There-Mini

Introducere

Proiectul meu constă într-un sintetizator embedded, controlat printr-un senzor de proximitate. Sintetizatorul va porni de la un număr de arpegiatoare, care funcționează ca preset-uri audio. Pornind de la aceste preset-uri, utilizatorul poate începe să modifice sunetul, folosindu-și mâna. Utilizatorul ciclează prin mai mulți parametri configurabili, apăsând pe niște butoane, și modifică acel parametru prin intermediul senzorului de proximitate.

Spre exemplu, să zicem că sistemul se află în starea în care parametrul ce poate fi modificat este volumul. Dacă utilizatorul apropie degetul/mâna de senzor, volumul crește, iar dacă îl depărtează, volumul scade.

Parametri configurabili ai sintetizatorului sunt:

- volum
- frecvență sonoră (pitch)
- viteză
- efecte (delay, phaser, flanger, reverb etc.)

Scopul sintetizatorului este de a îmi oferi oportunitatea să experimentez cu un proiect de inginerie de sunet. Am vrut să creez un fel de sandbox, într-un mediu embedded, pentru parametrizarea și modificarea diferitelor aspecte ale undelor sonore.

Conceptual, am pornit de la ideea tereminului (poză mai jos), care, funcțional, se comportă similar cu proiectul meu. Structural însă, tereminul emite sunet prin interferența cu câmpul electromagnetic, fenomen pur analogic. Proiectul meu este strict digital.

Proiectul mi se pare util din două motive principale:

1. Este o “jucărie” foarte fun ce oferă oportunitatea de a crea o infinitate de sunete.
2. Nu am observat să se pună accentul, până acum, la facultate pe concepte tangente ingineriei de sunet, așa că am vrut să experimentez în domeniu pe cont propriu.

Descriere generală

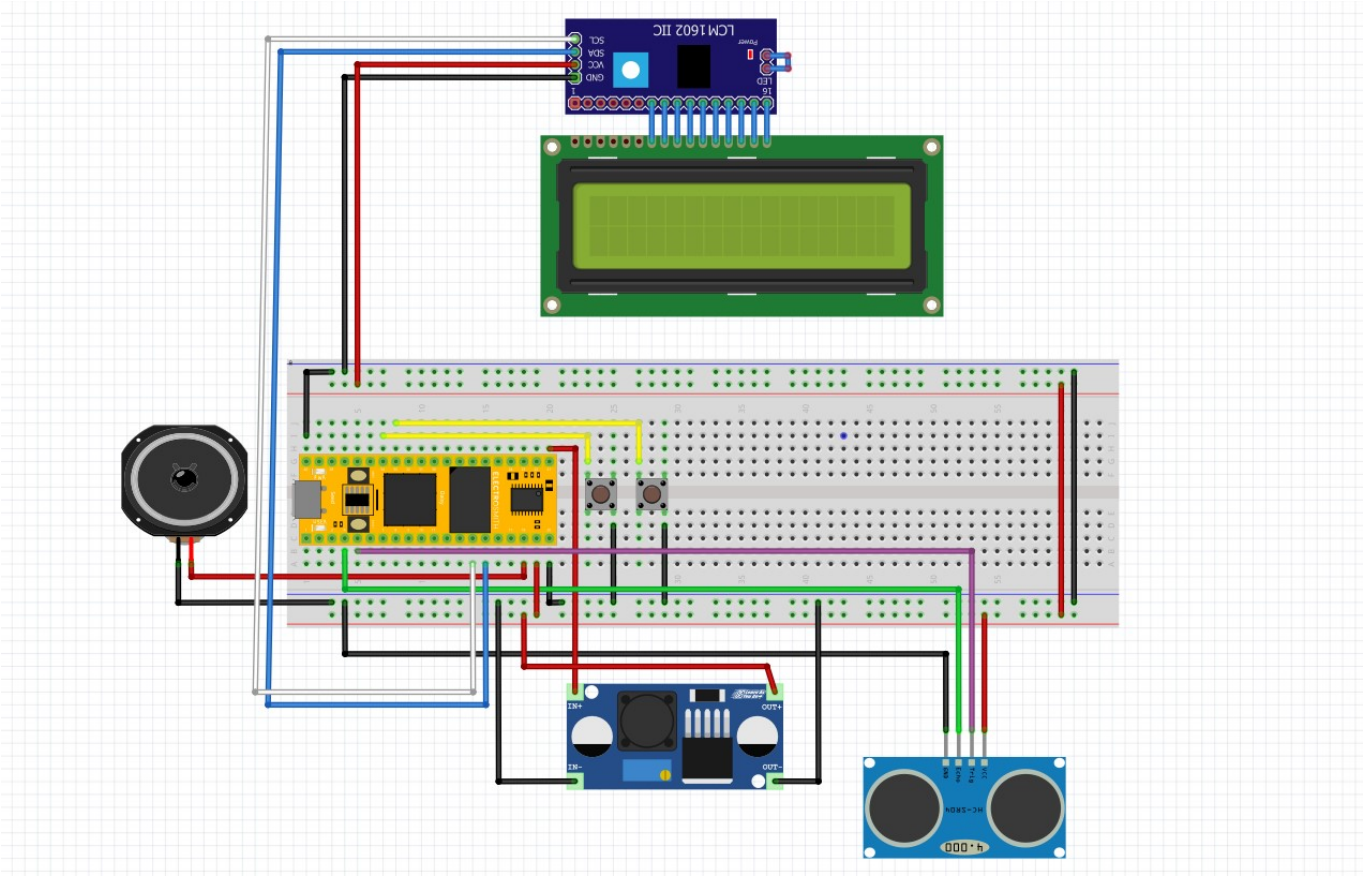
Arhitectura sistemului este destul de simplă. Un microcontroller specializat pentru a crea sintetizatoare preia date de la un senzor de proximitate și un set de butoane tactile și redă sunet către un difuzor. De asemenea, starea programului este afișată pe un display LCD, indicând utilizatorului parametrul curent.

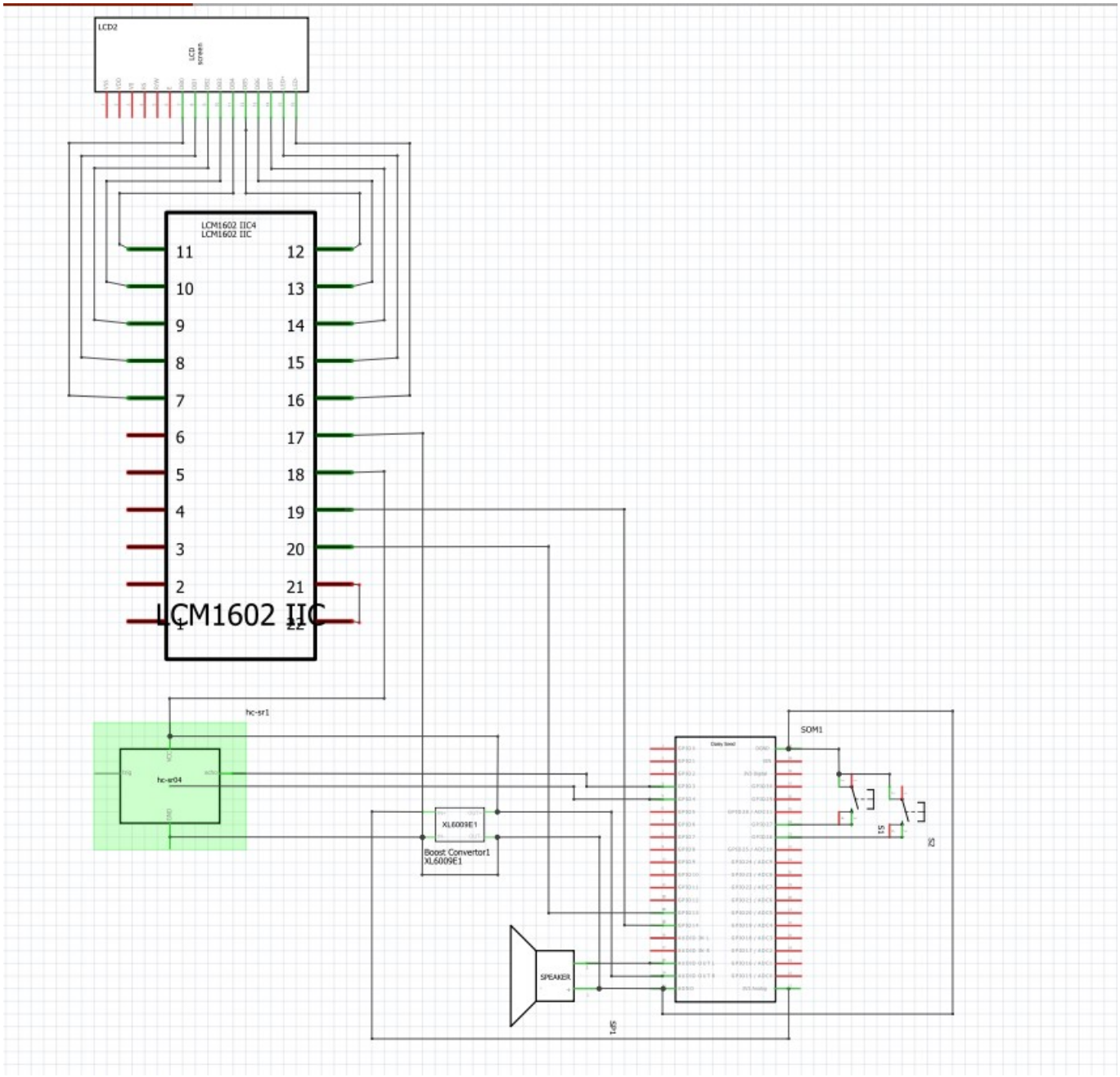
Hardware Design

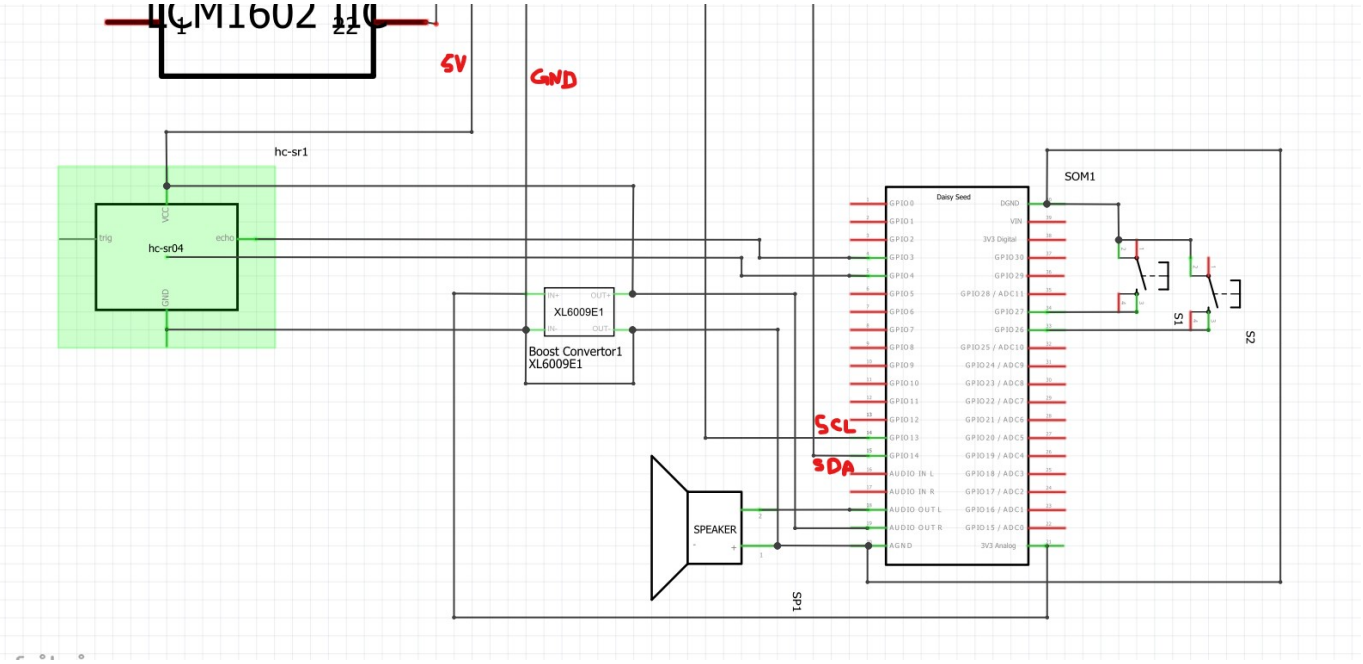
Listă Componente

Componentă	Link
Daisy Seed Microcontroller	https://www.modularsquare.com/shop/electrosmith/daisy-seed-2/
Breadboard	https://cleste.ro/breadboard-830-puncte-mb-102-mb102.html
Set Jumper Wires	https://cleste.ro/set-140-fire-jumper.html
Display LCD 1602 I2C	https://cleste.ro/ecran-lcd-1602-iic-i2c.html
Set Rezistențe	https://cleste.ro/set-rezistene-100buc-e4-3.html
Buton Tactil x 4	https://cleste.ro/butoane-tactile-6x6x5mm.html
Senzor Ultrasonic	https://cleste.ro/senzor-ultrasonic-hc-sr04.html
Speaker 3W	https://cleste.ro/visaton-2916-speaker-mini-5-cm-3-w-50-ohm-180-hz-to-17000-hz.html
Boost Converter XL6009	https://cleste.ro/modul-ridicare-tensiune-xl6009.html?gad_source=1

Schemă Electrică


LCM1602 IIC
GND
VCC
SDA
SCL
LED
IN+
IN-
+OUT
-OUT


LCD2
LCD screen
LCM1602 IIC4
LCM1602 IIC
11
12
10
13
9
14
8
15
7
16
6
17
5
18
4
19
3
20
2
21
1
22
LCM1602 IIC
hc-sr1
hc-sr04
XL6009E1
Boost Convertor1
XL6009E1
SOM1
SPEAKER



Notă: Deoarece schematicul full e destul de blurat, am atașat o poză cu zoom pe conexiunile către uC. Am scris de mână ce reprezintă conexiunile care ies din cadrul schematicului mărit. Interfațarea dintre adaptorul I2C și ecranul LCD este 1:1.

Montaj

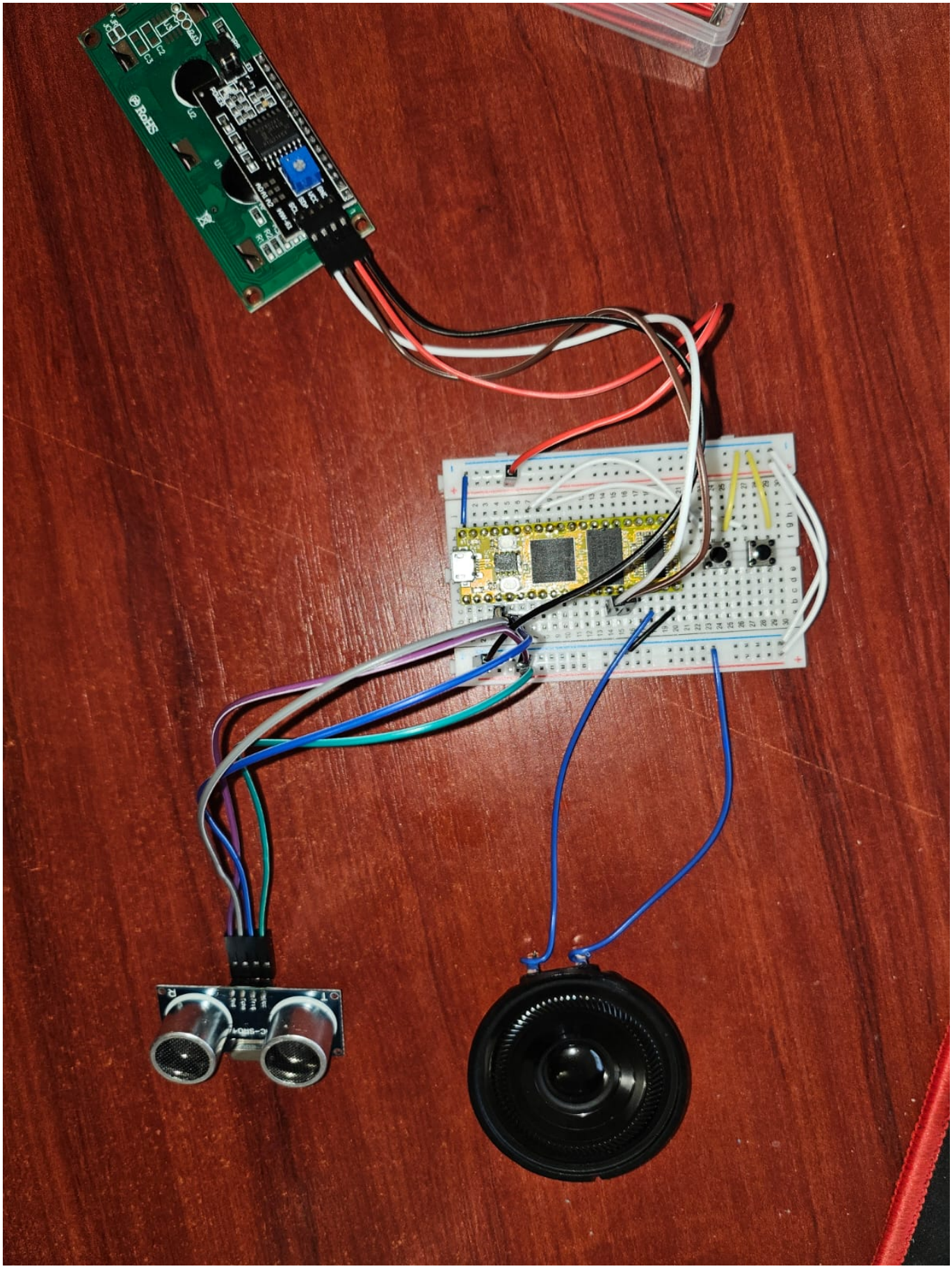
Am conectat toate ground-urile între ele.

Pentru HC-SR04, pin-ul de trigger l-am conectat la pinul D4 de pe uC, iar cel de echo la D3. Am ales acești 2 pini din două motive: sunt toleranți la 5V și nu au alte funcții de comunicare.

Pentru display-ul 1602, am conectat pinii de SCL și SDA la pinii D13 și D14 de pe Daisy Seed. Acești pini au funcția suplimentară de SCL, respectiv SDA pentru comunicarea I2C cu display-ul.

Pentru speaker, terminalul negativ s-a dus la ground, iar cel pozitiv în out[0][SIZE] de pe Daisy, un pin special pentru output audio.

Butoanele le-am conectat la 2 pini ce nu aveau funcții suplimentare pe Seed, D26 și D27. Astfel pot să înregistrez triggere de pe acei pini fără să interferez cu alte componente.



Software Design

Mediu de dezvoltare: Arduino IDE

Biblioteci 3rd party: DaisyDuino

Sursa:

demo.ino

```
#include "DaisyDuino.h"

DaisyHardware hw;
static Oscillator osc;

size_t num_channels;

const int trigPin = D4;
const int echoPin = D3;
const int baud = 9600;

int distance;

float pitch, amp = 100;

Switch pwrButton, effectButton;

int on = 0, effect = 0;

void MyCallback(float **in, float **out, size_t size) {
  float sine_signal;

  pwrButton.Debounce();
  effectButton.Debounce();

  osc.SetFreq(pitch);
  osc.SetAmp(amp * on);
    for (size_t i = 0; i < size; i++) {
      sine_signal = osc.Process();
      out[0][i] = sine_signal;
      out[1][i] = sine_signal;
    }
}

void setup() {

  pwrButton.Init(1000, true, 26, INPUT_PULLUP);
  effectButton.Init(1000, true, 27, INPUT_PULLUP);

  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
  Serial.begin(baud);

  float sample_rate;

  hw = DAISY.init(DAISY_SEED, AUDIO_SR_48K);
  num_channels = hw.num_channels;
  sample_rate = DAISY.get_samplerate();
  osc.Init(sample_rate);

  osc.SetWaveform(osc.WAVE_SIN);
  osc.SetFreq(440);
  osc.SetAmp(1);

  DAISY.begin(MyCallback);
}

void loop()
{
  pwrButton.Debounce();
  effectButton.Debounce();

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  distance = pulseIn(echoPin, HIGH);
  delay(50);

  if (effect == 0) {
    pitch = min((220.0 * ((float) distance / 1200)) + 220.0, 440.0);
  } else if (effect == 1) {
    amp = min((float) distance / 1200, 100.0f);
  }

  if (pwrButton.Pressed()) {
    on = !on;
    Serial.println(on ? "Power: ON" : "Power: OFF");
  }
  if (effectButton.Pressed()) {
    effect = !effect;
    Serial.print("Effect: ");
    Serial.println((!effect) ? "Pitch" : "Amp");
  }
}
```

Concluzii

Am rămas cu două mari lecții după finalizarea proiectului:

1. Data viitoare, pentru un proiect de genul, ar trebui să folosesc un uC cu mai multă documentație / mai multe exemple disponibile pe internet. Pe lângă că a fost imposibil să găsesc o bibliotecă pentru ecranul LCD pe I2C care să compileze pe ARM, nici nu am găsit exemple bune pentru lucruri precum întreruperi, timere sau ADC.
2. Trebuie să mă apuc mai din timp pentru un proiect embedded. Toate componentele hardware de care am avut nevoie le-am obținut abia vinerea dinaintea PM Fair, ceea ce a dus la un weekend plin de research / implementare software.

Download

- There-Mini

Jurnal

19.05: Senzorul și LCD-ul se alimentează la 5V, iar plăcuța are output de 3.3V. Am vorbit cu un coleg care are un boost converter, mâine îl atașez și testez componentele. Butoanele și difuzorul merg cum trebuie, testat. 19.05: Am realizat că semnalul difuzat de pinul echo al lui HC-SR04 este de fapt un semnal digital, așa că nu o să am nevoie de un pin de ADC.

Bibliografie/Resurse

1. https://github.com/electro-smith

Export to PDF