Tufts UNIVERSITY

COMPUTER ASSISTED MUSICAL PERFORMANCE

LOCAL PROJECTS

Mathias Barth, Daniel Katz, Matthew Langen, Jack O'Shea, Patrick Zwierzynski

Goal

Make musical performances more accessible to people with no formal music background.

Instrument design went through several iterations:

- Guitar was initially envisioned as an air guitar with cameras and a computer vision (CV) program tracking the user's movements
- Adapted to user holding a brightly colored broomstick to improve the CV tracking
- Finally decided that capacitive sensors on a rod would be more accurate, efficient, and easier to implement

Components and Communications

Networking

- A Raspberry Pi broadcasts a dedicated Wifi network
- Open Sound Control (OSC) messages are sent between instruments and the host

Host Machine / "Langenizer"

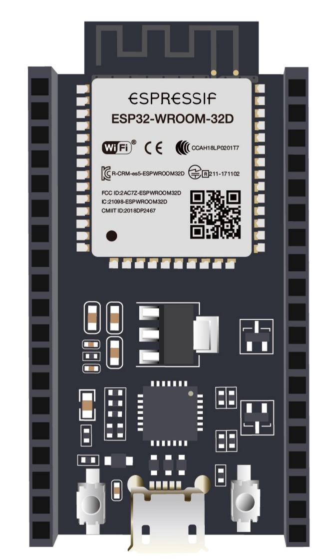
- Creates virtual MIDI ports for each ESP32 instrument
  - Creates illusion for the Digital Audio Workstation (DAW, we are using Ableton Live) that there actually is a guitar, bass, and drums connected
- Python program accepts OSC messages
- Records state of all instruments —> outputs MIDI through virtual ports that makes sense compositionally

Web Interface

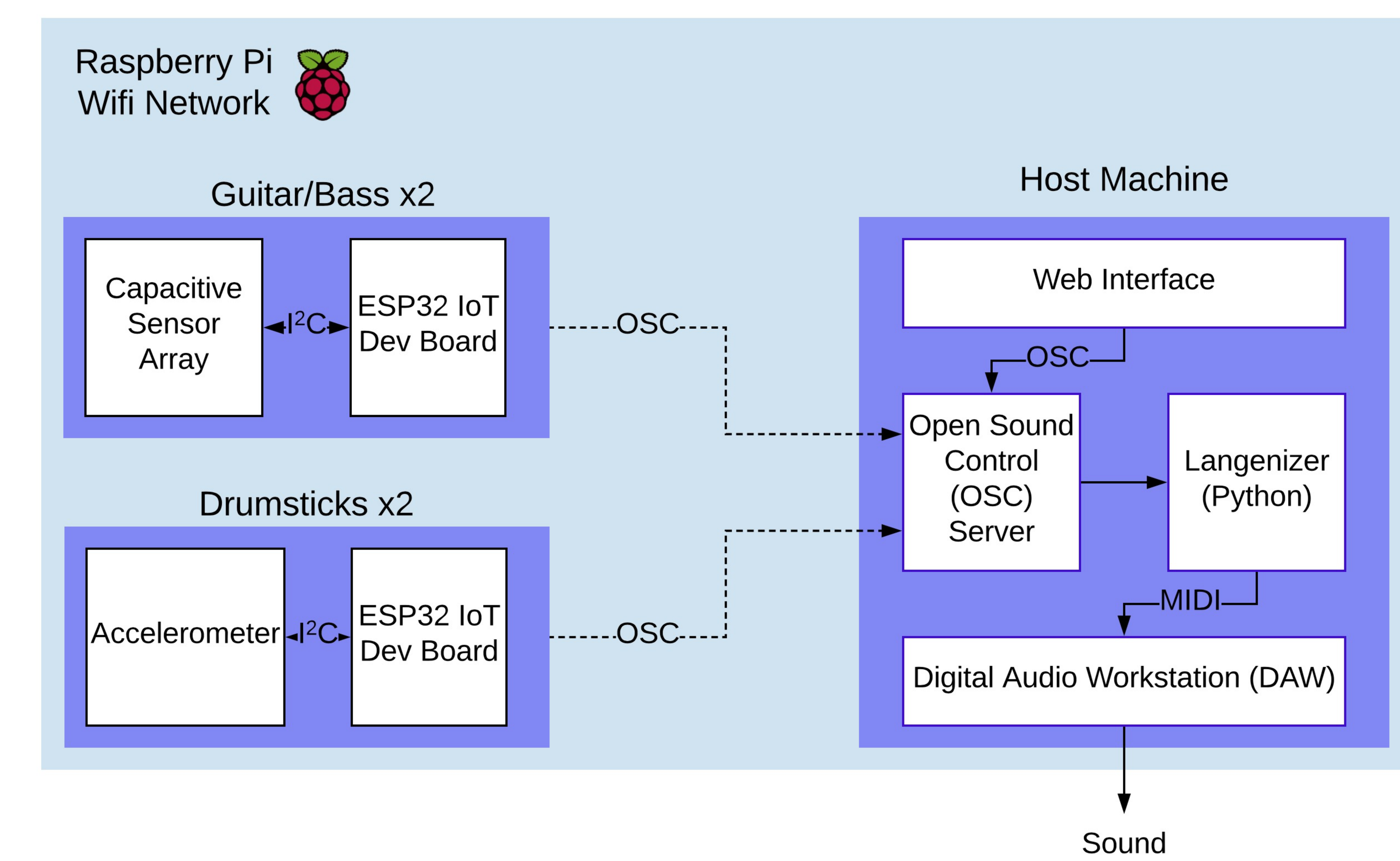
- Sends selected genre and key to Langenizer, updates instruments and notes accordingly

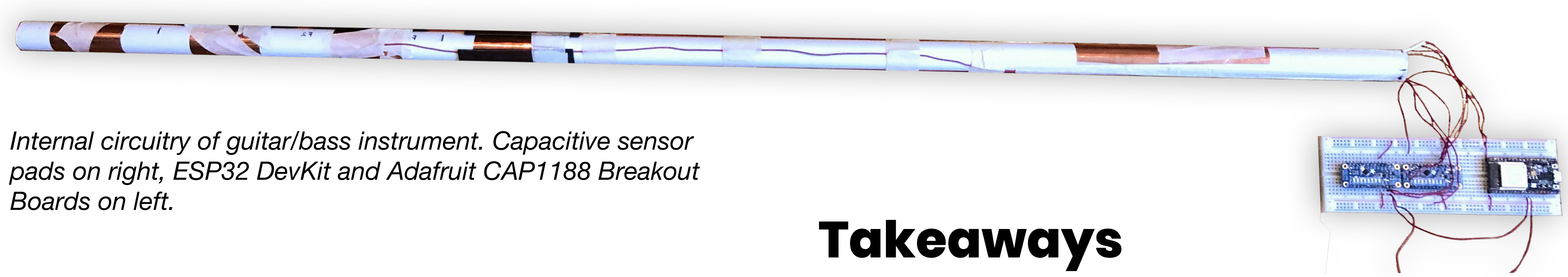
Instruments

Instruments powered by ESP32 IoT boards:



- Run Python subset called MicroPython
  - Used a fork called Loboris MicroPython for extra networking features
  - Thin documentation led to development delays
- Send Open Sound Control (OSC) messages to host via Wifi
- Interface with sensor circuits via I2C protocol





Internal circuitry of guitar/bass instrument. Capacitive sensor pads on right, ESP32 DevKit and Adafruit CAP1188 Breakout Boards on left.

Drumsticks

- Use 3-axis accelerometers
- When an accelerometer detects a spike:
  - Motors provide haptic feedback to the drummer
  - An OSC message is sent to the host

Guitar and Bass

- Use capacitive sensors — the same tech that powers touch screens — to detect touches on the frets and strumming
- OSC messages are sent when the user strums or changes frets
- Instead of ridges delineating frets, haptic feedback from a motor informs users when they change frets

manatee

The Manatee Design Team wishes to thank our project sponsor, Sundar Raman, as well as professors Ming Chow, Sammi Durrani, James Intriligator, and Ronald Lasser

Takeaways

- Thinly documented MicroPython slowed development — even more thinly documented Loboris fork threw a wrench in development:
  - Switched for extra networking features, then the I2C inexplicably stopped working
  - Readings were now little-endian instead of big-endian. This change was not documented and difficult to diagnose
- Achieving quality and consistent sensor data outside of controlled settings can be extremely difficult
  - The capacitive sensors were particular difficult to get right — our hats are off to the engineers at Samsung and Apple
  - Long wires running the length of the instrument
- A number of "assistive" performance features were tested, such as fixing rhythm or adding drum patterns
  - Most of these were foregone — turns out the more you "assist" users, the less they feel they are in control and the less satisfying the overall experience becomes