BUG

- Be Your Guard -

Abstract

- Why Making BUG
- How BUG Works
- Proof of Concept
- To Improve
- Credit

Why Making BUG

When it comes to home safety, security camera is always the top choice.

Why Making BUG

When it comes to home safety, security camera is always the top choice.

But video is hard for resource constrained devices to store or analyze at realtime.

Why Making BUG

So BUG makes use of other sensor-able materials to record signals from environment as events, such as:

- Sound
- Position
- Luminosity

Why Making BUG

Most time a meaningful event for human is composed by multiple single-material events.

Why Making BUG

Most time a meaningful event for human is composed by multiple single-material events.

EX: Opening a door might be accompanied with

- Specific Sound
- Different Position
- Different Luminosity

Why Making BUG

Most time a **meaningful event** for human is composed by multiple single-material events.

EX: **Opening a door** might be accompanied with

- Specific Sound
- Different Position
- Different Luminosity

Why Making BUG

Most time a meaningful event for human is composed by multiple single-material events.

EX: Opening a door might be accompanied with

- Specific Sound
- Different Position
- Different Luminosity

Why Making BUG

Most time a meaningful event for human is composed by multiple single-material events.

EX: Opening a door might be accompanied with

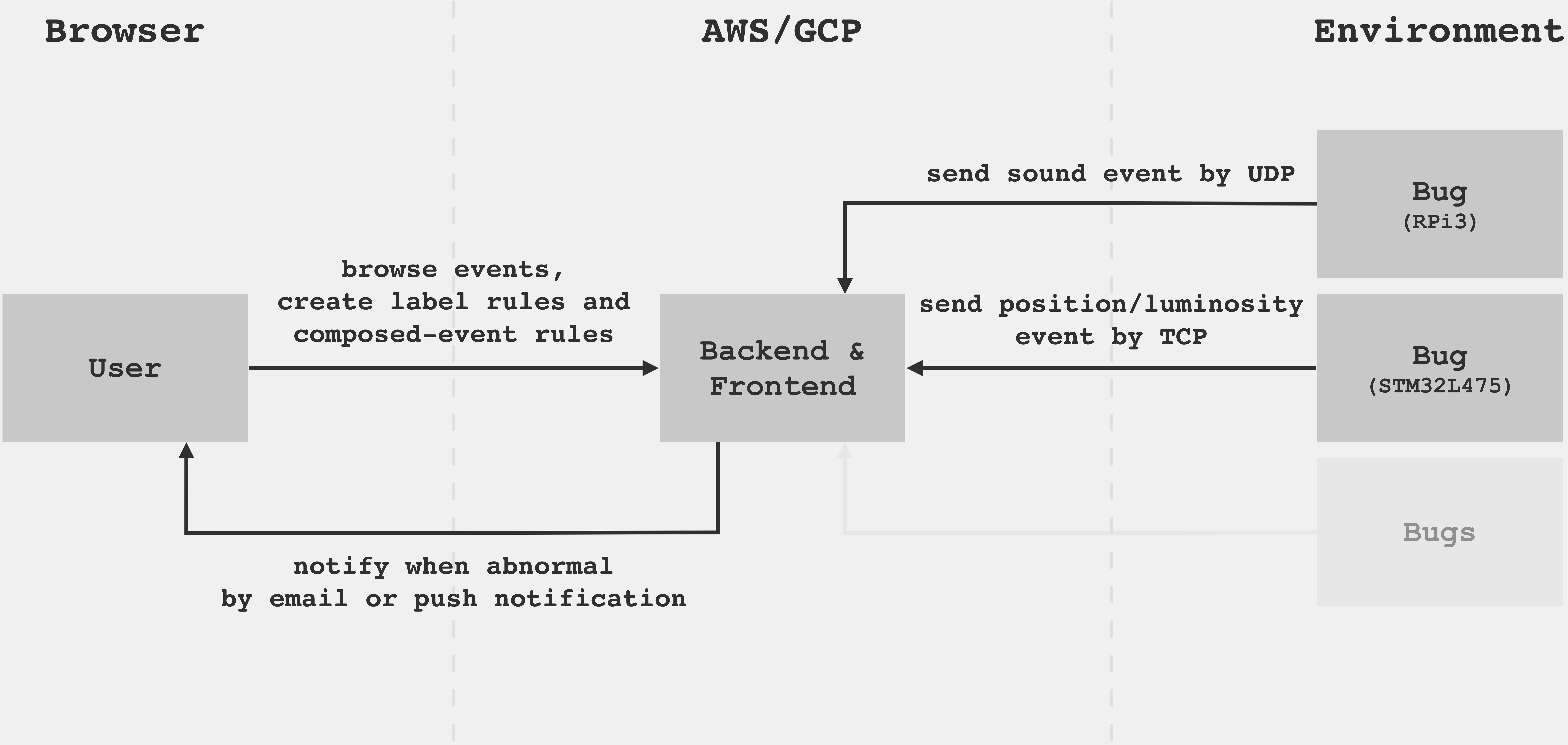
- Specific Sound
- Different Position
- Different Luminosity

Why Making BUG

So BUG provides user an interface to define rule about:

- How to label single-material events
- How to compose the labelled events to be human readable ones

How BUG Works



BUG - Bug(STM32L475)

Multithreaded Actor Model

BUG - Bug(STM32L475)

Multithreaded Actor Model

- WIFI - keep connection to AP

BUG - Bug(STM32L475)

Multithreaded Actor Model

- WIFI - keep connection to AP
- TCPClient - keep connection to Backend

BUG - Bug(STM32L475)

Multithreaded Actor Model

- WIFI - keep connection to AP
- TCPClient - keep connection to Backend
- Position Tracker - track position by sensing acceleration repeatedly

BUG - Bug(STM32L475)

Multithreaded Actor Model

- WIFI - keep connection to AP
- TCPClient - keep connection to Backend
- Position Tracker - track position by sensing acceleration repeatedly
- Luminosity Tracker - track luminosity by sensing photoresistor repeatedly

BUG - Bug(STM32L475)

Multithreaded Actor Model

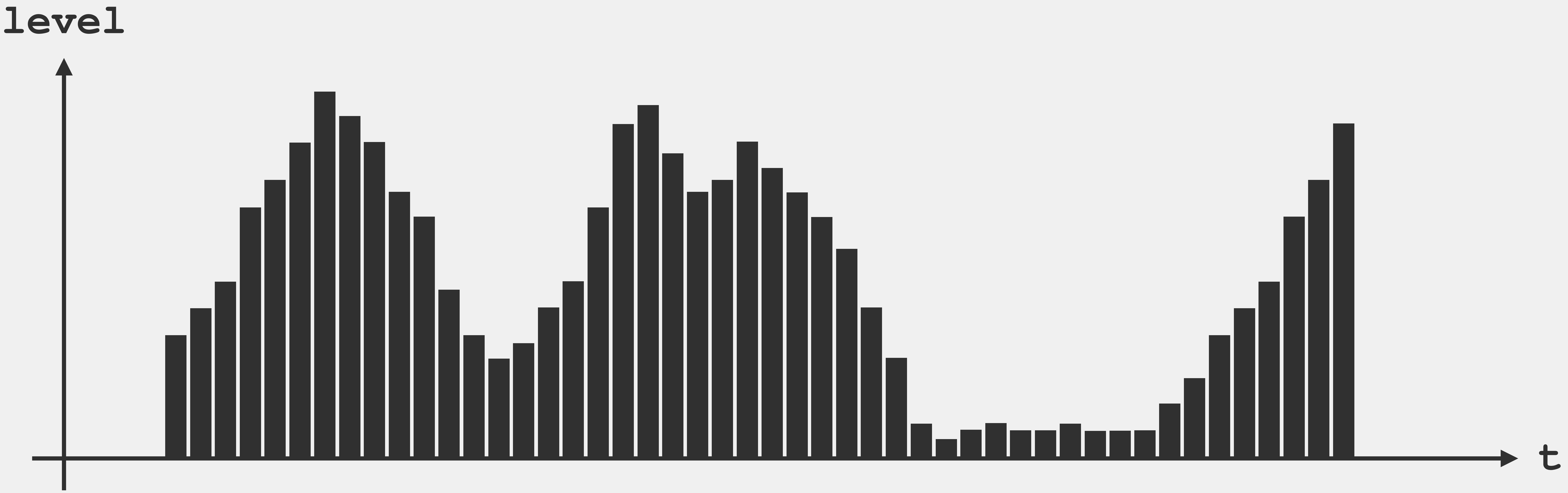
- WIFI - keep connection to AP
- TCPClient - keep connection to Backend
- Position Tracker - track position by sensing acceleration repeatedly
- Luminosity Tracker - track luminosity by sensing photoresistor repeatedly

BUG - Bug(STM32L475)

Filter noise using noise gate algorithm

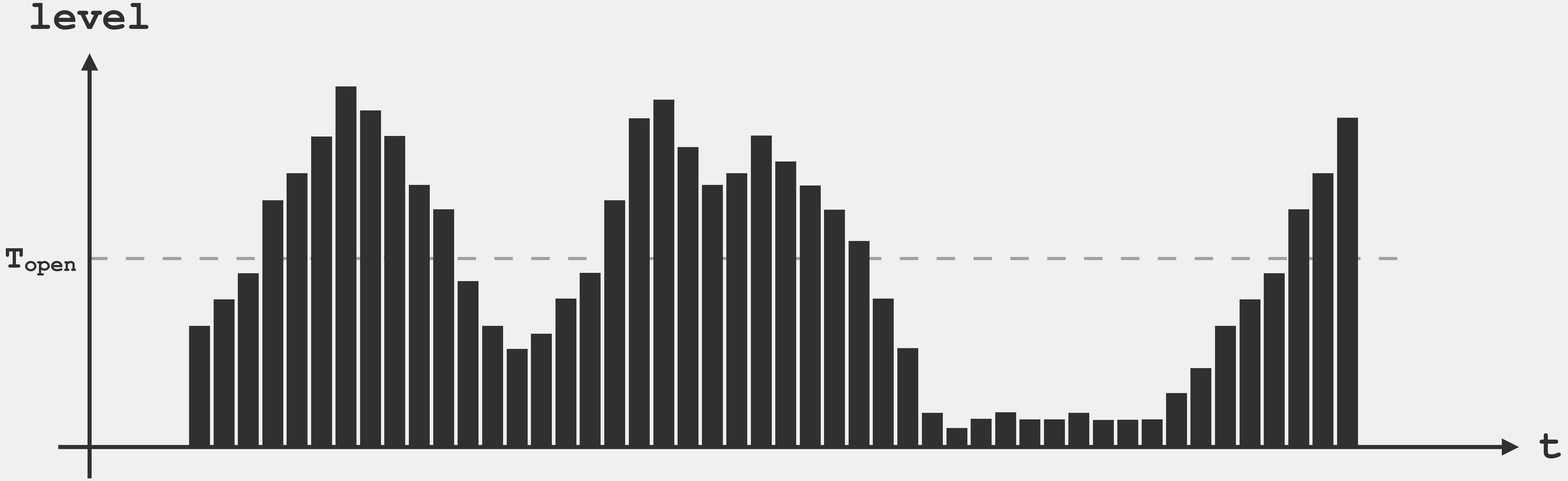
BUG - Bug(STM32L475)

Filter noise using noise gate algorithm



BUG - Bug(STM32L475)

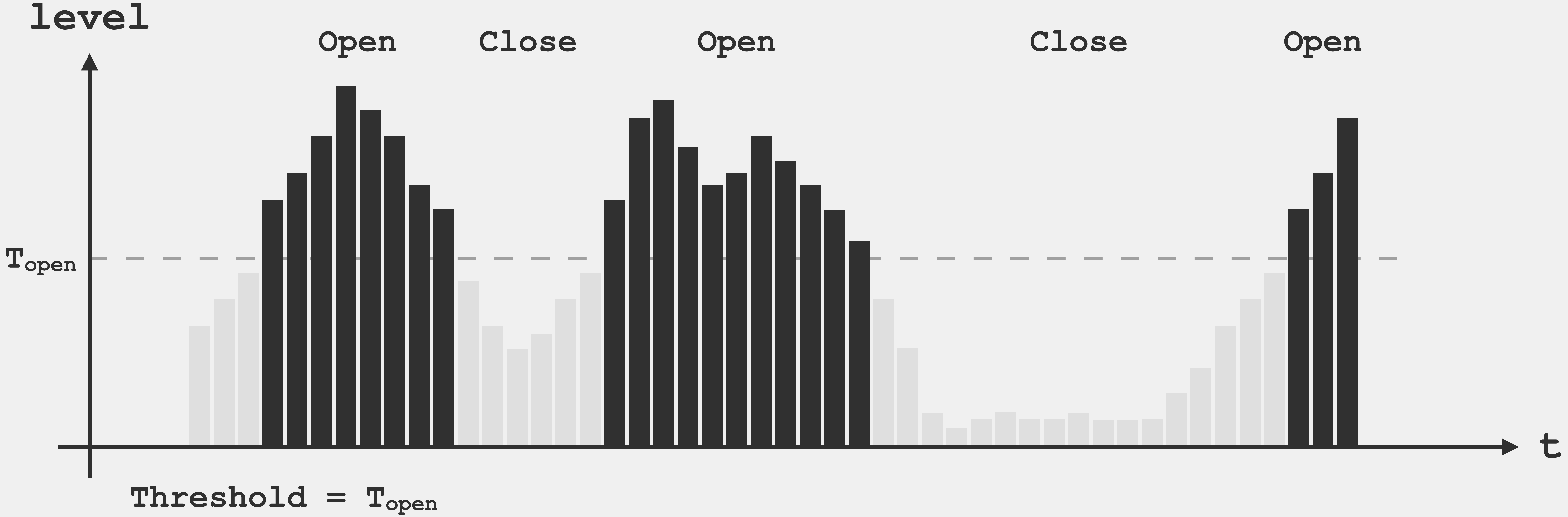
Filter noise using noise gate algorithm



Threshold = T_{open}

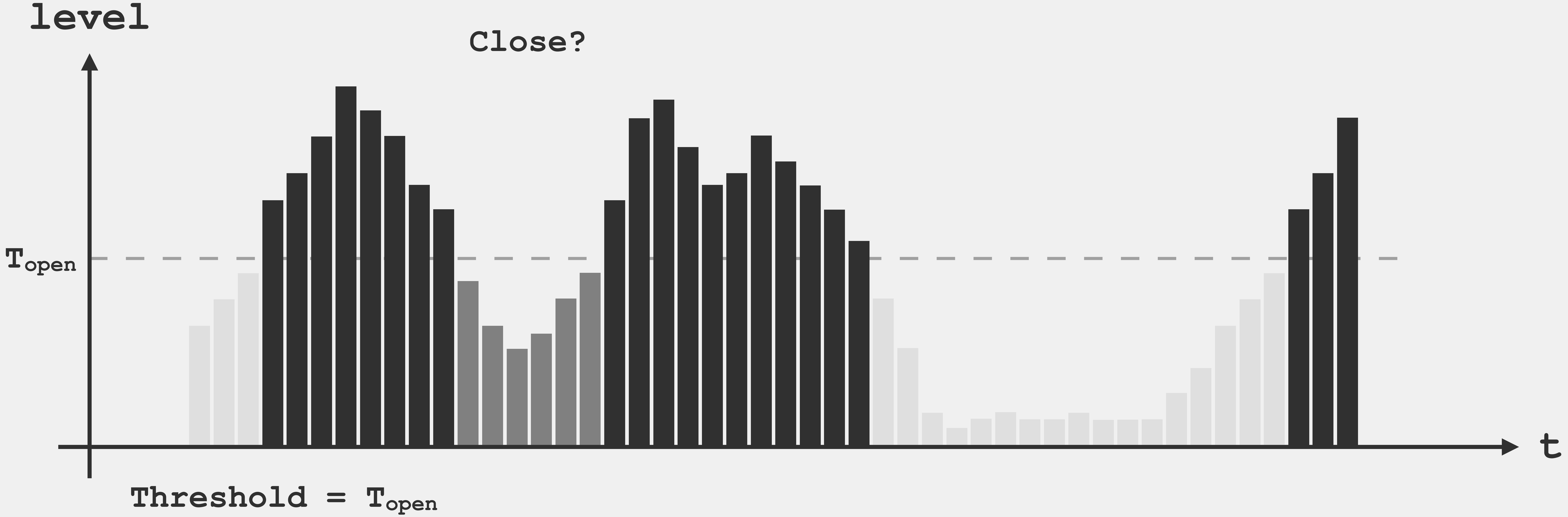
BUG - Bug(STM32L475)

Filter noise using noise gate algorithm



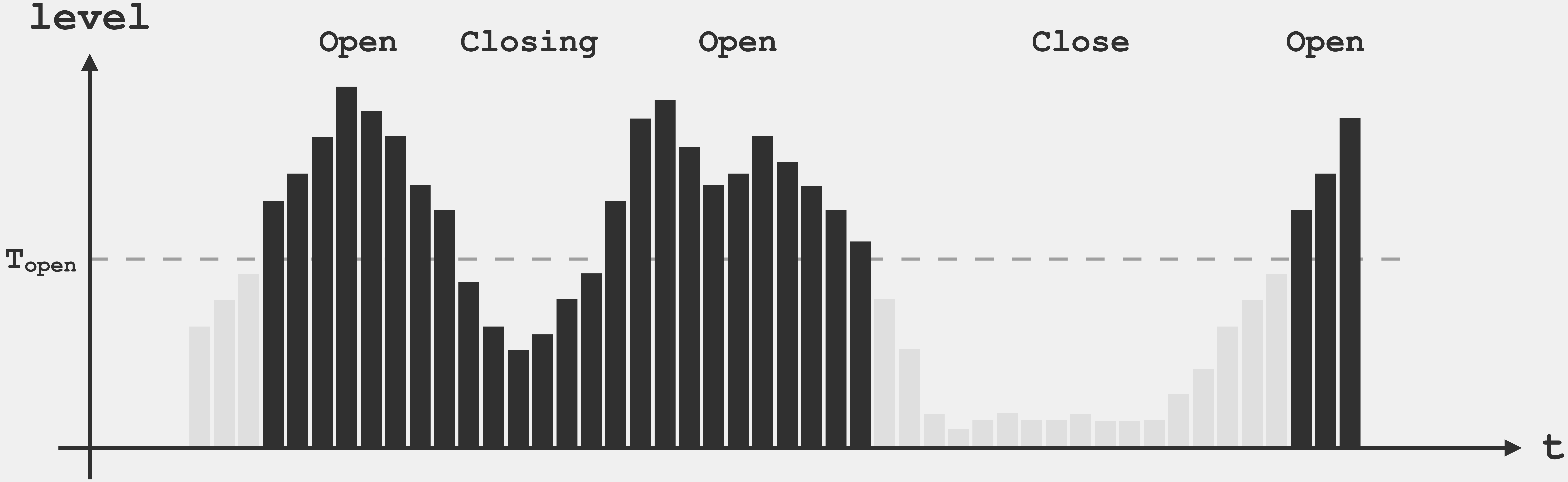
BUG - Bug(STM32L475)

Filter noise using noise gate algorithm



BUG - Bug(STM32L475)

Filter noise using noise gate algorithm



Threshold = T_{open}

Release Sample Count = 10

BUG - Bug(RPi3)

- Use USB microphone to sense sound
- Filter noise using noise gate algorithm
- Send data through UDP

BUG - Backend & Frontend

- Backend
  - Written in Rust
    - With tokio async runtime
    - With actix actor model
  - Use PostgreSQL
- Frontend
  - Written in Typescript with Next.js

Proof of Concept

To Improve

- Implement luminosity tracker
- Implement composed events composing
- Implement email notification and push notification
- Use TLS/DTLS between Bug and Backend
- Offline mode

Credit

- Audio Processing for Dummies