

SHA2017 Badge

hardware and software

Anne Jan Brouwer

Bas van Sisseren

NLUUG najaarsconferentie

16 November 2017

Who

- Anne Jan Brouwer
NoProtocol BV
high level architecture
- Bas van Sisseren
Quarantainenet BV
low level embedded



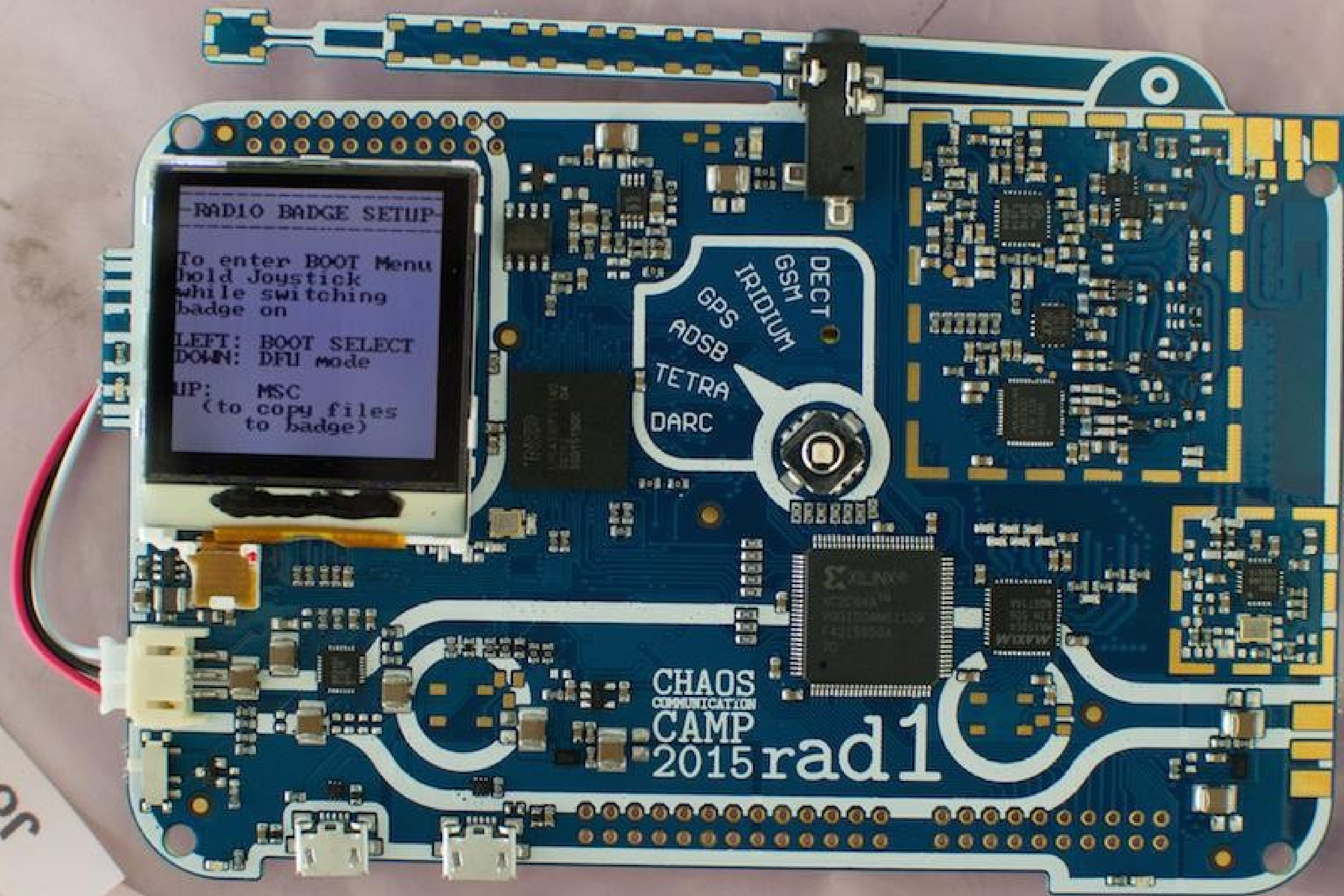
European hacker events

- Every 4 years in NL
 - since 1993
- Other “odd” years in DE
 - since 1999
- On “even” years smaller ones
- SHA2017



Event badges

- CCCamp 2011
 - r0ket
- EMF TiLDA
 - Many iterations
- CCCamp 2015
 - rad1o



RADIO BADGE SETUP

To enter BOOT Menu
hold Joystick
while switching
badge on

LEFT: BOOT SELECT
DOWN: DFU mode

UP: MSC
(to copy files
to badge)

CHAOS
COMMUNICATION
CAMP
2015rad1

Joystick

The first glue
dot goes to a
lower corner
Make

SHA2017 Badge

- Core values:
 - Function as a badge
 - Augment event
 - Easy to hack
- Distribute 3500+ awesome devices
 - All attendees, when they enter event



Brainstorms

- CCCamp augustus 2015
 - Big OLED screen
 - OpenWRT
- ETH0 Winter 2016
 - E-ink
 - ESP32
- EMF camp augustus 2016
 - MicroPython



Electromagnetic
Field 2016

Menu

Name

CHRISTIAN

TILDA
V3.141592653

Hardware

Proof of Concept

SHA

+2017

SHA2017 Badge

PoC

September 2016

- Arduino (Atmega328)
- Linksprite driver breakout 2.04" e-ink
- GDEH029A1 e-ink 2.9"
- Analog Joystick and push button





Hardware

Revisions

SHAA

+2017



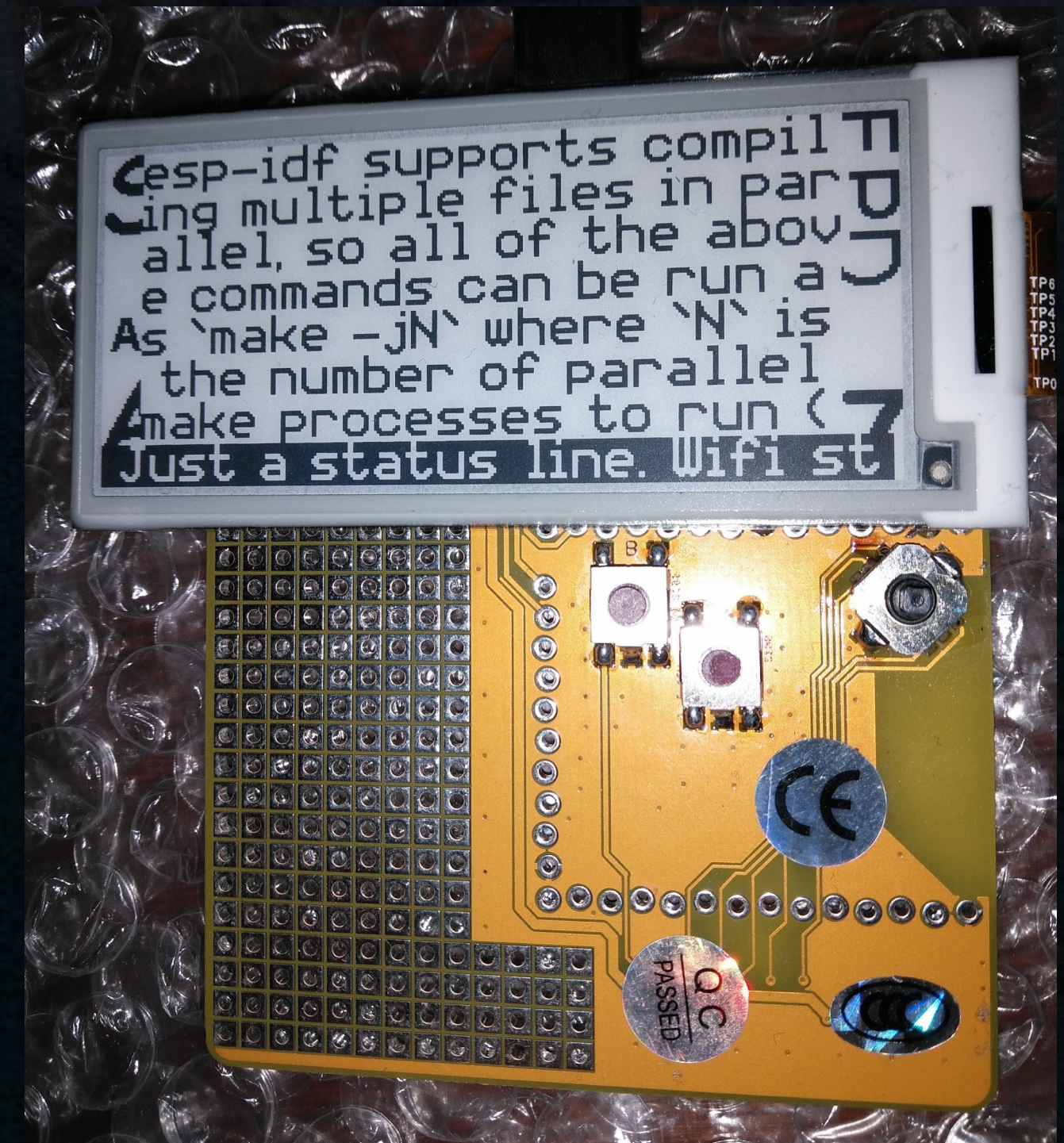
2017
Anne Jan
Brouwer
DEMO
DEMO

Waaron niet ook weer wat tekst
NickName
Anne Jan

SHA2017 Badge

v 0.0.x December 2016

- ESP32
- GDEH029A1
- CP2102 USB-serial
- Digital joystick & buttons
- Development area





SHA2017 Badge

v 0.1.0 Februari 2017

- ESP32
- GDEH029A1
- CP2102
- CPT112S Capacitive touch + proximity
- FXL6408 I/O expander



Demo menu
text demo 1
text demo 2
greyscale image 4
partial update test
ADC test
LEDs demo
UGFX demo

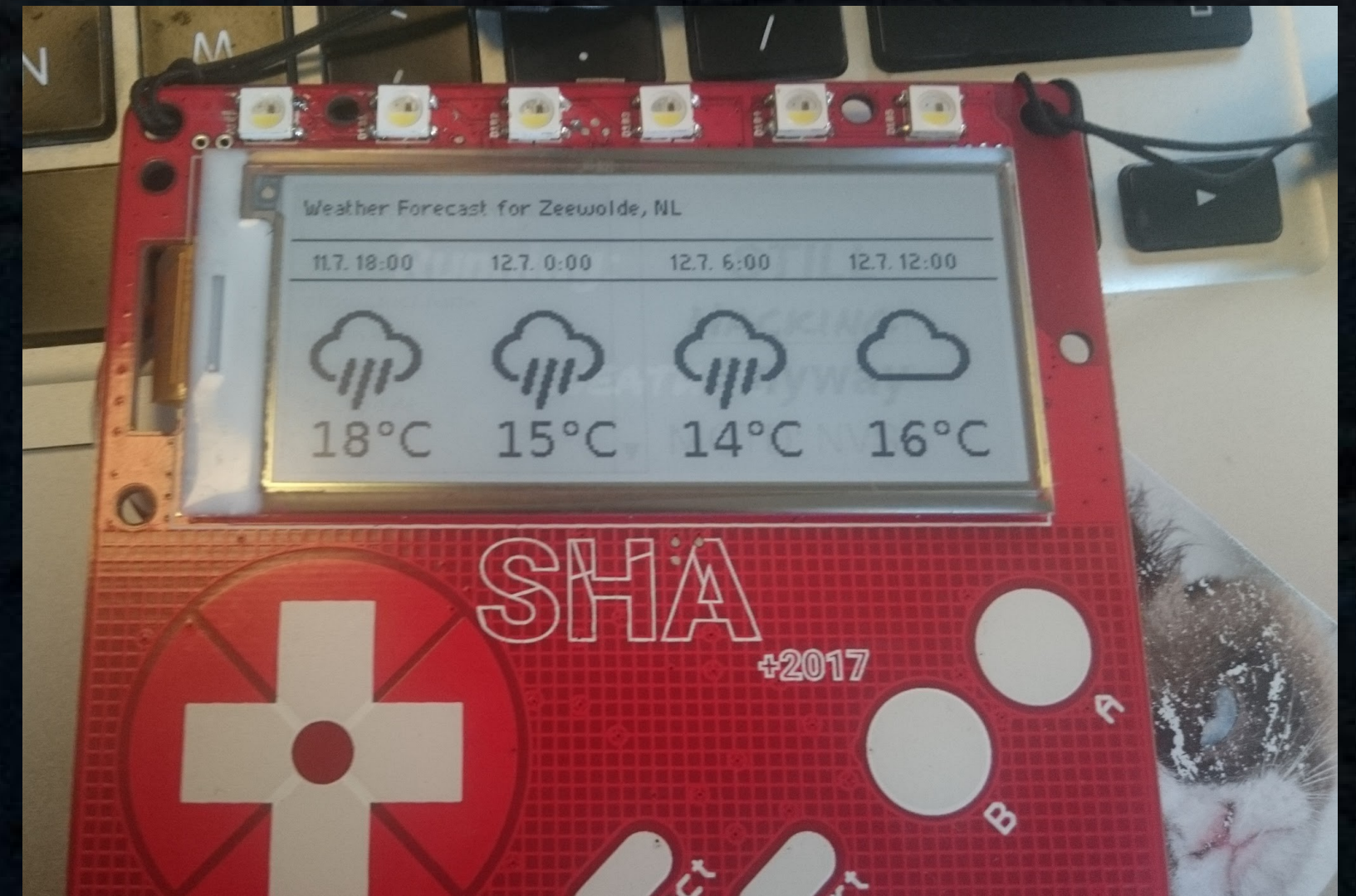
SHIA

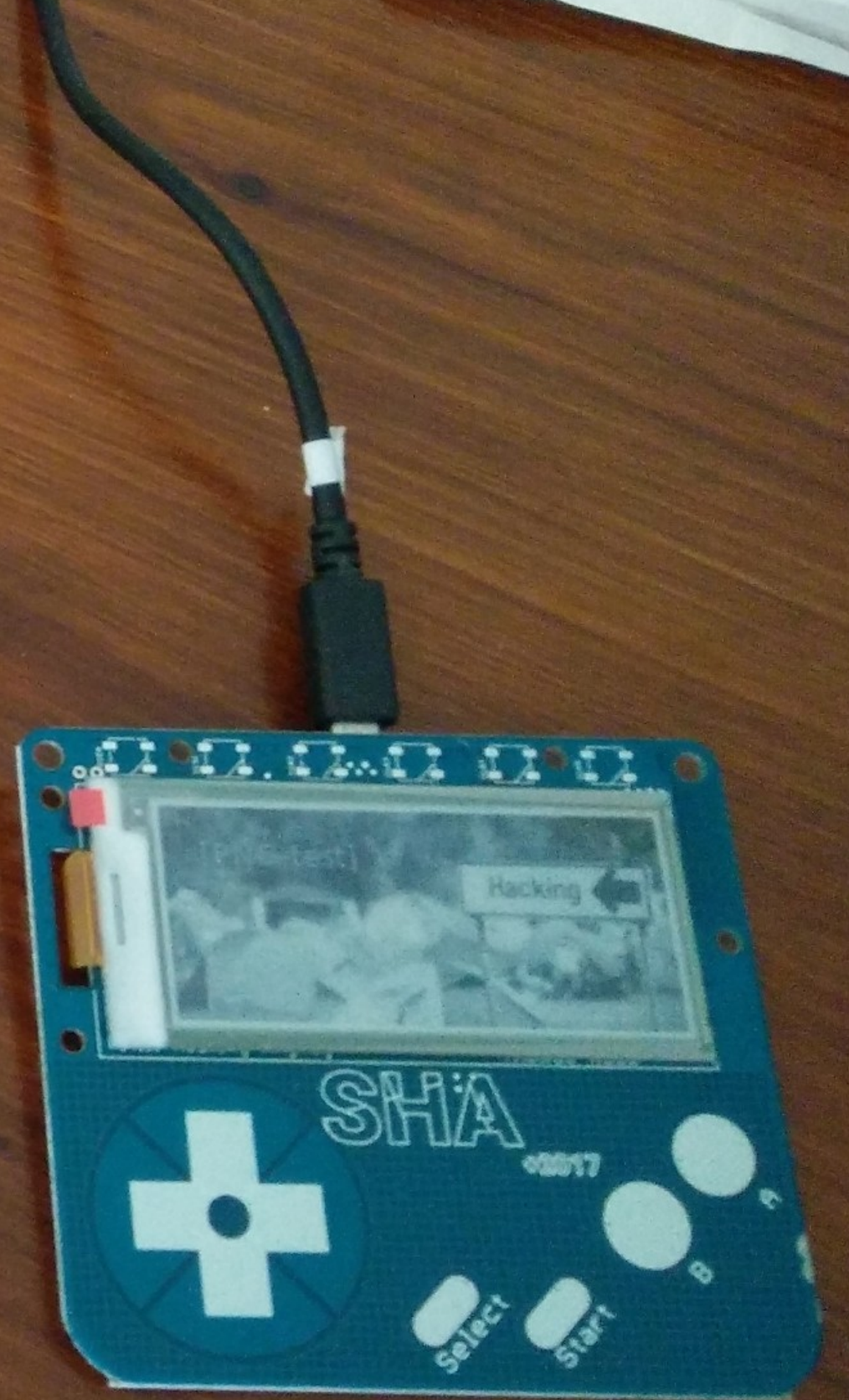
+2017

SHA2017 Badge

v 1.0.x May 2017

- ESP32
- DEPG0290B1
- CP2102
- MPR121QR2 Capacitive touch + I/O







Hardware

Overview

SHAA

+2017

SHA2017 Badge

- ESP32
 - Dual core 240 MHz Tensilica LX6
 - WiFi + BTLE
- DEPG0290B1
 - AMEPD
 - 296x128 pixels 1 bit
- CP2102
 - USB-serial
- MPR121QR2
 - Touch controller and IO multiplexer
- TP4056
 - Battery charger



Minigrip



Minigrip



Minigrip



Minigrip

BADGE

Hi, and welcome to your Badge kit!

It contains:

- Badge with e-ink display
- Battery
- Adhesive hook-and-loop pad
- 6x SK6812 RGBW LEDs
- Vibration Motor
- Lanyard

To get started, you can just power the badge, either by plugging in a micro-USB cable, or by connecting the battery. The LEDs and motor are included for DIY assembly.

Your badge will work fine without the LEDs and vibration motor, but your experience will be a lot more fun with those parts on the board! Check the other side of this flyer for a short guide on how to assemble the badge.

When you boot the badge for the first time, there might be a day zero firmware update available for you. The badge will inform you of such shamelessness and will likely choose to automatically update itself.

The update may take a little while, so grab another Mate and start reading the wiki for the latest info on the badge and an introduction to how to get started hacking it!

Feel free to drop by the badge support desk at the info tent if you're having trouble, or if you'd like to help out! Opening hours are on the wiki.

The hatchery, our app-repo, is at <https://badge.sha2017.org>

Our wiki page is located at: <https://wiki.sha2017.org/w/badge>

And our git-repo can be found at <https://github.com/SHA2017-badge>

You can chat with us through IRC in [#sha2017-badge](#) on [freenode](#)

Be sure to tweet your cool hacks to [@SHA2017Badge](#). We can't wait to see what amazing hacks and eggs you come up with!

+4973

ASSEMBLING THE BADGE

LEDs

MOTOR

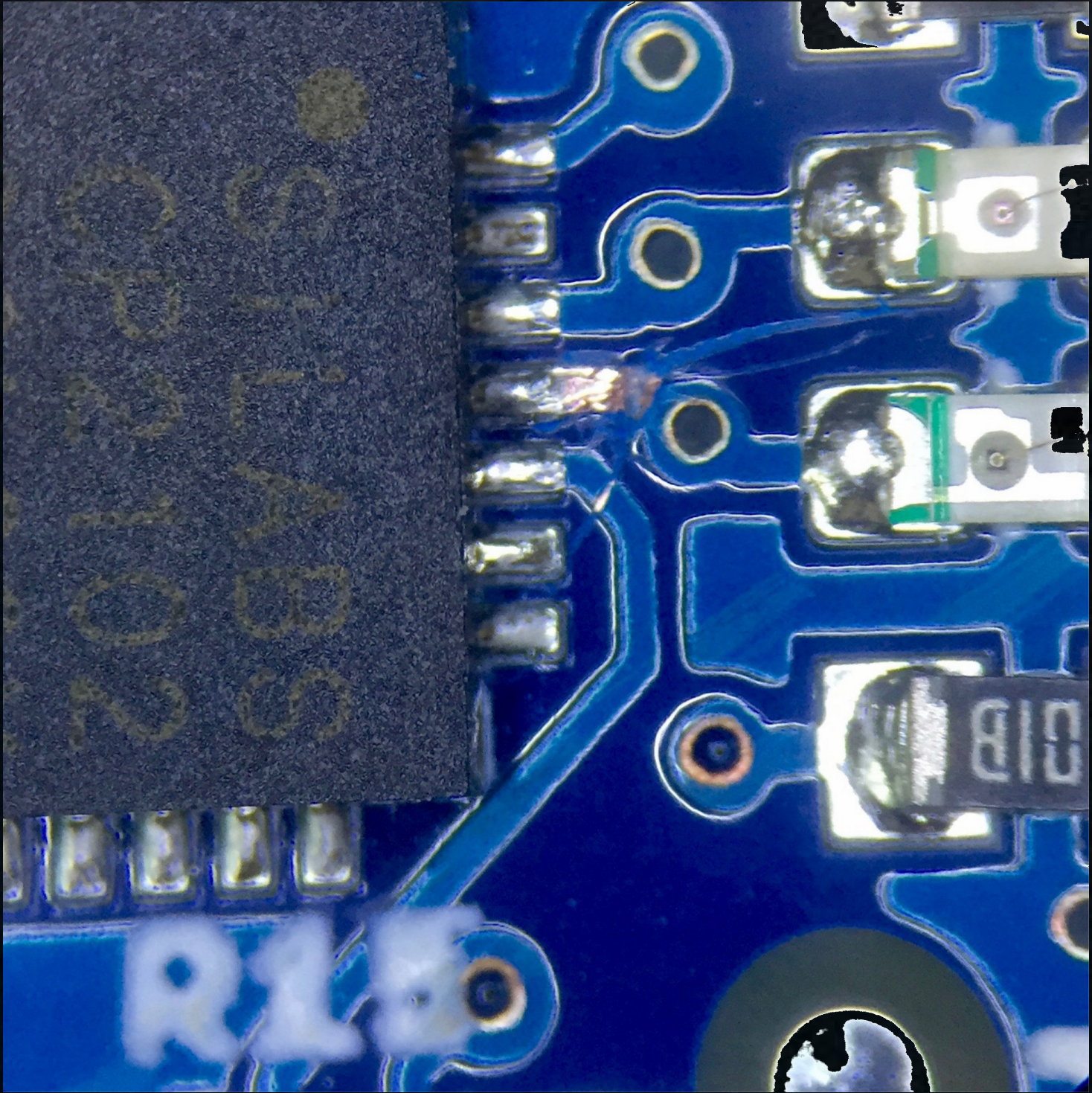
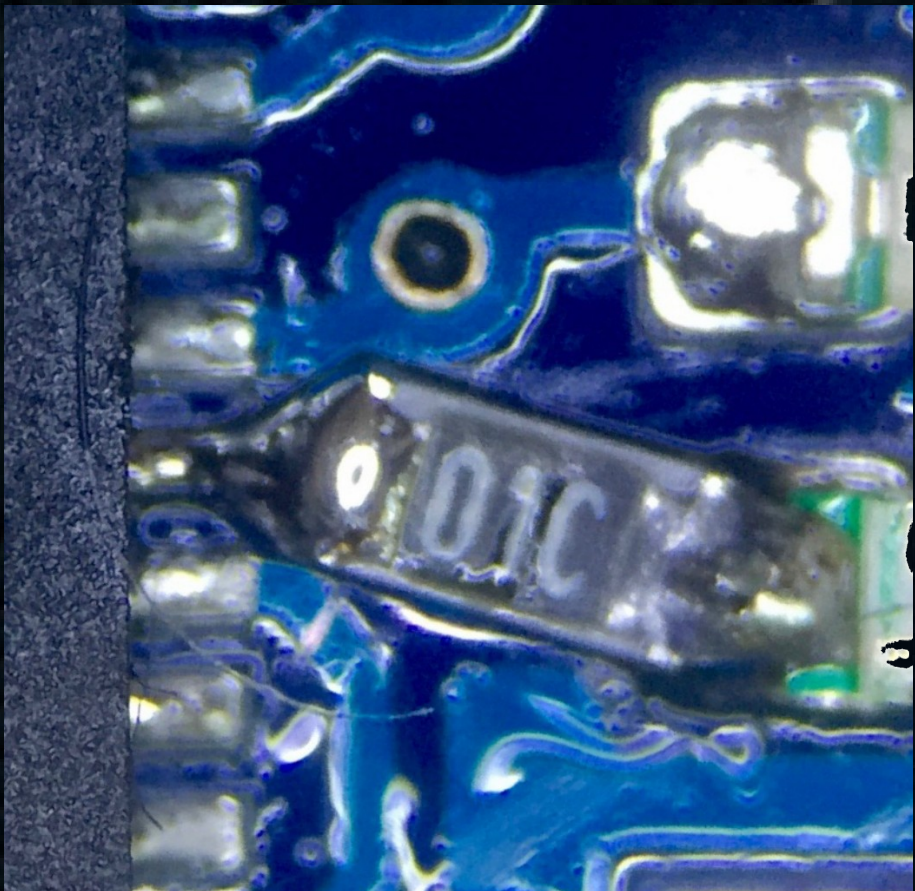
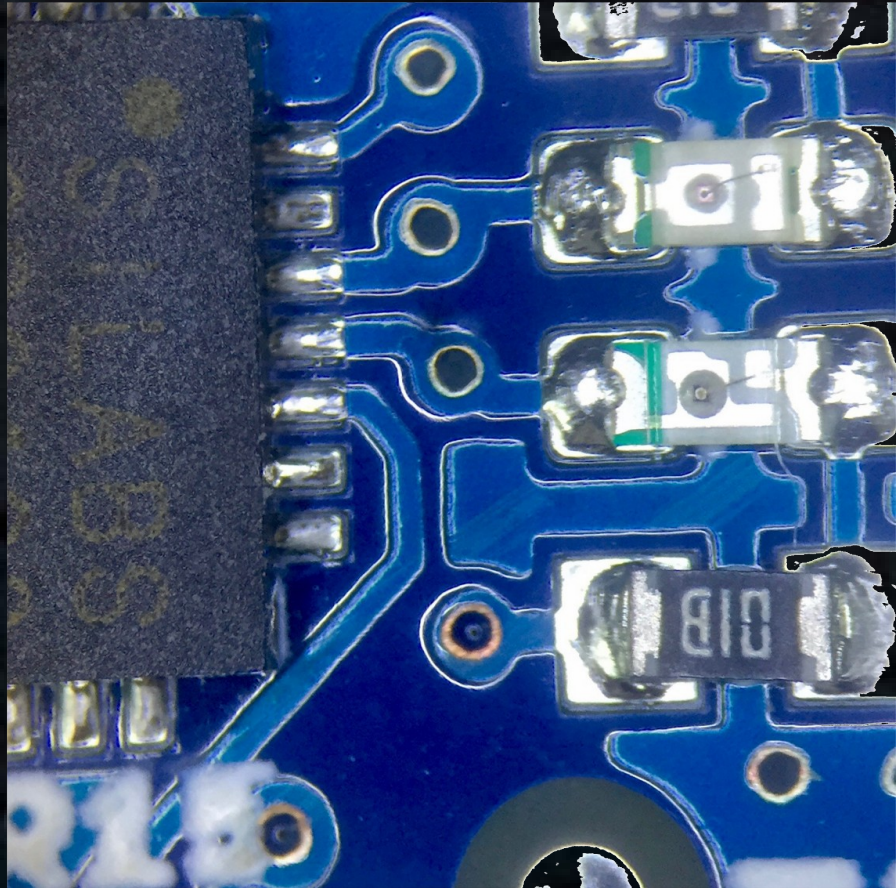
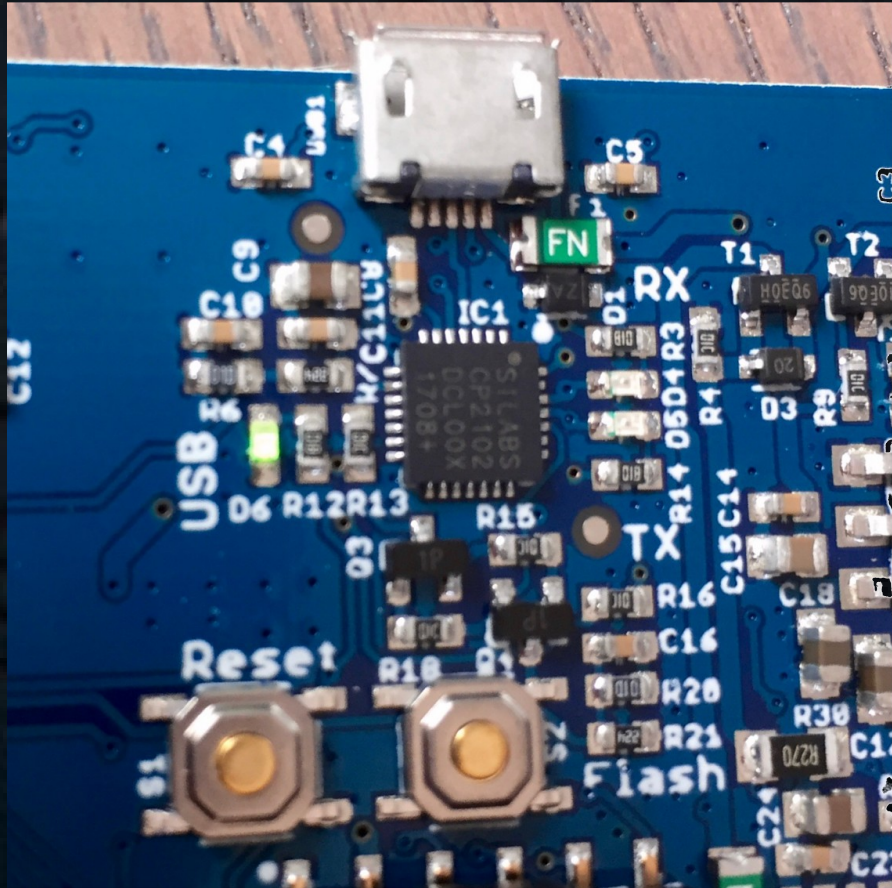
- The LEDs have angled corners. The yellow corner is also pointing towards the display.
- Do not touch the display.

BladeVPS
DuraSSD



SHA2017 Badge

- 4000 bags
- 4000 flyers
- 4000 lanyards
- 4000 batteries
- 8000 pieces of velcro
- 24000 RGBW LEDs
- 4000 vibrators







Software

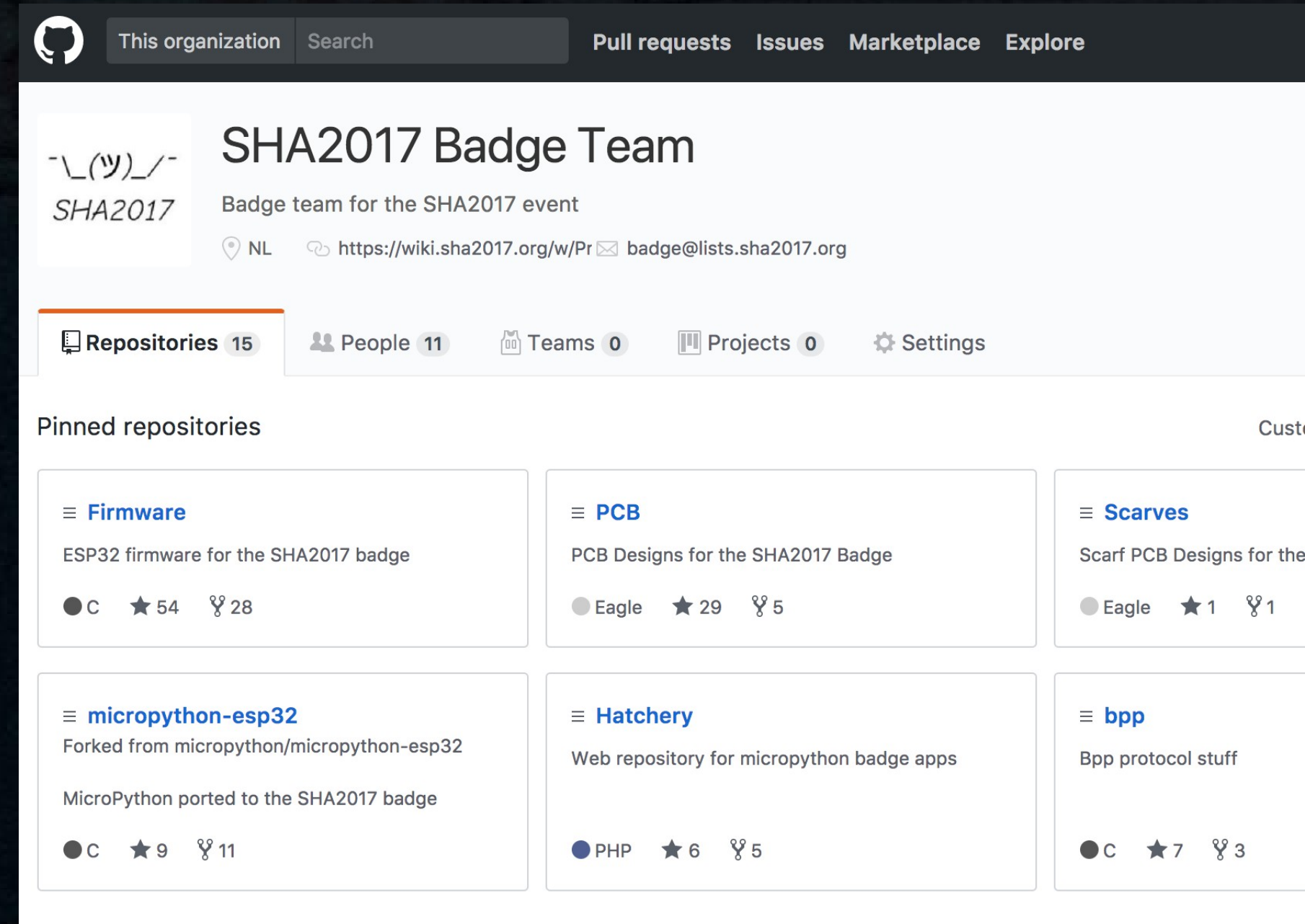
Overview

SHAA

+2017

SHA2017 Badge

- Low-level
 - Hardware drivers
 - OTA update
 - BPP
- MicroPython
- Applications



Development

- Separate repositories
- Workflow for cooperation
 - Use pull requests
 - Github issues
- Put everything in CI

build passing

Development

- ESP-IDF
 - Extremely nice for developing
- Kconfig
 - Build time options
 - Hardware revisions
 - Displays

SHA Badge e-ink default display type

Use the arrow keys to navigate this window or press the hotkey of the item you wish to select followed by the <SPACE BAR>. Press <?> for additional information about this

(X) DEPG0290B1

() GDEH029A1

<Select>

< Help >

SHA2017 Badge

E-ink “limitations”:

- 1 bit
- Slow display updates



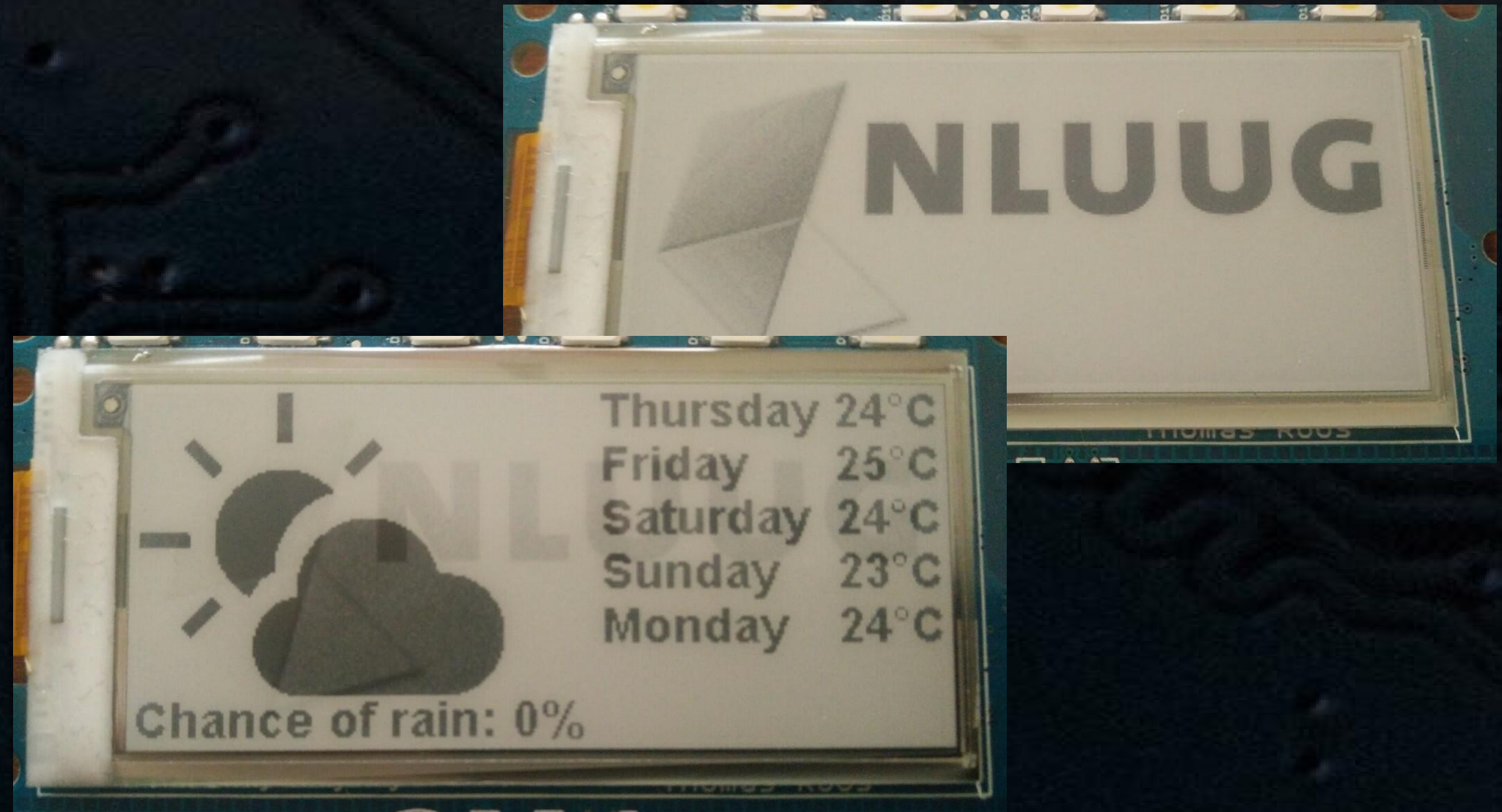
E-ink update magic

```
// GDEH029A1
unsigned char LUTDefault_part[30] = {
    0x10, 0x18, 0x18, 0x08, 0x18, 0x18, 0x08, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x13, 0x14, 0x44, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

unsigned char LUTDefault_full[30] = {
    0x02, 0x02, 0x01, 0x11, 0x12, 0x12, 0x22, 0x22, 0x66, 0x69,
    0x69, 0x59, 0x58, 0x99, 0x99, 0x88, 0x00, 0x00, 0x00, 0x00,
    0xF8, 0xB4, 0x13, 0x51, 0x35, 0x51, 0x51, 0x19, 0x01, 0x00
};
```

Faster e-ink

- “Burn-in”
- Ghosting



Greyscale

Use ghosting to our advantage.

- split image in planes
- start with black
- 1st plane: 16 cycles
- 2nd plane: 8 cycles
- and so on...



more E-ink magic

```
// DEPG0290B1
unsigned char LUTDefault_full[70] = {
    0xA0,0x90,0x50,0x00,0x00,0x00,0x00,
    0x50,0x90,0xA0,0x00,0x00,0x00,0x00,
    0xA0,0x90,0x50,0x00,0x00,0x00,0x00,
    0x50,0x90,0xA0,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x0F,0x0F,0x00,0x00,0x00,
    0x0F,0x0F,0x00,0x00,0x02,
    0x0F,0x0F,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00
};
```

SHA2017 Badge

- LEDs
stream of bytes ($g_0, r_0, b_0, w_0, g_1, r_1, b_1, w_1, \dots$)
- Vibrator motor
on/off

SHA2017 Badge

- Power management:
 - CPU Sleep Mode
 - SDcard, LEDs
- Power measurement:
 - Battery
 - USB

SHA2017 Badge

Capacitive Touch Buttons

Configurable per input channel:

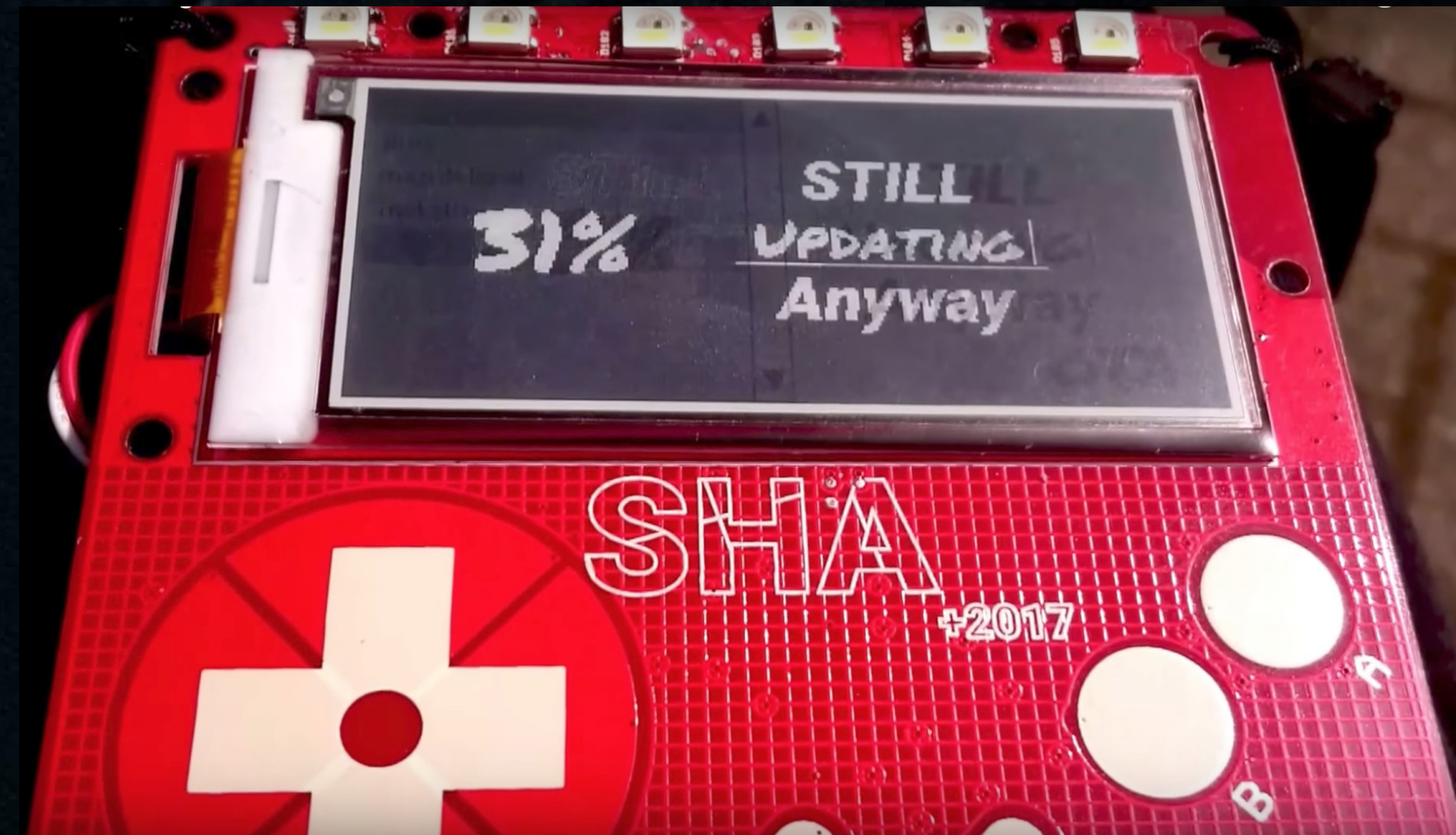
- 'touch' level
- 'release' level

Badge first boot



OTA updates

- 2 Firmware partitions
- HTTPS
with certificate pinning
- Update notification



MicroPython

Why?



- Based on Python
- Quicker development of “apps”
- Easy to use for target audience

No need for complicated toolchains to develop apps

- Ecosystem

MicroPython

micropython/esp32

- Implementation based on:
<https://github.com/micropython/micropython-esp32>
- Still very experimental
- Many libraries available
- Weird quirks
 - own vfs (clashed with esp-idf vfs)
 - date time (not unix timestamp)

MicroPython

Simple API for all peripherals:

- **LEDs:**

```
badge.leds_send_data(b'\x04\x01\x01\x01' * 6)
```

- **Vibrator motor:**

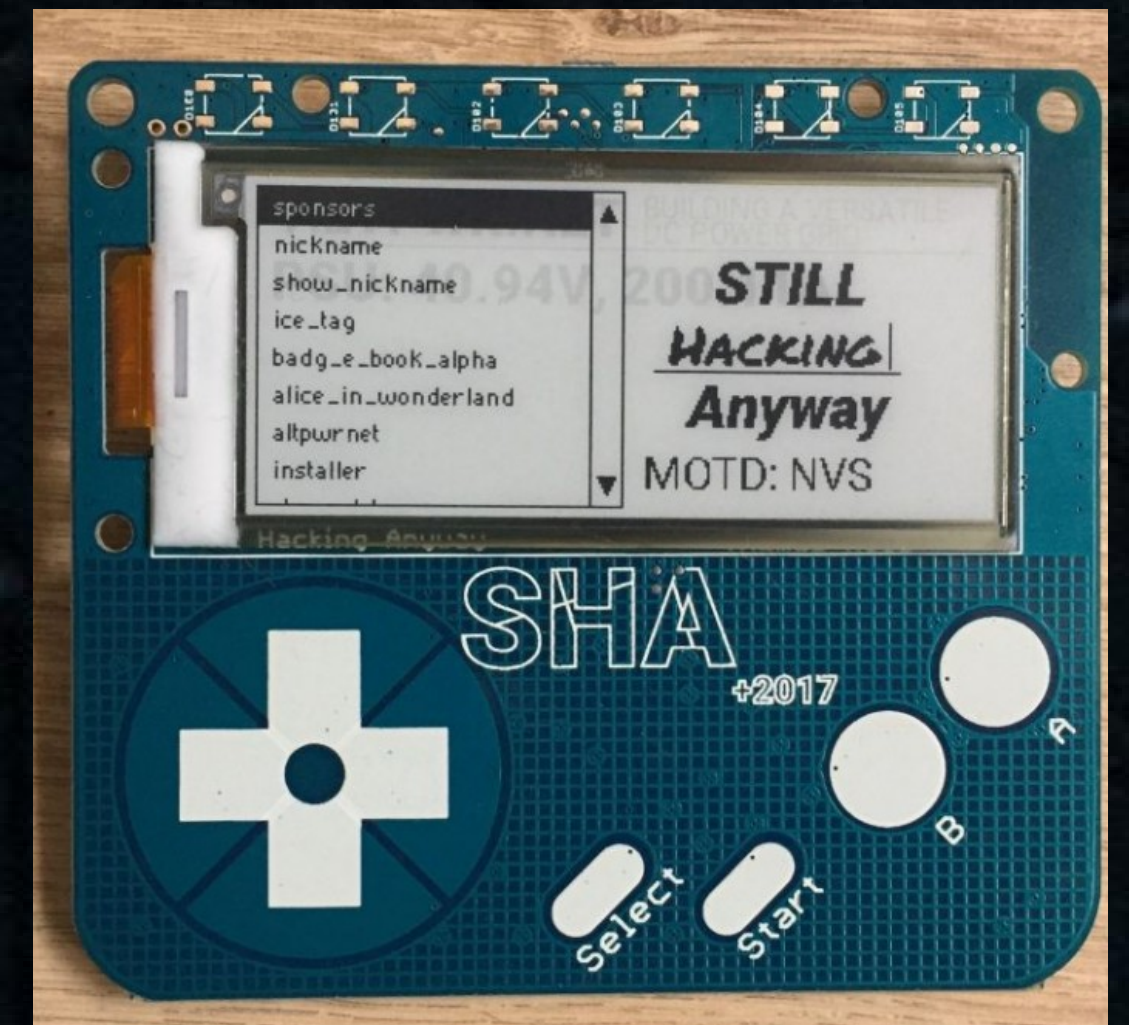
```
badge.vibrator_activate(0xc0a)
```

- **SDcard:**

```
badge.mount_sdcard()
```

MicroPython

- Boot state read from NVS
 - Run named application
 - OTA update
 - First boot



MicroPython

Multiple graphics libraries:

- uGFX
 - TiLDA uGFX widgets
- Special PNG loader
- Special JPG loader
- 3 gzip implementations?



MicroPython emulator

micropython/unix

- uGFX SDL
- Buttons, LEDs and buzzer

```
clang: warning: argument unused during compilation: '-pthread' [-Wunused-command-line-argument]
```

```
__TEXT __DATA __OBJC others dec hex  
684032 94208 0 4295009764 4295788004 1000c85e4
```

```
✓ ~/Projects/shabadge/Firmware/micropython/unix [esp32|+ 2...2▶ 3]
```

```
[16:52 $ ./micropython
```

```
MicroPython v1.8.6-1528-gc93a32af-dirty on 2017-06-16; darwin version
```

```
Use Ctrl-D to exit, Ctrl-E for paste mode
```

```
[>>> import woezel
```

```
[>>> woezel.install('game_of_life')
```

```
Installing to: /Users/annejan/.micropython/lib/
```

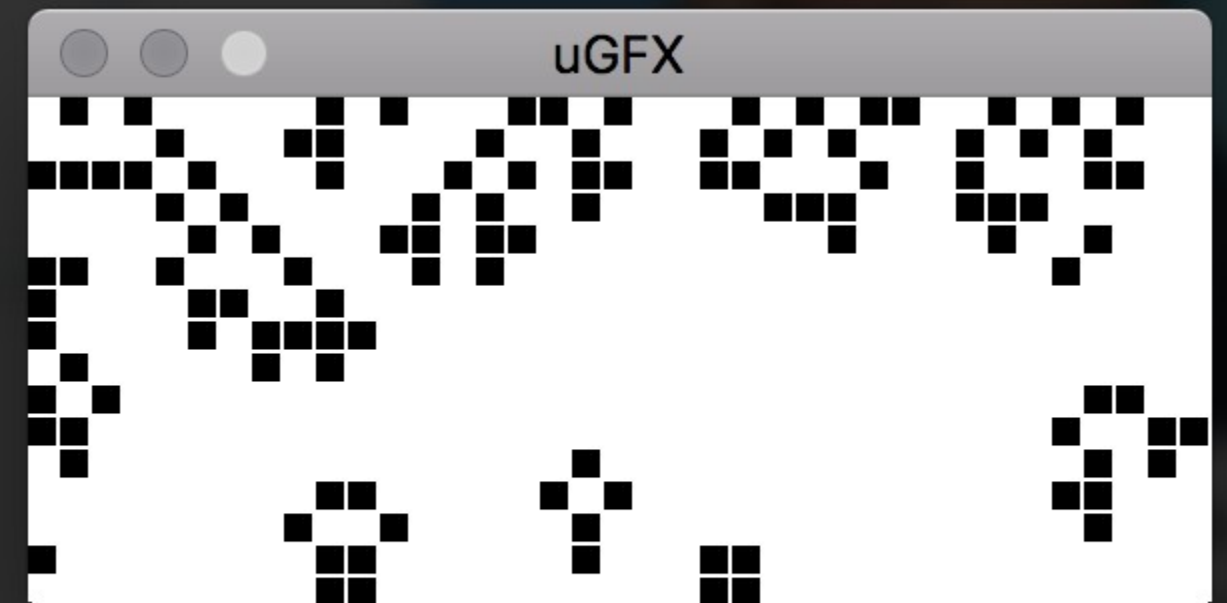
```
Warning: badge.sha2017.org SSL certificate is not validated
```

```
Not (yet) checking if game_of_life already installed
```

```
Not (yet) checking if version 3 already installed
```

```
Installing game_of_life 3 from https://badge.sha2017.org/eggs/game_of_life59419c9739563.tar.gz
```

```
[>>> import game_of_life
```



Woezel

- Fork of upip
- Version handling
- Separate folders
- Listing
- Searching
- TLS validation ussl
for *.sha2017.org

SHA2017 Hatchery

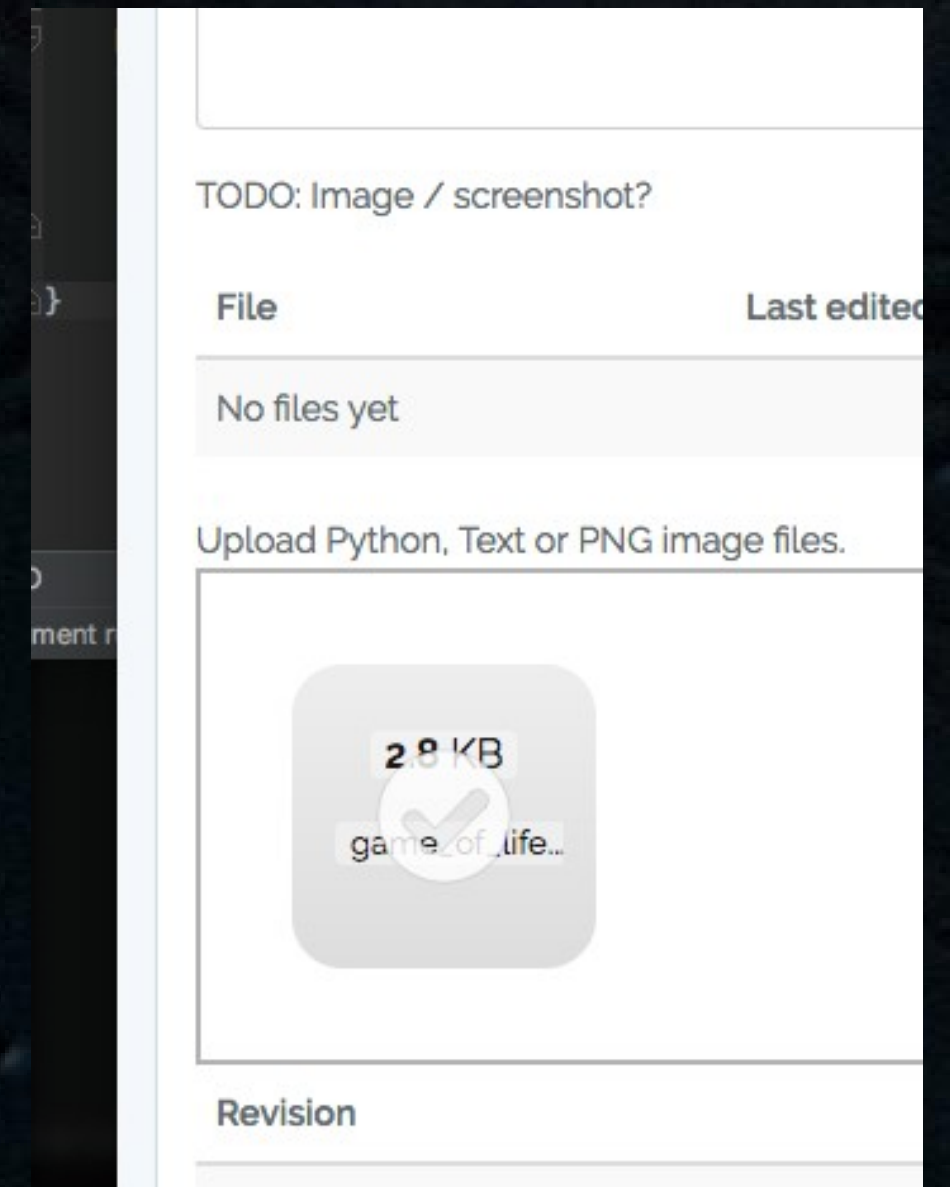
Contributors: 296 Eggs: 330

[WIKI CODING HELP](#)
[GITHUB](#)
[TWITTER](#)

Name	Revision	Size of zip	Size of content	Category	Last release
Rain	22	1983	4371	utility	11 hours ago
NLUUG Schedule	1	8553	32493	event_related	5 days ago
ov_departures	12	2097	7059	utility	1 week ago
Bootmanager	18	1406	3690	utility	1 week ago
Thermostat	129	5430	15675	uncategorised	1 day ago
TempSense	28	1441	3368	unusable	3 weeks ago
Badgeric	5	1292	2815	wearable	1 month ago
/dev/plant	4	548	590	data	1 month ago
ShowNickAndTags	7	1089	2773	event_related	1 month ago
SpaceStateV2	2	1137	2288	uncategorised	2 months ago
EasyMQTT	2	906	1500	data	2 months ago
MQTT Example	1	737	1056	utility	2 months ago

Hatchery

- eggs
- json
- revisions
- category
- status
- dependencies



Description

Weather app, thanks doebi

Category

Event related

Status

Working

Dependencies

Game of Life
wlan
Mandelbrot
spiral
TEXT...

Dependants

- t1

TODO: Image / screenshot?

Publish ☒ Save

File	Last edited	Size in bytes	add
weather.py	2017-07-12 07:52:03	3354	delete
__init__.py	2017-07-12 07:52:03	27	delete

Upload Python, Text or PNG image files.

Revision	Published
16	No
15	Yes
14	Yes
13	Yes
12	Yes

Hatchery

- CodeMirror editor
 - vim, emacs bindings
- pyflakes
 - code checker
- Hacked minigzip
 - minimize dictionary

Register

Name	
E-Mail Address	
Password	
Confirm Password	
Preferred editor	notepad.exe
Register	

Hatchery Eggs NLUUG Schedule fahrplan.py

fahrplan.py

Content

```
1  ### Fahrplan Module for Event schedule Egg
2  ### Author: f0x, edited by Renze
3  ### License: GPLv3
4
5  ### Screen is divided in three columns, each representing a track
6  ### each column represents a time period of 2 hours
7
8  import ugfx, badge, wifi
9  import json, time
10 import urequests as requests
11
12 display_width  = 295
13 display_height = 128
14
15 selected      = 0
```

Hatchery

- Easy to use, quick to make
- Laravel 5.4
- PHP 7.1+
- Unit tested
 - No guarantee

build

passing

coverage

89%

code climate

4.0

codacy

A

SHA2017

Zeewolde

3650 participants

4 days



SHA2017 Badge Bar

- 500+ Kg “stuff”
- Badge Bar tent
- Last rework at day 0
 - A badge for every visitor



- LEDs and vibrator soldering
- Re-flashing firmware



>256 apps

±40 tagged 'event related'

- Schedule
- SHA2017Game > 700 participants > 150 winners
- Village apps
- Logos
- LED eyesore



Countering malware

Options:

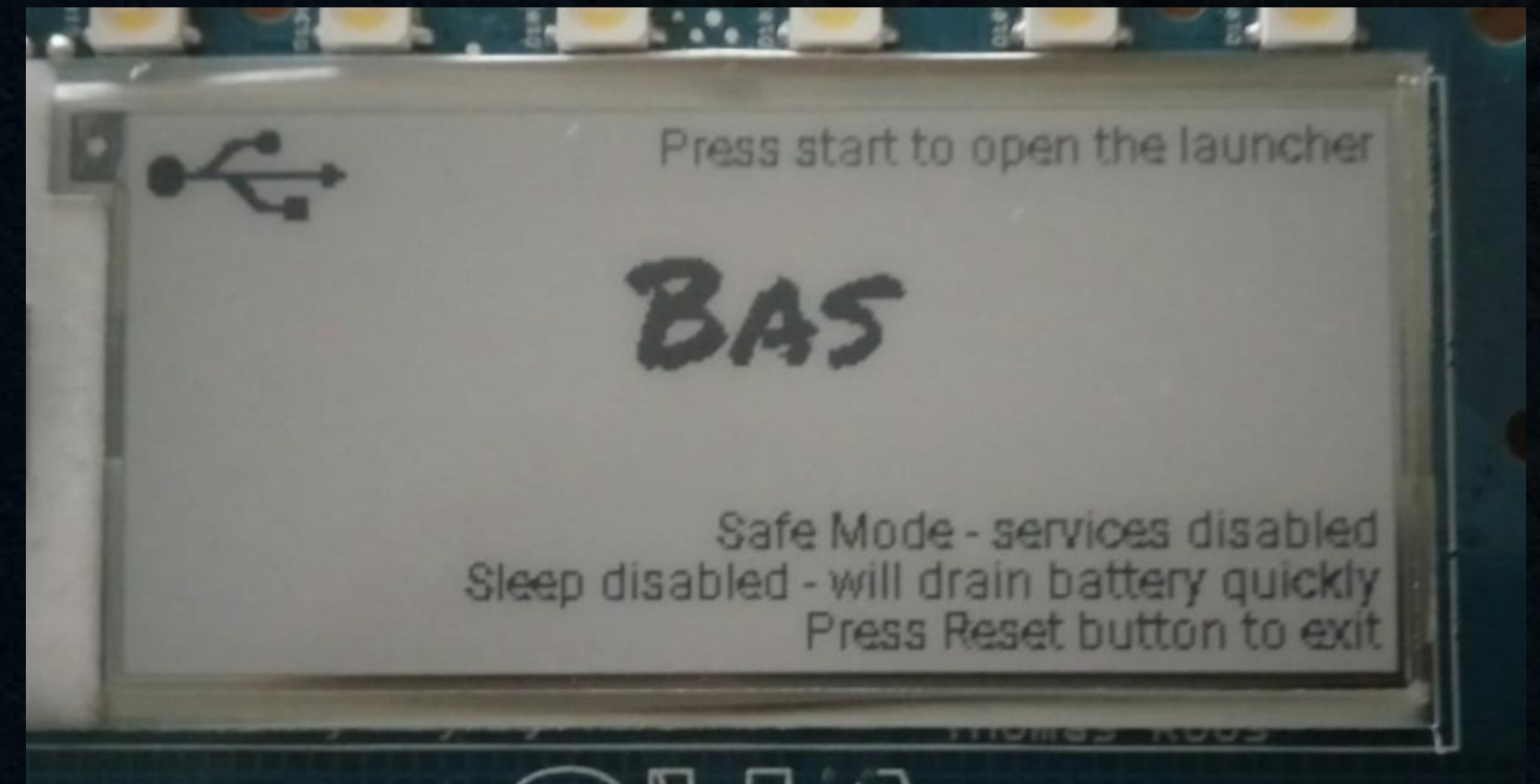
- Stricter APIs
- Blocking known bad apps
- or ...



Safe mode

Boot while pressing 'Start' key. This disables:

- Background services
- Loading of unknown modules
- Sleeping



SHA2017 Badge



SHA2017 Badge Team

@SHA2017Badge

Follow

To whoever pwnd our flashing station:
we salute and thank you! Please meet
the badge team! Come and say hi
(again)!

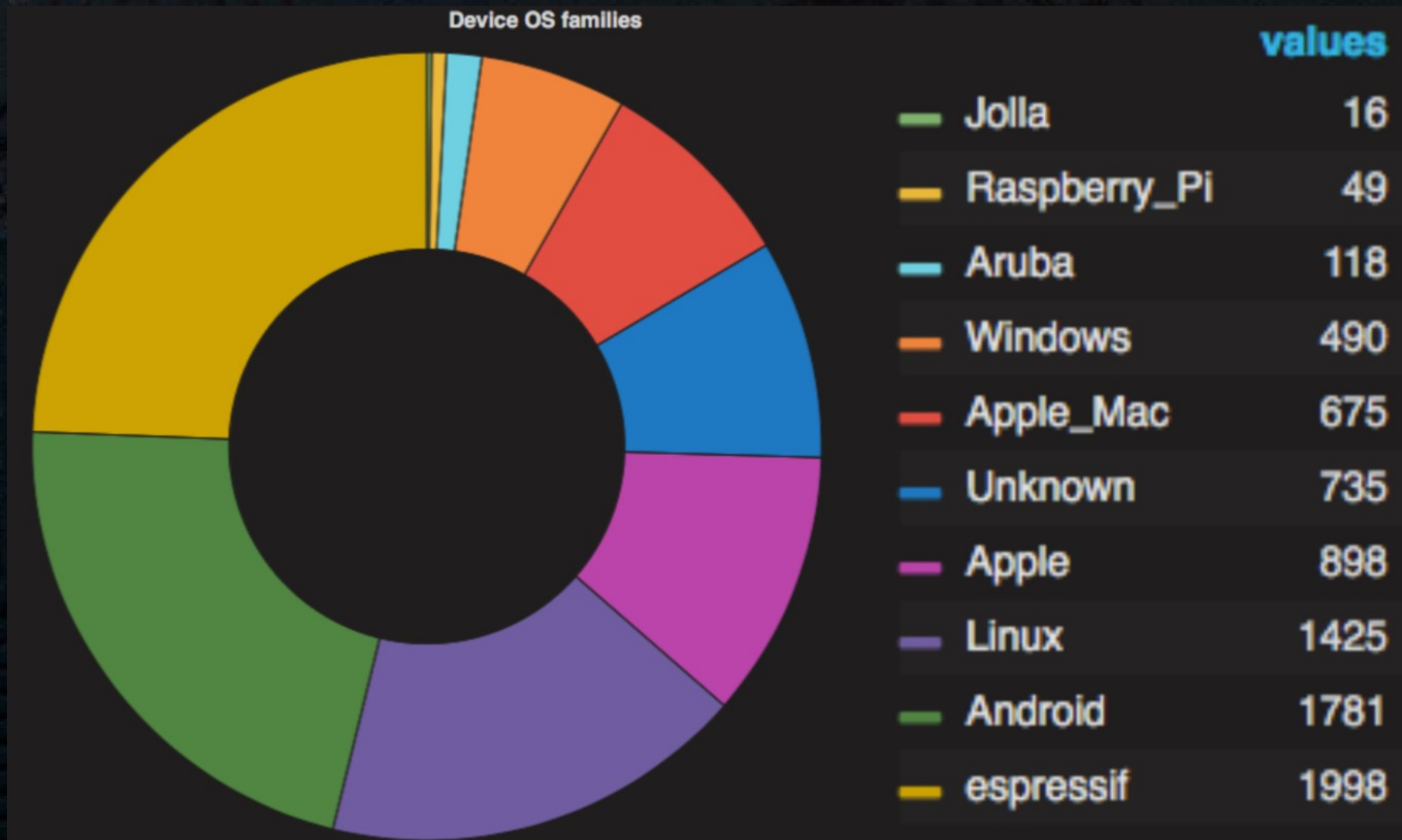


10:13 am - 6 Aug 2017

SHA2017 Badge

● Firmware		937 commits 17 contributors
● uGFX		61 commits 6 contributors
● MicroPython		1242 commits 22 contributors
● Autoflasher		90 commits 3 contributors
● Hatchery		200 commits 5 contributors
● Apps	330 eggs	15796 commits 296 contributors

SHA2017 Badge



Questions?

<https://badge.sha2017.org>

SHA

+2017

MicroPython

UGFX example:

```
ugfx.clear()

ugfx.fill_circle(64, 64, 48, ugfx.BLACK)

ugfx.rounded_box(128, 32, 152, 32, 8, ugfx.BLACK)

ugfx.string_box(128, 32, 152, 32, 'SHA Badge',
               'Roboto_Black22', ugfx.BLACK, ugfx.justifyCenter)

ugfx.flush(ugfx.LUT_FULL)
```

MicroPython

PNG example:

```
ugfx.clear()

# uGFX's PNG loader
ugfx.display_image(0, 0, '/media/hacking.png')

# alternative: own PNG loader
badge.eink_png(0, 0, '/media/hacking.png')

ugfx.flush(ugfx.GREYSCALE)
```