QUICK REFERENCE · TK101

# PDM Microphone

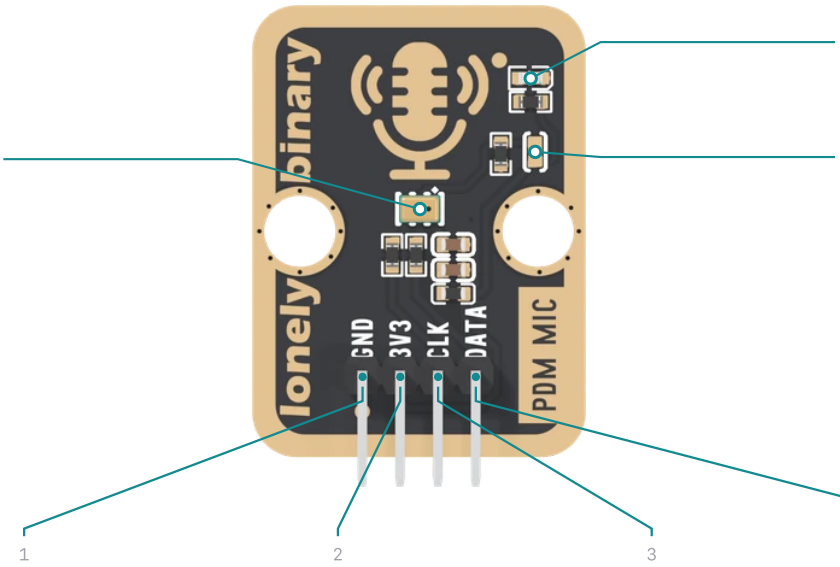
A MEMS microphone that sends sound as a stream of single bits, read by an ESP32-S3's I2S in PDM mode.

**3.3 V only, and a chip with PDM input**

The microphone runs from 1.62 to 3.6 V: **power it from 3V3, never 5 V.** It needs a microcontroller that reads PDM, such as the ESP32-S3 on its I2S0. An Uno or a Nano has no I2S at all.

## Pins, and which to wire

Drawn from above, header at the bottom. The ESP32 drives CLK; the microphone answers with one bit per clock pulse on DATA. Sound goes in through the small hole in the microphone's lid.




**Red power LED**
Lit on 3V3. It says nothing about the clock.

**L/R jumper, open**
L/R held LOW by 10 kΩ. Matters only for two mics.

**PDM microphone**
Sound enters through the hole in its lid. Keep it open.

| Front | GND | 3V3 | CLK | DATA |
|---|---|---|---|---|
| Back | GND | 3V3 | CLK | DATA |
| ESP32-S3 | GND | 3V3 | PDM CLK | PDM DIN |

**The numbers are small, and that is right**

The microphone gives −26 dBFS for a loud voice a few centimetres away. Speech at arm's length is about a thousandth of full scale. Subtract the average first, which removes a small offset, then multiply: the sketches apply a gain.

## How big a sample, for how loud a sound

### A loud voice, close

**1,642** of 32,767 at full scale

94 dB SPL is -26 dBFS.

### Speech at arm's length

**33** of 32,767 at full scale

60 dB SPL is -60 dBFS.

### A quiet room

**3** of 32,767 at full scale

40 dB SPL is -80 dBFS.

A peak in the tens is speech, and in the thousands is somebody talking into the hole. A reading that never moves from zero, or never drops below thousands, is a wiring fault, not a quiet room.

Worked out from the datasheet's sensitivity, −26 dBFS at 94 dB SPL (LinkMems M072 v1.4), not measured. The sound levels are typical figures. Sketch compiled for the ESP32-S3 with esp32 core 3.3.10.

## How loud is the room

Arduino IDE · Board: ESP32S3 Dev Module · USB CDC On Boot: Enabled · Tools > Serial Plotter

```
#include <ESP_I2S.h>

const int PIN_CLK = 5, PIN_DATA = 6;
const int RATE = 16000;
I2SClass mic;
int16_t samples[512];                    // 32 ms at 16 kHz

void setup() {
  Serial.begin(115200);
  mic.setPinsPdmRx(PIN_CLK, PIN_DATA);
  mic.begin(I2S_MODE_PDM_RX, RATE, I2S_DATA_BIT_WIDTH_16BIT,
            I2S_SLOT_MODE_MONO);
}

void loop() {
  size_t n = mic.readBytes((char *)samples, sizeof(samples)) / 2;
  if (n == 0) return;

  long sum = 0;                          // the offset first
  for (size_t i = 0; i < n; i++) sum += samples[i];
  int offset = sum / (long)n;

  int peak = 0;
  for (size_t i = 0; i < n; i++) {
    int a = abs(samples[i] - offset);
    if (a > peak) peak = a;
  }
  Serial.print("peak:");
  Serial.println(peak);                  // 0 to 32767
}
```

```
peak:31
peak:1240
peak:27
```

The Serial Plotter at 115200: a flat line for the room, a spike when you speak.

## Wiring

| | FRONT | ESP32-S3 |
|---|---|---|
| 1 | **GND** | **GND** |
| 2 | **3V3** | **3V3** |
| 3 | **CLK** | **GPIO 5** |
| 4 | **DATA** | **GPIO 6** |

Any free GPIO works for CLK and DATA; the ESP32-S3 receives PDM on I2S0 only, and the library chooses it. Keep the wires short: the clock runs at one or two megahertz.

## Where the sound gets in

Through the 0.5 mm hole in the microphone's lid, on the front. Do not cover it, and never blow compressed air into it, wash the board or clean it ultrasonically: the datasheet forbids all three. A speaker near the microphone, playing what it hears, feeds back into a howl; use headphones, or record first and play afterwards.

## When it does not work

### The reading is always 0

No clock or no data. Check CLK and DATA against the sketch, and GND.

### The numbers barely move when I talk

That is normal: speech is about a thousandth of full scale. Apply a gain.

### A howl from the speaker

Feedback. Use headphones, move the speaker away, or lower the gain.

### The sketch stops at the first read

DATA is not arriving. Check the wire, and that the pins in the sketch are the pins wired.

### A steady value that never drops

The offset. Subtract the average of each block before taking the peak.

### It worked on 5 V once, then stopped

5 V is over the microphone's 3.6 V maximum. Replace it and use 3V3.

## Specifications

| | | | |
|---|---|---|---|
| **Microphone** | LMD2718T261-OA1 (LinkMems), PDM | **Supply** | 3.3 V; 1.62 to 3.6 V |
| **Sensitivity** | −26 dBFS at 94 dB SPL | **Noise** | SNR 58 dB(A) |
| **Overload** | 120 dB SPL | **Clock** | 1.0 to 4.8 MHz, standard mode |
| **Wiring** | 2 signal wires, CLK and DATA | **Board** | 22.4 × 30.4 mm, header fitted |