

Videos

Sesión 1

- concierto solenoide:
https://www.youtube.com/watch?v=g_hiz-Kx0kM&list=PL341E603FB13FE2D3&ab_channel=reduzent
- Processing & Kinect Sensor: Finger Tracking + OSCP5 + Pure Data + Arp2600
https://www.youtube.com/watch?v=X5WGyJPS_5s
- Experimento de resonancia acustica:
https://www.youtube.com/watch?v=vvJAgUBF4w&ab_channel=brusspup
- Graffiti Laser: <http://www.graffitiresearchlab.com/blog/projects/laser-tag/>
- Delicate boundaries: <http://csugrue.com/delicateboundaries/>
- Licuadora que funciona con gruñidos: <https://youtu.be/6DDkwdPaYmk>
- Almacenador de gritos: <https://youtu.be/Ta7rN5TeKzw>
- Drawdio: <https://www.youtube.com/watch?v=Ein9asQgfB8> y <https://www.youtube.com/watch?v=HYg8iycYZNs&t=2s>
- sensor pez: <https://twitter.com/unloquer/status/1099095464291508224>
- planta feliz: <https://www.youtube.com/shorts/YAsNlonSBac>
- juego de cubos en una caja : <https://www.youtube.com/shorts/YZ7aAG4DWZk>
- pelotica: <https://www.youtube.com/shorts/9eLff37aS4U>
- pez automata: <https://youtu.be/HH9IDf5W-gU>
- concierto de 8bits: <https://youtu.be/nE3JVpOwWuU>
- Lista de reproduccion:
<https://www.youtube.com/playlist?list=PL36dJutVa9QZj95urcHqORXI1mH-VnsEN>
- galería de algunos sensores: <https://twitter.com/jero98772/status/1383031921484165120>
- multimedia de jugetes: https://wiki.unloquer.org/personas/jero98772/taller_explora/multimedia
- Taller de aire y vestuario: <https://www.flickr.com/photos/37012247@N06/49257858968/>

Sesión 2

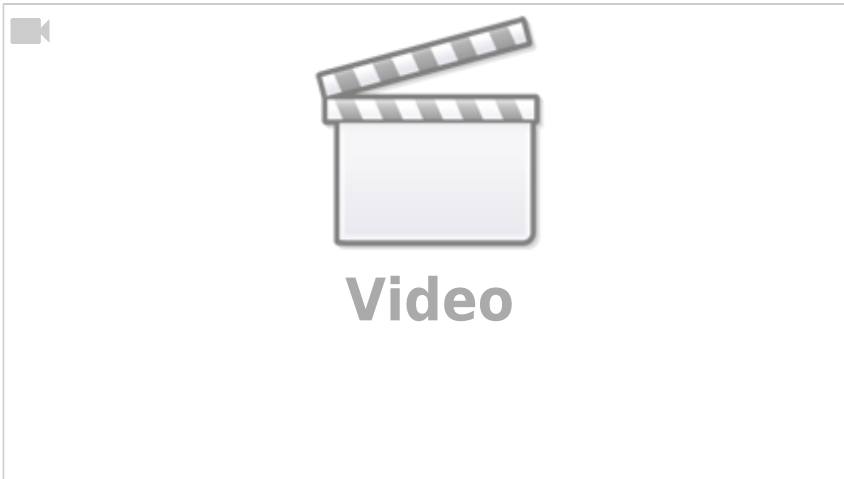
Para generar ideas:

Sensores de calidad del aire y weareables: <https://youtu.be/8ZFxrDkVOFk>

Sesión 3

Para crear los conceptos:

Comprender el mundo a través de los datos:



¿Para que sirven los datos?

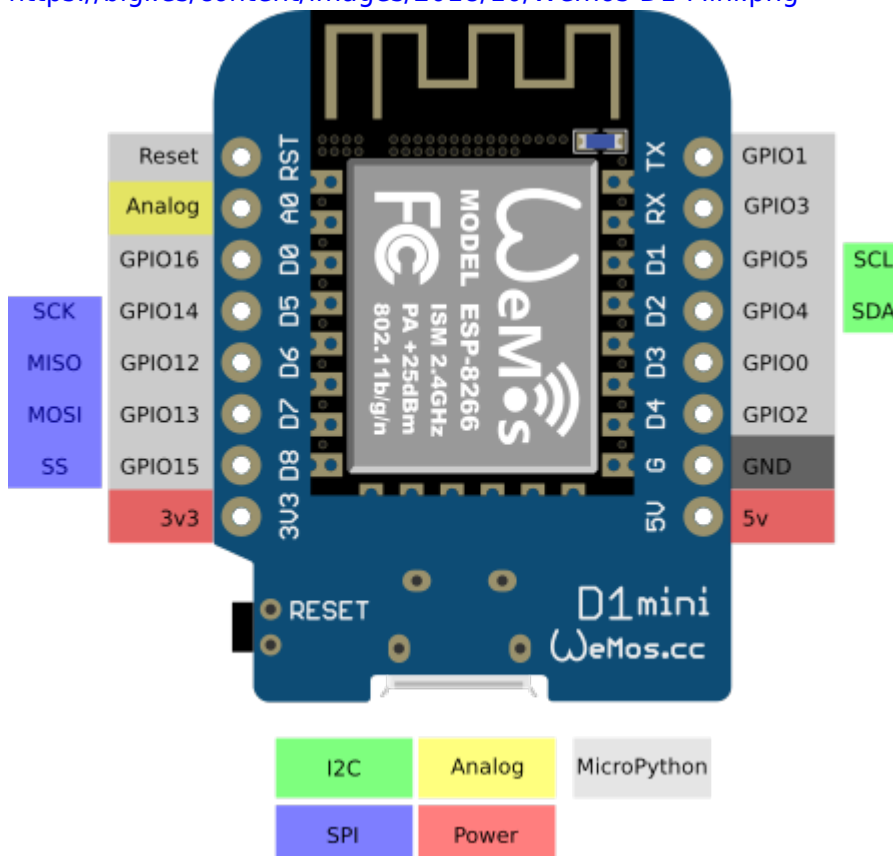
<https://www.behance.net/gallery/82992729/Dear-Data-Postcard>

http://www.visualcomplexity.com/vc/project_details.cfm?id=666&index=27&domain=Music

Componentes

ESP 8266

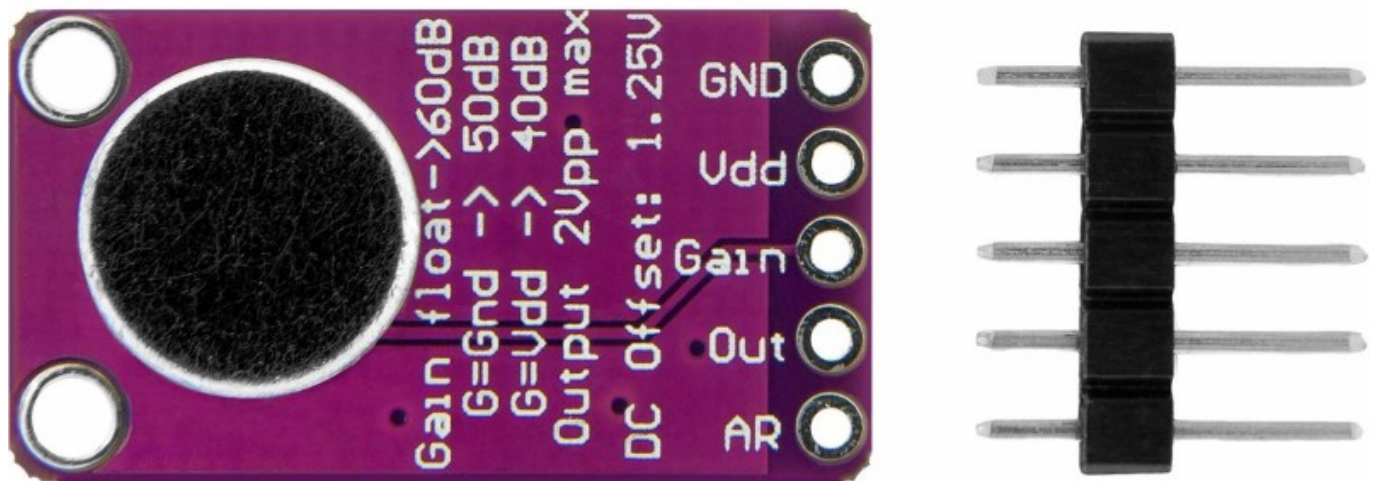
<https://bigl.es/content/images/2018/10/Wemos-D1-Mini.png>



Micrófono

Info del Micrófono

https://maxelectronica.cl/3891-thickbox_default/modulo-max9814-sensor-de-sonido-microfono-electret-20-20khz.jpg

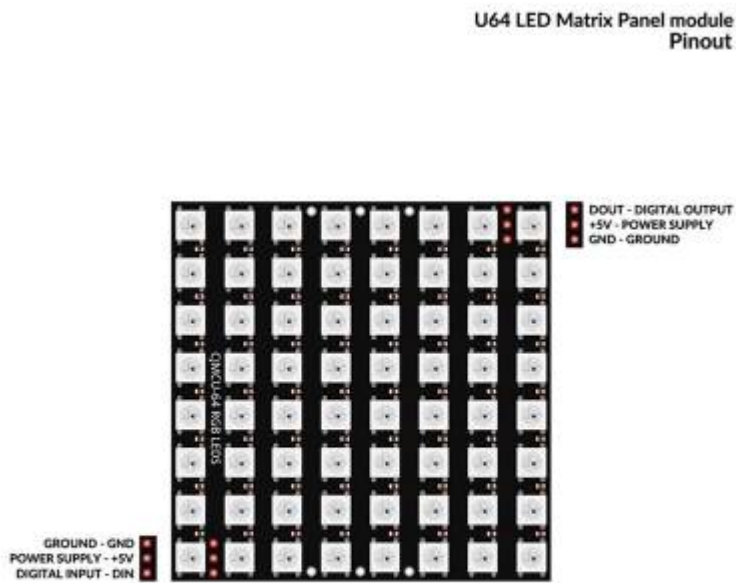


Amplificador

<https://www.maximintegrated.com/en/products/analog/audio/MAX9814.html>

Matriz de leds

<https://img.bestdealplus.com/ae04/kf/H0a03f4ded2694b5f8ec2324c68521a6b5.jpg>



Aplicación Android

<https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinoroid2&hl=es&gl=CO>

Entorno para desktop

Windows: <https://www.arduino.cc/en/software>

Github y Código

Repositorios de github: <https://github.com/unloquer/ETSesnor>

Repositorio del código sesión 2: intensidad

<https://github.com/unloquer/ETSesnor/blob/main/src/src.ino>

Descargar repositorio codigo para programar una "imagen" en la matriz

<https://github.com/unloquer/ETSesnor>

[ejemplo.ino](#)

```
#include <FastLED.h>
#define LED_PIN D3
#define LED_TYPE WS2812B
#define COLOR_ORDER GRB
#define amarillo CRGB::Yellow
#define negro CRGB::Black
#define rojo CRGB::Red
#define azul CRGB::Blue
#define maplv1 0x00FF00
#define maplv2 0x00AA00
#define maplv3 0xFFFF00
#define maplv4 0xFFE994
#define maplv5 0xFFAA00
#define maplv6 0xEC9BA4
#define maplv7 0xE1AA00
#define maplv8 0xFF00FF
#define maplv9 0x00DAFE
#define maplv10 0x0181FE

const uint8_t matrixWidth = 8;
const uint8_t matrixHeight = 8;
#define NUM_LEDS (matrixWidth * matrixHeight)

int BRIGHTNESS = 60;
CRGB leds[matrixWidth * matrixHeight];

int loop_cnt = 0;
const int sampleWindow = 50; // Sample window width in mS (50 mS = 20Hz)
unsigned int sample;

void setup() {
    Serial.begin(115200);
    LEDs.addLeds<LED_TYPE,LED_PIN,COLOR_ORDER>(leds,NUM_LEDS);
    FastLED.setBrightness(BRIGHTNESS);
}

#define ESCENAS 1
CRGB matrix[ESCENAS][8][8] = {
    {
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
        {azul, azul, azul,azul, azul, azul, azul,azul},
    }
}
```

```
    {azul, azul, azul, azul, azul, azul, azul, azul},
    {azul, azul, azul, azul, azul, azul, azul, azul},
  },
};

void loop() {
  for(int i = 0; i< matrixHeight; i++) {
    for(int j = 0; j< matrixWidth; j++) {
      leds[i*matrixWidth + j] = matrix[loop_cnt%ESCENAS][i][j];
    }
  }
  unsigned long startMillis = millis(); // Start of sample window
  unsigned int peakToPeak = 0;

  unsigned int signalMax = 0;
  unsigned int signalMin = 1024;

  // collect data for 50 mS
  while (millis() - startMillis < sampleWindow)
  {
    sample = analogRead(0);
    if (sample < 1024) {
      if (sample > signalMax)
      {
        signalMax = sample;
      }
      else if (sample < signalMin)
      {
        signalMin = sample;
      }
    }
  }
  peakToPeak = signalMax - signalMin;
  int changeBrightness = map(peakToPeak, 20, 500, 0, 100);
  FastLED.setBrightness(changeBrightness);
  FastLED.show();
  Serial.println(peakToPeak);
  loop_cnt++;
  FastLED.clear();
}
```

Descargar repositorio codigo para programar varios estados en la matriz

Referentes

- https://www.reddit.com/r/FastLED/wiki/index/user_examples

From:

<https://wiki.unloquer.org/> -

Permanent link:

<https://wiki.unloquer.org/proyectos/talleres/ets/recursos?rev=1638028440>

Last update: **2021/11/27 15:54**

