

# **Memoria proyecto: Desarrollo de un dispositivo de grabación remota de conversaciones mediante tecnología láser**

Desarrollado por: Rocco Mancin, Vicente Tenconi

## **Introducción**

En el contexto de las investigaciones encubiertas y las operaciones de inteligencia, las fuerzas del orden requieren tecnologías avanzadas que permitan obtener información de manera discreta, segura y eficaz. La captación remota de conversaciones o sonidos sin necesidad de intervenir físicamente en el lugar objetivo constituye una herramienta estratégica de gran valor.

## **Objetivo del proyecto**

Este proyecto tiene como objetivo principal la realización de una prueba de concepto (PoC) para evaluar la viabilidad de un dispositivo de grabación de conversaciones o sonidos remotos mediante el uso de un láser, construido utilizando componentes económicos y fácilmente disponibles online. El sistema propuesto pretende registrar conversaciones a distancia a través de la detección de las vibraciones generadas en superficies reflectantes, principalmente ventanas, por las ondas sonoras que produce una conversación.

## **Descripción general del funcionamiento**

El principio físico que sustenta este dispositivo es la vibrometría láser. Cuando las ondas sonoras producidas por una conversación impactan sobre superficies ligeras, como los cristales de una ventana, estas superficies vibran de forma imperceptible pero constante. Un haz láser dirigido hacia la superficie puede captar dichas vibraciones a través de los cambios en la fase, frecuencia o posición del haz reflejado.

## **Fase de investigación y estudio preliminar**

En primer lugar, se realizó una investigación teórica sobre tecnologías de captación óptica de sonido, vibrometría láser y procesamiento de señales.

Las principales conclusiones de esta fase fueron:

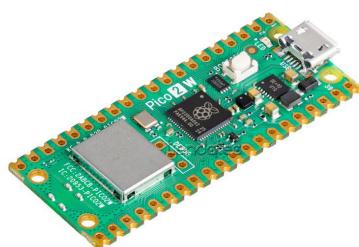
- La viabilidad teórica del uso de láser para captar vibraciones acústicas está ampliamente documentada.
- La principal dificultad técnica radica en la eliminación del ruido y la calibración del sistema para distintas condiciones ambientales y distancias de grabación.
- Las soluciones viables pasan por una combinación de hardware de alta precisión y algoritmos avanzados de procesamiento de señal.

## Selección de componentes

Tras el estudio preliminar, se inició la selección de los componentes:

- **Láser de clase 2 (650 nm, rojo visible):** se seleccionó por su buena visibilidad en las pruebas de laboratorio y facilidad en encontrar fotodiodos compatibles.
- **Fotodiodo BPW34:** para captar las ligeras variaciones de intensidad en el haz reflejado.
- **Amplificador de señal MAX9814:** para amplificar la señal del fotodiodo antes de su digitalización.
- **Raspberry Pi Pico:** como microcontrolador para grabación a través del ADC y procesamiento de señal inicial.
- **Ordenador portátil:** Para procesar datos utilizando algoritmos de conversión de audio y reducción de ruido.

Adicionalmente, hemos utilizado una placa de pruebas y cables eléctricos.

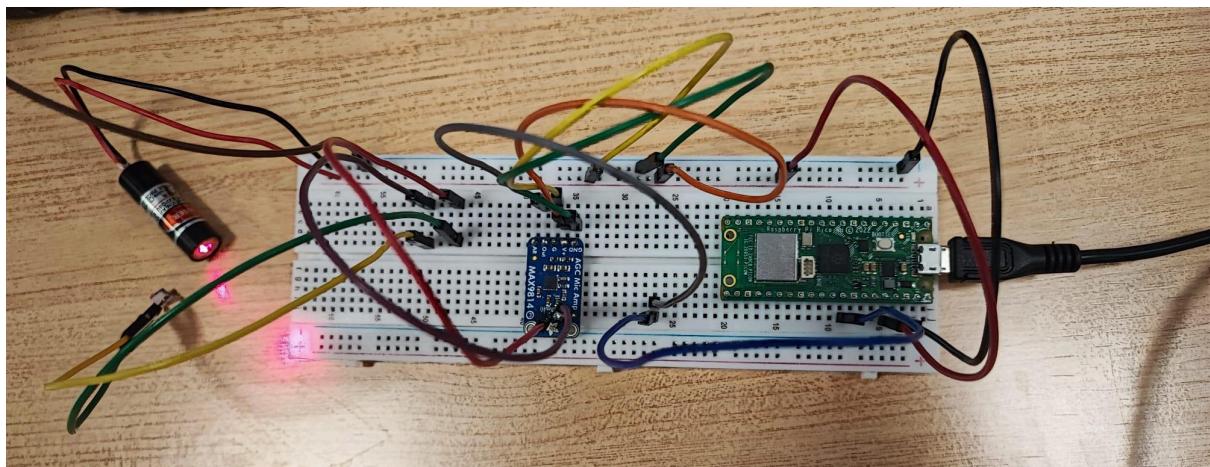


## Diseño de prototipos

La fase de diseño de prototipos constituyó el primer paso práctico hacia la validación de los principios teóricos estudiados durante la investigación preliminar. Dado que el proyecto se encontraba en una etapa de prueba de concepto, se optó por construir el prototipo utilizando componentes montados sobre placa de pruebas. Esta metodología permitió realizar ajustes rápidos y facilitar la experimentación con distintos parámetros y configuraciones de hardware.

Se diseñó el prototipo en módulos independientes:

- **Módulo de emisión:** compuesto por el láser montado sobre un soporte ajustable que permitió probar distintas configuraciones de alineación.
- **Módulo de medición y amplificación:** compuesto por el fotodiodo y el amplificador.
- **Módulo de adquisición y transmisión:** compuesto por el Raspberry Pi Pico, que enviaba los datos digitalizados al ordenador portátil para su análisis.



## Desarrollo de software y firmware

### Desarrollo del firmware para adquisición de datos

Se diseñó y programó un firmware, que actuó como el sistema embebido de adquisición de datos.

Las tareas principales del firmware fueron:

- Leer la señal proveniente del amplificador a través del convertidor analógico-digital (ADC) integrado de la Raspberry Pi Pico.
- Formatear los datos de adquisición en paquetes de datos.
- Transmitir los datos capturados a la computadora principal mediante el protocolo UDP a través de la conexión WiFi.

El uso de UDP permitió una transmisión rápida y ligera de datos, aceptando una pequeña posibilidad de pérdida de paquetes.

En seguida el código del firmware:

```
#include <hardware/adc.h>
#include <hardware/regs/adc.h>

#include <WiFi.h>
#include <WiFiUdp.h>

#define STASSID ""
#define STAPSK ""
#define CYCLE_DURATION 200 // cada 200 micros = 5000 cada segundo
#define SAMPLES_PER_PACKET 128

IPAddress remoteAddress = IPAddress(192, 168, 1, 156);
unsigned int remotePort = 8888; // local port to listen on

WiFiUDP Udp;

void setup() {
    Serial.begin(115200);

    WiFi.begin(STASSID, STAPSK);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(500);
    }

    Serial.print("Connected! IP address: ");
    Serial.println(WiFi.localIP());

    adc_init();
    adc_gpio_init(27); // Enable ADC on GPIO26 (ADC0)
    adc_select_input(1); // Select the first ADC channel
}

uint16_t replyBuffer[SAMPLES_PER_PACKET+8];
unsigned int _previousMicros, _currentMicros;
int numSample = 0;

void loop() {
    _currentMicros = micros();
    if (_currentMicros - _previousMicros > CYCLE_DURATION) {

        uint16_t result = adc_read();
        replyBuffer[numSample] = result;
        numSample++;

        _previousMicros = _currentMicros;

        if (numSample == SAMPLES_PER_PACKET) {
            Udp.beginPacket(remoteAddress, remotePort);
```

```

        Udp.write((uint8_t*)replyBuffer, SAMPLES_PER_PACKET * sizeof(uint16_t));
        Udp.endPacket();

        numSample = 0;
    }

}

```

## Desarrollo del código para la recepción y almacenamiento de los datos

En el lado de la computadora de procesamiento, se desarrolló un script en Python encargado de recibir los paquetes de datos UDP enviados por la Raspberry Pi Pico. Este script cumplía las siguientes funciones:

- Recibir y desempaquetar los datos transmitidos.
- Guardar los datos binarios en bruto (raw data) directamente en un archivo de almacenamiento.

Este enfoque permitió una separación clara entre la etapa de adquisición y la de procesamiento, facilitando la repetición de análisis y conversiones posteriores sobre los datos almacenados sin necesidad de realizar nuevas adquisiciones.

A seguir el código del script de recepción de datos raw:

```

import socket
import datetime

UDP_IP = "0.0.0.0"
UDP_PORT = 8888

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))

print("Listening...")

f = open(datetime.datetime.now().strftime('%Y-%m-%d_%H:%M:%S') + '.raw', 'wb')

first_byte = None
try:
    while True:
        data, _ = sock.recvfrom(4096) # buffer size is 4096 bytes
        if first_byte == None:
            first_byte = datetime.datetime.now()
            print("First packet received")
        f.write(data)

except KeyboardInterrupt as e:
    last_byte = datetime.datetime.now()

```

```

num_bytes = f.tell()
f.close()

print(f'First byte: {first_byte}')
print(f'Last byte: {last_byte}')
print(f'Num bytes: {num_bytes}')
print(f'KBytes per sec: {(num_bytes / 1024) / (last_byte-first_byte).total_seconds() }')

```

Para poder trabajar con herramientas de audio convencionales y facilitar el análisis, fue desarrollado un script de conversión de datos capaz de leer los archivos de datos raw y convertirlos al formato PCM big endian de 16 bits. Además, el script de conversión aplicaba un filtrado digital a los datos capturados, atenuando las frecuencias fuera del rango 250 Hz - 1333 Hz. Sucesivamente, se utilizó la herramienta de línea de comandos **ffmpeg** para encapsular los datos en un contenedor de audio estándar WAV, compatible con las aplicaciones de edición y análisis de audio.

A seguir el código del script de conversión de datos raw:

```

import sys
import numpy as np
import scipy.signal as sg

if len(sys.argv) != 2:
    print("Numero argomenti errato")
    sys.exit()

infile = open(sys.argv[1], 'rb')
outfile = open(f'{sys.argv[1]}.conv', 'wb')
chunk_size = 2

#####

def apply_band_pass_filter(data, lowcut, highcut):
    b, a = sg.butter(4, lowcut / (5000 / 2.), 'low')
    x_fil = sg.filtfilt(b, a, data)

    b, a = sg.butter(4, highcut / (5000 / 2.), 'high')
    x_fil = sg.filtfilt(b, a, x_fil)

    return x_fil

audio_data = []

while True:
    chunk = infile.read(chunk_size)
    if chunk:
        chunk = int.from_bytes(chunk, byteorder='little', signed=False)
        chunk = (chunk << 4) - 2**15
        chunk = int.to_bytes(chunk, length=2, byteorder='little', signed=True)

        audio_data.append(chunk)

    if len(audio_data) == 1024:
        outfile.write(audio_data)
        audio_data = []

```

```

if not infile.read(1): # eof reached
    break
infile.seek(infile.tell()-1) # -1 to revert read(1)

else:
    break

infile.close()

audio_data = apply_band_pass_filter(audio_data, 250, 1333)
audio_data = b''.join([ int.to_bytes(int(chunk), length=2, byteorder='little',
signed=True) for chunk in audio_data ])

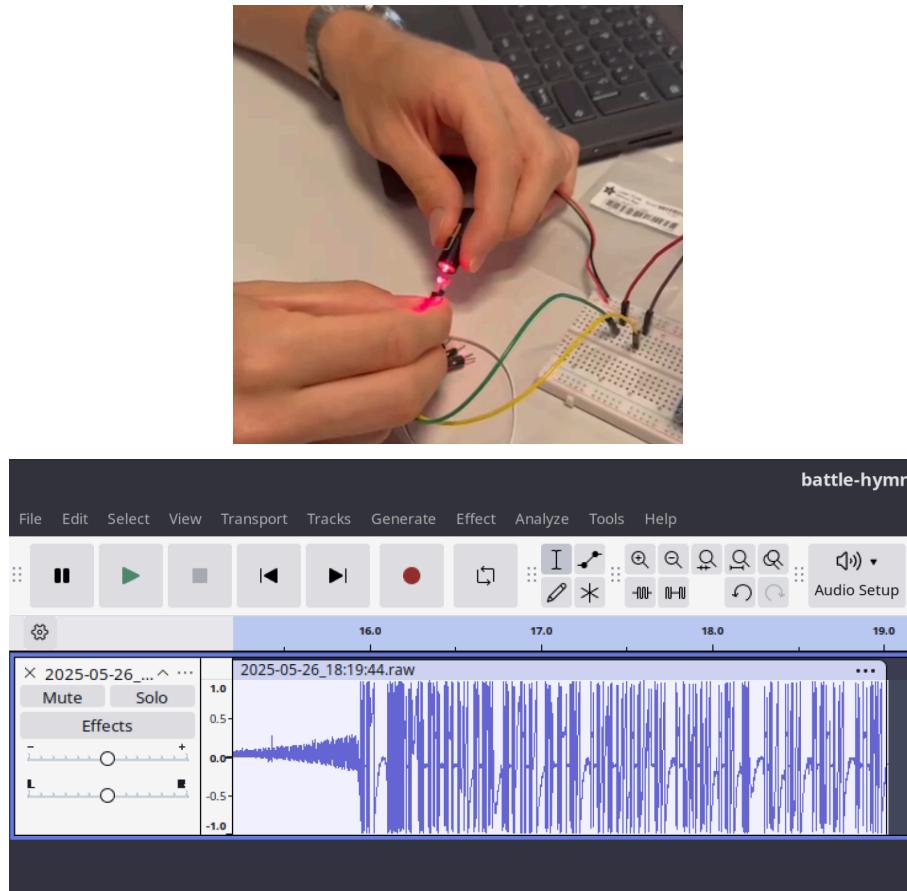
outfile.write(audio_data)
outfile.close()

```

## Pruebas en entornos controlados

Gracias a este diseño flexible fue posible:

- Verificar la correcta detección de las variaciones del haz láser, como es posible ver en las siguientes imágenes:



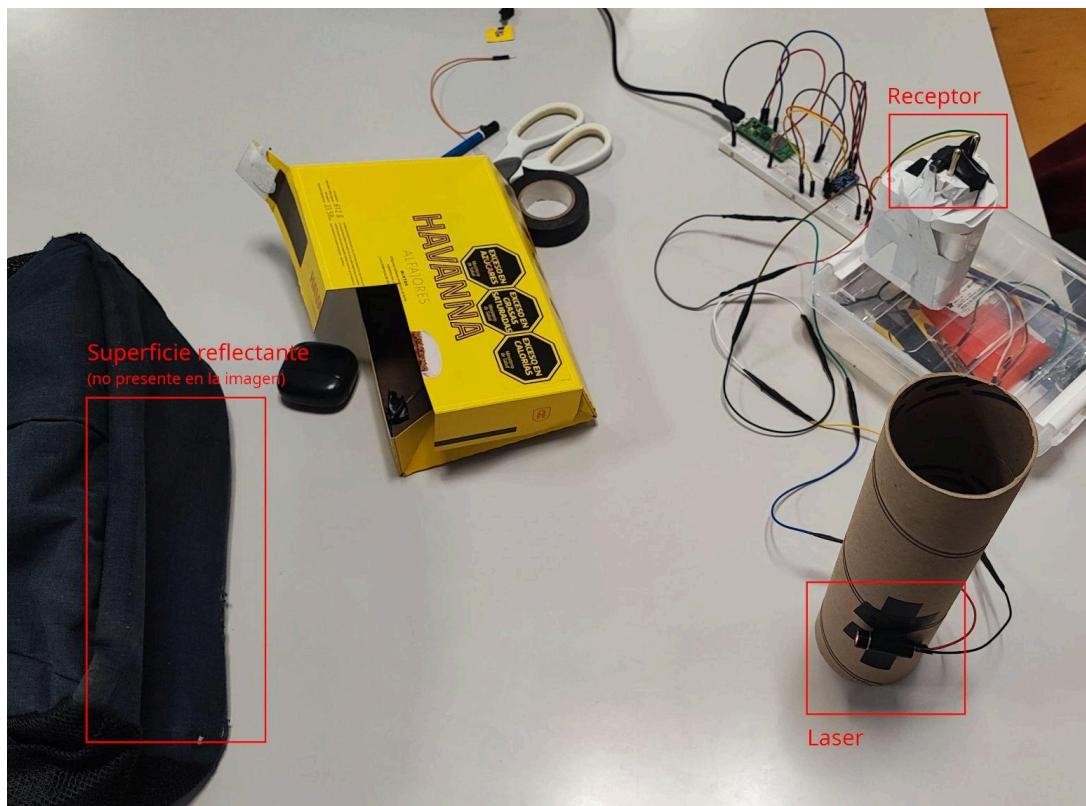
- Ajustar la ganancia de amplificación en función de la distancia y la intensidad de las vibraciones.
- Evaluar la efectividad de los primeros filtros digitales implementados para eliminar ruidos de fondo.

Con el prototipo preliminar operativo, se llevaron a cabo pruebas experimentales en entornos controlados. En todos los casos se utilizó como superficie reflectante el cristal de la pantalla de un teléfono móvil que reproducía música.

## Escenario 1

En este escenario se utilizó un ambiente interior con luz ambiental normal y sin precauciones específicas para evitar interferencias.

El resultado fue una grabación de audio de baja calidad, con algunas palabras inteligibles.

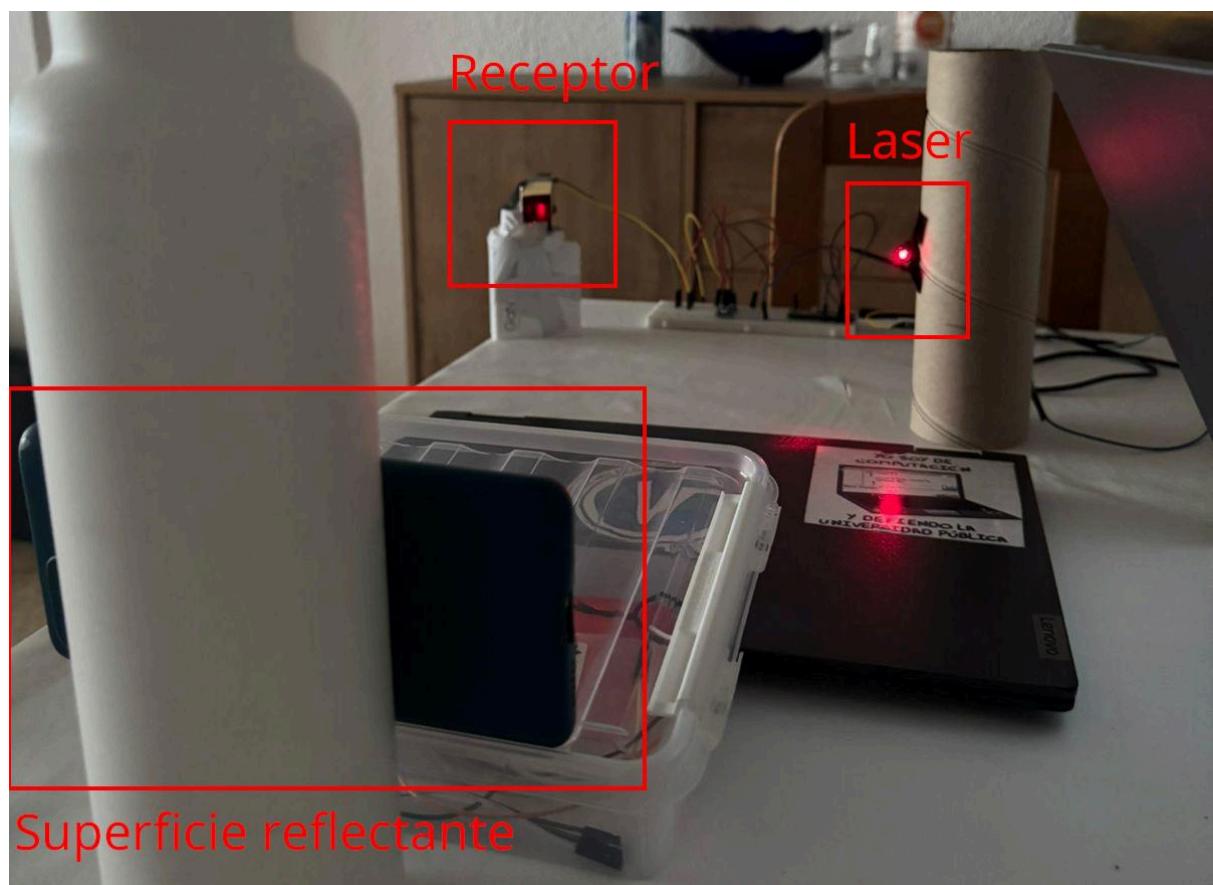


## Escenario 2

En este escenario se utilizó un ambiente interior con luz ambiental normal y una pequeña caja para reducir las interferencias por la luz ambiental.

El resultado fue una grabación de audio de peor calidad en comparación con el escenario 1. Más tarde se descubrió que el material interno de la caja era parcialmente reflectante, lo que significaba que la caja sí estaba reduciendo la luz ambiental que llegaba al sensor, pero

al mismo tiempo estaba introduciendo interferencia adicional de la luz láser que se reflejaba en el interior de la caja.



### Escenario 3

En este escenario se utilizó un ambiente interior casi completamente oscuro

El resultado fue una grabación de audio de mucha mejor calidad que en casos anteriores, lo que indica que la fuente principal del ruido probablemente es la luz ambiental.

Este escenario se probó utilizando la misma configuración que el escenario anterior pero quitando la caja.

## Problemas encontrados y soluciones implementadas

Durante el desarrollo surgieron varios problemas técnicos significativos. A continuación, se detallan los principales y las soluciones adoptadas:

### Dificultades de alineación láser

**Problema:** Alinear correctamente el láser con el fotodiodo resultó ser un problema, ya que pequeños cambios provocaban la pérdida total de la señal reflejada.

**Solución:** Se utilizó un soporte móvil al que se pudo fijar el láser mediante cinta adhesiva. Moviendo el soporte móvil y el punto de fijación se pudo alinear correctamente el láser.

### Interferencias por luz ambiental

**Problema:** La luz ambiental provocaba interferencias en la detección del movimiento de la superficie reflectante.

**Solución:** Inicialmente se construyó una pequeña caja que pudiera proteger el sensor de la luz ambiental. Más tarde se descubrió que la caja construida estaba hecha de un material inadecuado (ya que era reflectante). Pruebas posteriores confirmaron que reducir la luz ambiental es un aspecto clave para aumentar la calidad de la grabación.

## Resultados obtenidos

Es posible grabar sonidos inteligibles, pero con el prototipo actual es necesario estar en condiciones favorables.

## Propuestas de mejora

Si bien el prototipo cumplió con los objetivos propuestos, se identificaron varias posibles mejoras:

- **Filtrado hardware adicional:** La adición de filtros analógicos de hardware previos al procesamiento digital permitiría una pre filtración de ruido, aliviando parte de la carga computacional y aumentando la calidad de la señal final.

- **Optimización del amplificador:** Utilizar amplificadores de mayor ancho de banda y menor ruido propio para mejorar la fidelidad de la captación de señal.
- **Adición de una óptica de colimación:** El uso de lentes colimadoras permitiría mantener la integridad del haz a distancias mucho mayores, aumentando el rango operativo del dispositivo y la calidad del sonido grabado.
- **Algoritmos de aprendizaje automático:** Incorporar algoritmos de IA podría permitir una mejor separación de señal y ruido, adaptándose en tiempo real a cambios en las condiciones ambientales.

## 12. Conclusiones

El proyecto ha demostrado la viabilidad técnica de la captación remota de conversaciones mediante tecnología láser. Está claro que para obtener grabaciones de calidad, especialmente a distancia, se necesitan componentes más caros y precisos, además de crear un sistema más estable.