Untouchable LIMiTS

Introduction

Looper Instrument & Metronome inside Theremin Synthesizer is a musical instrument that combines the functionality of a synthesizer, an optical-sensor-based theremin, and a loop station, all processed on an ESP32-S3 microcontroller.

- **What it does:** It generates real-time audio waveforms (sine, square, triangle, sawtooth) that the user can modulate. The pitch (frequency) is controlled either “in the air” by moving a hand over a ToF laser sensor, or via a DIY capacitive touch keyboard. The device allows recording a multi-bar audio sequence and playing it back in a continuous loop, offering the ability to play “over” your own recording (overdubbing).
- **The starting idea:** The desire to build an electronic instrument that does not rely on expensive physical keyboards, while integrating a “loop pedal” directly into the internal memory to be able to have a jam session with yourself.

General description

The project is structured around the **ESP32-S3-WROOM-1 (N16R8)** microcontroller, which runs the audio synthesis on one processor core and the user interface/metronome on the other core, ensuring zero-latency performance.

System Architecture (Main Blocks):

- **Input & Navigation Module:** An I2C OLED display and an EC11 rotary encoder form the graphical user interface for selecting waveforms and ADSR envelope settings.
- **Sensory Module (The Instrument):**
  - A **VL53L1X** Time-of-Flight sensor communicates via I2C to measure hand distance with millimeter precision, mapping the data to a musical scale.
  - A capacitive keyboard (aluminum foil strips) connects directly to the ESP32's internal hardware touch pins.
- **Looping:** Tactile push buttons trigger audio recording directly into the 8MB PSRAM.
- **Metronome:** An independent active buzzer with volume and bpm configurable. Being separate from the sound generating loop it won't interfere with the recording.
- **Audio Output Module (The “Hacky” DAC):** The digital signal (PDM/PWM) generated by the ESP32 passes through a 1st-order hardware Low-Pass RC filter for smoothing. It is then amplified by an **LM386** module and routed directly to a pair of wired headphones via a modified 3.5mm jack splitter.

Hardware Design

Bill of Materials (BoM):

- **1x ESP32-S3-DevKitC-1 (N16R8):** The main microcontroller (16MB Flash, 8MB PSRAM for the audio buffer).
- **1x VL53L1X Time-of-Flight Sensor:** For the Theremin functionality (I2C).
- **1x 0.96” OLED Display:** For the system menu (I2C).
- **1x EC11 Rotary Encoder:** With built-in push button, for menu navigation.
- **1x LM386 Audio Amplifier Module:** To amplify the filtered signal for headphones.
- **3x 10k Linear Potentiometers (B10K):** For dynamic volume control and LFO/filter adjustments.
- **1x Passive Buzzer:** For the integrated dual-core metronome.
- **3-4x Tactile Push Buttons (Momentary):** 12x12mm, for loop triggering and UI.
- **Passive Components (RC Filter & Decoupling):**
  - 1x `104` Ceramic Capacitor (0.1µF)
  - 1x `475` Ceramic Capacitor (4.7µF) - For power supply stabilization on the breadboard.
  - Resistors (220Ω - 330Ω for the audio filter, 1kΩ for the buzzer).
- **1x 3.5mm Audio Jack Splitter Cable:** to connect headphones to the breadboard without destroying the original headphone plug.
- **DIY Materials:** Aluminum foil and cardboard for the capacitive keyboard.
- **Miscellaneous:** Breadboards (minimum 2 tied together), Dupont jumper wires (M-M, M-F)

Block Schema

Software Design

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării şi surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi şi structuri pe care plănuiţi să le implementaţi
- (etapa 3) surse şi funcţii implementate

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Export to PDF