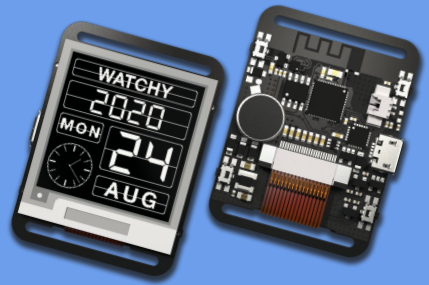
Designing and Building a Watch Face for E-ink with Watchy

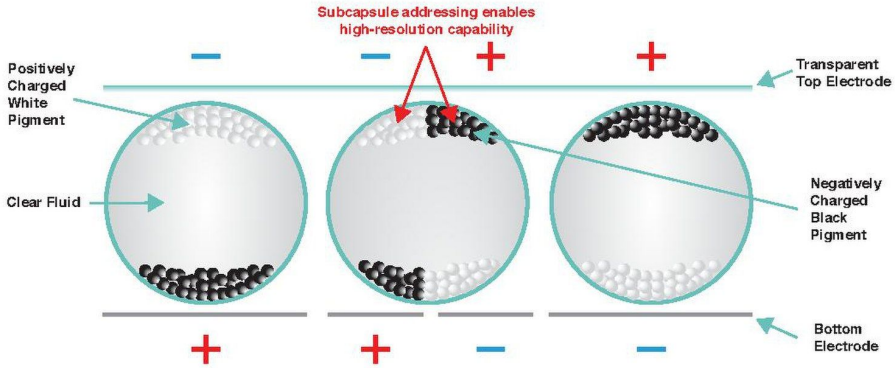
Teardown 2023

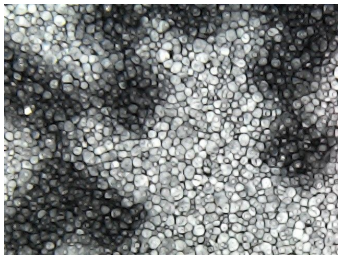
SQFMI

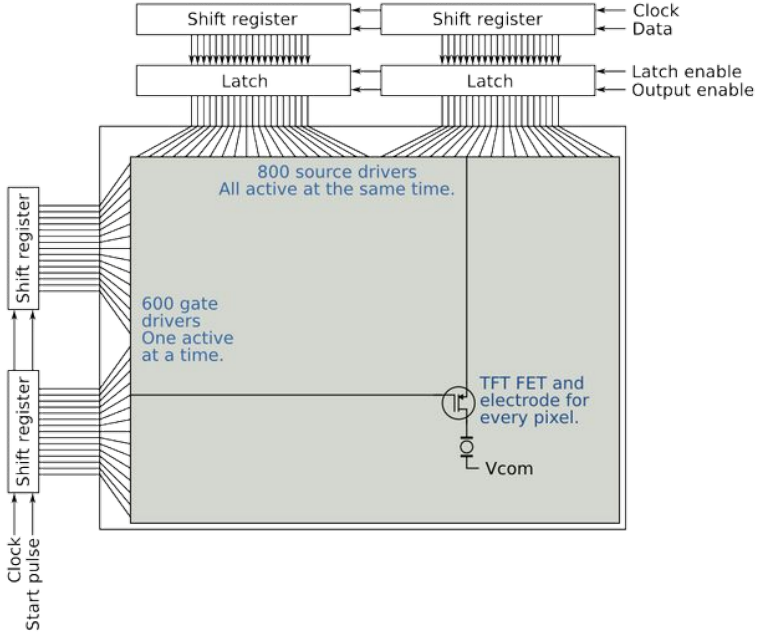


E-Ink Primer

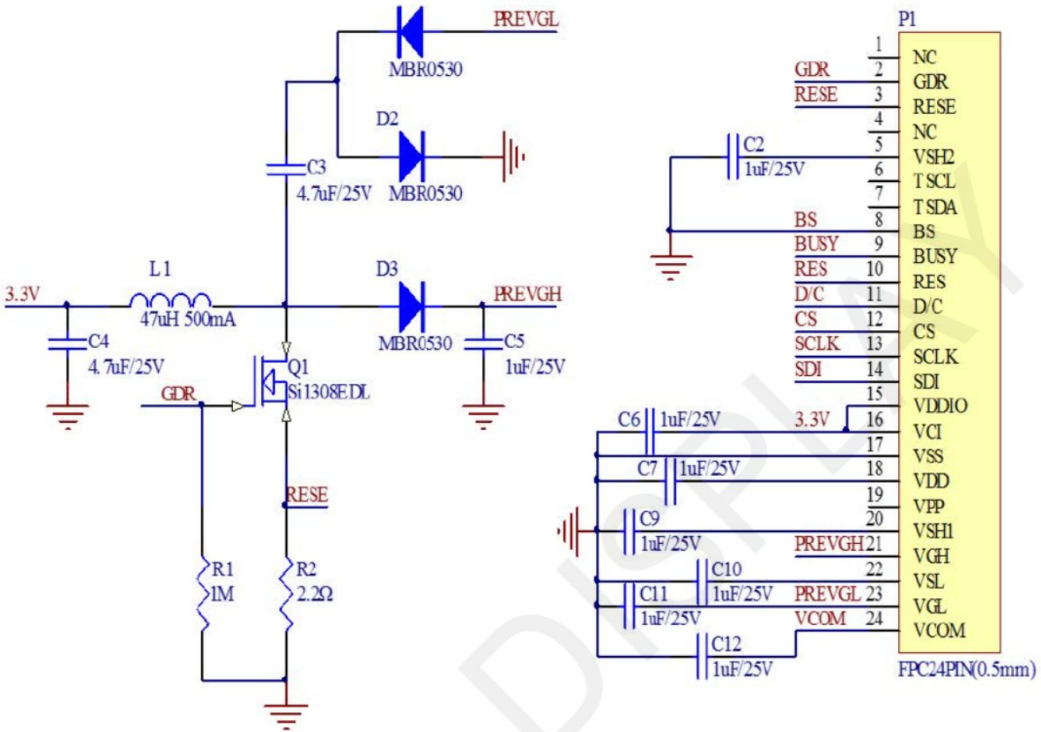
How does it work?

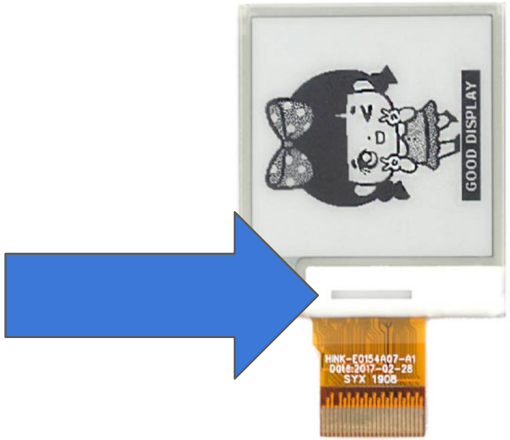






Driving Circuit

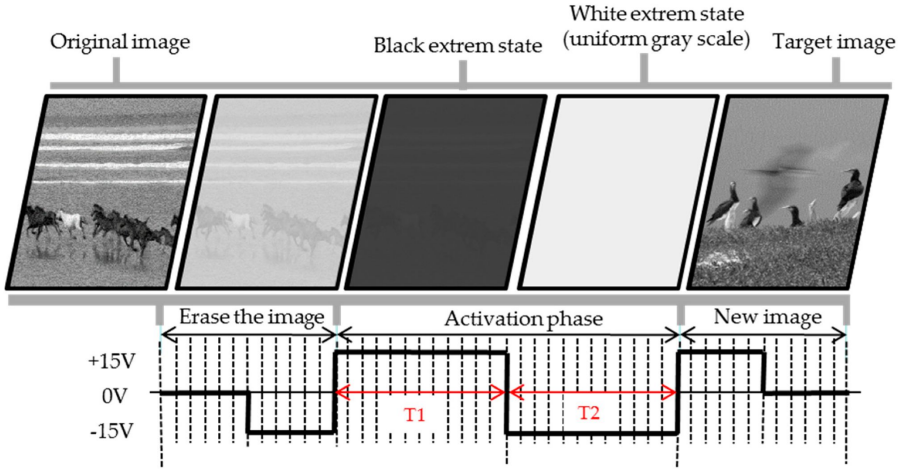




Partial vs Full Refresh

Update type	Global Update	Fast Update	Partial Update
Initialization	Panel initialization	Panel initialization	Panel initialization
Supported colors	Three-colors (BWR) and monochrome (BW)	Monochrome (BW)	Monochrome (BW)
Image data upload	Full screen image New image buffer with all 0x00 buffer	Full screen image Previous image buffer and new image buffer	Partial window image Previous image buffer and new image buffer
Panel update	Whole screen Flashing effect	Whole screen Fast mode	Partial window Fast mode
Image quality	Optimal quality	Possible ghosting	Possible ghosting (worse than fast update)
Upload image speed	Slow	Slow	Fast
Update image speed	Slow	Fast	Fast
Overall speed	Slow	Faster	Fastest

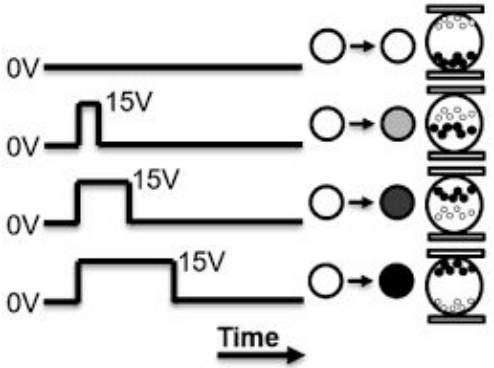
LUTs and Waveforms



1. Prevent DC buildup in the panel
2. Drive the particles ard towards the top and bottom layers i.e. black/white
3. Target voltage from frame

LUTs

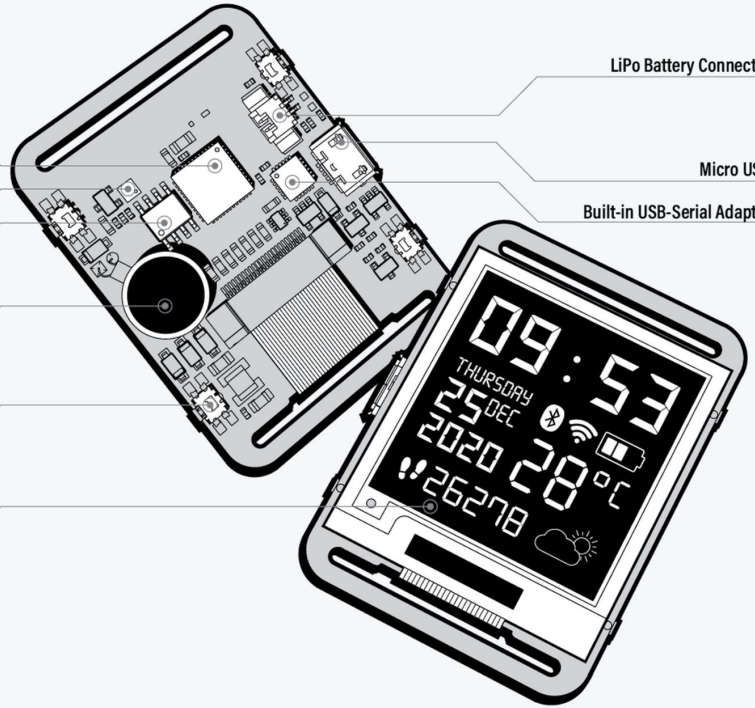
- Voltage
- Timing
- Temperature
- WW/WB/BW/BB



Watchy

WATCHY

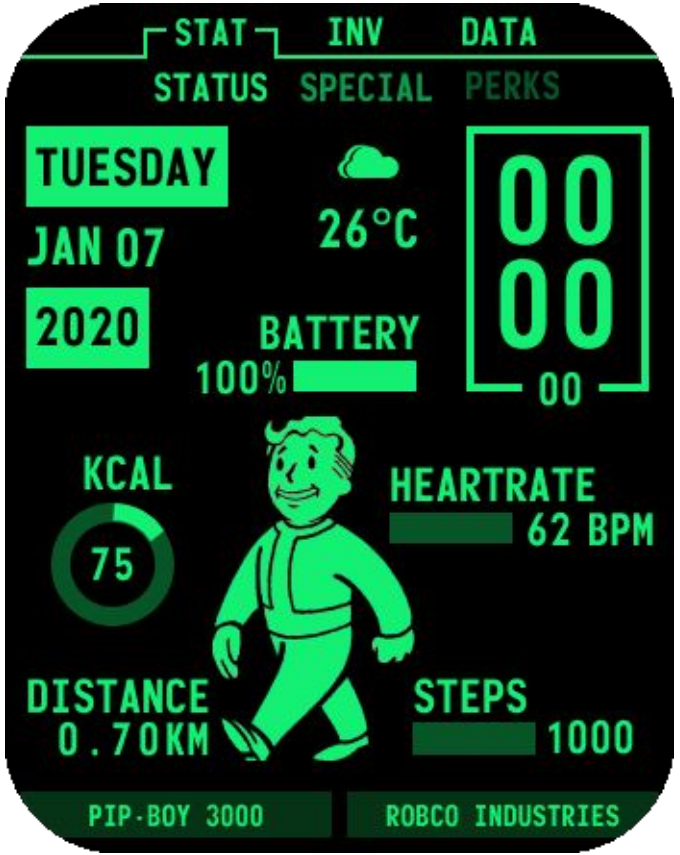
AN OPEN SOURCE E-PAPER WATCH

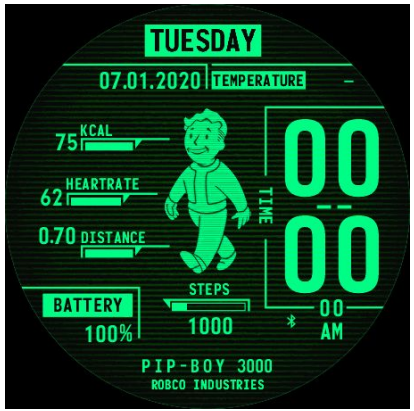


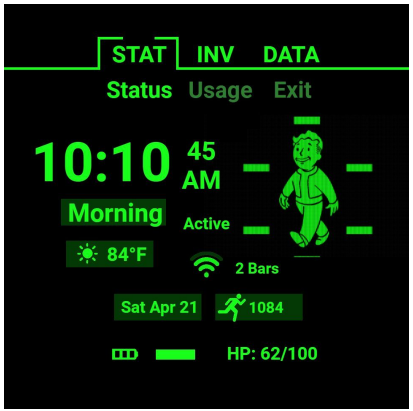
SQFMI

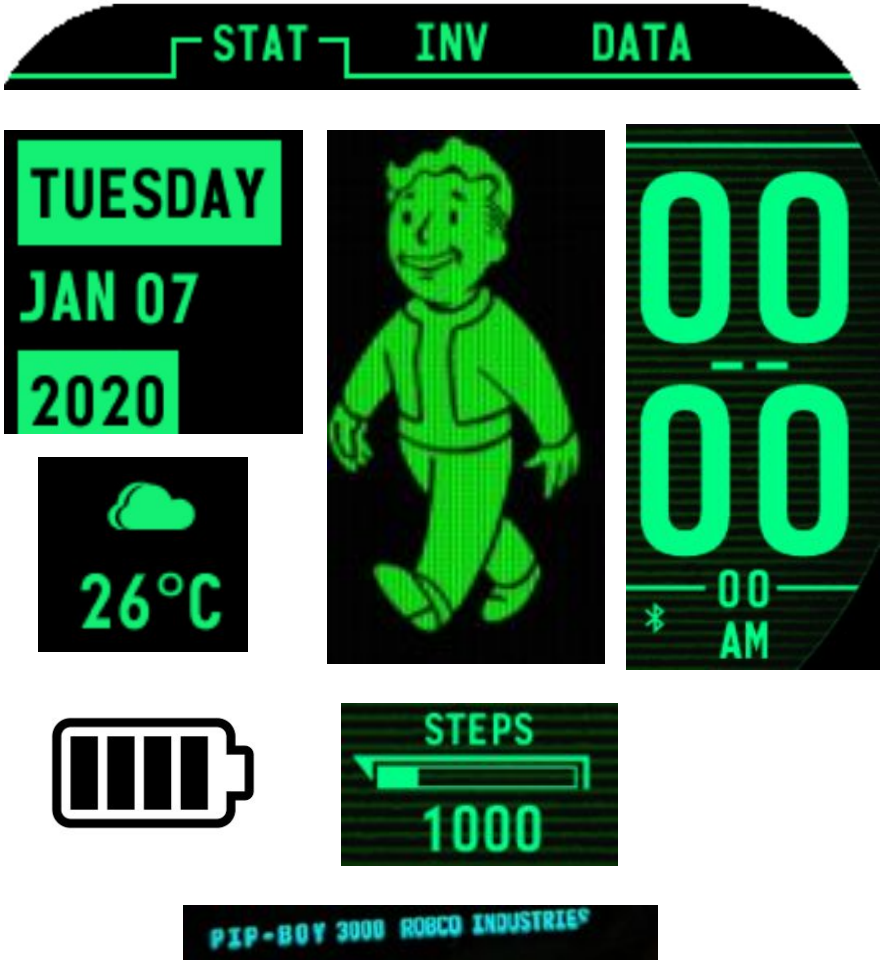
Watch Face

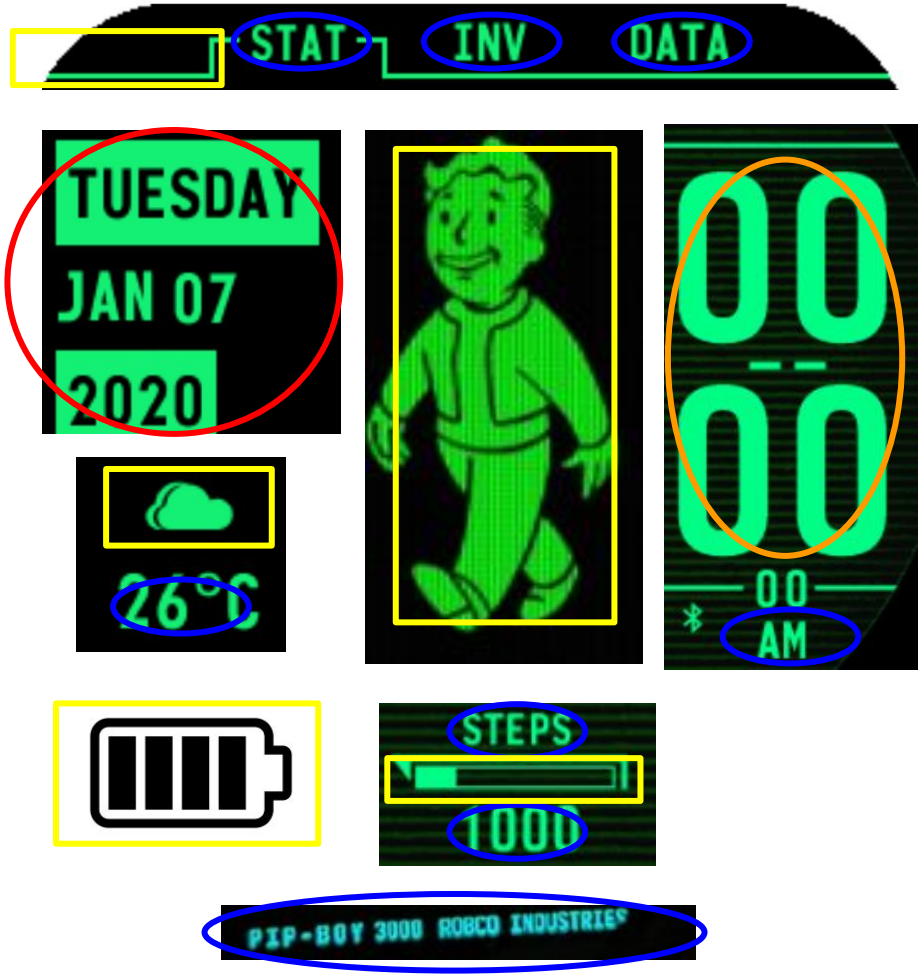
Concept - PIP Boy

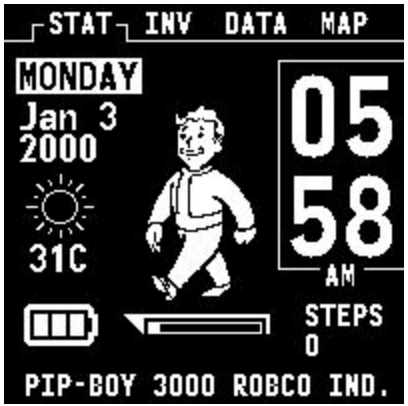







STAT
INV
DATA
TUESDAY
JAN 07
2020
26°C
00
00
00
AM
STEPS
1000
PIP-BOY 3000 ROBCO INDUSTRIES


STAT
INV
DATA
TUESDAY
JAN 07
2020
00
00
00
AM
STEPS
1000
PIP-BOY 3000 ROBCO INDUSTRIES


STAT INV DATA MAP
MONDAY
Jan 3
2000
31C
05
58
AM
STEPS
0
PIP-BOY 3000 ROBCO IND.

Fonts

- Convert from TTF to Byte Array (*.ttf to *.h)
  - https://rop.nl/truetype2gfx/
- Every font size must be converted
  - Every character is a glyph (byte array)
  - No font scaling at runtime
- Avoid fonts with anti-aliasing (gray will appear as black)

Images

- Convert from image to byte array
  - https://javl.github.io/image2cpp/
- Brightness/Alpha threshold converts pixel to either white or black
- Dithering “fakes” gray by tricking the eye through diffusing dots (error diffusion)
- Use BW bitmaps for pixel perfect images

Original

Binary

Original

Binary

Dithered

Software

- Arduino IDE
- Libraries
  - **ESP32-Arduino**
  - **Watchy Lib**
    - GxEPD2 (E-ink display driver)
    - Adafruit GFX Library
    - Arduino_JSON
    - DS3232RTC/Rtc_Pcf8563
    - WiFiManager

Code

https://github.com/sqfmi/watchy-pipboy

Links

https://watchy.sqfmi.com/

https://shop.sqfmi.com/

Best Practices

- Minimize updates to save power
- Use partial refresh where possible
- Use full refresh to clear out ghosting/artifacts