# Turning the Microbit into a BLE microphone

## A development story

Colin Evrard, Javad Rahimipetroudi

mind

an open source of wisdom

# About me

- **Embedded Software Engineer**
  - Background in general C/C++
    - Recent shift in embedded
  - Started @Mind Jan 2025
  - Jack of all trades (except webdev)
  - Fan of reproducibility, scientific method

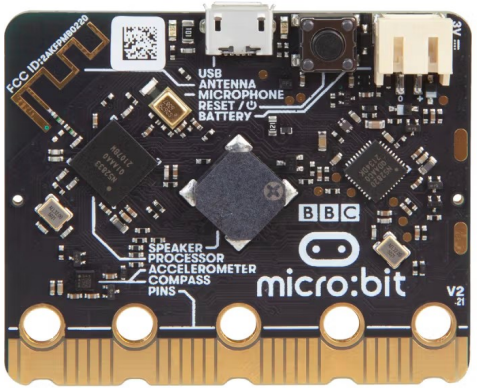
mind
an open source of wisdom

# Context

- **Got to know Zephyr @Mind**
  - From simple demos
  - To basic BLE
  - To a small contribution
- **Mind cultivates expertise**
  - More significant contribution as training
  - Looked for accessible boards to work on
  - BBC MicrobitV2 lying around


# Intro to the microbit v2

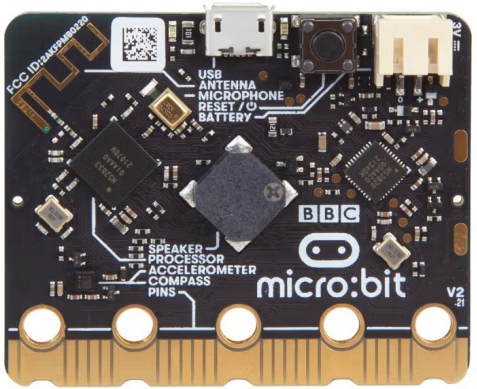
- **Tied to ARM history**
  - Spiritual successor to the BBC Micro
- **Education oriented board**
- **Easy to access**
- **From kindergarten to university**
- **Specs:**
  - Nordic nRF52833 Cortex-M4 32 bit with FPU
  - Flash 512KB
  - RAM 128KB
  - Clock Speed 64MHz
  - 2.4GHz radio




https://microbit.org/buy/bbc-microbit-single/

# Why use that board

- Zephyr support
- Mems analogic microphone
- Increase Zephyr awareness
- Strong existing ecosystem

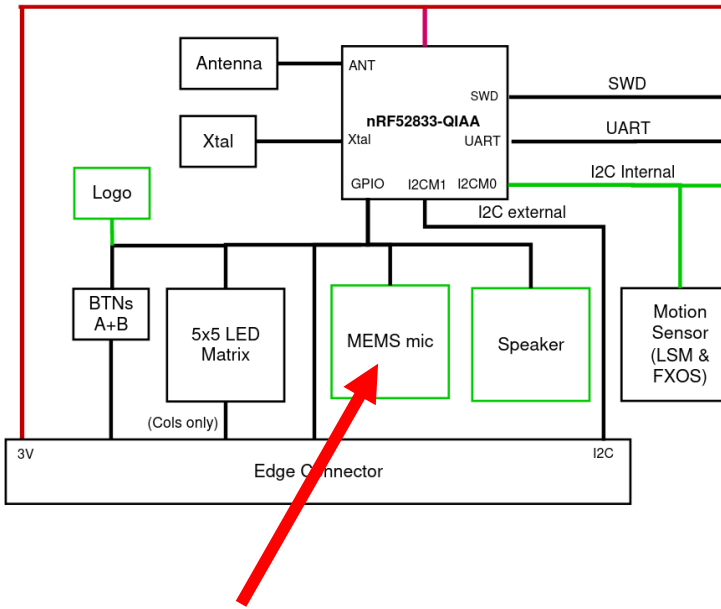



https://microbit.org/buy/bbc-microbit-single/

# Microbit Microphone

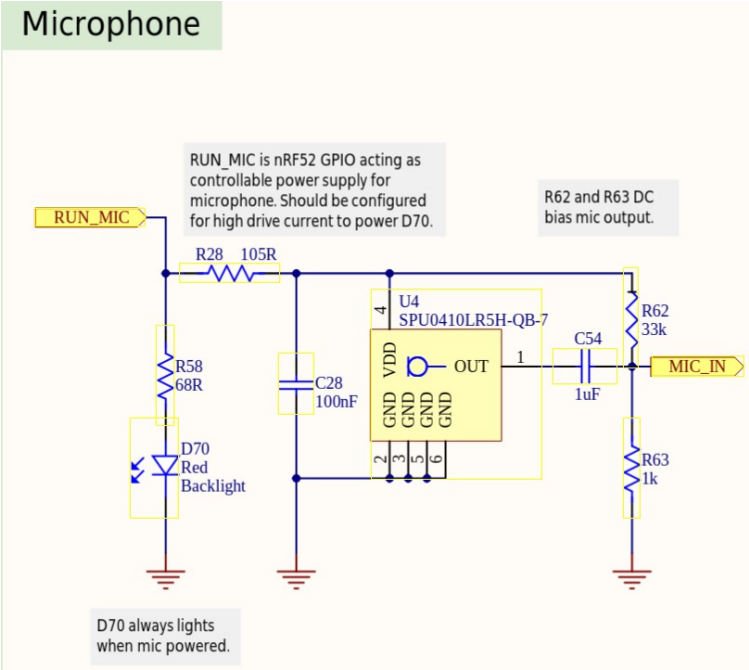
- **Comparison**
  - Microbit has a microphone and a speaker
  - Zephyr support for most sensors
  - Not for the Mems Microphone
  - Weird…
- **Ask other Zephyr enthusiast**
  - Titouan Christophe
  - Work on MIDI in Zephyr

https://tech.microbit.org/hardware/



# Zephyr Audio input support

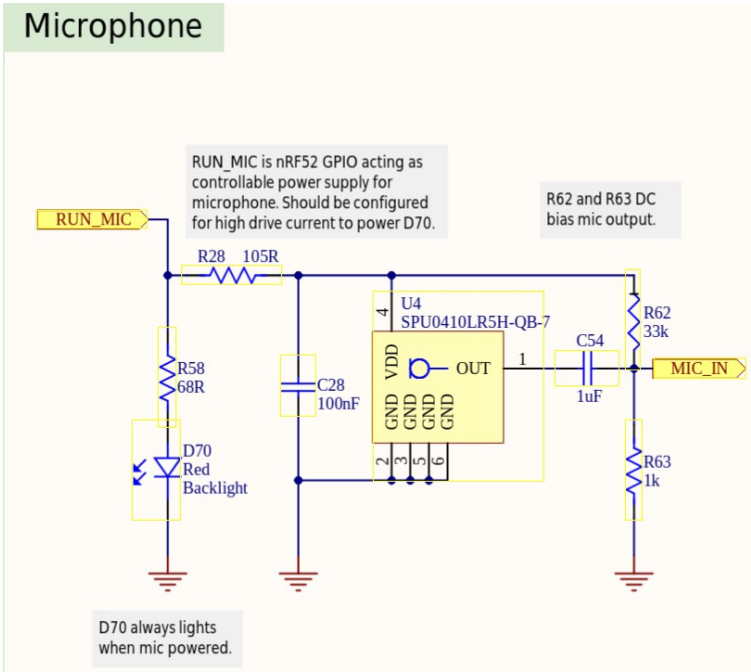
- **Wide Zephyr support**
  - DAI, DMIC, PDM, Audio codecs
  - Data in digital format
- **MicrobitV2**
  - Mems microphone
  - Seemingly hard wired to ADC
  - Needs GPIO input for power
  - ADC Audio input support ?




https://github.com/microbit-foundation/microbit-v2-hardware

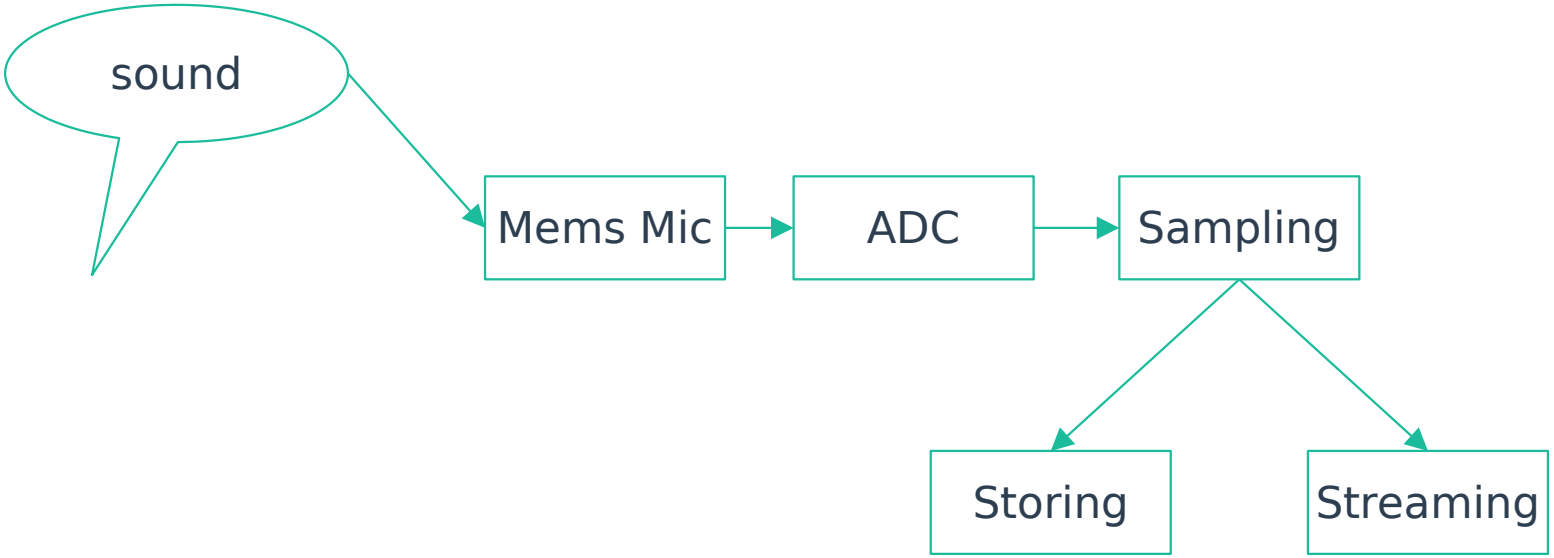
# Zephyr Audio input support

- **ADC Audio Input**
  - No direct support
- **SW PCM sampling**
  - Needs attention to details
  - Sampling frequency CPU limited
  - Timing needs to be precise
  - Storing vs Streaming




https://github.com/microbit-foundation/microbit-v2-hardware

# Project taking form



# Initial scope

- **Implement a recorder/speaker logic**
  - Record PCM samples from the mems mic
  - Replay the samples on the small speaker through PWM
- **Naive implementation**
  - Fix a sample rate (4KHz is a good starting point)
  - Fix max duration of recording with corresponding buffers
  - Use `k_busy_wait` to time precisely the samples
  - Replay the samples on the speaker

# Naive approach problems

- **Non-zero copy (No DMA)**
- **k_busy_wait hogs all resources**
- **Speaker quality not viable**
  - Difficult to estimate recording quality
  - Transfering wav sample in base64 over serial
    - Easy to do thanks to zephyr rich ecosystem
    - Still an involved process
    - Quality feedback still difficult

# Improving the Microbit Microphone

- **Sound sampling side**
  - Live streaming → Less buffer space
  - Use DMA
  - Use Re-entrant timer for sample pacing
  - Use `adc_read_async` instead of `adc_read_dt`
- **Get the sound out**
  - PCM samples from DMA
  - Encode to LC3
  - Stream over BAP Broadcast source profile

# Code

```
for (int sound_sample_idx = 0; sound_sample_idx < MAX_NUM_SAMPLES; sound_sample_idx++) {
    k_timer_status_sync(&sampling_timer);
    ret = k_poll(poll_evts, 1, K_NO_WAIT);
    if (ret != 0) {
        printk("Failed to poll during sampling %d\n", ret);
        break;
    }

    k_poll_signal_reset(&adc_signal);
    adc_seq.buffer = &(send_pcm_data[sound_sample_idx]);
    ret = adc_read_async(adc_spec.dev, &adc_seq, &adc_signal);
    if (ret != 0) {
        printk("Failed to read adc, sample #%d, code %d\n", sound_sample_idx, ret);
        sound_sample_idx = 0;
        break;
    }
}
```

# Code

```
for (int sound_sample_idx = 0; sound_sample_idx < MAX_NUM_SAMPLES; sound_sample_idx++) {
    k_timer_status_sync(&sampling_timer);
    ret = k_poll(poll_evts, 1, K_NO_WAIT);
    if (ret != 0) {
        printk("Failed to poll during sampling %d\n", ret);
        break;
    }

    k_poll_signal_reset(&adc_signal);
    adc_seq.buffer = &(send_pcm_data[sound_sample_idx]);
    ret = adc_read_async(adc_spec.dev, &adc_seq, &adc_signal);
    if (ret != 0) {
        printk("Failed to read adc, sample #%d, code %d\n", sound_sample_idx, ret);
        sound_sample_idx = 0;
        break;
    }
}
```

# BAP Microphone

- BAP = Basic Audio Profile
- Part of BLE specification
- Support is increasing
- Unicast/Broadcast profile
- Source/Sink profile
- LC3 codec support (also in Zephyr !)

# Improving the Microbit Microphone

- **Implementation**
  - Integrate Zephyr BAP broadcast sample
  - Fix scope (8KHz, LC3, No USB input)
  - Test out !

# Testing BAP on Linux

- **Debian 12**
- **Enable `bluetoothd` experimental features**
- **Use pipewire with lc3 plugin**
- **LC3 audio sink shows up**
- **Still can’t receive Microbit broadcast :(**

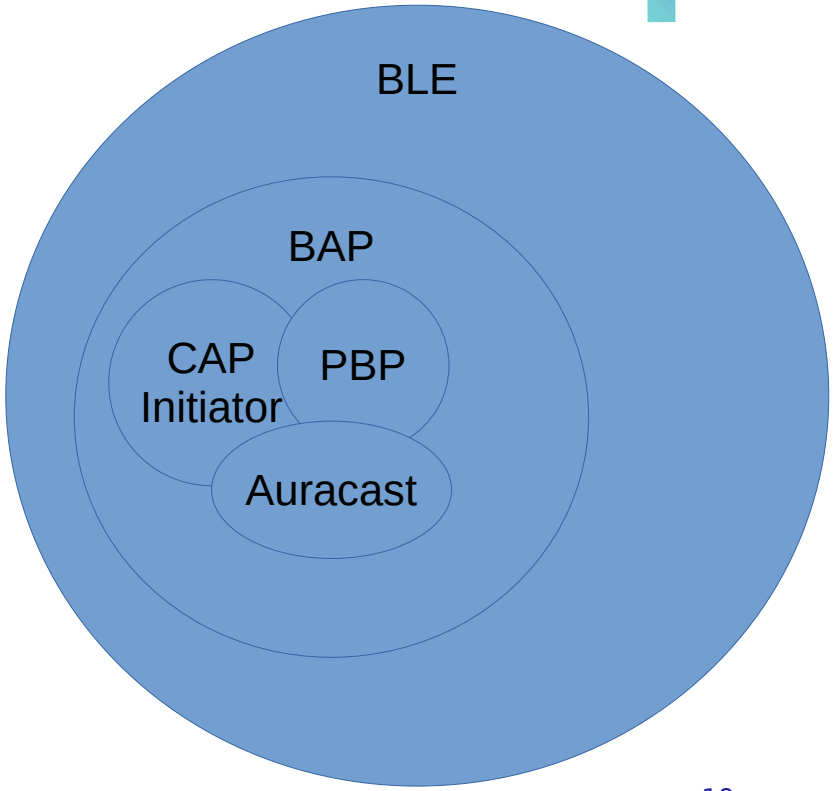
# Bumble

- **Python CLI tool by google**
- **Full Python bluetooth stack**
- **Flashed NRF52840 w/ hci_usb**
  - Enabled necessary features
  - Could see broadcasts with it

https://github.com/google/bumble

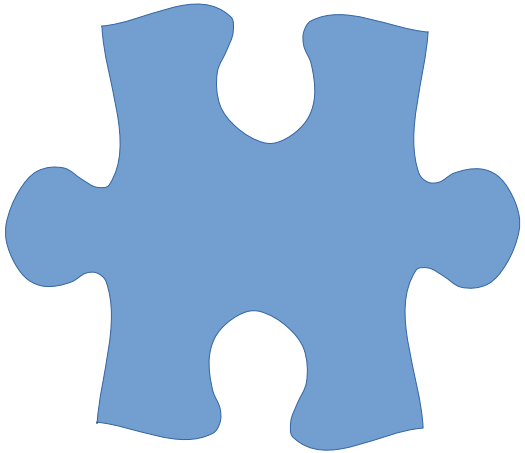
# Bluetooth LE audio complexity

- **Non-exhaustive diagram**
  - Complex system
  - Time consuming
  - Manageable
  - Clean reference material
- **Difficulty is elsewhere**
  - No easy to use reference implem
    - Zephyr could be considered a reference
  - Different name
  - Auracast, etc



# No demo

- **Everything looks fine**
  - Broadcast shows up in bumble
  - Stays in “pending”
  - Scanning shows it as  “started”
- **Implementation difference**
  - Must be the root cause
  - Who is right ? Bumble ? Zephyr ?
  - Difficult to know without reference



# Source code

- **Needs a bit of cleanup**
  - Will be uploaded to the mind gitlab in a few days
  - https://gitlab.com/essensium-mind/fosdem-26-microbit-microphone
  - Empty for now, come back in a few days after cleanup

# Conclusion

- **Zephyr**
  - Generic PCM sampler subsystem
- **BLE is difficult**
  - Let’s show the complete demo next year
- **Zephyr samples rule**
  - Hands-on approach
  - Allows to dive in complex topics

# Thank you for listening.

# Questions ?

mind

an open source of wisdom