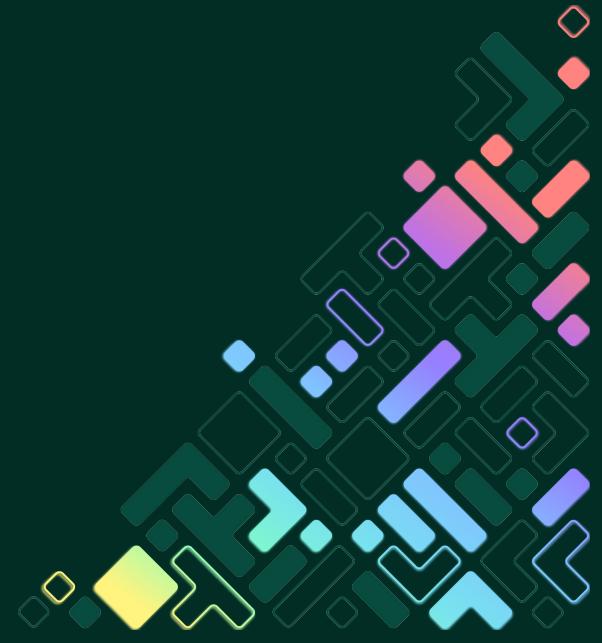




# Zephyr® Project

## Developer Summit





Zephyr® Project  
Developer Summit

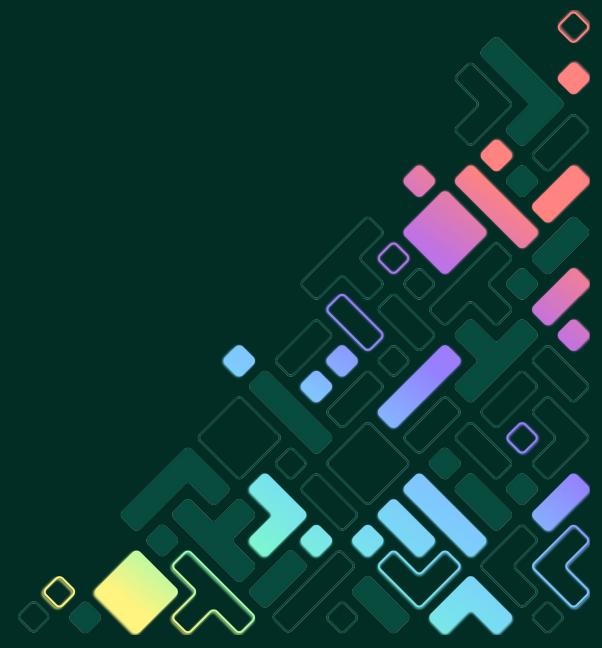
# Z LED Frame - "Illuminating Art with Zephyr"

Jakub Duchniewicz, Tietoevry

@JDuchniewicz



#EmbeddedOSSummit

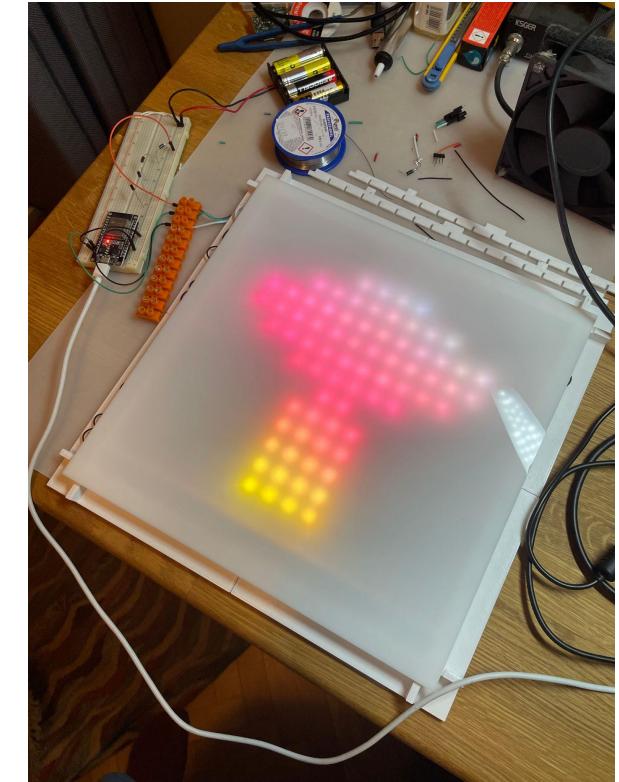




# Zephyr® Project

## Developer Summit

- Caveat: no BT, Li-Po batteries covered here (I also don't hang the Frame on the wall...)





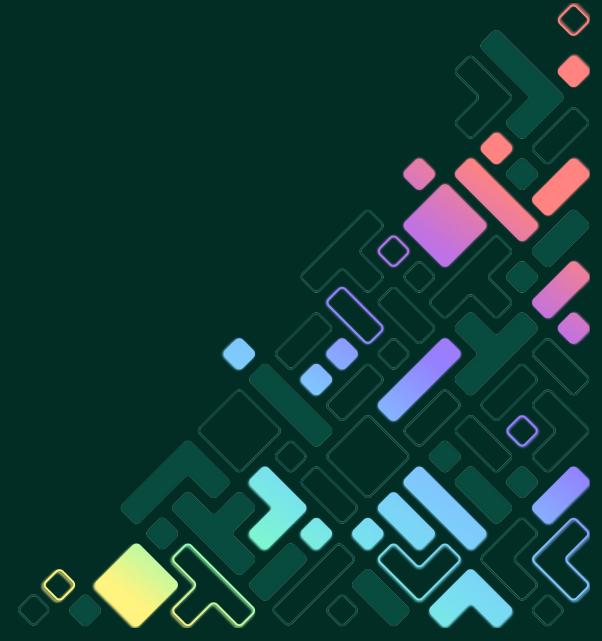
Zephyr® Project  
Developer Summit

# What is covered?

Z LED Frame - "Illuminating Art with Zephyr"



#EmbeddedOSSummit





## Accessible DIY project:

- why do it?
- steps
  - prototyping framework
  - LED handling with Zephyr
  - Basic HTTP networking with Zephyr
  - 3D designing and printing stories
- hurdles along the way
- solutions and tips
- future work



Zephyr® Project  
Developer Summit

> whoami



#EmbeddedOSSummit





# Zephyr® Project

## Developer Summit

> [jduchniewicz](#)

 **tietoevry**

- Developing 5G NR L1 processing chip at Tietoevry for Undisclosed Client :)  
ex Samsung, ex Intel, ex GSoC
- Tinkering around with Zephyr, Rust, Game Engines and Linux on embedded devices, after hours developing Sticky Piston Studios
- Blogging @ [www.jduchniewicz.com](http://www.jduchniewicz.com)
- Graduate of MSc in Embedded Systems at KTH Royal Institute of Technology and University of Turku
- In my free time doing a ton of sports: wind/kite/surf/skiing/climbing/unicycle  
also parenting a big German Shepherd dog :)



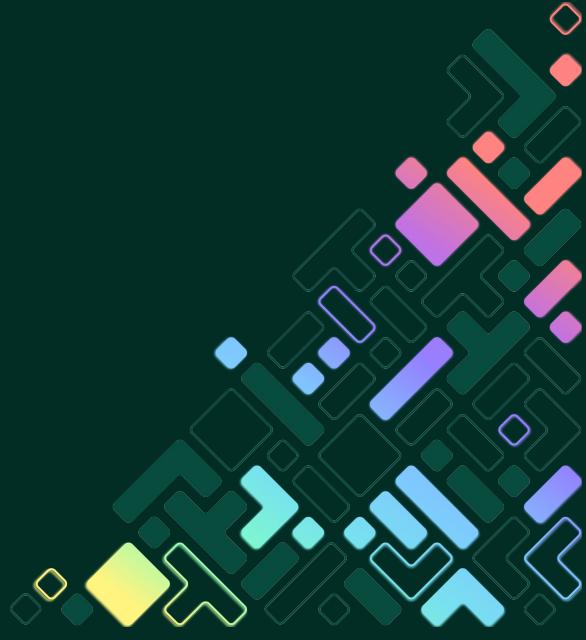


Zephyr® Project  
Developer Summit

# Why do it?



#EmbeddedOSSummit





Zephyr® Project

Developer Summit

- Not having enough art around me
- Also, not enough cool projects made by myself
- Craving something that gives more visible effects than my daily job :)
- Wanting to learn some new skills, FreeCAD, 3D printing, Zephyr and networking (since I do something similar at work)
- Being inspired by projects like ZS Watch or alike



Zephyr® Project  
Developer Summit

# How did I do it?



#EmbeddedOSSummit

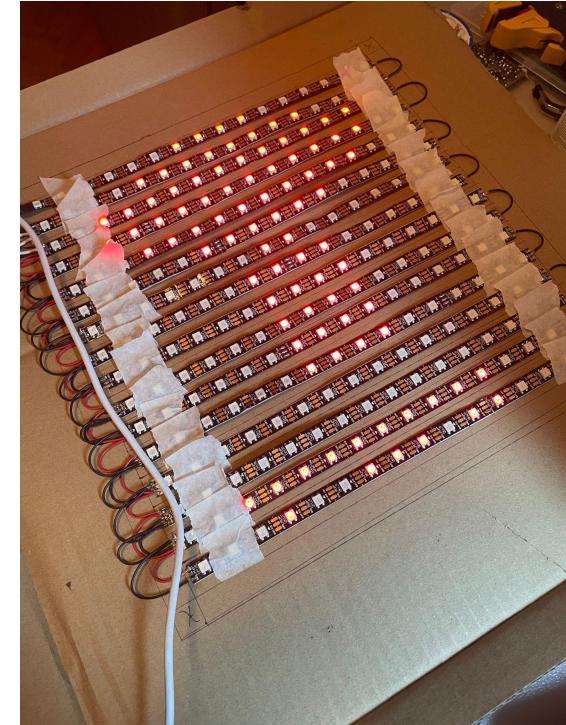




# Zephyr® Project

## Developer Summit

- Step 0 - Ideation:
  - Find some inspirations, similar projects  
(in my case DIY/shiny things attract everyone)
  - Choose the brain of the system - the MCU  
(or whatever else suits you)  
*think how much power, GPIOs etc you need*
- Step 1 - Validation (prototyping):
  - Check if the MCU of choice **is supported!**
  - Experiment with minimal examples - simply basing on Zephyr samples covers most of your ideas (*in rare cases you may find yourself contributing a new peripheral demo*)
  - Start with a prototype - cardboard will (*usually*) suffice





- **Step 2 - SW MVP (or a prototype):**
  - Once you know that the project is feasible, start with an MVP - SW is usually easier to get going and see something quick
  - The code/functionality does not need to be perfect and complete (yet)
- **Step 3 - HW (MVP and semi-complete product):**
  - Having SW to put inside cases we can print/craft cases/parts and test them alongside the working SW
- **Step 4+ - SW extensions + HW assembly:**
  - Having the case printed out, we can assemble the HW, solder remaining connections and "solidify" the prototype
  - Now we can focus on the "soft" part of the project :)

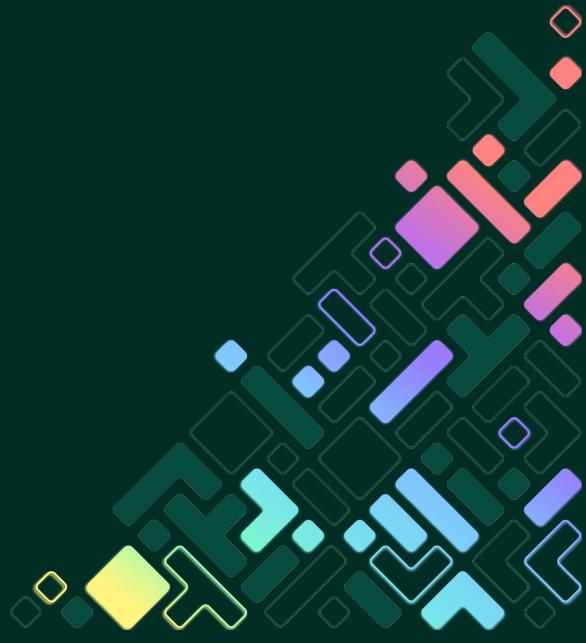


Zephyr® Project  
Developer Summit

# Step 0: Ideation



#EmbeddedOSSummit

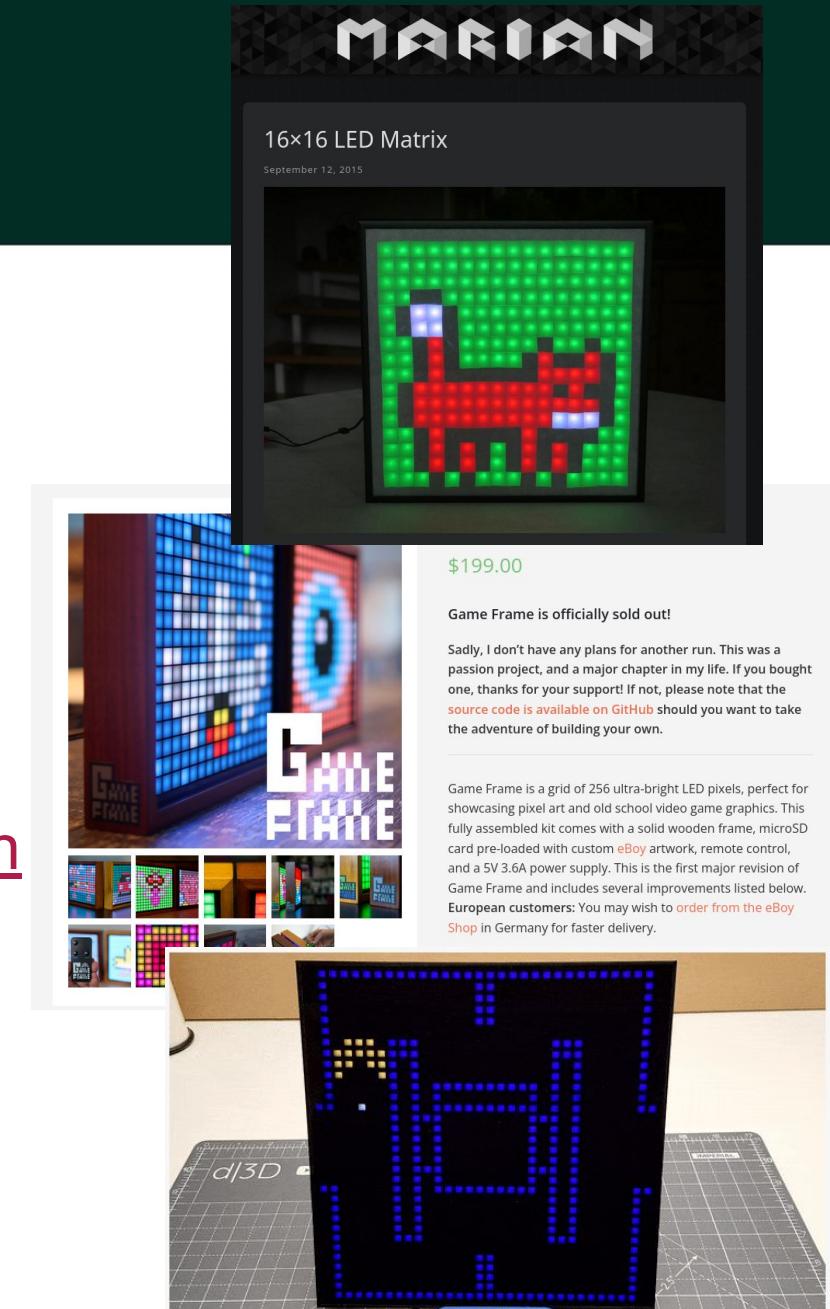




# Zephyr® Project

## Developer Summit

- Several similar projects already exist:
  - <https://marijan42.de/article/ledmatrix/>  
LED matrix using RPi, very solid DIY without any 3D printing!
  - <https://www.ledseq.com/product/game-frame/>  
Entire product, sold by 199 USD run on Teensy MCU with Open Source firmware
  - <https://www.daftmike.com/2018/03/led-matrix-animation-frame.html>  
32x32! frame with 3D printed parts on RPi Zero
- But... *none of those run Zephyr*, do they?





Zephyr® Project

Developer Summit

- Initially started with ESP32C6 which turned out to have no Zephyr support (yet!) - *I should have read article on ESP32C6 reality check*
- There have been major changes in Espressif's HAL (Hardware Model Version 2) - *still did not update the project to use it...*
- If you still want to use it or contribute:
  - GH discussion - [discussions/62138](#)
  - issue tracking ESP32 support - [issues/29394](#)

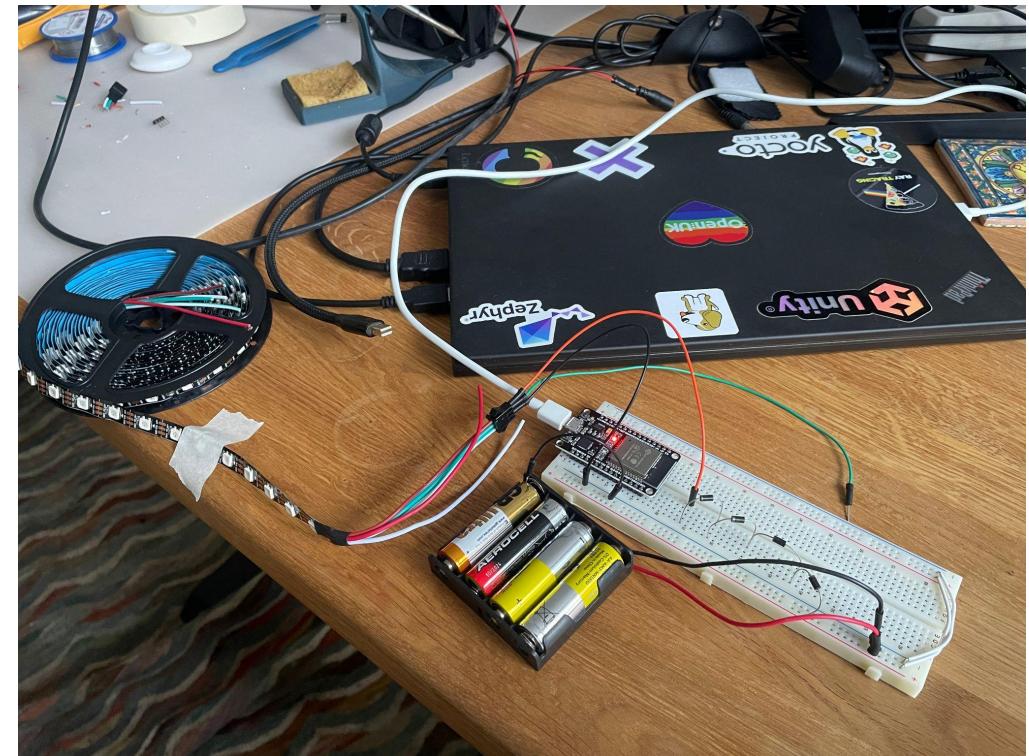




# Zephyr® Project

## Developer Summit

- After troubles with the RISC-V board (*sad trumpet sound*) - chose the good ol' ESP32
- Amazingly, LED connectivity and programming worked instantaneously
- Spot the battery pack and some diodes - know where this will lead?





Zephyr® Project  
Developer Summit

# Step 1: Validation



#EmbeddedOSSummit

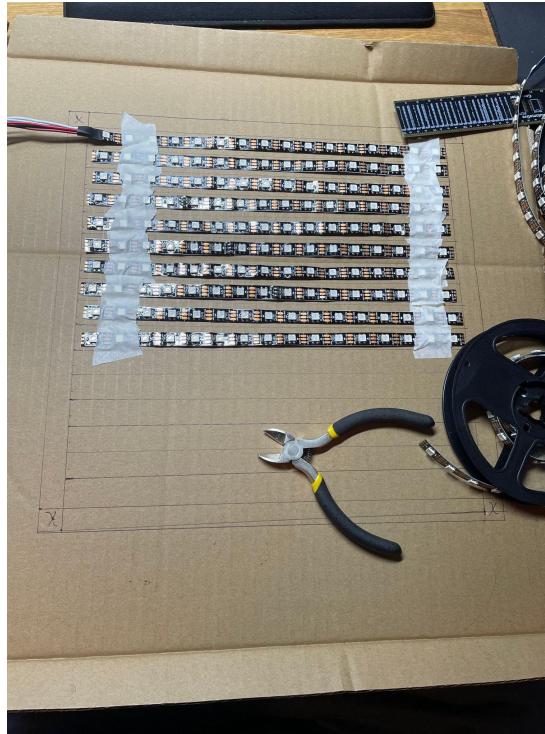
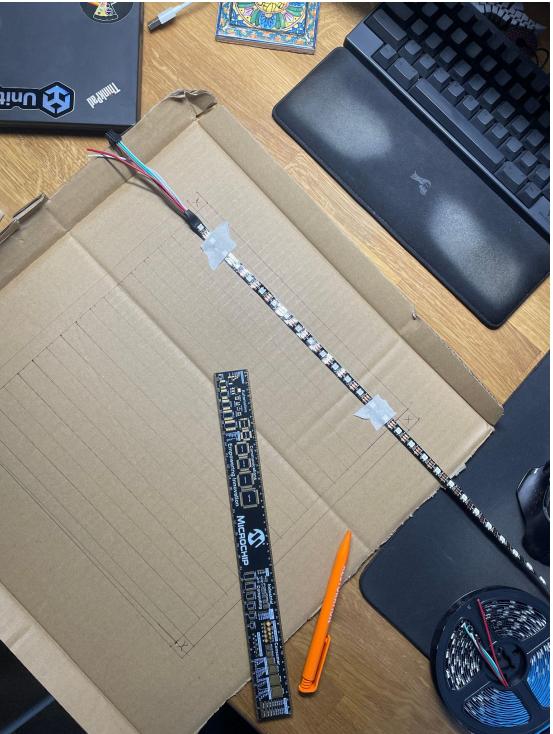




# Zephyr® Project

## Developer Summit

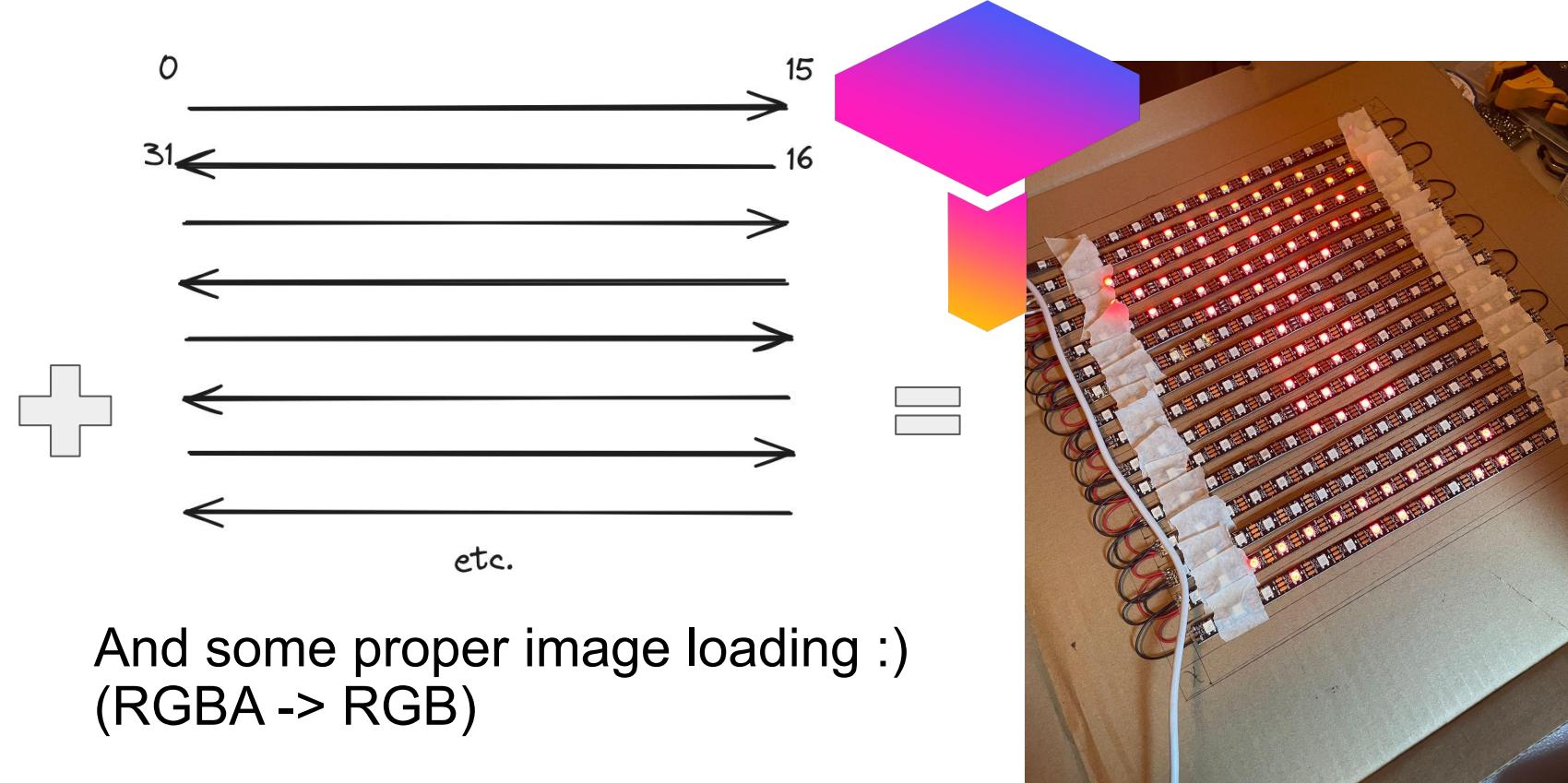
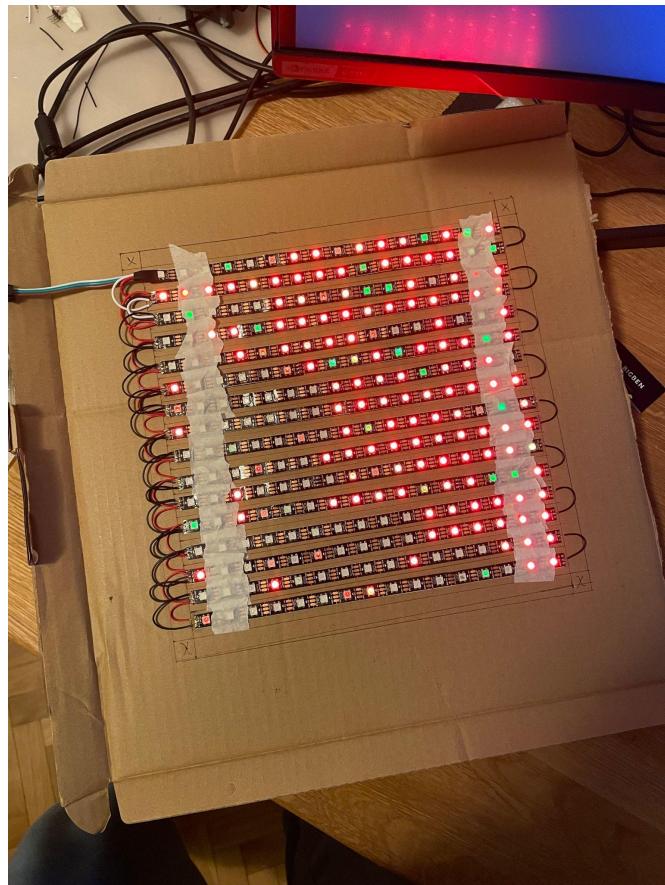
- From a single strip      to multiple strips      to a grid





# Zephyr® Project

## Developer Summit



And some proper image loading :)  
(RGBA → RGB)

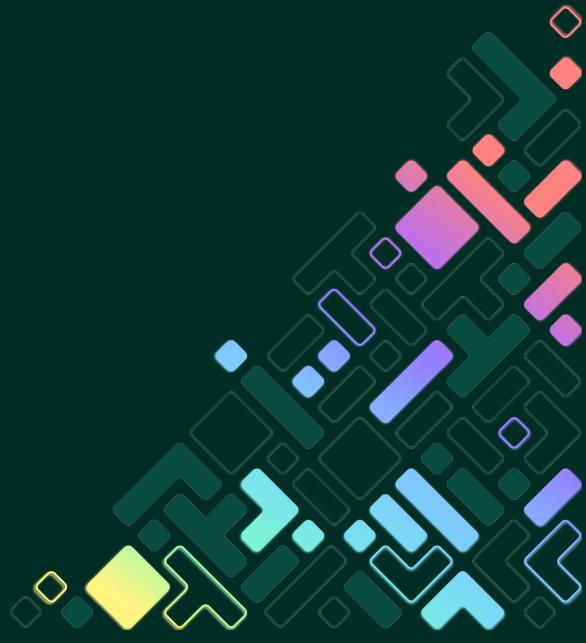


Zephyr® Project  
Developer Summit

# Step 2: Software Minimum Viable Product



#EmbeddedOSSummit





# Zephyr® Project

## Developer Summit

- LED handling is *super easy* - just use the driver! (It just works.)

```
#define STRIP_NODE DT_ALIAS(led_strip)

static const struct device *const strip = DEVICE_DT_GET(STRIP_NODE);
```

```
if (device_is_ready(strip))
{
    LOG_INF("Found LED strip device %s", strip->name);
}
```

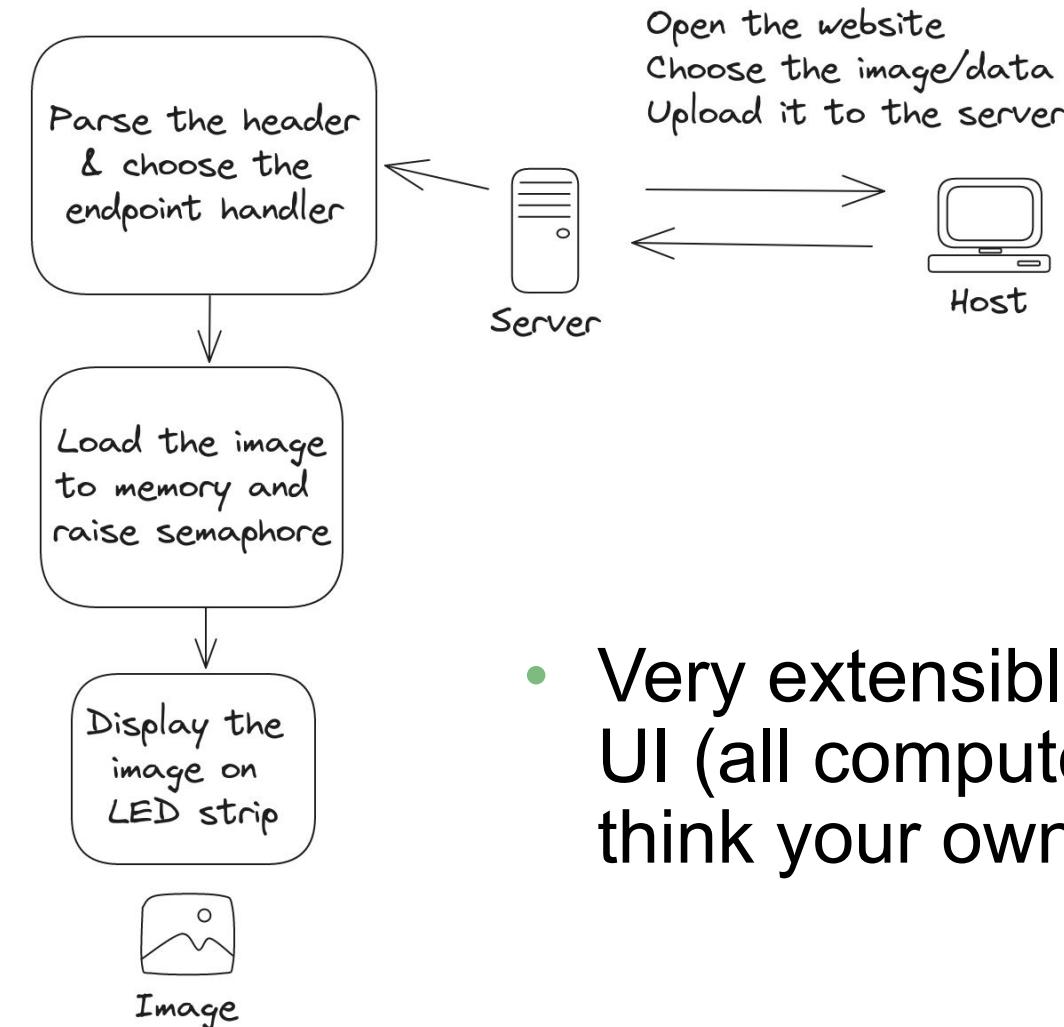
- And load the data!

```
// Update the LED strip with the new image
rc = led_strip_update_rgb(strip, pixels, STRIP_NUM_PIXELS);
```



# Zephyr® Project

## Developer Summit



- Simple server over HTTP
- 2 - way HTTP API communication
- We can offload processing to client (ESP32 is not a H100 NVIDIA GPU :)
- Very extensible - can add more APIs, go crazy on UI (all computed client-side)  
think your own shiny smart-home



# Zephyr® Project

## Developer Summit

- We have great demos - [dumb\\_http\\_server](#) and ([mt counterpart](#))
- However they are fairly basic, need to get creative about handling custom endpoints, data processing etc.

```
static endpoint_t valid_endpoints[NUMBER_OF_ENDPOINTS] =  
{  
    {"/", GET},  
    {"api/image", POST},  
    {"api/mode/preset", POST},  
    {"api/mode/calibration", POST},  
};
```



- Have seen
- We have counters
- However handling

```
static endpoint_t endpoints[] = {  
    {"/", GET},  
    {"api/image", GET},  
    {"api/mode/prev", GET},  
    {"api/mode/call", GET},  
};  
  
static int parse_header(char *buf, int buf_size, char *endpoint, int endpoint_size, method_t *method)  
{  
    char *ptr = buf, *delim_pos = NULL;  
    int len = 0;  
    if (strstr(ptr, "GET"))  
    {  
        *method = GET;  
        ptr += 4; // +1 for whitespace  
    }  
    else if (strstr(ptr, "POST"))  
    {  
        *method = POST;  
        ptr += 5;  
    }  
    else  
    {  
        LOG_ERR("Unknown method found!");  
        return -1;  
    }  
  
    delim_pos = strchr(ptr, ' ');  
    len = delim_pos - ptr;  
    if (len + 1 > endpoint_size)  
    {  
        LOG_ERR("Too long endpoint name! Max allowed: %d", endpoint_size);  
        return -1;  
    }  
  
    strncpy(endpoint, ptr, len);  
    *(endpoint + len) = '\0'; // null-terminate it
```

```
static int handle_endpoint(int client, char *endpoint_buf, method_t method, char *buf, int buf_len)
{
    // GET /
    if (strncmp(valid_endpoints[0].name, endpoint_buf, strlen(endpoint_buf)) == 0)
    {
        // here we don't need to check any return values and can directly return the page to render
        LOG_ERR("Sending the data to the client!");
        (void)sendall(client, content, sizeof(content));
        return 0;
    }
    else if (strncmp(valid_endpoints[1].name, endpoint_buf, strlen(endpoint_buf)) == 0)
    {
        LOG_ERR("Handling POST /api/image");
        static bool read_header = false;
        uint8_t length = 0u;

        k_mutex_lock(&mode_mutex, K_FOREVER);
        mode = NETWORK;
        k_mutex_unlock(&mode_mutex);

        if (read_header)
        {
            // for now this will inform that we handled all
            if (recv_img_offset + buf_len >= MAX_IMAGE_SIZE)
            {
                LOG_ERR("Stop accepting - Trying to send more data than Image can accept.");
                length = MAX_IMAGE_SIZE - recv_img_offset;
                if (length > 0)
                {
                    memcpy(received_image + recv_img_offset, buf, length);
                }
            }
        }
    }
}
```

```
    rcv_img_offset = 0;
    read_header = false;

    // Give the semaphore to notify that a new image is ready
    k_sem_give(&image_semaphore);
    LOG_ERR("Gave the semaphore");
    // Wait for the image to be processed
    k_sem_take(&image_semaphore, K_FOREVER);
    LOG_ERR("Took the semaphore");
    return 1;
}

memcpy(received_image + rcv_img_offset, buf, buf_len);

rcv_img_offset += buf_len;
return -1;
}

// until get the magic number, keep reading the header 0xBADAD00B
// there is a slight chance that the magic number will be split between two packets - but for now we don't handle it
char *pos = strstr(buf, magic_number);
if (pos)
{
    LOG_ERR("Found the magic number!");
    read_header = true;
    // read the remainder as the image data
    rcv_img_offset = 0;
    length = buf_len - (pos - buf) - strlen(magic_number);
    memcpy(received_image, pos + strlen(magic_number), length);
    rcv_img_offset += length;
}
LOG_ERR("Still parsing the header");
return -1;
```



Zephyr® Project

Developer Summit

- Some ESP32/network specific tips:
  - Be sure to add `CONFIG_ESP_HEAP_MEM_POOL_REGION_1_SIZE=1024` to `prj.conf`
  - Also, in the `dumb_http_sample_mt` be sure to add the IPv6 support: `CONFIG_NET_CONFIG_NEED_IPV6=y` otherwise expect some weird crashes - yet a mystery to be solved
  - Make sure that IPv4 address that you assign in the `prj.conf` is free, otherwise the server won't start!



# Zephyr® Project

## Developer Summit

- That's the basic HTML webpage that is served when you successfully run the **dumb\_http\_demo/mt**

Warning: this is a content sample. Proceed to [Zephyr About page](#) for up to date information.

### What is Zephyr™ Project?

The Zephyr™ Project, is a Linux Foundation hosted Collaboration Project, an open source collaborative effort uniting leaders from across the industry to build a best-in-breed small, scalable, real-time operating system (RTOS) optimized for resource constrained devices, across multiple architectures. The Zephyr Project's goal is to establish a neutral project where silicon vendors, OEMs, ODMs, ISVs, and OSVs can contribute technology to reduce the cost and accelerate time to market for developing the billions of devices that will make up the majority of the Internet of Things

The Zephyr Project is perfect for building simple connected sensors, LED wearables, up to modems and small IoT wireless gateways. Because the Zephyr OS is modular and supports multiple architectures, developers are able to easily tailor an optimal solution to meet their needs. As a true open source project, the community can evolve the project to support new hardware, developer tools, sensor and device drivers. Enhancements in security, device management capabilities, connectivity stacks and file systems can be easily implemented.

The Zephyr kernel is derived from Wind River's commercial VxWorks Microkernel Profile for VxWorks. Microkernel Profile has evolved over 20 years from DSP RTOS technology known as Virtuoso. The RTOS has been used in several commercial applications including satellites, military command and control communications, radar, telecommunications and image processing. The most recent example of the technology's success is the successful Philae Landing on Comet Churyumov-Gerasimenko and the accompanying Rosetta Orbiter.

- You can run JavaScript on the client side to do the heavier processing - in our case image pre-processing.



# Zephyr® Project

## Developer Summit

```
// Create an image element
var image = new Image();
image.onload = function() {
    // Create a canvas element
    var canvas = document.createElement('canvas')
    canvas.width = 16;
    canvas.height = 16;
    var ctx = canvas.getContext('2d');

    // Draw the image onto the canvas (resizing it to
    ctx.drawImage(image, 0, 0, 16, 16);

    // Get the pixel data from the canvas
    var imageData = ctx.getImageData(0, 0, 16, 16);
    var pixelData = imageData.data;
    console.log(pixelData);

    // Create a Uint8Array to store the raw binary dat
    var binaryData = new Uint8Array(16 * 16 * 3);
    var newIdx = 0;
    for (var i = 0; i < pixelData.length; i += 4, newIdx += 3) {
        binaryData[newIdx] = pixelData[i];          // Red
        binaryData[newIdx + 1] = pixelData[i + 1];   // Green
        binaryData[newIdx + 2] = pixelData[i + 2];   // Blue
    }
}
```

```
<div class="container">
    <input type="file" id="fileInput">
    <button onclick="processImage()">Convert</button>
    <button onclick="uploadProcessedData()">Upload</button>

    <div class="preview-container">
        <div class="preview" id="originalPreview">
            <h2>Original Image</h2>
            
        </div>
        <div class="preview" id="convertedPreview">
            <h2>Converted Image</h2>
            
        </div>
        <button onclick="setMode('preset')">PresetPattern</button>
        <button onclick="setMode('calibration')">CalibrationPattern</button>
    </div>
```



# Zephyr® Project

## Developer Summit

```
// Create an image element
```

```
var image = new Image();
```

```
image.onload = function() {
```

```
// Create a canvas element
```

```
var canvas = document.createElement('canvas');
```

```
canvas.width = 16;
```

```
canvas.height = 16;
```

```
var ctx = canvas.get
```

```
// Draw the image on the canvas
```

```
ctx.drawImage(image, 0,
```

```
// Get the pixel data
```

```
var imageData = ct
```

```
var pixelData = im
```

```
console.log(pixelDa
```

```
// Create a Uint8Arr
```

```
var binaryData = ne
```

```
var newIdx = 0;
```

```
for (var i = 0; i < pixelData.length; i += 4, newIdx += 3) {
```

```
    binaryData[newIdx] = pixelData[i]; // Red
```

```
    binaryData[newIdx + 1] = pixelData[i + 1]; // Green
```

```
    binaryData[newIdx + 2] = pixelData[i + 2]; // Blue
```

```
}
```

Browse... rust.png Convert Upload



PresetPattern

CalibrationPattern



# Zephyr® Project

## Developer Summit

Scratching my head with the very dim diodes and some of them even not lighting, green barely lit, blue not even lighting up

Light Emitting Diode Colours

**Hypothesis** - some color-calibration or timing issues

**The actual problem?**

Battery pack discharged itself during many hours of testing the strip -.-

Typical LED Characteristics			
Semiconductor Material	Wavelength	Colour	$V_F @ 20mA$
GaAs	850-940nm	Infra-Red	1.2V
GaAsP	630-660nm	Red	1.8V
GaAsP	605-620nm	Amber	2.0V
GaAsP:N	585-595nm	Yellow	2.2V
AlGaP	550-570nm	Green	3.5V
SiC	430-505nm	Blue	3.6V
GaInN	450nm	White	4.0V



Zephyr® Project  
Developer Summit

# Step 3: Hardware MVP



#EmbeddedOSSummit

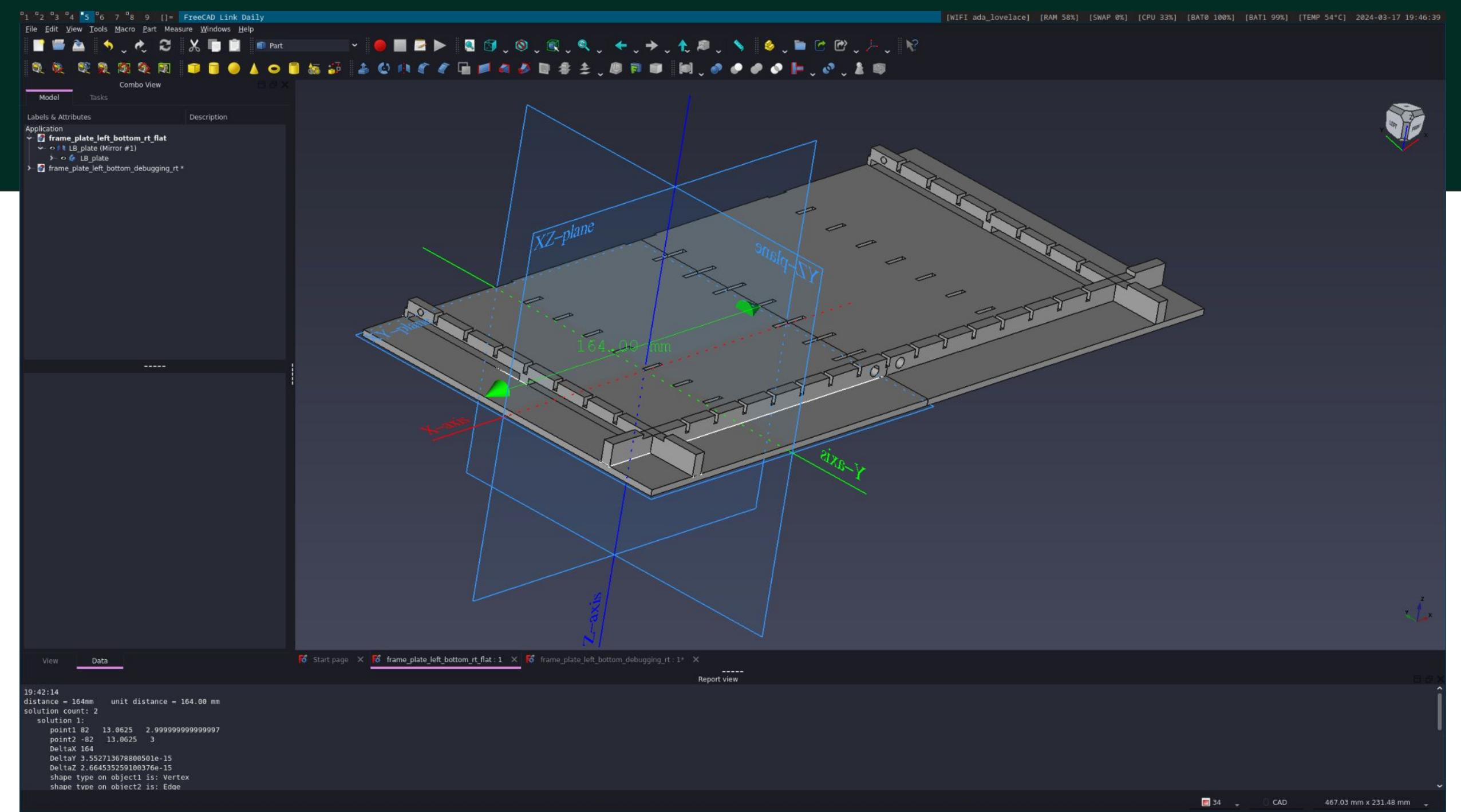




Zephyr® Project

Developer Summit

- Where to start? **3D Design!**
- Somebody once told me (*♪ the world is gonna roll me ♪*) - don't learn FreeCAD - it's difficult and not intuitive(!)
- Guess what I did :)?





Zephyr® Project

Developer Summit

- Reality - vanilla FreeCAD turned out to be lacking core features - changes in the topology (e.g. renaming :)) and meaningful errors



- Thankfully, there is a [fork from realthunder](#) that works around TNP problem, allows for disjoint bodies and good error debugging

[https://www.reddit.com/r/FreeCAD/comments/11dkx75/where\\_to\\_find\\_recompute\\_failure\\_details/](https://www.reddit.com/r/FreeCAD/comments/11dkx75/where_to_find_recompute_failure_details/)



Zephyr® Project

Developer Summit

- DIY spirit in mind - composable and IKEA style HW - *print them yourself, assemble them yourself*
- Parts list:
  - Backplate Left + Right (similar)
  - Vertical criss-cross strips
  - Horizontal criss-cross strips
  - tiny plate connectors
  - Frame edges + PSU and ESP32 mount
  - PSU, power switch, cable
  - ESP32 + cables
  - Plexiglass 2mm 80% opaqueness 30x30cm
  - LED strip
  - Filament - YMMV - half of the 1kg spool? *I used whole for prototyping*



Zephyr® Project  
Developer Summit

# 3D Printing



#EmbeddedOSSummit

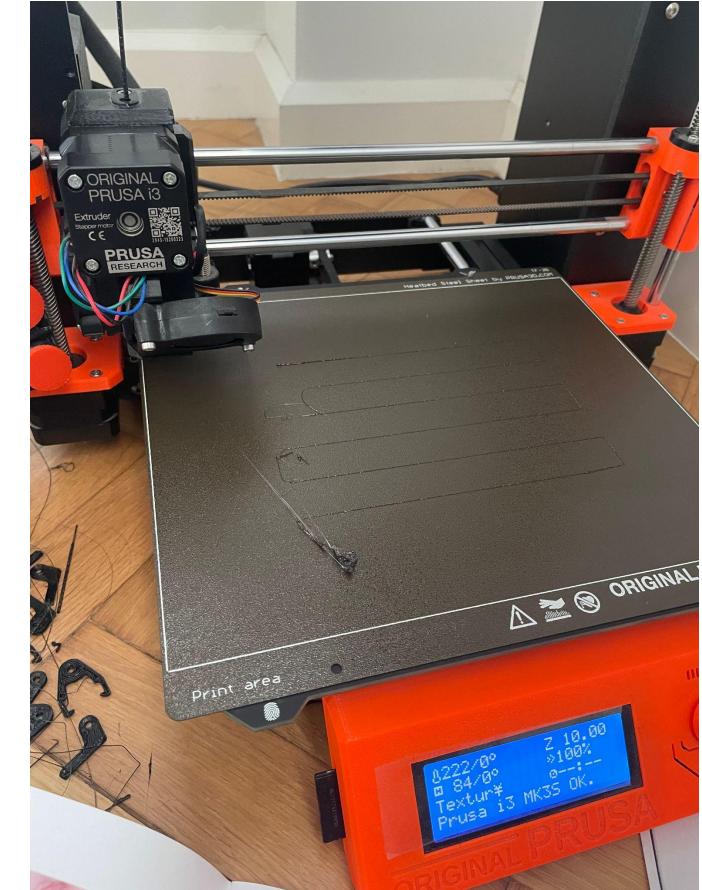




# Zephyr® Project

## Developer Summit

- 3D printing is *usually* not hard, PETG or PLA are *usually* fun
- Unless...
- You start getting spurious Y-axis failures, you check the belt tightness, clean the bearings and find that Z-offset is **utterly rubbish**
- You clog the nozzle and have to perform the **cold pull**





# Zephyr® Project

## Developer Summit

- 3D printing is *usually* not hard, PETG or PLA are *usually* fun
- Unless...
- You start getting spurious Y-axis failures, you check the belt tightness, clean the bearings and find that Z-offset is **utterly rubbish**
- You clog the nozzle and have to perform the **cold pull**





# Zephyr® Project

## Developer Summit

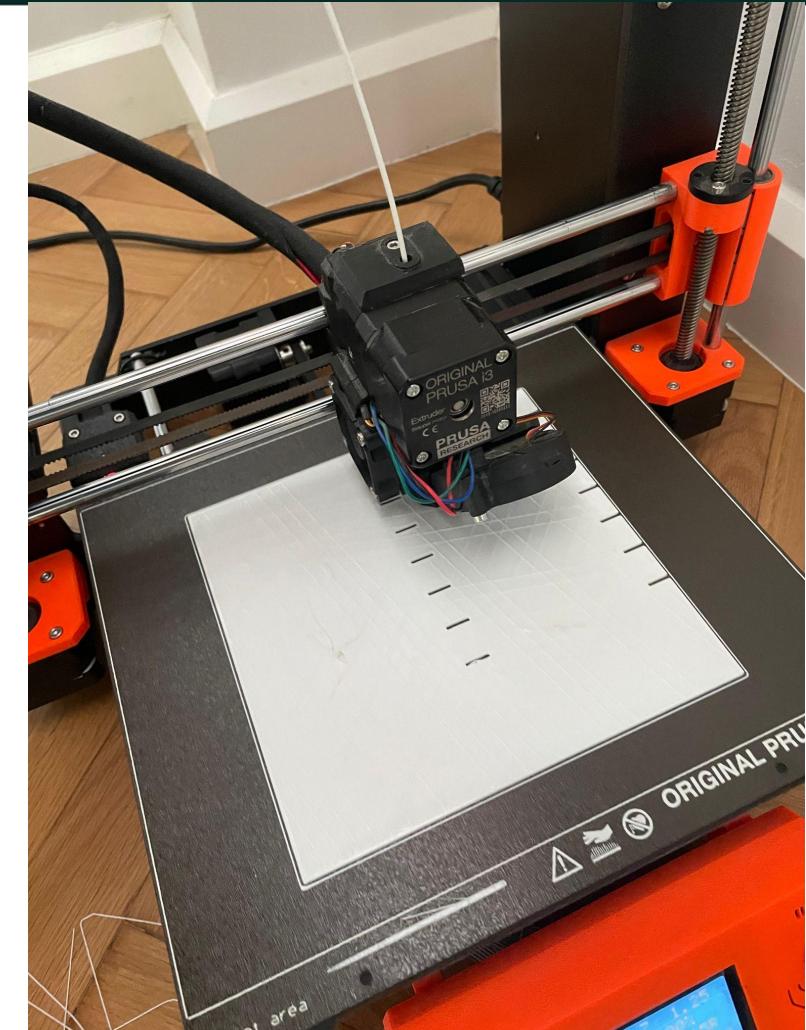
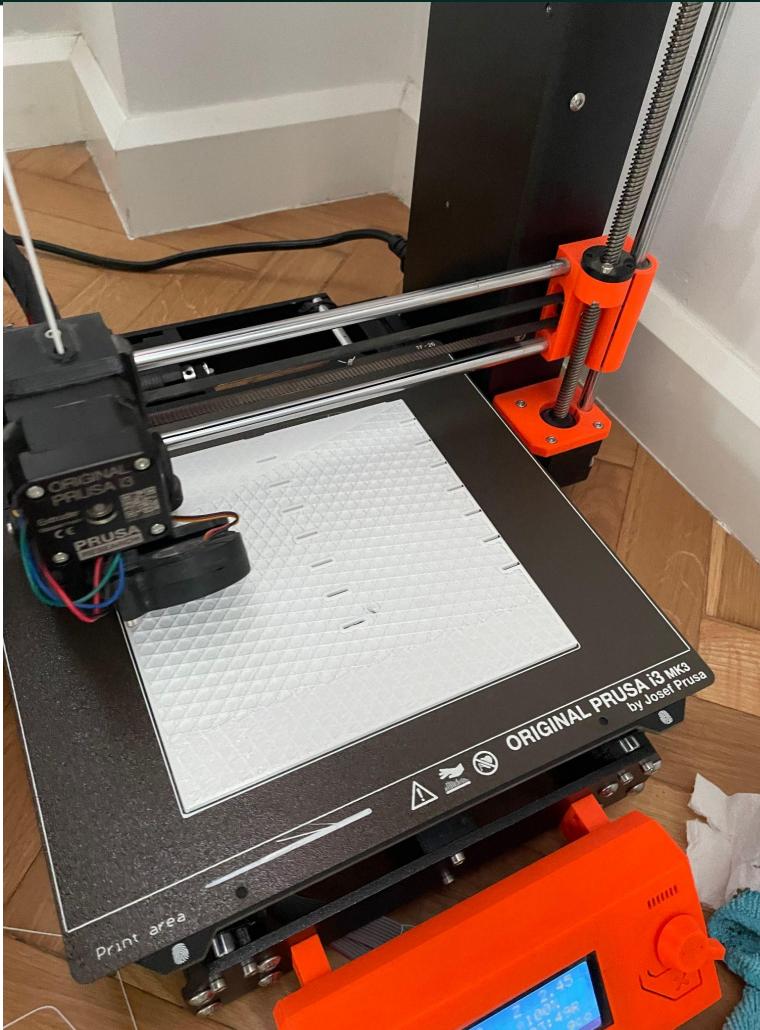
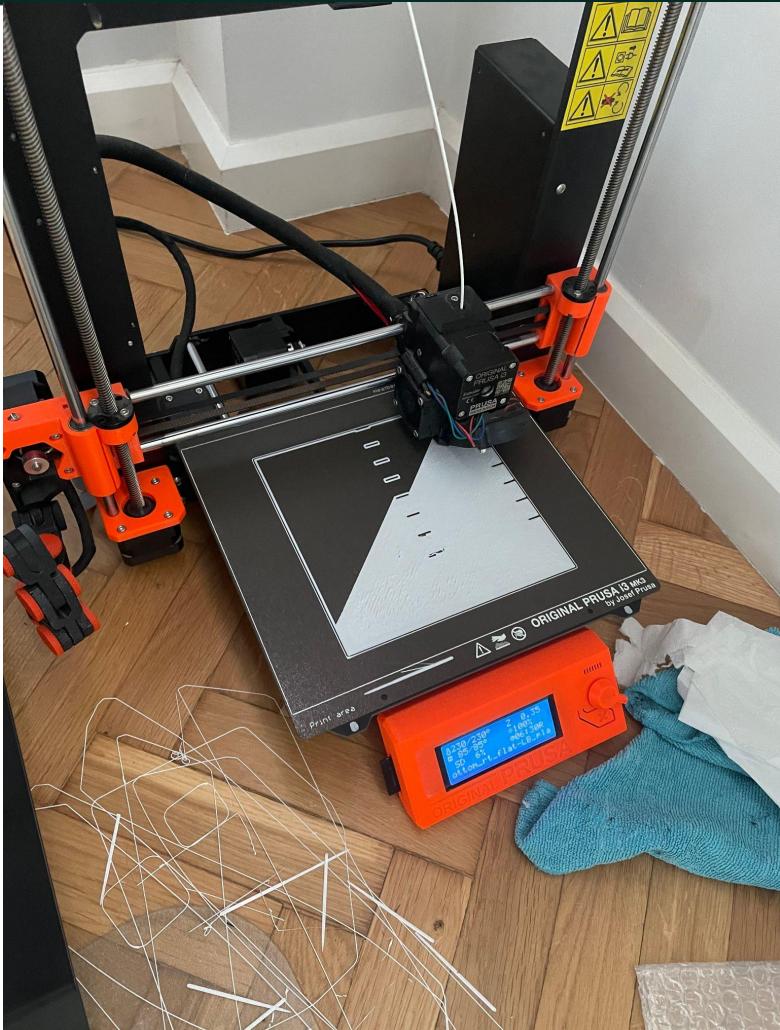
- 3D printing is great! PLA are usually good
- Unless...
- You start getting errors like this. You check the bearings and find they're not greased. It's **rubbish**
- You clog the nozzle. You clean it with the **cold pull**





# Zephyr® Project

## Developer Summit





Zephyr® Project

Developer Summit

Some takeaways from this part:

- always take layer height (width) into account when measuring
  - usually around 2mm
  - so if designing a hole, make it 2mm wider in radius
- Print very small slices and *validate your assumptions* as early as possible - IKEA mindset
- Bridges are fine, better to have a bridge then PETG supports that are a pain to remove!



Zephyr® Project  
Developer Summit

# Assembly



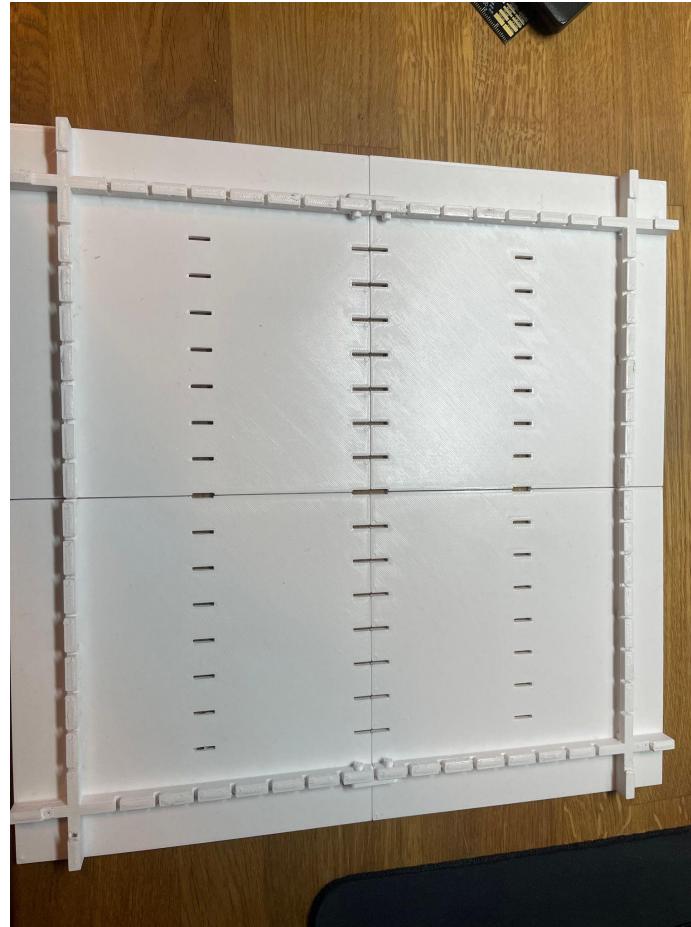
#EmbeddedOSSummit





# Zephyr® Project

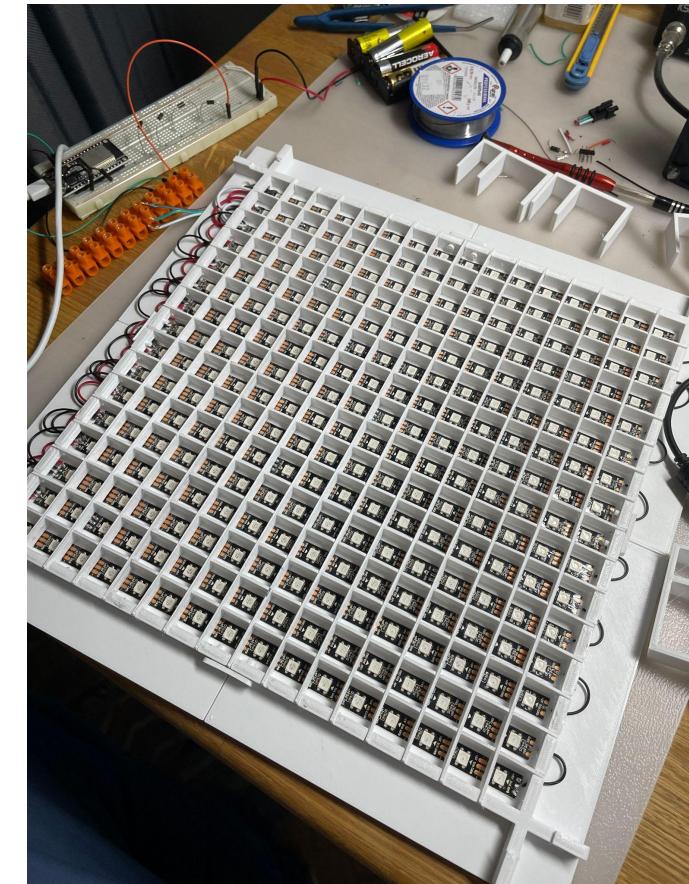
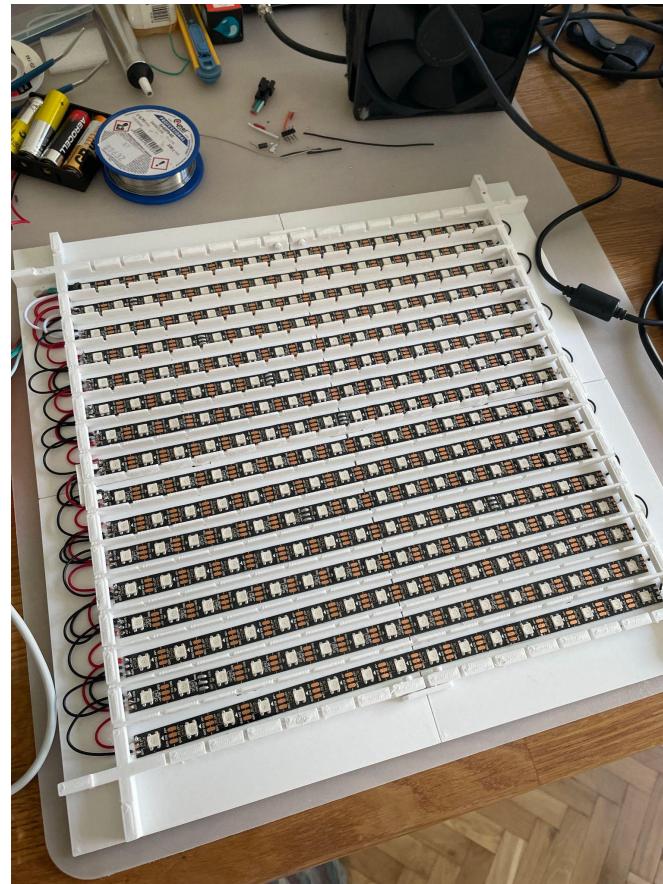
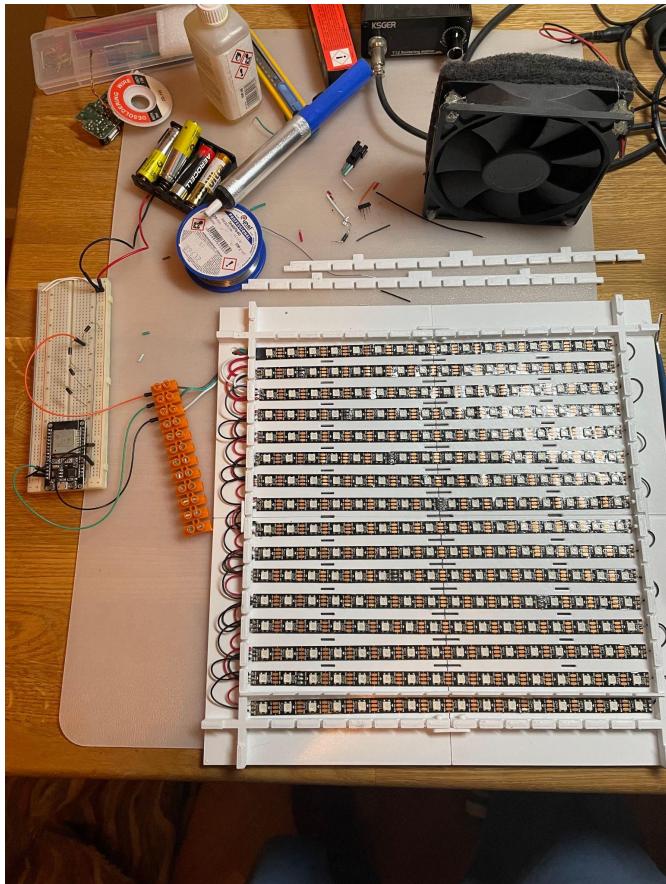
## Developer Summit





# Zephyr® Project

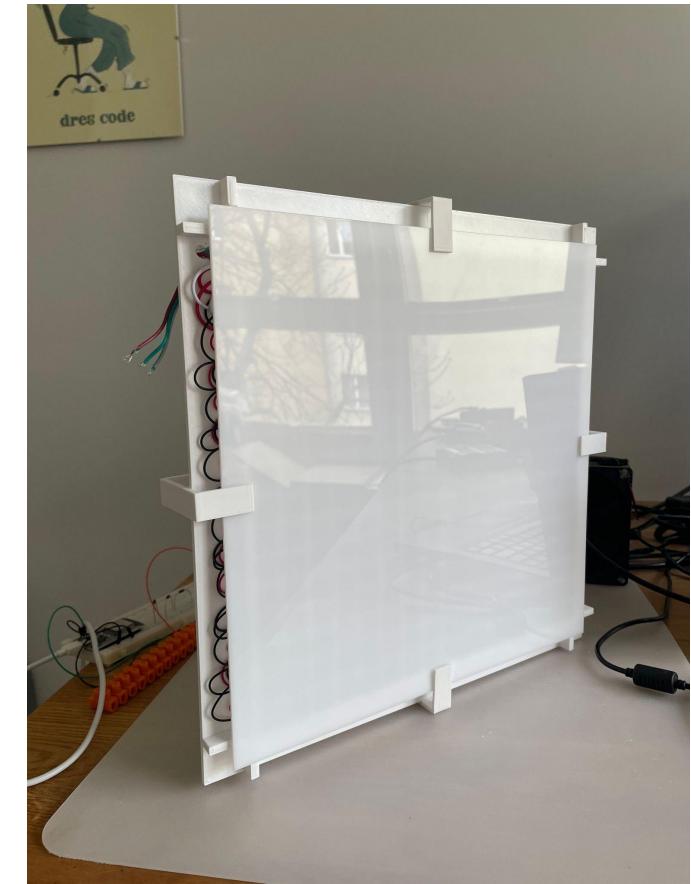
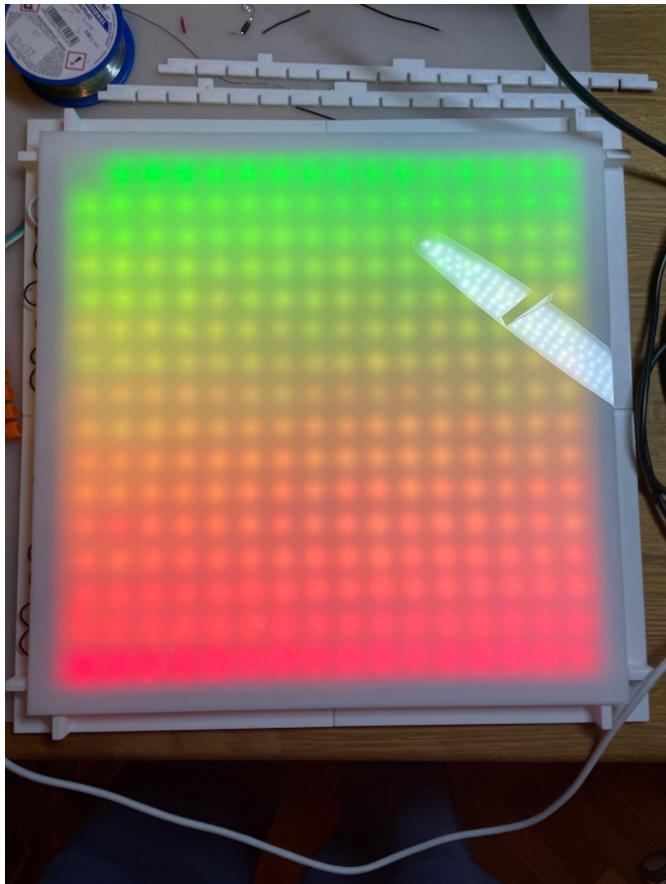
## Developer Summit





# Zephyr® Project

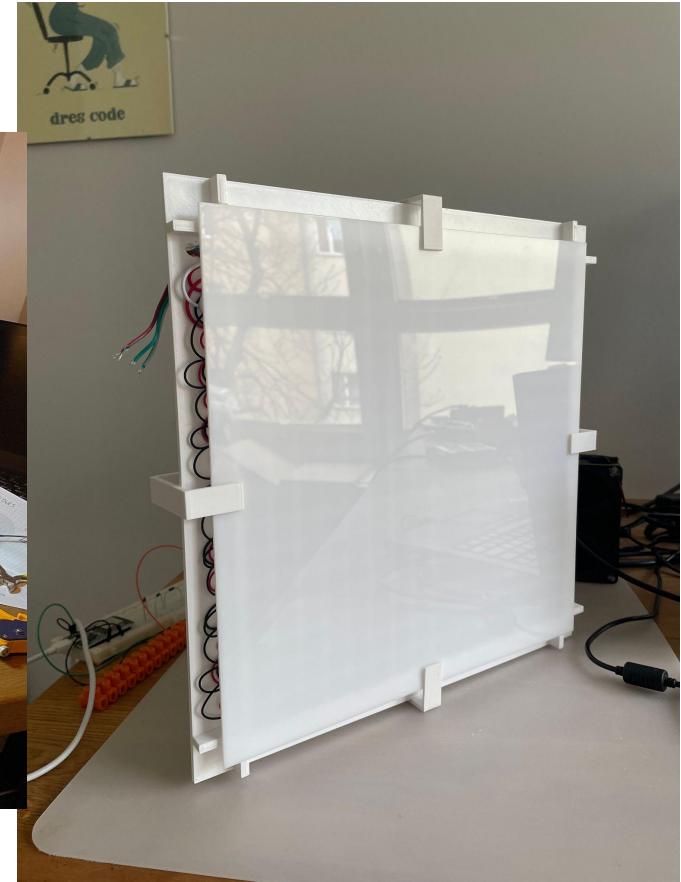
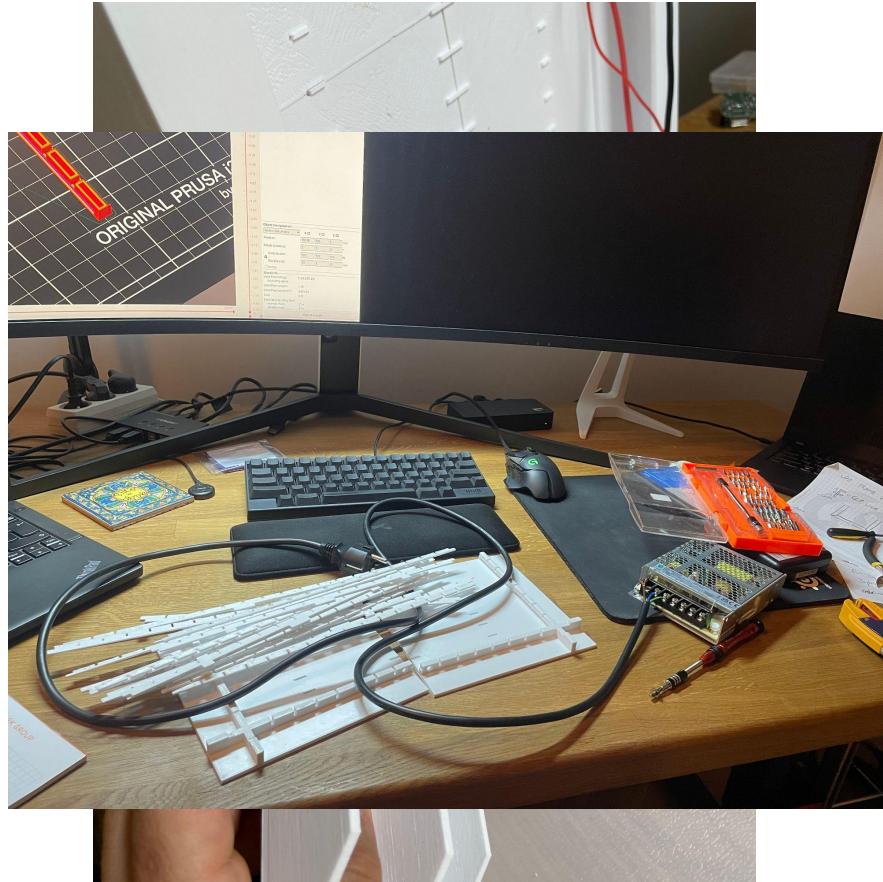
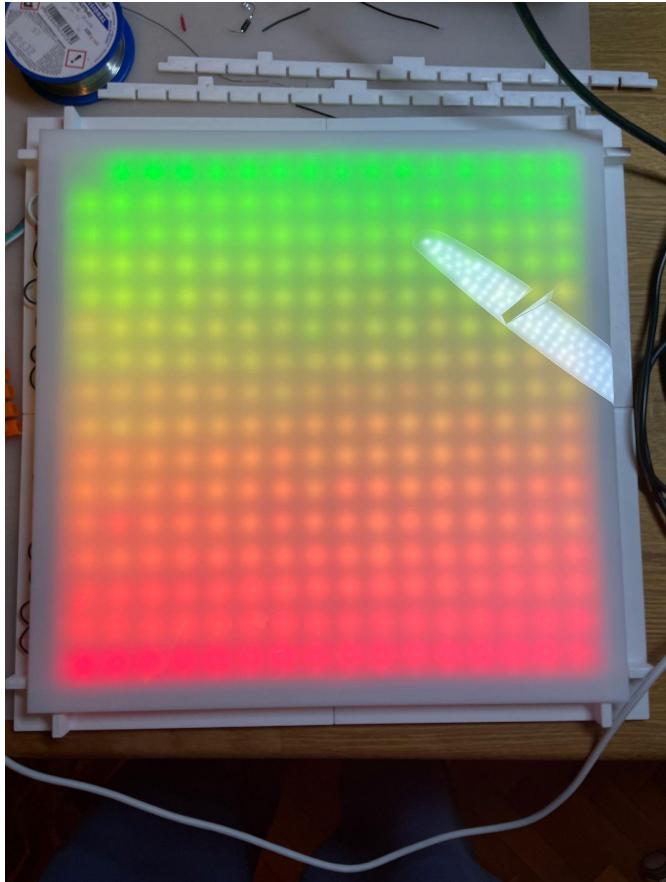
## Developer Summit





# Zephyr® Project

## Developer Summit

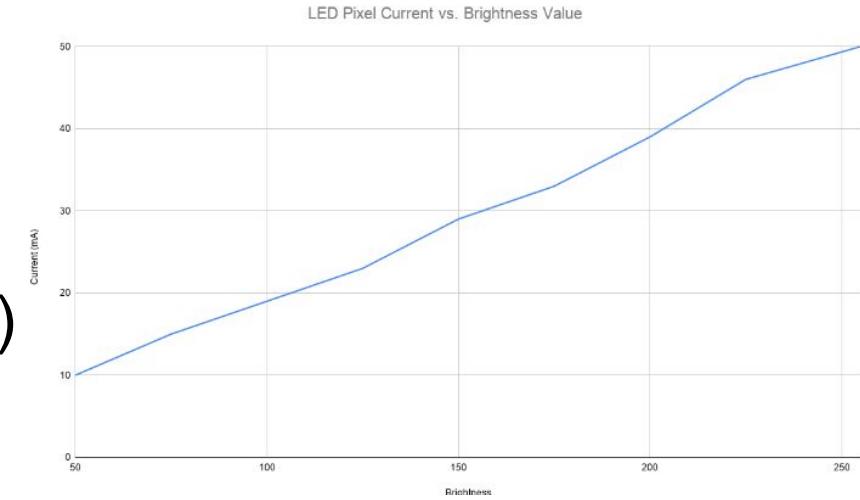




# Zephyr® Project

## Developer Summit

- Power measurements
- Running at 5V input:
  - 1.1W at Idle - MCU sleeping
  - 6W for your typical image (think Zephyr logo)
  - 16W for Link image
  - 22W for full brightness - **max I = 4.4A**
  - 13W half brightness
- Seems like my 16A @ 5V beefy PSU is a slight overkill but according to the graph I could test the limits of my PSU





Zephyr® Project  
Developer Summit

# Step 4+: Software Extensions





Zephyr® Project

Developer Summit

- With the MVP done and the project assembled we can:
  - Bash our heads against the wall debugging spurious networking crashes :)
  - Add new endpoints and tweak the UI
  - and of course:

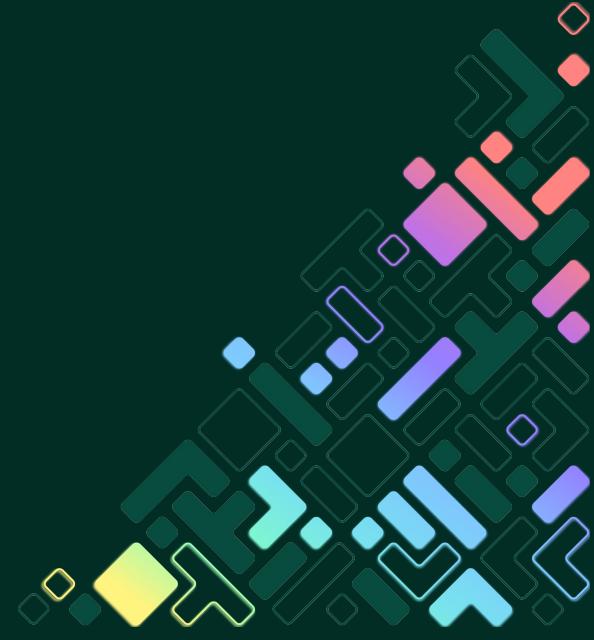


Zephyr® Project  
Developer Summit

# Things left out



#EmbeddedOSSummit





Zephyr® Project

Developer Summit

- Adding animations - had trouble to downconvert GIFs on the client using browser API
- Automatic Wi-Fi connection and connection restoring + LED to signal successful connection?
- Persistent memory storage of several samples and loading the sample on bootup
- UI improvements - cleaner interface
- Streaming games over Wi-Fi - using the Frame as an external screen
- Scaling down power requirements - using smaller PSU



Zephyr® Project  
Developer Summit

# Takeaways?



#EmbeddedOSSummit





Zephyr® Project

Developer Summit

- Check for board support **early**
- Hack around and rush for prototype - **validate**
- **Contribute to Zephyr documentation :)**
- Ask when in doubt - Discord is super friendly place
- Don't be afraid of new things - grow!
- Don't listen to people on the internet ↖\_(ツ)\_↗
- 3D printing and designing are an iterative process and you will fail a lot in your measurements :)



### Acknowledgements:

- Big thanks to all the folks in the networking chat on the Zephyr discords, especially:
  - jukkar - for providing me with tips and tricks that I shared and I \*will\* put into the documentation :)
  - Bell - for replying instantaneously!
  - sylvio - for giving me the magical prj.conf flags for ESP32 :)
- folks from Espressif who continue to improve RISC-V boards support (still a way to go there, so this is a small call to action - also for me :D)



Zephyr® Project

Developer Summit

## Further reading:

- FreeCAD recommended fork  
[https://github.com/realthunder/FreeCAD\\_assembly3/](https://github.com/realthunder/FreeCAD_assembly3/)
- Effortless printer calibration (really :))  
<https://www.printables.com/model/251587-stress-free-first-layer-calibration-in-less-than-5>
- Neopixels resource (WS281B)  
<https://wp.josh.com/2014/05/13/ws2812-neopixels-are-not-so-finicky-once-you-get-to-know-them/>
- Excellent RISC-V + Zephyr readings  
<https://danielmangum.com/categories/risc-v-bytes/>

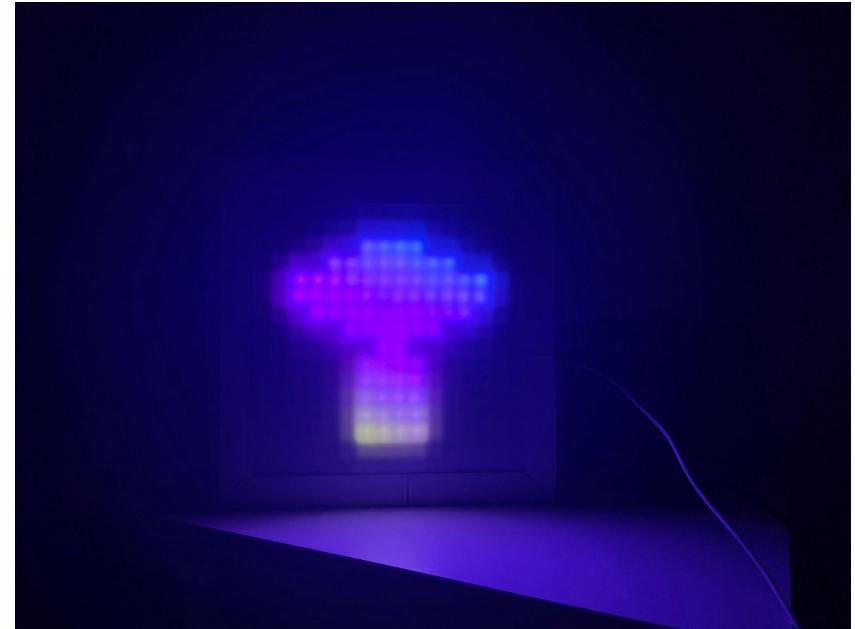


Zephyr® Project

Developer Summit

Blogpost series about this project:

- First part - Introduction and prototype
  - Second part - more about software
  - Third part - 3D design, 3D printing and assembly - TBA
  - Fourth part - extensions - TBA
  - More? (*maybe*)
- 
- Repo link: <https://github.com/JDuchniewicz/zled-frame>
  - HW repo link: <https://github.com/JDuchniewicz/zled-frame-hw>



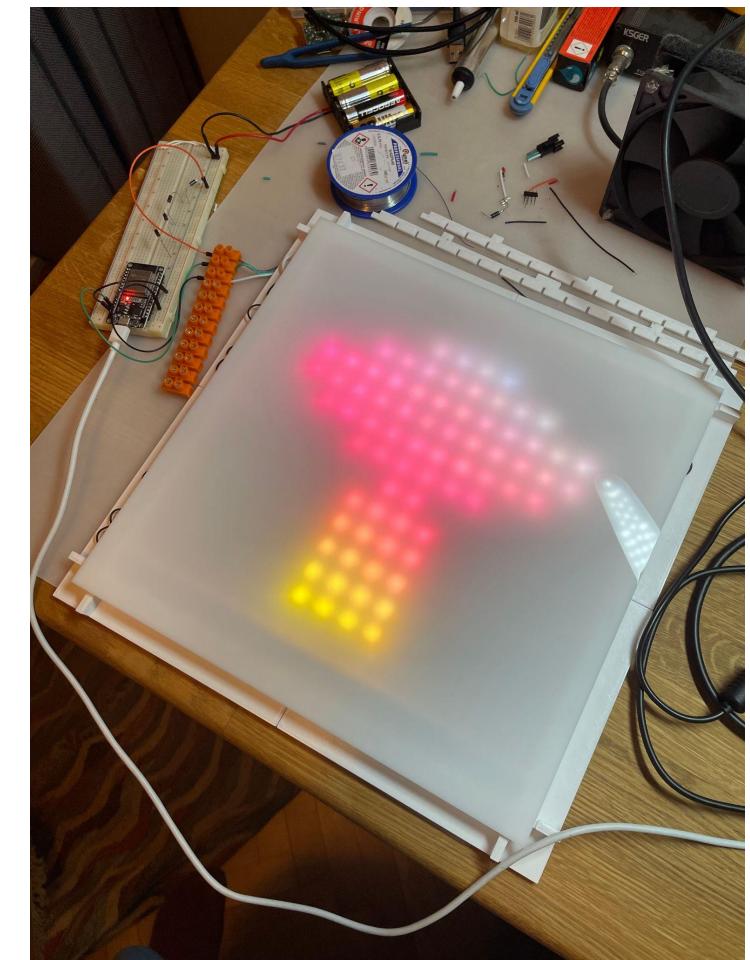
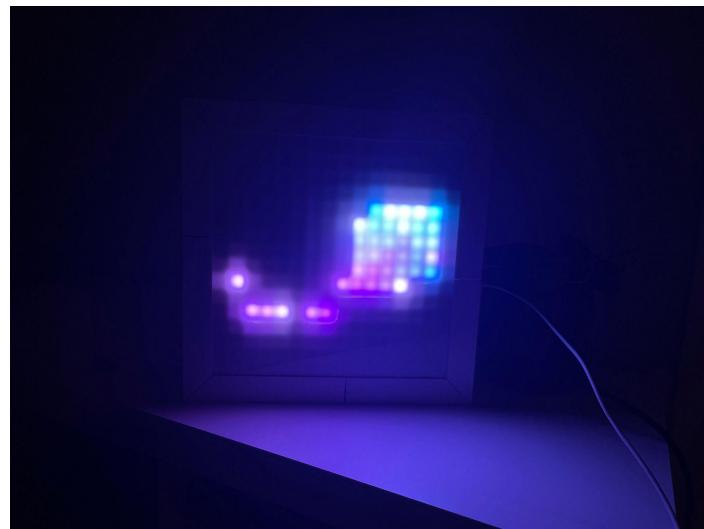


# Zephyr® Project

## Developer Summit

Of course all of this is Open Source (duh) and Open Hardware - at least the parts that I designed

*Remakes are welcome and encouraged :)*





Zephyr® Project  
Developer Summit



# Thank you. Questions?

Z LED Frame - "Illuminating Art with Zephyr"

[j.duchniewicz@gmail.com](mailto:j.duchniewicz@gmail.com)  
[www.jduchniewicz.com](http://www.jduchniewicz.com)  
[www.stickyistonstudios.com](http://www.stickyistonstudios.com)



#EmbeddedOSSummit

