- A clear, insulating carrier fluid

Typical black & white system:

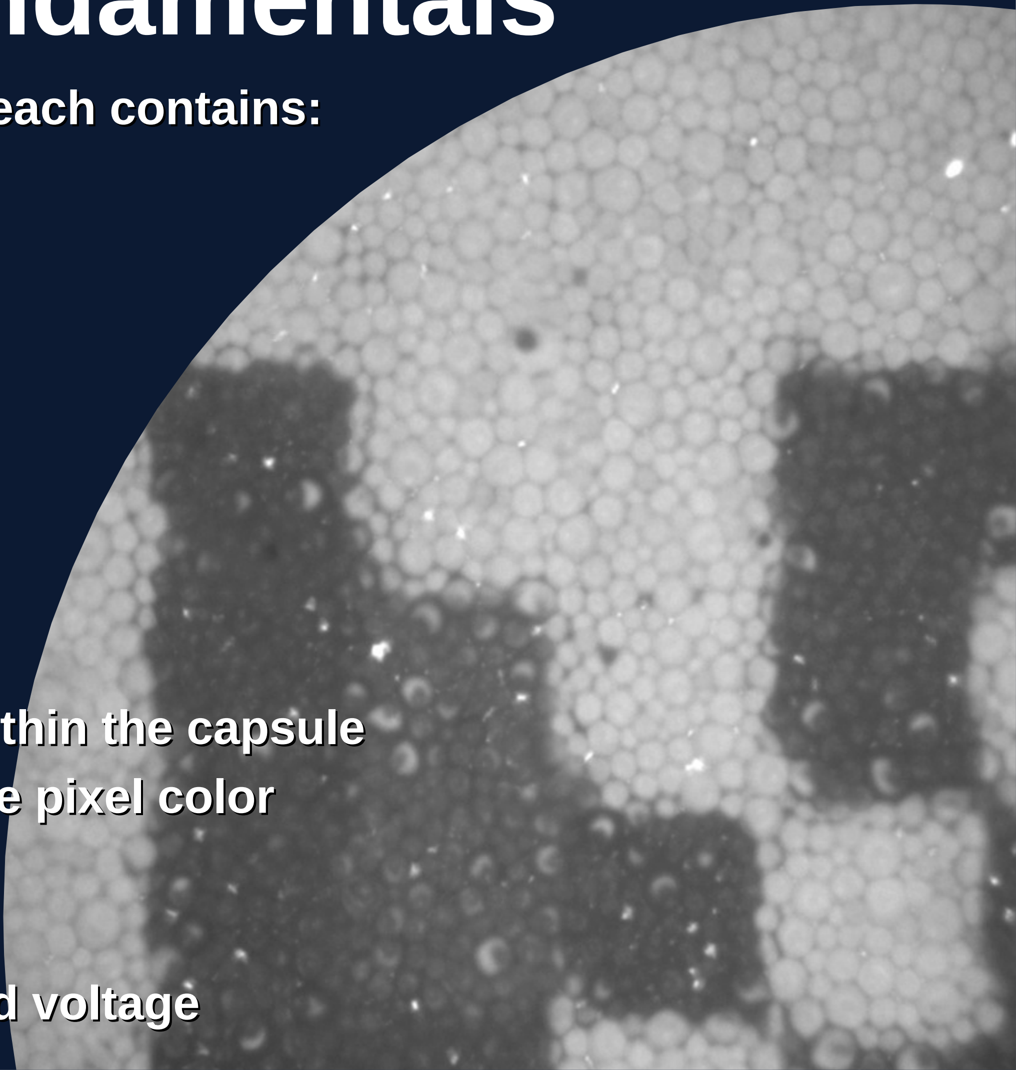
- White particles: negatively charged
- Black particles: positively charged

Image formation

- Applied electric field moves particles within the capsule
- Particles visible at the top surface define pixel color

Capsule state is bi-stable:

- Particles remain in place without applied voltage



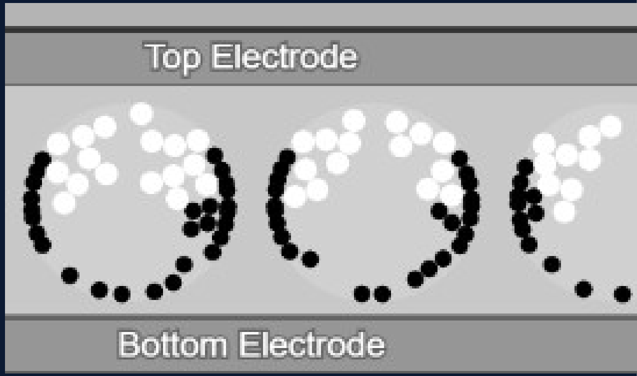
Particle Polarity and Motion

Electrophoretic particle behavior

- Pigment particles carry a defined **electric charge**
- Applied voltage creates an **electric field** across the capsule
- Particle motion depends on:

    Polarity (direction)

    Field strength and fluid viscosity (speed)



Implications for driving

- Voltage polarity controls motion direction
- Voltage magnitude and pulse duration control displacement
- Overdriving causes overshoot, ghosting, and stress

More Than Black & White

Approach 1: Color filters

RGB color filter on top of B/W E-Paper

Pros:

- No impact on update speed
- Full grayscales and mixed colors possible

Cons:

- Reduced white reflectance
- Lower contrast and weak colors
- Additional optical and manufacturing complexity

More Than Black & White

Approach 2: Multiple pigment particles

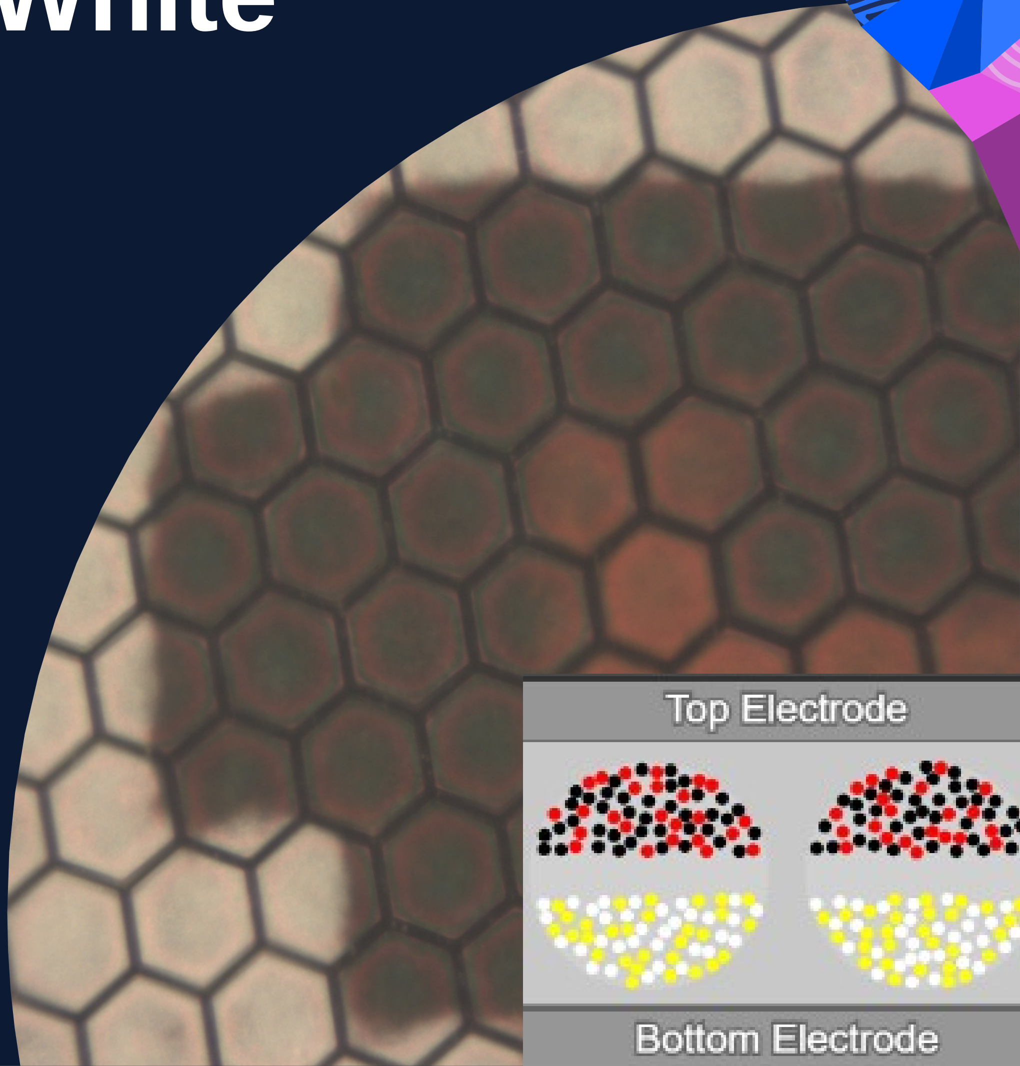
More than one pigment type inside the capsule

- Black/White/Red/Yellow
- White/Yellow/Cyan/Magenta

Constraints:
- Only two electrical polarities available
- Particles distinguished by mobility / mass

Pros:
- High contrast, paper-like appearance

Cons:
- Very slow updates (seconds to minutes)
- Little to no color mixing



Building a Screen from E-Paper Foil

Driving constraints

- Unlike LCDs, multiplexing is not feasible
- Only a very limited number of passive segments possible
- Practical displays require a TFT active-matrix backplane

Implications

- Custom TFT backplanes → expensive tooling
- Per-pixel storage enables controlled waveforms
- Requires VCOM offset for DC balance

Controlling an E-Paper Foil with TFT Backplane

A) “dumb” Driver IC on the TFT panel

External controller generates:

Gate selection signals & source driving signals

Controller needs to know everything about waveforms and voltages

B) Cost-optimized panels integrate a driver + controller IC directly on the TFT glass

Host system sends:

Image data (SPI / I²C)

Refresh command

Waveform execution handled inside the IC

Controlling an E-Paper Foil with TFT Backplane

A) “dumb” Driver IC on the TFT panel

External controller generates:

Gate selection signals & source driving signals

Controller needs to know everything about waveforms and voltages

B) Cost-optimized panels integrate a driver + controller IC directly on the TFT glass

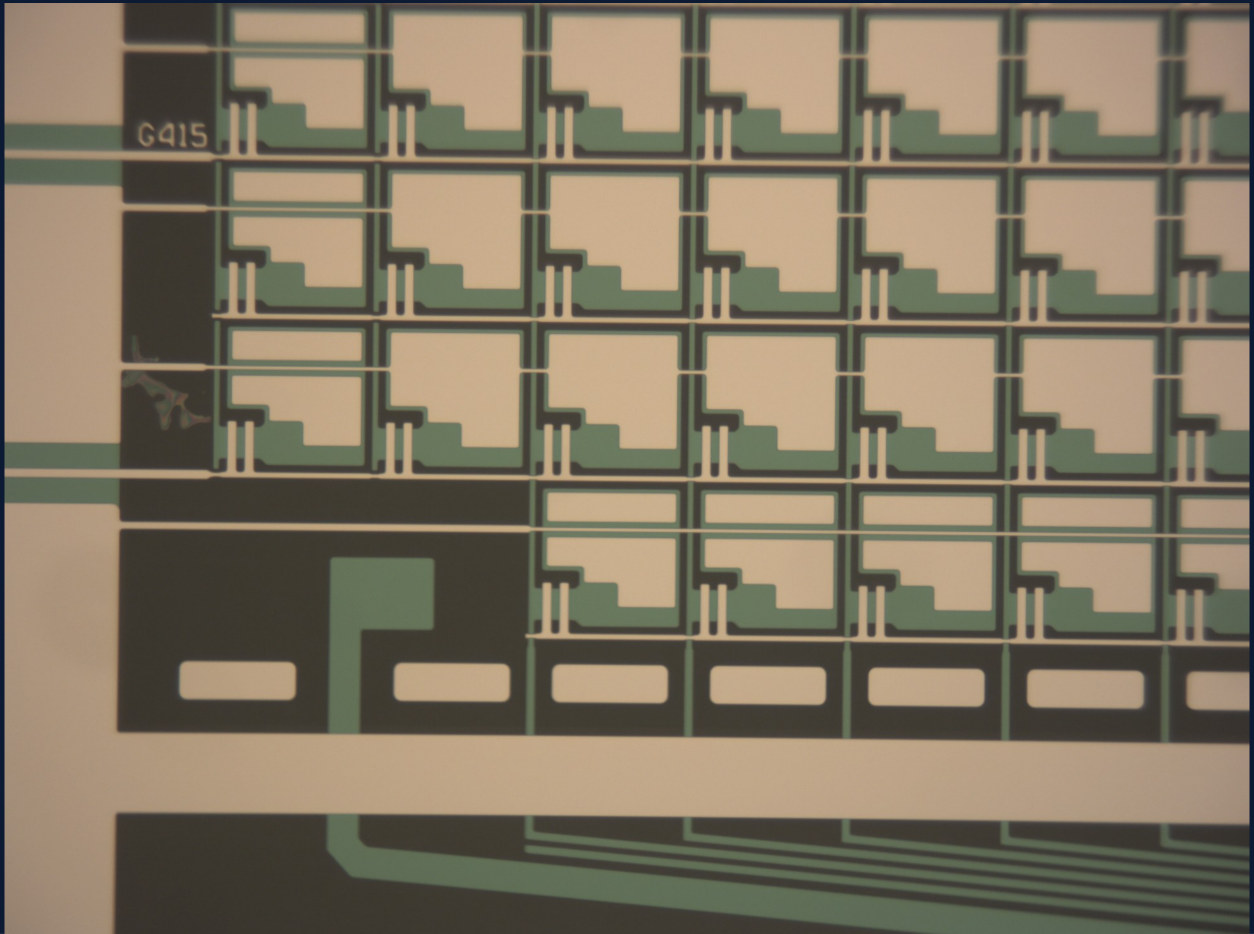
Host system sends:

Image data (SPI / I²C)

Refresh command

Waveform execution handled inside the IC

Controlling an E-Paper Foil with TFT Backplane

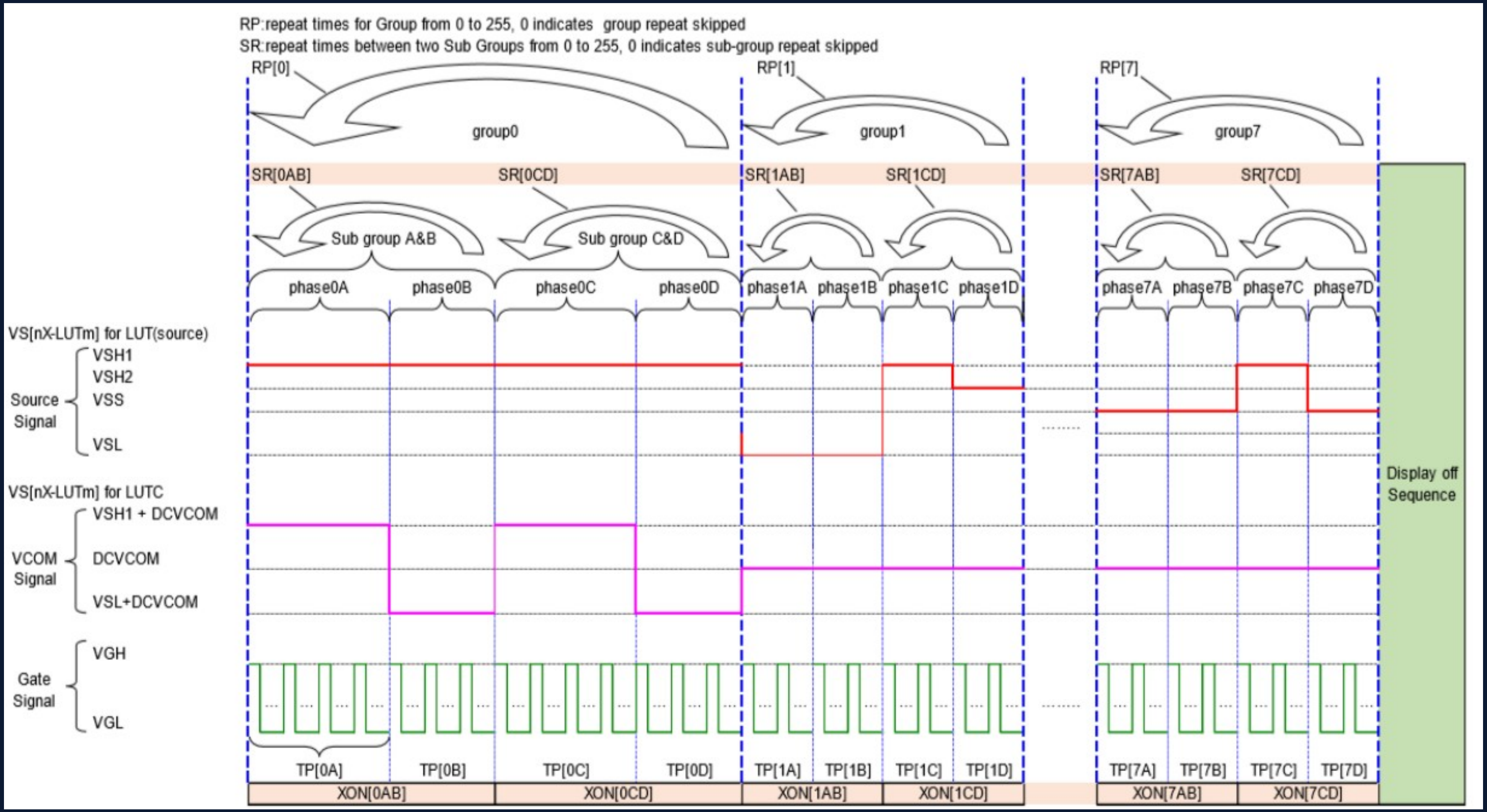


From Waveforms to Look-Up Tables (LUTs)

E-Paper driving requires complex physical waveforms

- Typical waveforms use 3–7 discrete voltage levels
- Combination of very short and very long pulses
- One global waveform for the TFT backplane
- Separate pixel waveforms for Black, White, Gray, or Colors
- More Waveforms for differential Updates: White → Black, Black → White,...
- Waveforms are compressed into Look-Up Tables (LUTs)
- Several complete sets of LUTs for different Temperature Ranges

Example LUT (Driver SSD1683)



Example LUT (Driver SSD1683)

RP: Repeat count for the whole block

VS…: Select one of 4 possible voltages

SR_AB: Repeat count for Step A/B

SR_CD: Repeat count for Step C/D

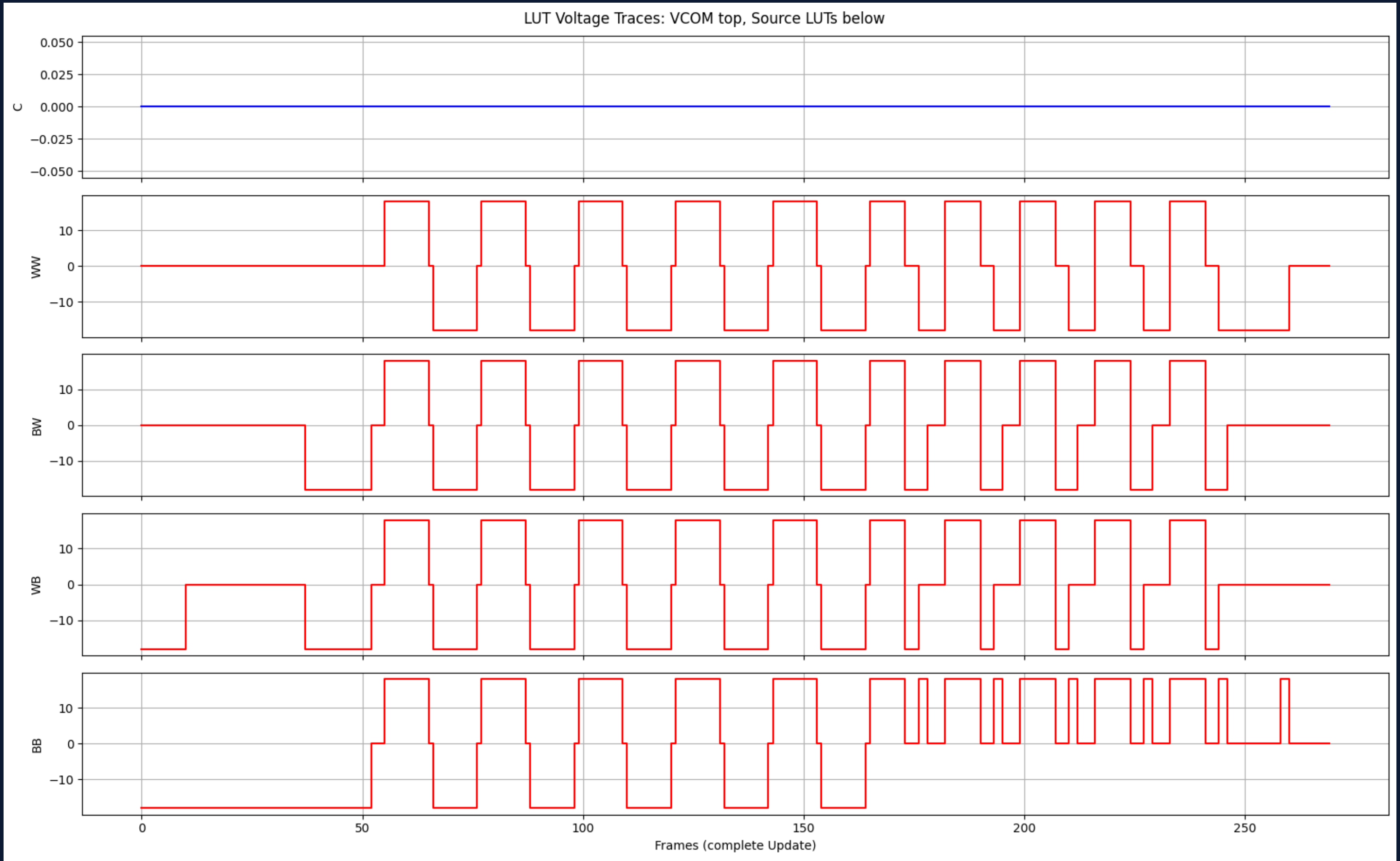
Note: Each E-Paper driver IC may have a slightly different LUT format

addr.	D7	D6	D5	D4	D3	D2	D
0	RP LUTC 0						
1	VS-0A-LUTC		TP LUTC 0A				
2	VS-0B-LUTC		TP LUTC 0B				
3	VS-0C-LUTC		TP LUTC 0C				
4	VS-0D-LUTC		TP LUTC 0D				
5	SR LUTC 0AB						
6	SR LUTC 0CD						
7	RP LUTC 1						
8	VS-1A-LUTC		TP LUTC 1A				
9	VS-1B-LUTC		TP LUTC 1B				
10	VS-1C-LUTC		TP LUTC 1C				
11	VS-1D-LUTC		TP LUTC 1D				
12	SR LUTC 1AB						
13	SR LUTC 1CD						
14	RP LUTC 2						
…	…						
…	…						
…	…						
50	VS-7A-LUTC		TP LUTC 7A				
51	VS-7B-LUTC		TP LUTC 7B				
52	VS-7C-LUTC		TP LUTC 7C				
53	VS-7D-LUTC		TP LUTC 7D				
54	SR LUTC 7AB						
55	SR LUTC 7CD						
56	RP LUTR 0						
57	VS-0A-LUTR		TP LUTR 0A				

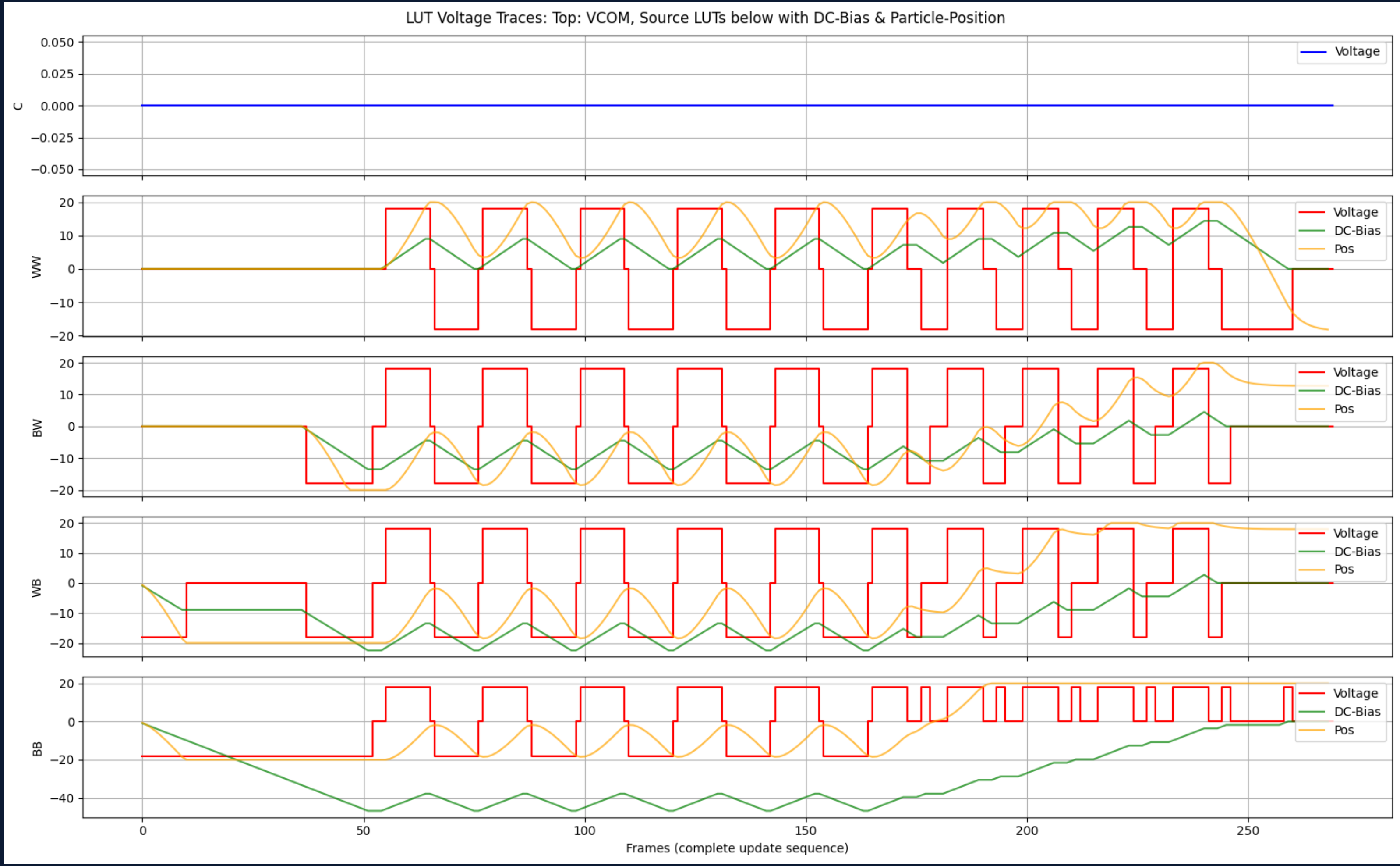
Example LUT (Driver SSD1683)

```
lut_4gray_data = bytes([
    0x01, 0x0A, 0x1B, 0x0F, 0x03, 0x01, 0x01,   # 000 RP LUTC 0
    0x05, 0x0A, 0x01, 0x0A, 0x01, 0x01, 0x01,   # 007 RP LUTC 1
    0x05, 0x08, 0x03, 0x02, 0x04, 0x01, 0x01,   # 014 RP LUTC 2
    0x01, 0x04, 0x04, 0x02, 0x00, 0x01, 0x01,   # 021 RP LUTC 3
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 028 RP LUTC 4
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 035 RP LUTC 5

    0x01, 0x0A, 0x1B, 0x0F, 0x03, 0x01, 0x01,   # 042 RP LUT WW 0
    0x05, 0x4A, 0x01, 0x8A, 0x01, 0x01, 0x01,   # 049 RP LUT WW 1
    0x05, 0x48, 0x03, 0x82, 0x84, 0x01, 0x01,   # 056 RP LUT WW 2
    0x01, 0x84, 0x84, 0x82, 0x00, 0x01, 0x01,   # 063 RP LUT WW 3
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 070 RP LUT WW 4
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 077 RP LUT WW 5

    0x01, 0x0A, 0x1B, 0x8F, 0x03, 0x01, 0x01,   # 084 RP LUT BW 0
    0x05, 0x4A, 0x01, 0x8A, 0x01, 0x01, 0x01,   # 091 RP LUT BW 1
    0x05, 0x48, 0x83, 0x82, 0x04, 0x01, 0x01,   # 098 RP LUT BW 2
    0x01, 0x04, 0x04, 0x02, 0x00, 0x01, 0x01,   # 105 RP LUT BW 3
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 112 RP LUT BW 4
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 119 RP LUT BW 5

    0x01, 0x8A, 0x1B, 0x8F, 0x03, 0x01, 0x01,   # 126 RP LUT WB 0
    0x05, 0x4A, 0x01, 0x8A, 0x01, 0x01, 0x01,   # 133 RP LUT WB 1
    0x05, 0x48, 0x83, 0x02, 0x04, 0x01, 0x01,   # 140 RP LUT WB 2
    0x01, 0x04, 0x04, 0x02, 0x00, 0x01, 0x01,   # 147 RP LUT WB 3
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 154 RP LUT WB 4
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 161 RP LUT WB 5

    0x01, 0x8A, 0x9B, 0x8F, 0x03, 0x01, 0x01,   # 168 RP LUT BB 0
    0x05, 0x4A, 0x01, 0x8A, 0x01, 0x01, 0x01,   # 175 RP LUT BB 1
    0x05, 0x48, 0x03, 0x42, 0x04, 0x01, 0x01,   # 182 RP LUT BB 2
    0x01, 0x04, 0x04, 0x42, 0x00, 0x01, 0x01,   # 189 RP LUT BB 3
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 196 RP LUT BB 4
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,   # 203 RP LUT BB 5

    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   # 210 leer
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   # 217 leer
    0x02,                                       # 224 FR
    0x00, 0x00,                                 # 225 XON
    0xFE                                        # 227 EOPT
])
```

Example LUT (Driver SSD1683)



Example LUT (Driver SSD1683)



Hacking Grayscales on a 2-bit Driver

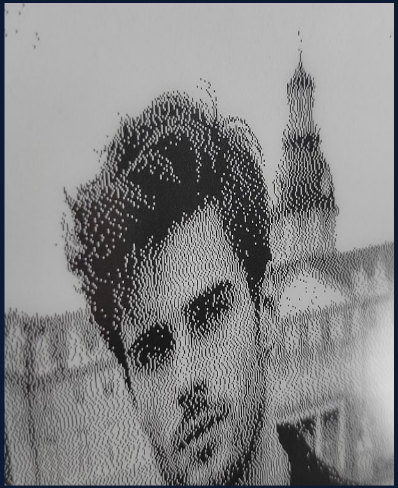
Driver stores 1 or 2 bits per pixel → max 2/4 LUTs

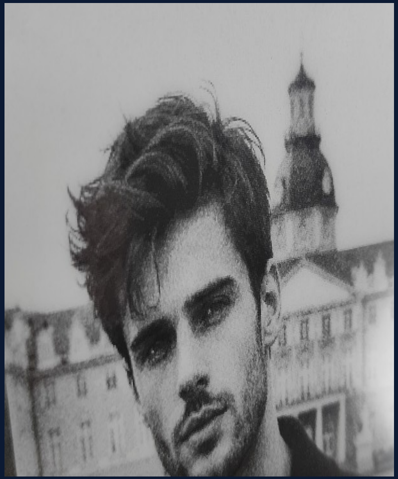
Phase 1: Display Black & White image
Only real black pixels stay black
All others forced to white

Phase 2: Load custom LUTs
Backplane LUT → GND/VCOM offset
Transparent LUT → does not alter pixels
Grayscale LUT → White → target grayscale (dynamic)

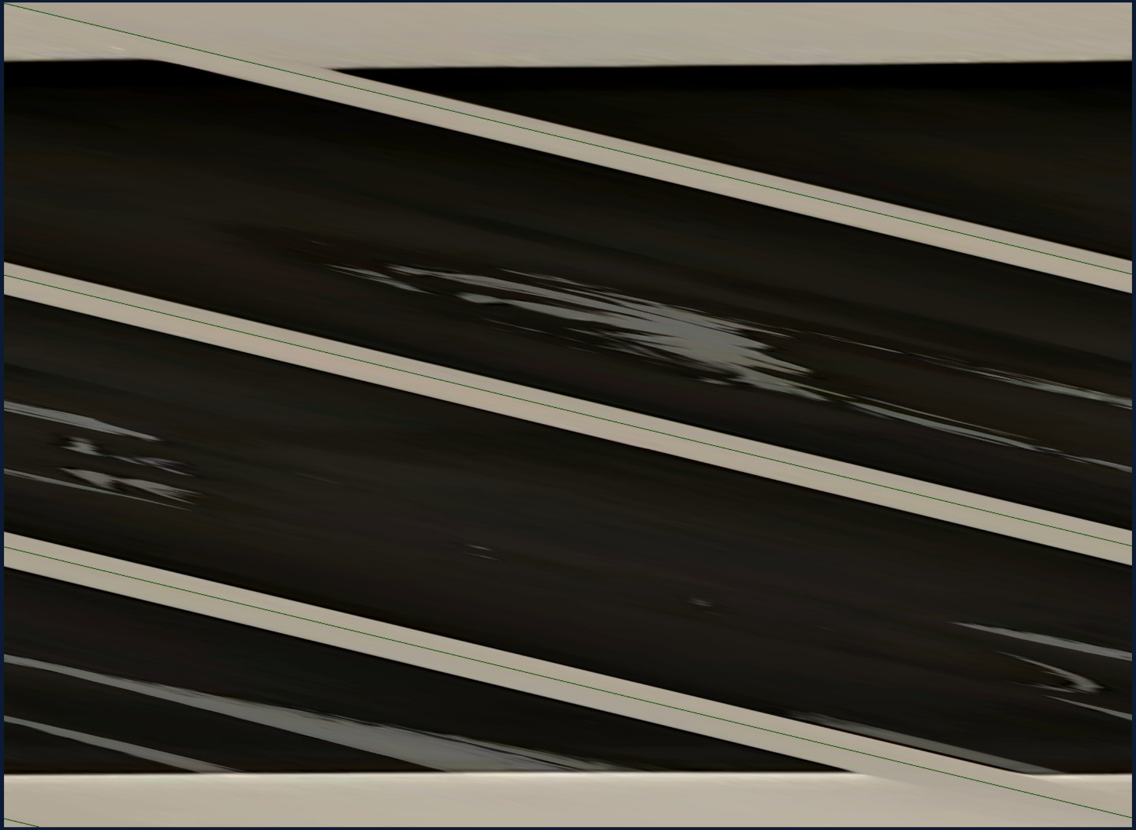
Phase 3: Incrementally write grayscale pixels
Only pixels set for grayscale X are updated
Repeat for all gray levels

Phase 4: Reapply remaining pure white pixels to reduce ghosting





Hacking Grayscales on a 2-bit Driver



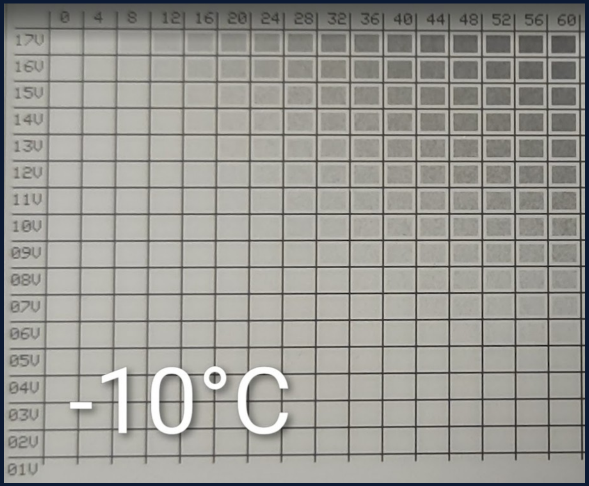
What's Next: Temperature

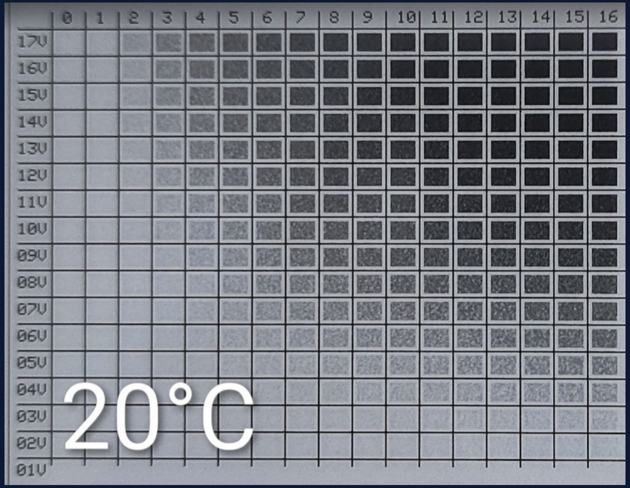
Measure E-Paper response to timing, voltage, and temperature

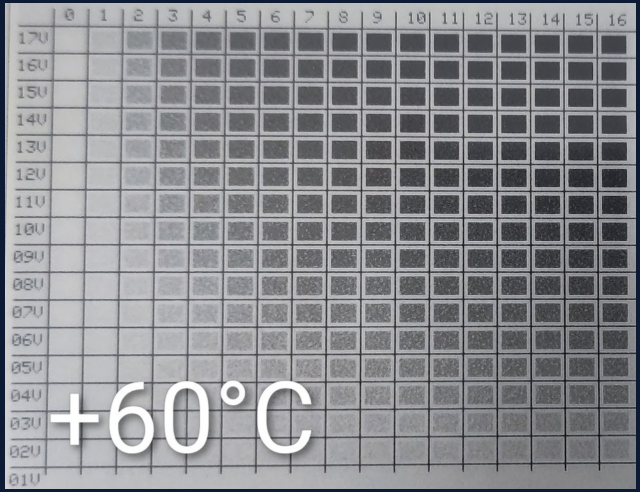
Use data to make LUT generation robust against temperature changes

Optional: build a particle simulator for waveform testing

High temperatures → risk of display damage if pulses not adjusted







What’s Next: Faster Grayscale & Color E-Paper

2-bit/pixel optimization:

- Update 3 grayscales per frame step

Incremental multi-phase updates:

- Step 1: write 4 base levels
- Step 2: adjust levels by half-steps
- Step 3: further fine steps for all shades

Future potential:

- Extend method to multi-color E-Paper

Creating Your Own LUT

- Don’t forget the VCOM-DC setting (Burn In!)
  - Sometimes you can not read it from OTP Memory
  - Some newer drivers can measure optimal DC offset automatically
- Avoid permanent DC offset on the panel
  - Switch off power as soon as possible
  - Use DC-balanced waveforms
    - E.g., if writing 200 ms at +17 V, apply 200 ms at −17 V before
    - For multi-voltage driving (−17V, −5V, 0V, +5V, +17V), calculate total DC and compensate
- Combine factory LUTs with custom LUTs
- Add shake phase at the beginning to unstuck particles
- Use VCOM LUT to double voltage → double speed (But no partial Update)

Q&A

Let's share LUTs and experiences

Collect tips, tricks, and measurement results

Discuss waveform designs, DC balancing, and multi-phase updates

Goal: build a shared knowledge base for E-Paper driving

Alex Wenger

Mail: alex@wenger-online.de

https://github.com/Blueloop/E-Paper-driving-Waveforms

https://matrix.to/#/#EPaper-OpenWaveforms:matrix.org