

Making Microcontrollers Sing

Building a Better Platform for Real-Time Audio on the ESP32

By Dirk Roosenburg



Common microcontrollers are often inconvenient for audio projects

Raspberry Pi

\$20 - \$40

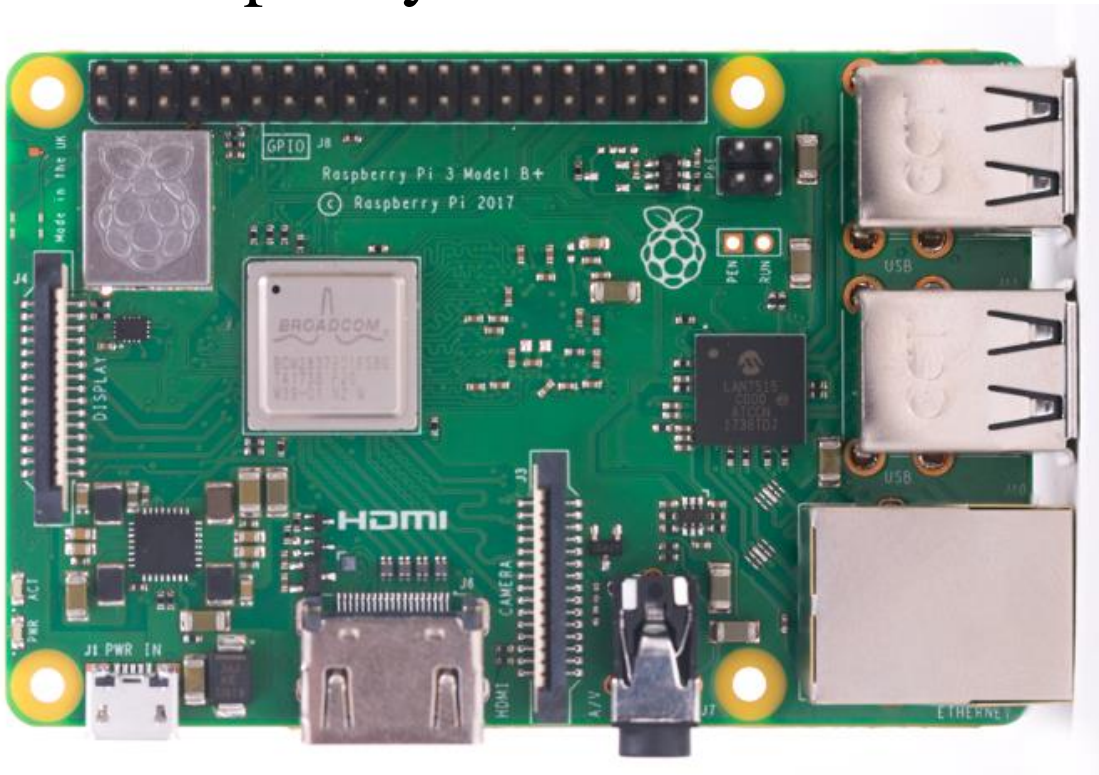


Image Source: raspberrypi.org [1]

Arduino

\$20 - \$40

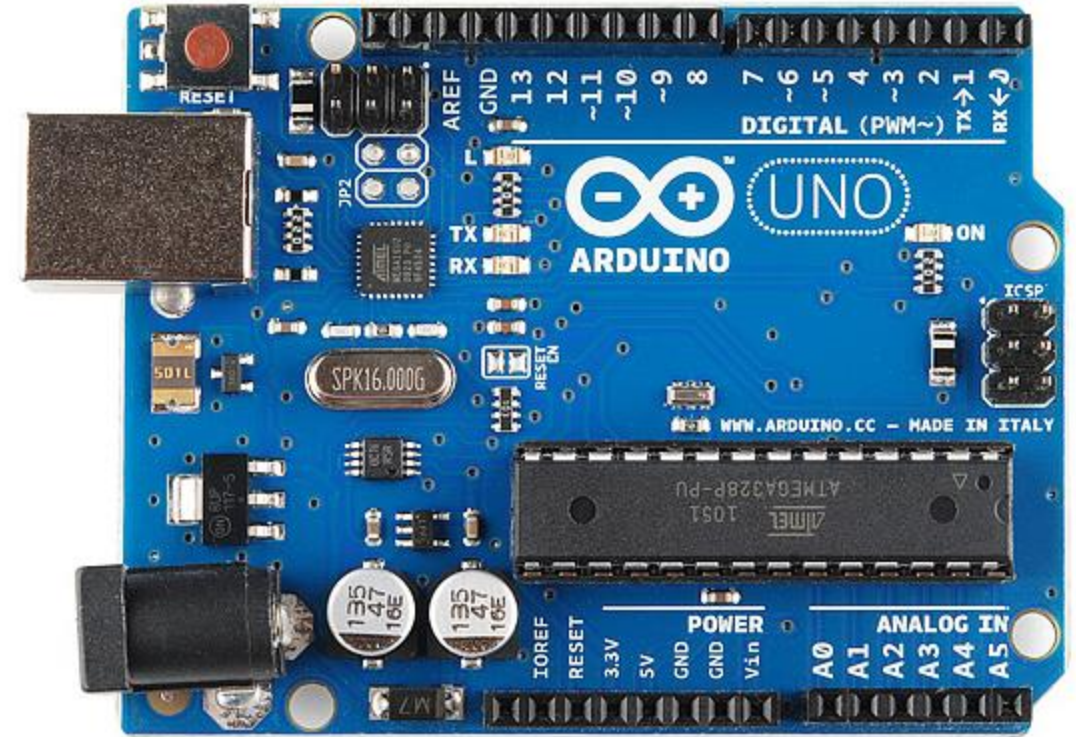


Image Source: sparkfun.com [2]

Platforms for audio exist, but they're expensive

Bela

\$150-\$200

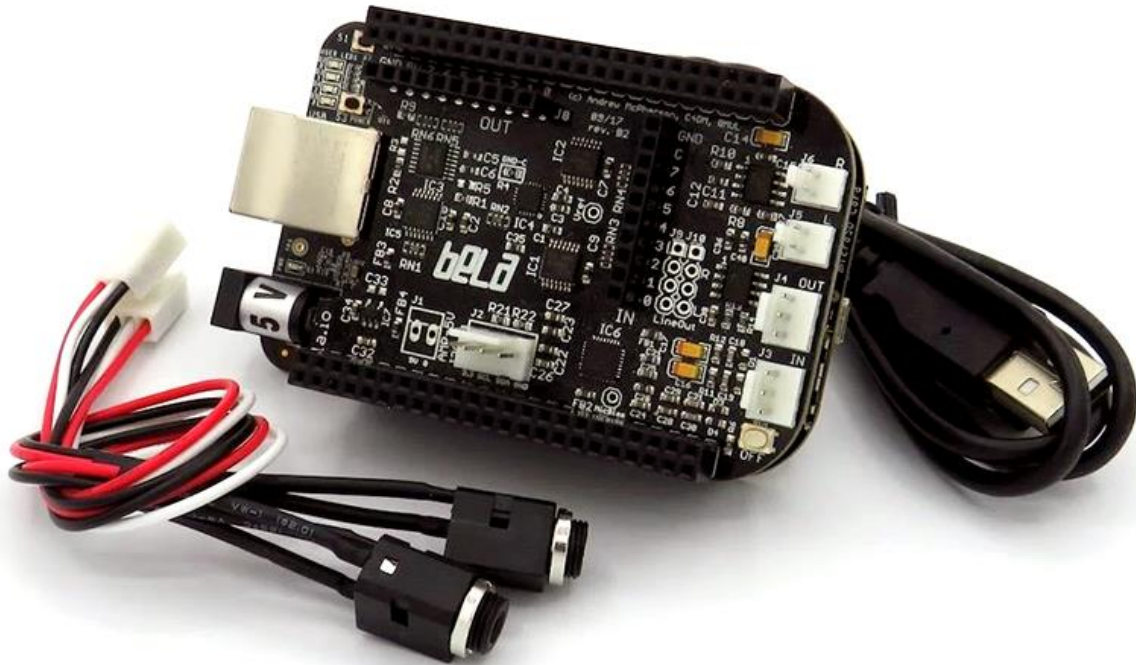


Image Source: bela.io [3]

Teensy - \$50 - \$100

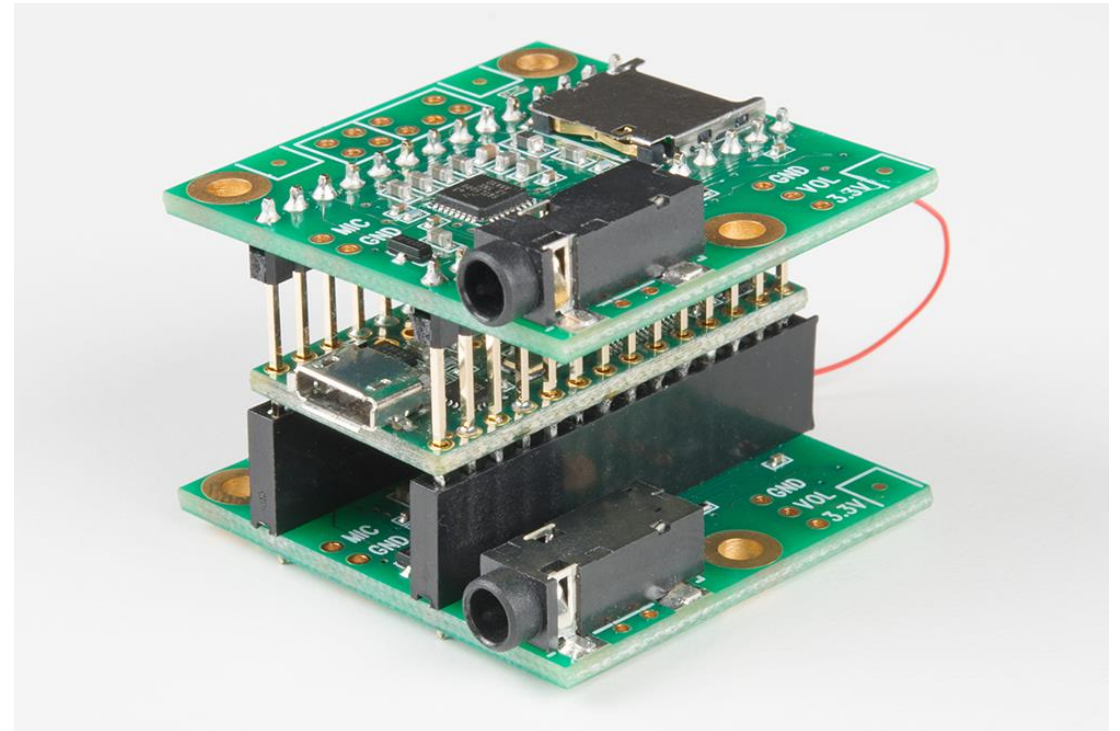
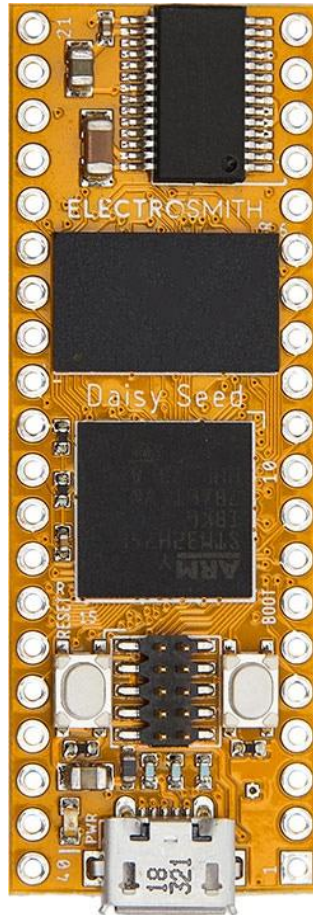


Image Source: sparkfun.com [4]

The Daisy: A New Challenger



Daisy

\$30

Image Source: electro-smith.com [5]

The ESP32: Great Potential in a Inexpensive Package



ESP32

\$5 module
~\$15 - \$30 boards

Image Source: botland.com [10]

A large selection of development boards provide many outboard options

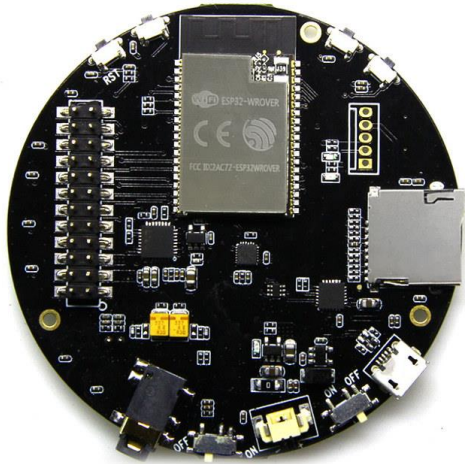


Image Source: cnx-software.com
[6]

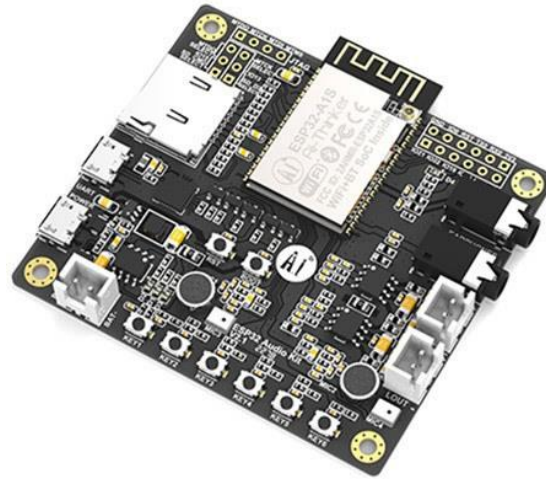


Image Source: cnx-software.com
[7]

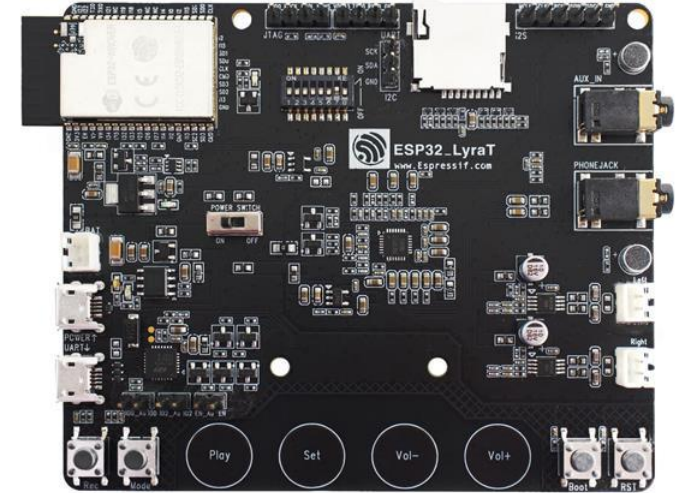
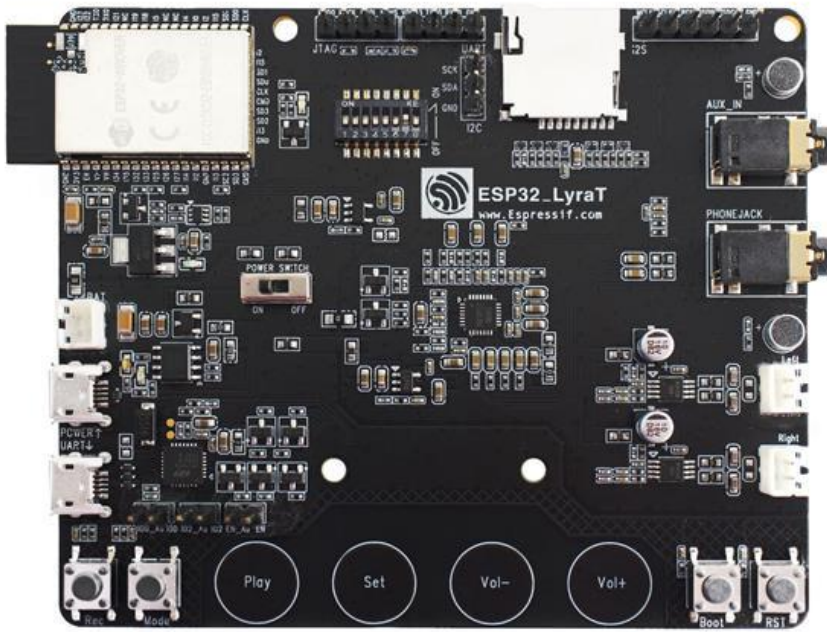


Image Source: mouser.com [8]

LyraT: a good platform for pedals and synth modules



- A synthesizer module or guitar pedal with custom DSP
- A Polyphonic Midi Synthesizer
- A Bluetooth Speaker

Image Source: mouser.com [8]

TTGo TAudio: a good platform for building instruments

- An instrument with visual feedback through LED's
- A self contained wearable instrument using accelerometer control

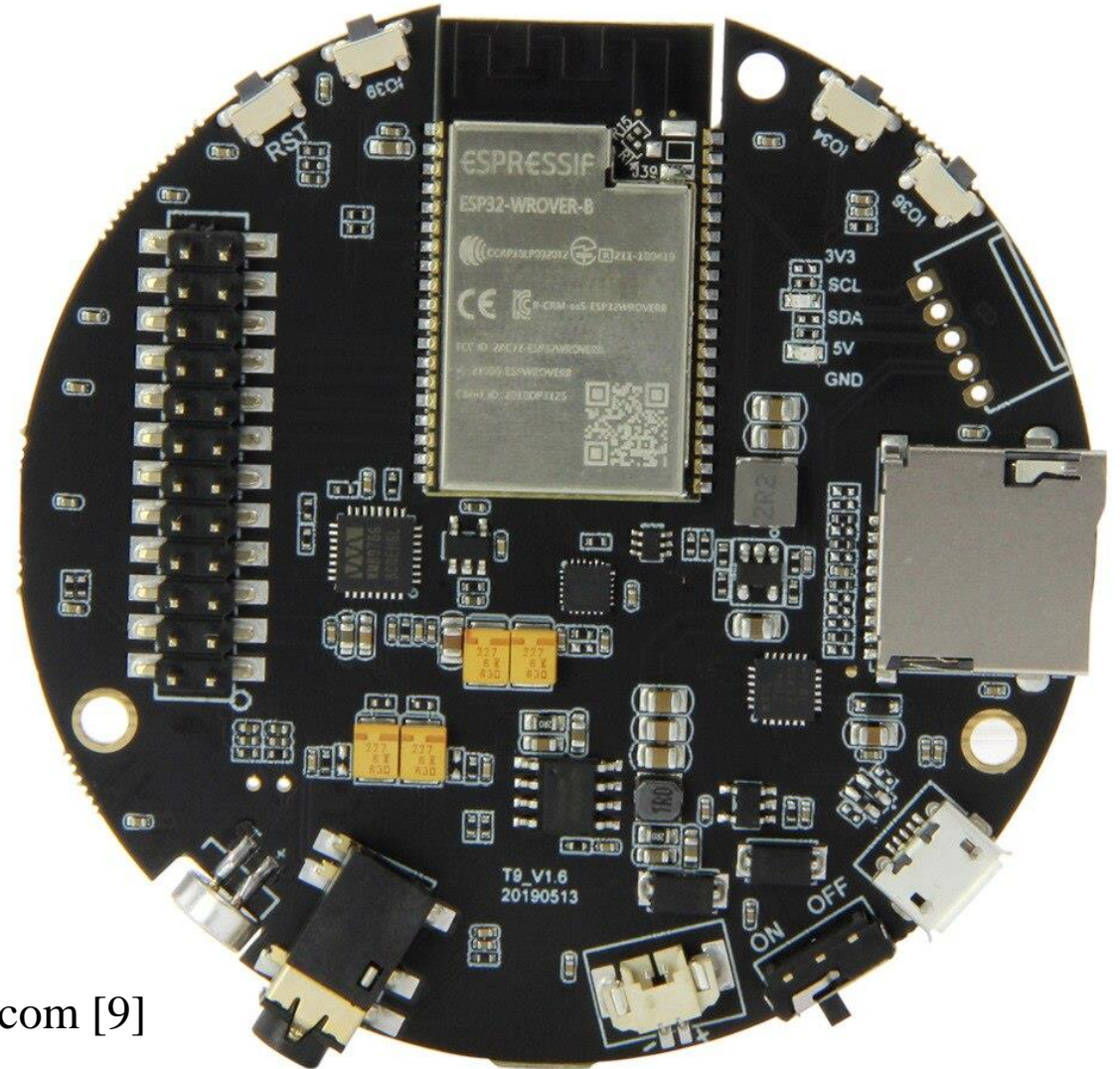
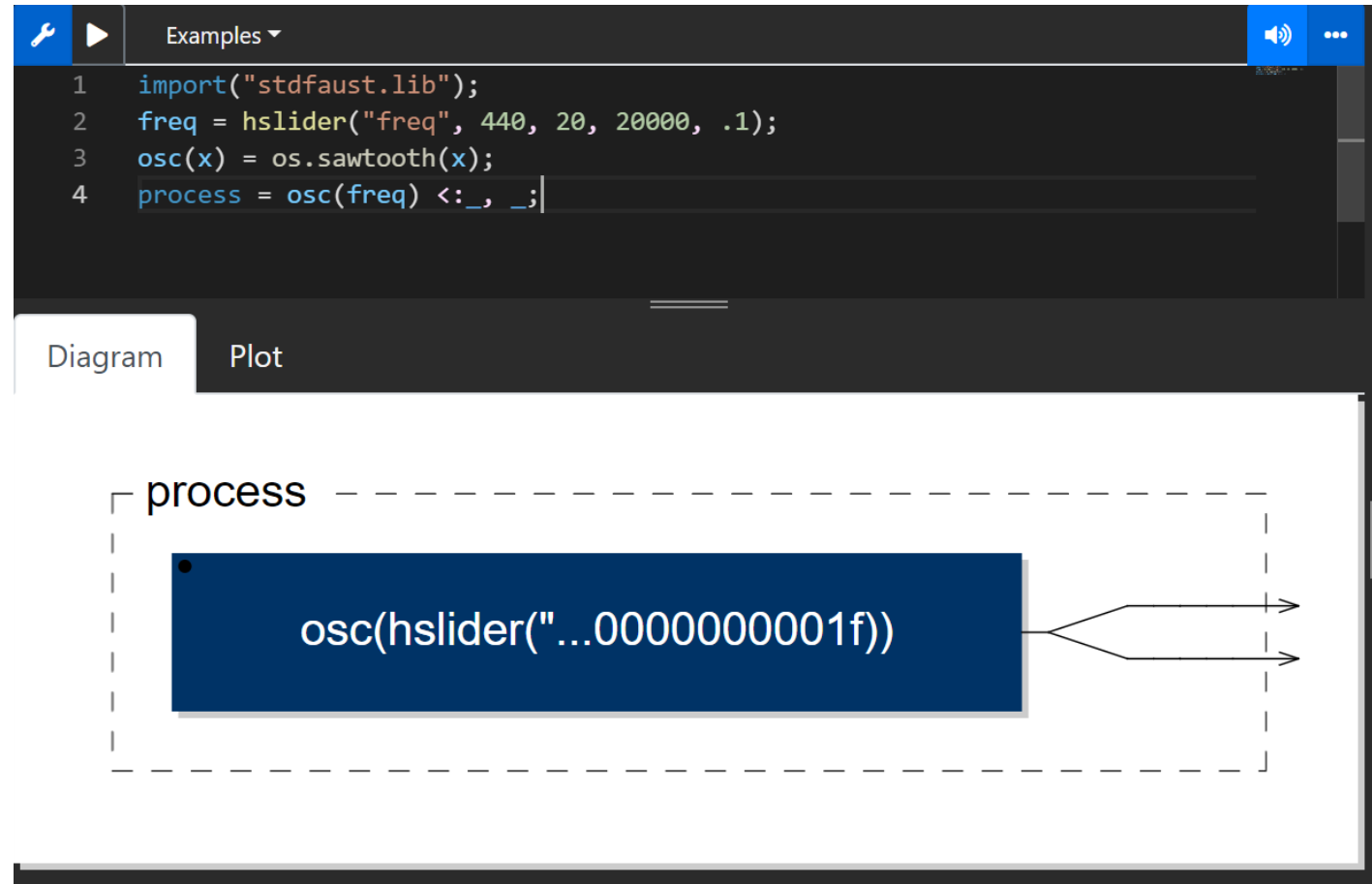


Image Source: aliexpress.com [9]

Faust allows for easy, low level DSP app creation

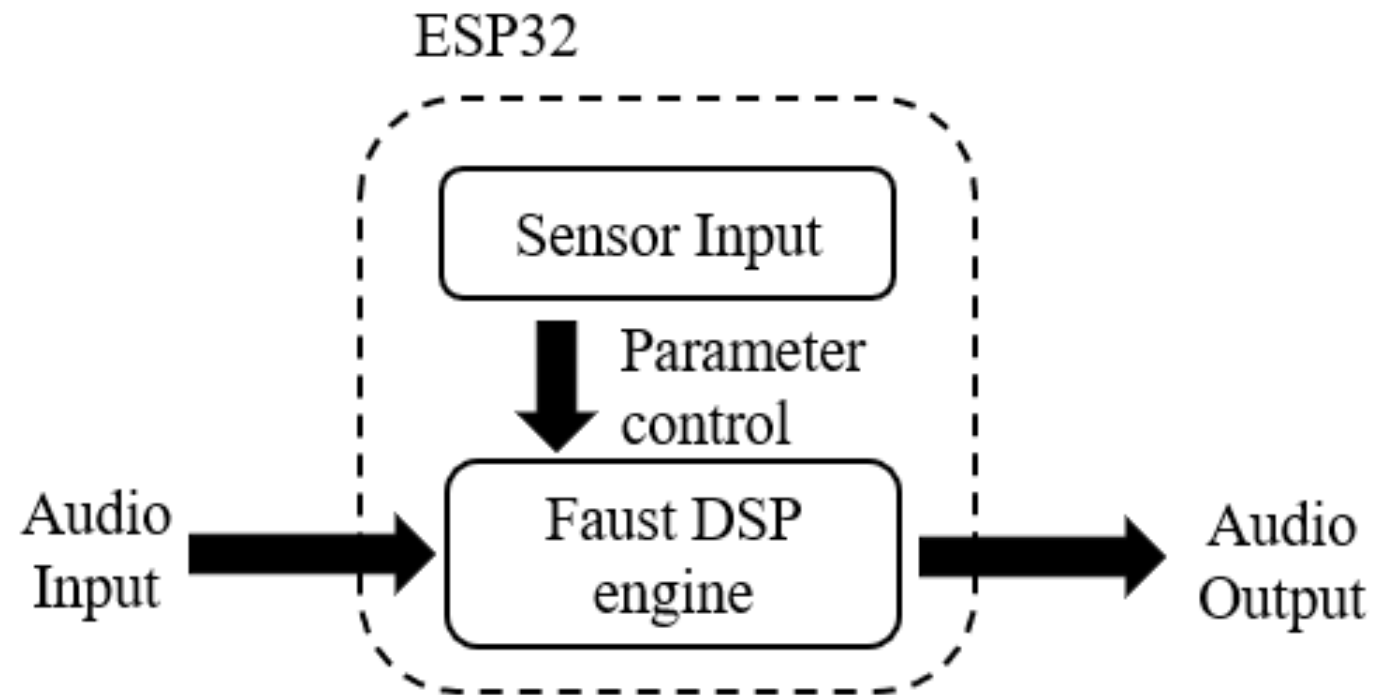


The screenshot displays the Faust IDE interface. The top section is a code editor with a dark background, showing the following code:

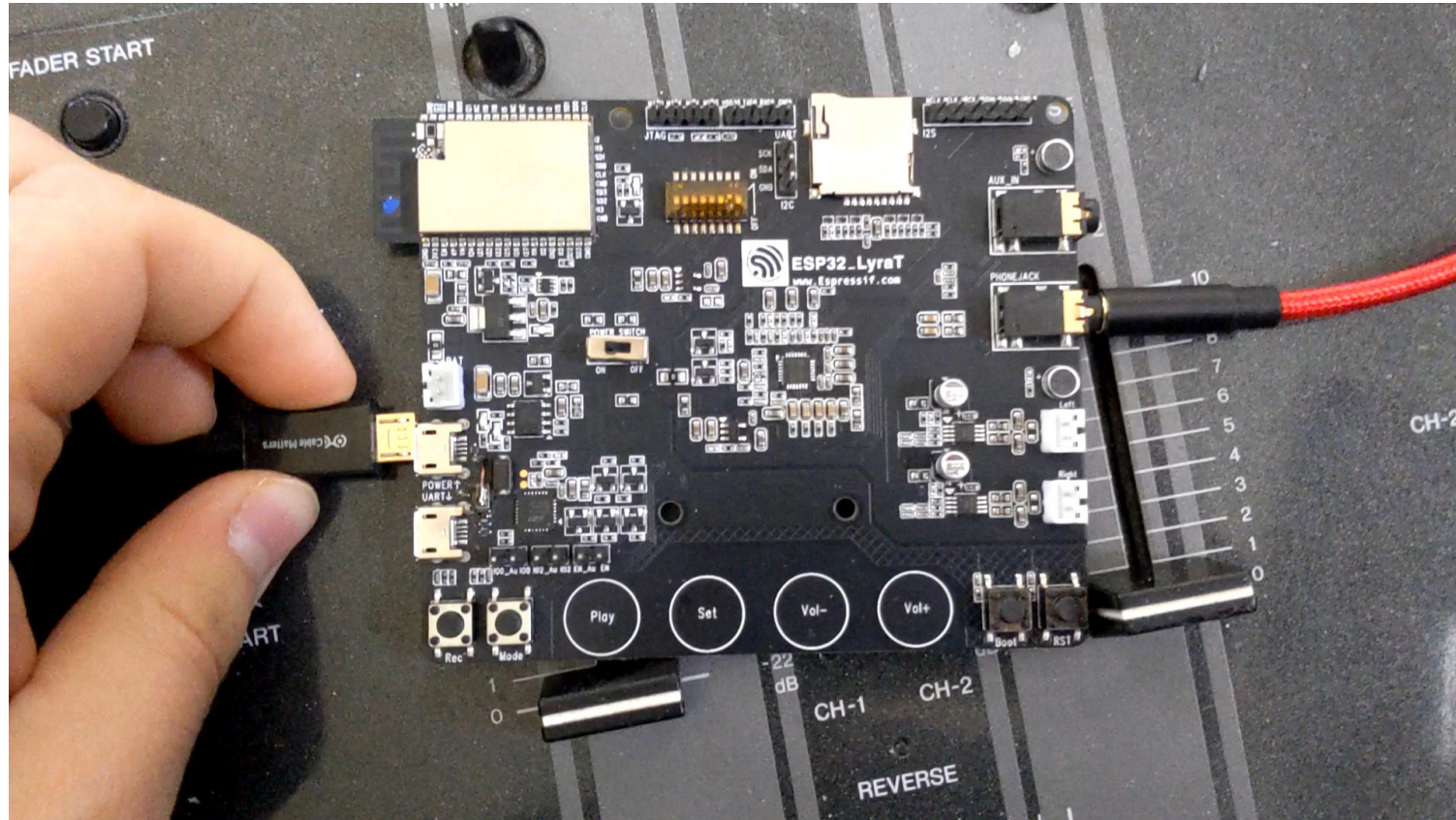
```
1 import("stdfaust.lib");  
2 freq = hslider("freq", 440, 20, 20000, .1);  
3 osc(x) = os.sawtooth(x);  
4 process = osc(freq) <:_, _;
```

Below the code editor, there are two tabs: "Diagram" and "Plot". The "Diagram" tab is currently selected, showing a block diagram. The diagram is enclosed in a dashed box labeled "process". Inside this box, there is a blue rectangular block containing the text "osc(hslider(...0000000001f))". To the right of this block, there are two parallel horizontal arrows pointing outwards, representing the output of the process.

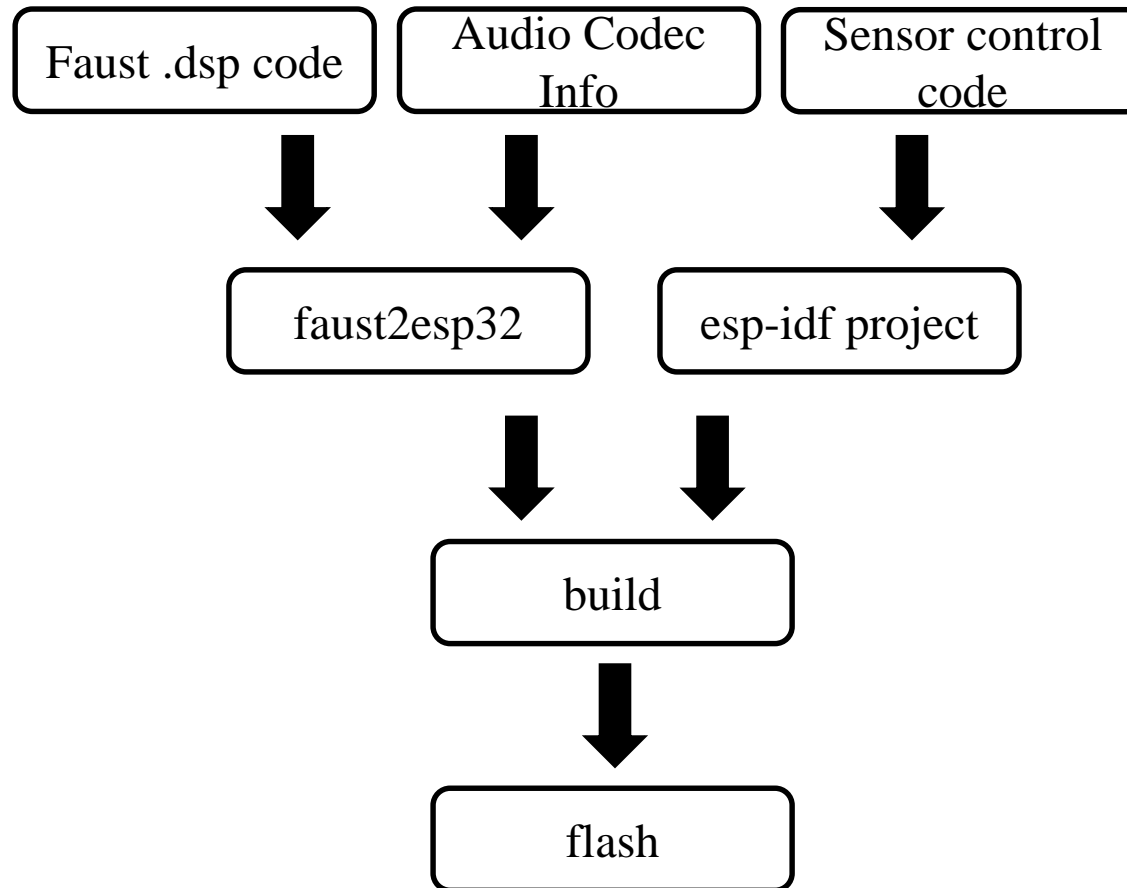
Faust can run in real-time on the ESP32



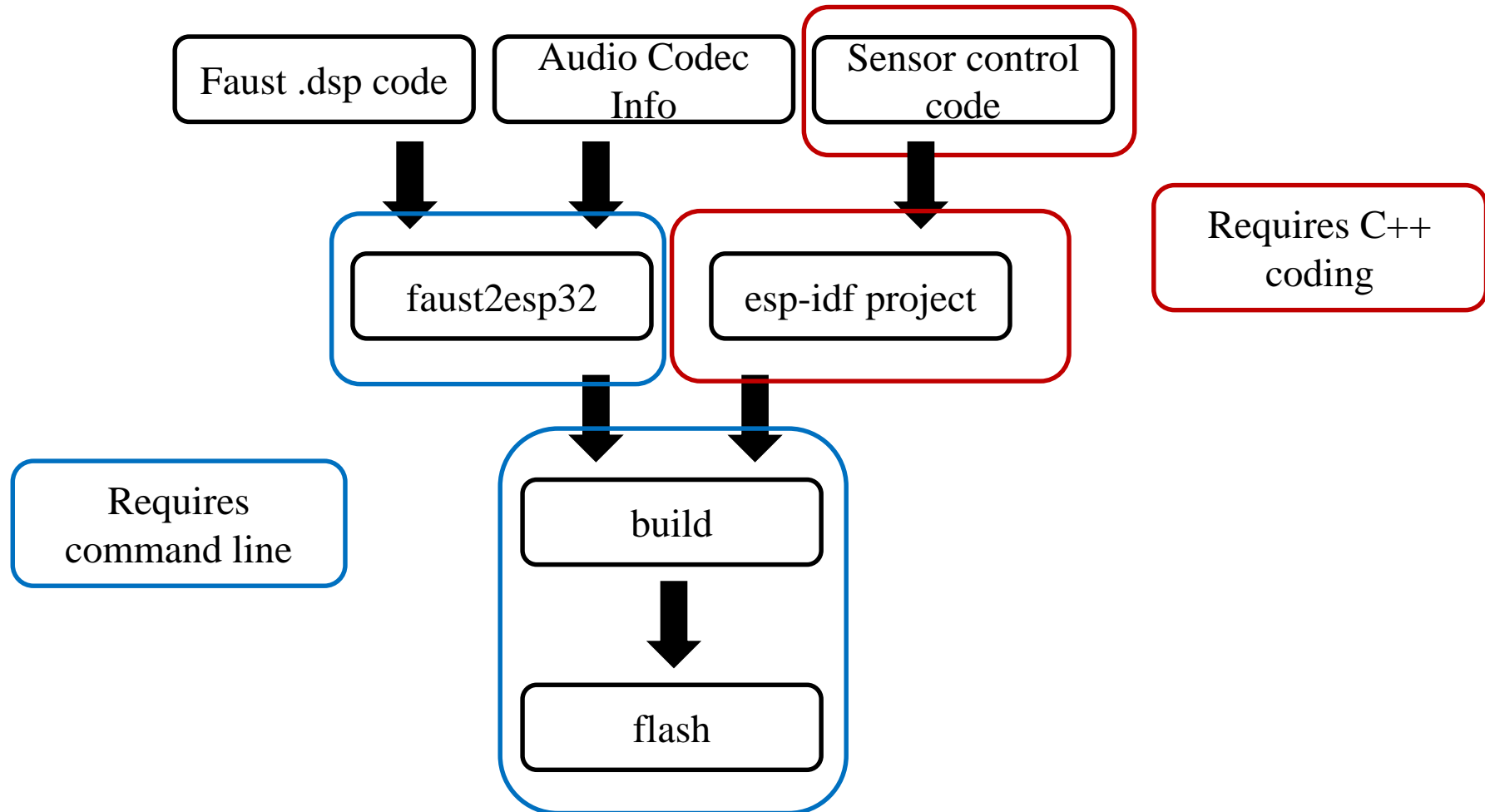
Faust can run in real-time on the ESP32



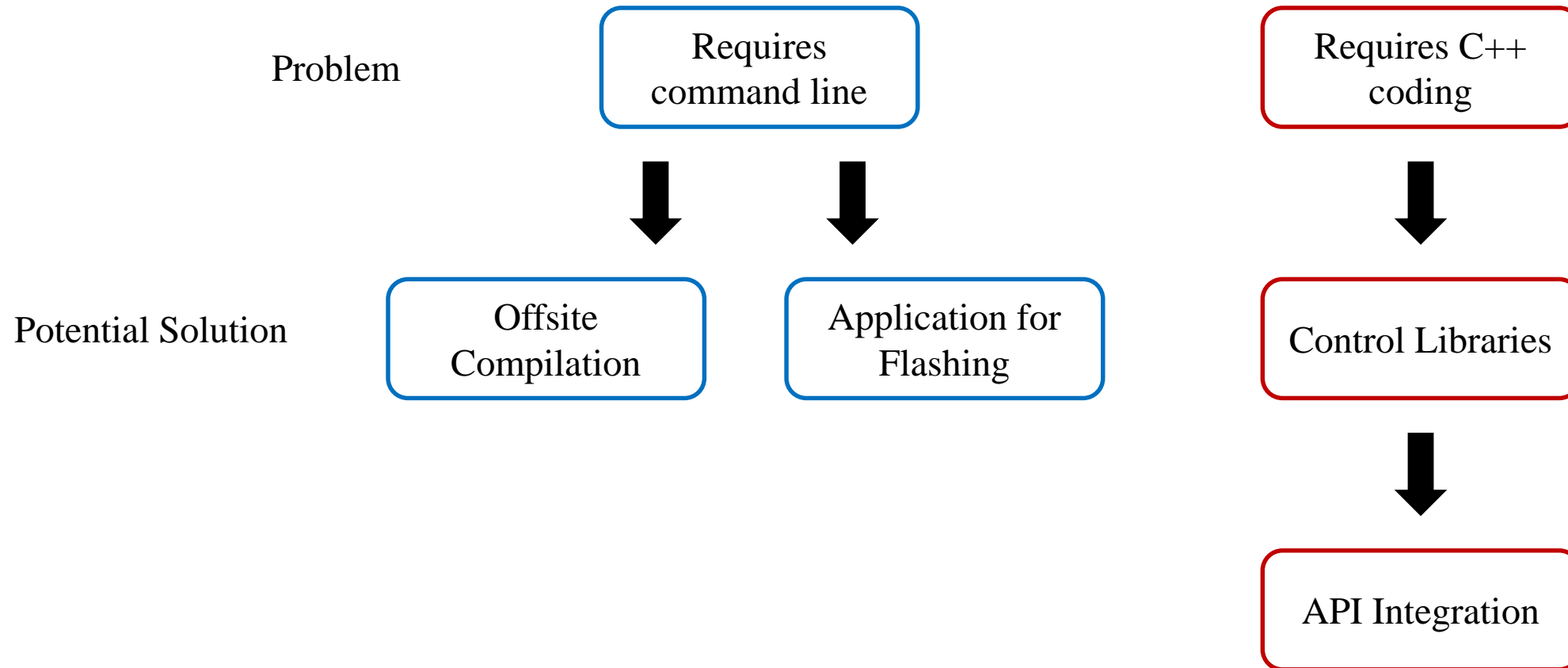
Typical Development is Tedious



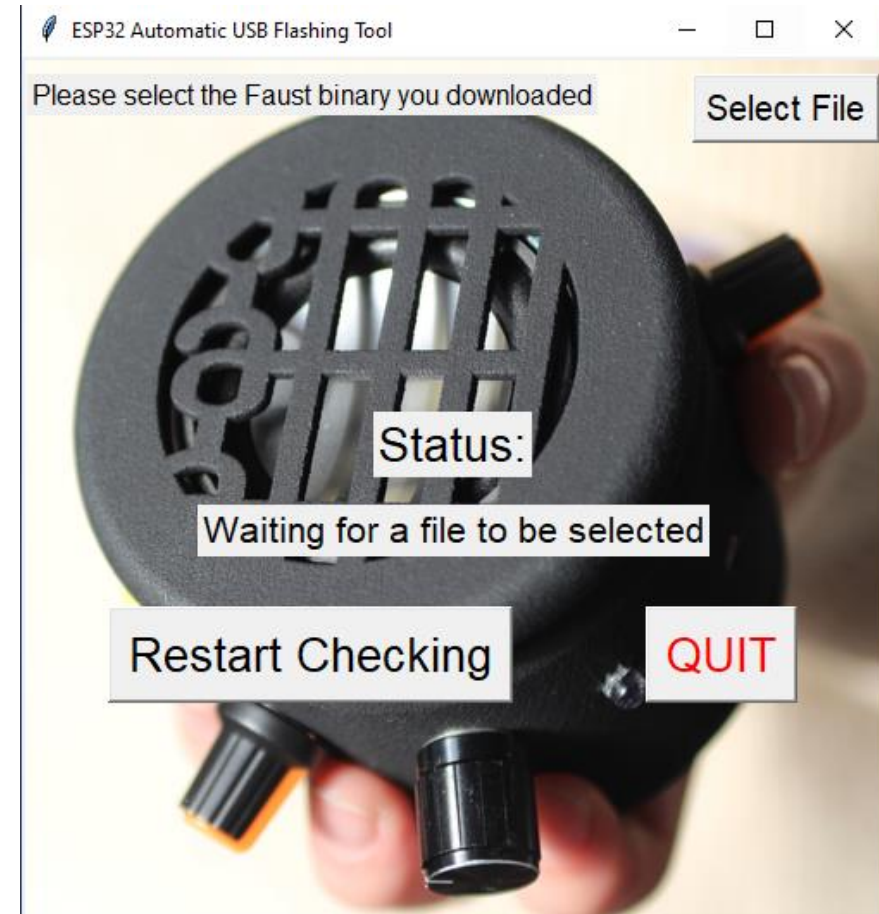
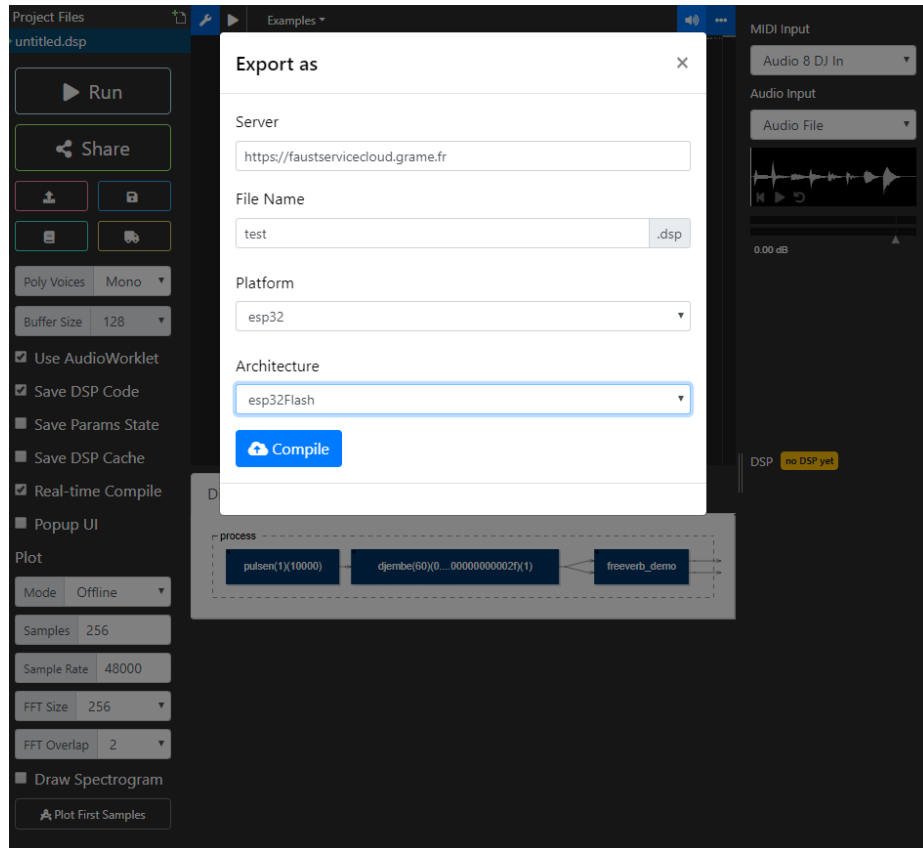
Typical Development is Tedious



Lowering the ESP32 Learning Curve



The command line can be removed, at a cost



A great deal of functionality has been added
in the past few months

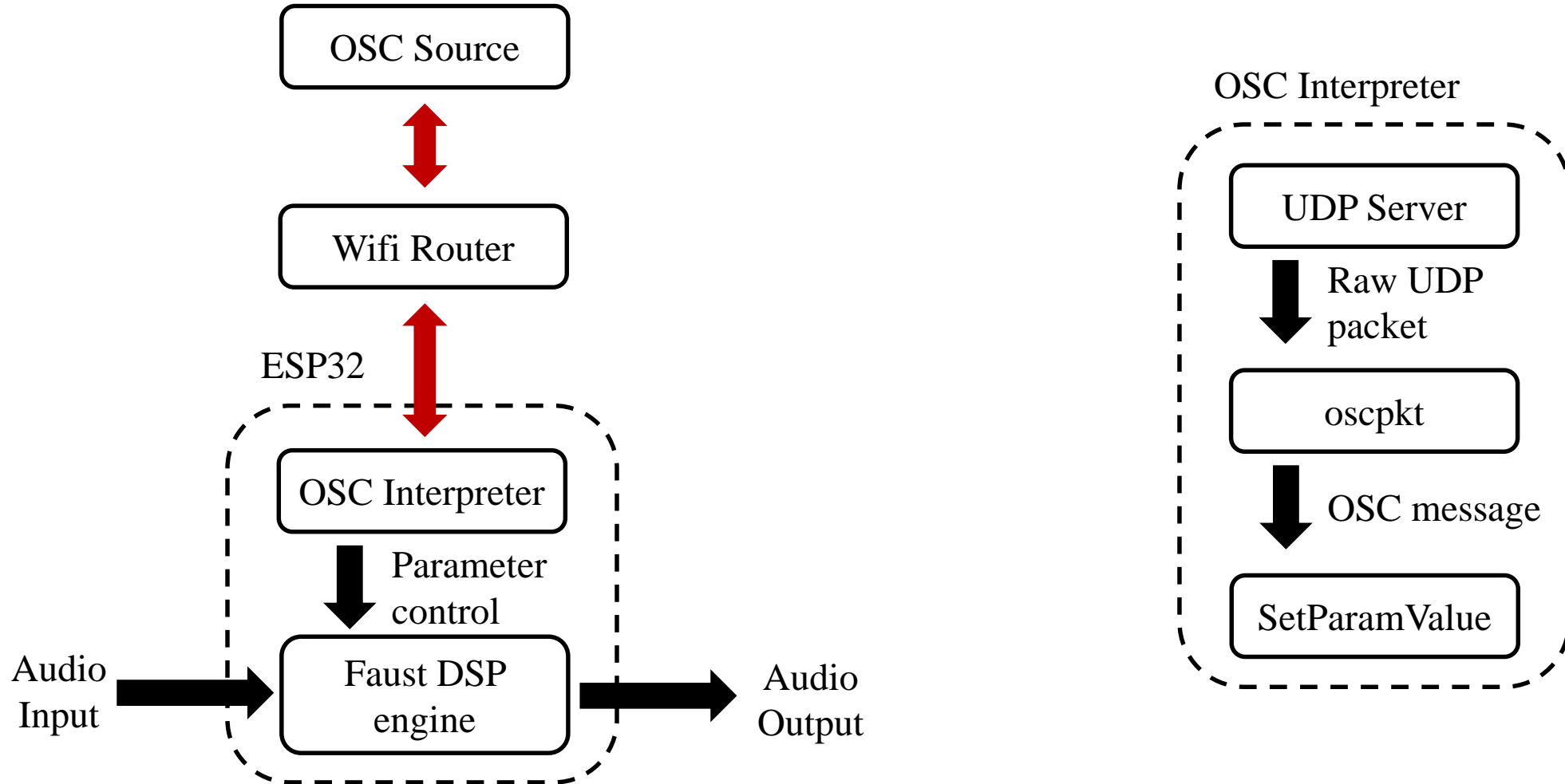
Midi & Polyphony

More Supported Audio Codecs

OSC over WiFi

Accelerometer & Gyroscope Support

How the OSC on the ESP32 Functions



OSC offers many interesting possibilities!

Pedal Controlled with TouchOSC

A Portable Organ for any Room

Future Research for OSC

Integration into the Faust API

Bidirectional Communication

Integrating with Sensor Readout

Multiple devices on the same network

Thank You!

Image Sources

- [1] - <https://www.raspberrypi.org/>
- [2] - <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>
- [3] - <https://bela.io/products/>
- [4] - <https://www.sparkfun.com/news/2055>
- [5] - <https://www.electro-smith.com/daisy/daisy>
- [6] - <https://www.cnx-software.com/2018/02/12/ttgo-taudio-v1-0-is-a-20-audio-board-with-esp32-wrover-module/>
- [7] - <https://www.cnx-software.com/2018/10/24/esp32-a1s-esp32-audio-development-kit/>
- [8] - <https://www.mouser.com/ProductDetail/Esspressif-Systems/ESP32-LyraT?qs=MLItCLRbWsxPzPCja546ZA==>
- [9] - <https://www.aliexpress.com/i/32846510254.html>
- [10] - <https://botland.com.pl/en/wi-fi-modules/7655-wifi-bluetooth-ble-esp-wroom-32d-smd.html>