



industriales
etsii

**Escuela Técnica
Superior
de Ingeniería
Industrial**

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Diseño y desarrollo de un equipo de medición de la cantidad de burbujas en el agua mediante sonda de doble punta

TRABAJO FIN DE GRADO

**GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA**



**Universidad
Politécnica
de Cartagena**

Autor: Juan Carlos Ronda Escudero
Director: Juan Antonio López Riquelme
Codirector: Nieve Pavón Pulido

Cartagena,

Índice

Capítulo 1 Introducción y Objetivos	Pág. 5
1.1 Introducción	Pág. 5
1.2 Objetivos	Pág. 6
1.3 Desarrollo de la memoria	Pág. 7
Capítulo 2 Estado del arte	Pág. 9
2.1 Introducción	Pág. 9
2.2 Canal experimental	Pág. 9
2.3 Flujos multifásicos y resalto hidráulico	Pág. 11
2.4 Mediciones actuales	Pág. 14
2.4.1 Sonda óptica	Pág. 14
2.4.2 Módulo optoelectrónico	Pág. 15
2.5 Problemática actual	Pág. 16
Capítulo 3. Descripción de la arquitectura Hardware y Software desarrollada	Pág. 17
3.1 Introducción	Pág. 17
3.2 Hardware	Pág. 17
3.2.1 Fuente de alimentación	Pág. 18
3.2.2 Sonda de resistividad	Pág. 19
3.2.3 ESP32	Pág. 20
3.2.4 LCD 1602 I2C	Pág. 21
3.2.5 Electrónica de acondicionamiento	Pág. 22
3.3 Software	Pág. 25
3.3.1 ESP32	Pág. 25

3.3.2 Software WLAN PC	Pág. 31
Capítulo 5. Resultados y conclusions	Pág. 39
5.1 Resultados	Pág. 39
5.2 Conclusiones	Pág. 41
5.3 Trabajo futuro	Pág. 41
Bibliografía	Pág. 43
Anexo Esquemático	Pág. 44
Anexo PCB	Pág. 45

Índice de figuras

Figura 2.1. Canal hidrodinámico Gunt HM 162	Pág. 10
Figura 2.2 Canal hidrodinámico del laboratorio de hidráulica	Pág. 11
Figura 2.3 Resalto hidráulico	Pág. 11
Figura 2.4 Rotura de oleaje	Pág. 12
Figura 2.5 Tipos de resaltos hidráulicos	Pág. 13
Figura 2.6 Esquema de instrumentación	Pág. 14
Figura 2.7 Sonda óptica de doble punta	Pág. 15
Figura 2.8 Módulo optoelectrónico	Pág. 15
Figura 3.1 Esquema de bloques del Hardware	Pág. 18
Figura 3.2 Fuente simétrica $\pm 12V$ 1A	Pág. 18
Figura 3.3 Ejemplo de geometría de la sonda	Pág. 19
Figura 3.4 Sonda de resistividad	Pág. 20
Figura 3.5 SoM ESP32	Pág. 20
Figura 3.6 LCD 16x2 I2C	Pág. 22
Figura 3.7 Oscilador	Pág. 23
Figura 3.8 Ejemplo divisor resistivo	Pág. 23
Figura 3.9 Acondicionamiento de señal	Pág. 24
Figura 3.10 Señal cuadrada y señal de entrada del microcontrolador	Pág. 24
Figura 3.11 PCB	Pág. 25
Figura 3.12 Crear nuevo proyecto	Pág. 31
Figura 3.13 Interfaz del programa	Pág. 32
Figura 3.14. Interfaz de PC tras recibir los datos	Pág. 37
Figura 3.15. Excel exportado	Pág. 38

Figura 4.1 Tabla de datos comparados	Pág. 40
Figura 5.2 Gráfica de datos comparados	Pág. 40

Capítulo 1

Introducción y Objetivos

1.1 Introducción del proyecto.

Una de las ramas de la Ingeniería Civil es la ingeniería hidráulica, la cual se encarga de diseñar y ejecutar obras relacionadas con el abastecimiento y distribución del agua.

Una de las mayores estructuras que un ingeniero hidráulico puede construir es una presa. Para el diseño de una presa son necesarios muchos estudios sobre la energía cinética y potencial de la corriente del agua, los resaltos hidráulicos, los sedimentos y las burbujas de aire que transporta el agua.

El aprovechamiento de la energía cinética del agua es la [energía hidráulica] denominadas energías renovables. Este tipo de energía es generada a partir de recursos naturales. La energía hidráulica genera un impacto mínimo sobre el medio ambiente y gracias al ciclo del agua es una energía inagotable, además es una energía limpia ya que no produce emisiones.

Para estudiar la dinámica del movimiento y la velocidad de las burbujas se utilizan equipos basados en fibra óptica, los cuales suelen tener un coste elevado.

En este proyecto se pretende diseñar un equipo para estudiar los fenómenos mencionados mediante una tecnología basada en la conductividad, reduciendo así el coste del equipo de forma considerable.

Una de las causas por la que se planteó este proyecto, además del elevado coste de adquisición del equipo de medición de la fibra y tratamiento de la señal, es la fragilidad de las sondas de fibra óptica.

El estudio principal donde se utilizan estas sondas es la medición de la cantidad de burbujas en el agua después de un resalto hidráulico, como una cascada o la apertura de la compuerta de un dique. Este entorno de flujo turbulento somete a la fibra óptica a fuertes impactos por la fuerza del agua y los sedimentos que esta arrastra. Esto unido a la fragilidad del vidrio conlleva una multitud de reparaciones que incrementan el coste y el tiempo.

Los ensayos se han realizado en el laboratorio de hidráulica del edificio ELDI de la Universidad Politécnica de Cartagena.

Se establecen parámetros para rastrear y detectar de entre todos los datos obtenidos los que corresponden a burbujas de aire. También se compararán los datos con los obtenidos con la sonda de fibra óptica que se mostrarán en el apartado de resultados del último capítulo.

1.2 Objetivos.

El objetivo general del proyecto es diseñar y desarrollar un equipo de medición de burbujas en el agua tras un resalto hidráulico para poder realizar estudios de la dinámica de movimiento y velocidad.

Para conseguir este objetivo, se proponen los siguientes objetivos específicos:

- Diseñar y desarrollar la sonda de medida.
- Estudio y selección de los componentes más adecuados para la medición.
- Diseño y desarrollo de la electrónica de acondicionamiento.
- Realizar una interfaz de usuario del sistema.

1.3 Desarrollo de la memoria.

Capítulo 1: Introducción

El Capítulo 1 presenta la motivación de crear este equipo y su fin, enumera los objetivos y describe la memoria.

Capítulo 2. Estado del Arte

En el Capítulo 2 se describe el equipo con el que se hacen mediciones, y el entorno donde se desarrollan los ensayos y la problemática que se plantea.

Capítulo 3. Descripción de la arquitectura Hardware y software desarrollado

En este capítulo explicamos detalladamente los componentes que forman parte de este sistema. También exponemos el trabajo realizado de hardware y el software desarrollado para integrar todos los componentes.

Capítulo 4. Resultados, conclusiones y trabajo futuro

En este último capítulo se exponen los resultados obtenidos, las conclusiones a las que podemos llegar en base a esos datos, los trabajos futuros posibles y las posibilidades de mejora.

Capítulo 2

Estado del Arte

2.1 Introducción.

En este capítulo se muestra el canal donde se realizan las pruebas ubicado en la Universidad Politécnica de Cartagena en el laboratorio de hidráulica.

Se hace una presentación de los flujos multifásicos y concretamente el resalto hidráulico ya que será el objeto de estudio. Se explica también el tipo de medida y el equipo con el que se toma.

Una vez explicado el objeto a estudiar y el medio por el cual se estudia, se plantea la problemática que presenta y por lo tanto la necesidad de crear un equipo que lo sustituya.

2.2 Canal experimental.

Los ensayos se realizaron en el laboratorio de hidráulica del edificio ELDI de la Universidad Politécnica de Cartagena, en un canal hidrodinámico [HM 162] de pendiente variable (Figura 2.1). Este canal, construido en Alemania, tiene una longitud efectiva de 12,5 m, un ancho de 30 cm y una altura de 45 cm.

El funcionamiento del canal se basa en una bomba centrífuga que junto con un sensor de caudal y un variador de frecuencia recircula agua a caudal constante hasta 132 m³/h. El caudal se ajusta en el cuadro de control y se regula automáticamente y la altura del agua se regula con una compuerta ajustable en altura mediante un husillo accionado por manivela.

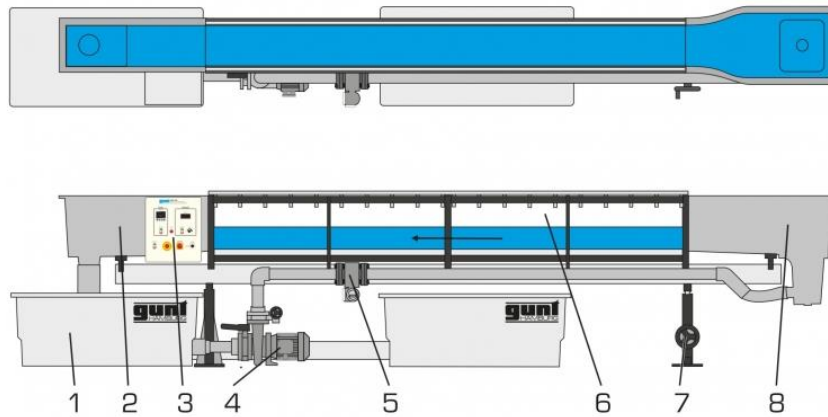


Figura 2.1. Canal hidrodinámico Gunt HM 162

1. Depósito de agua.
2. Elemento de salida. Situado al final del canal de ensayo donde recoge y filtra el agua por gravedad.
3. Caja de distribución. En ella se encuentra un sistema tipo Scada conectado a un PLC y un variador de frecuencia donde se controla todo el canal.
4. Bomba centrífuga.
5. Caudalímetro electromagnético.
6. Sección de ensayo 12,5 x 0,45 x 0,3 m.
7. Ajuste de inclinación. Se ajusta electrónicamente.
8. Elemento de entrada y cámara de remanso.



Figura 2.2. Canal hidrodinámico del laboratorio de hidráulica del edificio ELDI de la UPCT

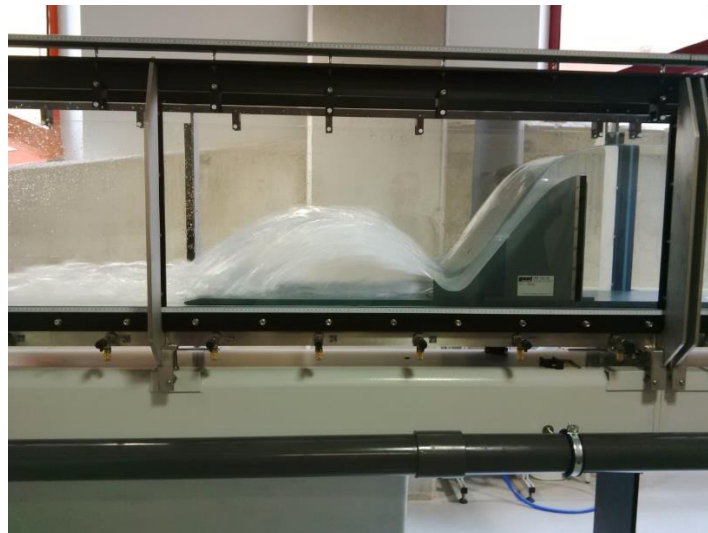


Figura 2.3 Resalto hidráulico

2.3 Flujos multifásicos y resalto hidráulico.

Un [flujo multifásico] es la mezcla de al menos dos fases, ya sean líquido, sólido o gaseoso.

Algunos ejemplos son la humedad que existe en el ambiente o cuando rompe una ola en la costa debido a la gran energía que lleva y al disminuir progresivamente el fondo o al encontrarse una variación abrupta, toda la energía se proyecta hacia la superficie.



Figura 2.4 Rotura de oleaje

Los flujos multifásicos se involucran en muchas ramas de la ingeniería ya sea la floculación en ingeniería química para tratamiento de aguas, la cavitación en ingeniería naval para el estudio de cascos y hélices de barcos, o la mezcla de aire y combustible en el motor de combustión interna.

Este proyecto se centra en un flujo multifásico de agua y aire. Este fenómeno se conoce como [resalto hidráulico] y se produce cuando un flujo cambia de velocidad bruscamente debido a la variación del terreno y genera una pérdida de energía en la superficie.

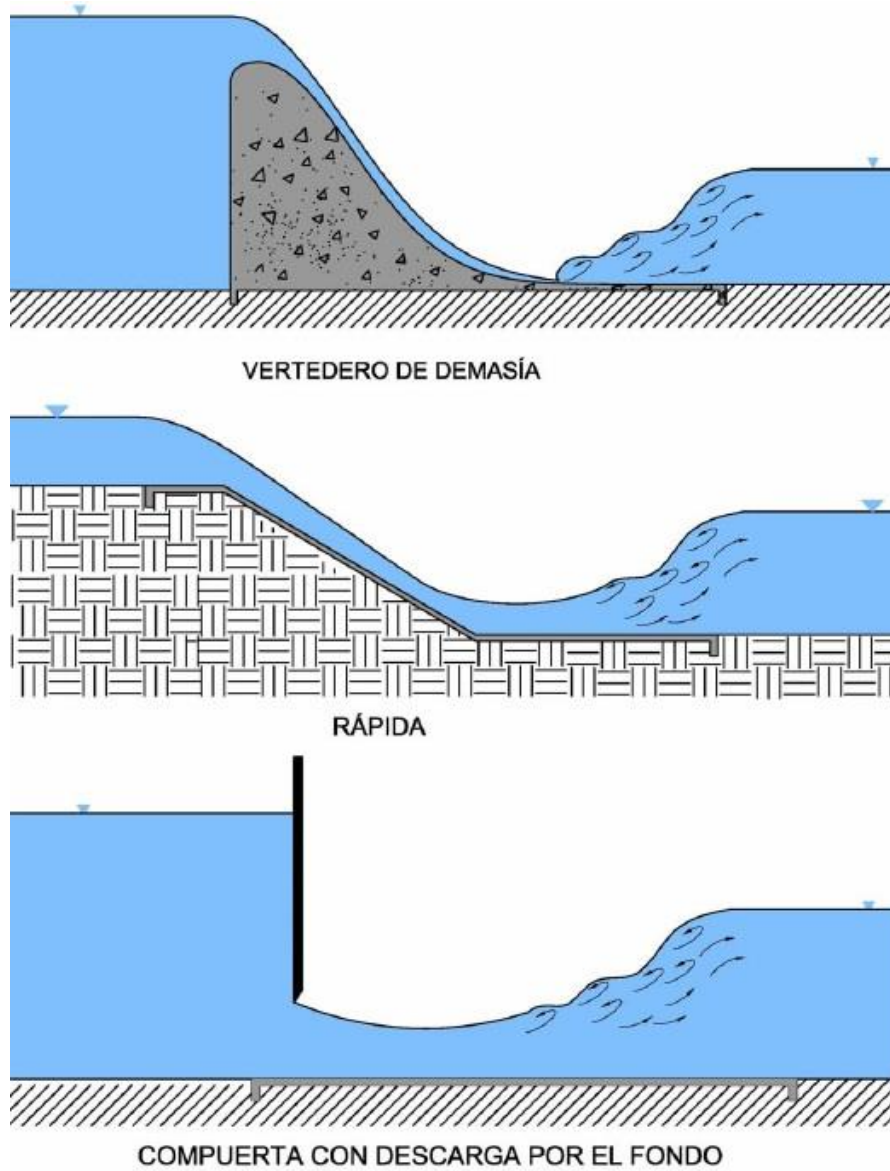


Figura 2.5 Tipos de resaltos hidráulicos

Este equipo es crucial para estudiar la dinámica de dichos flujos para diseñar y reforzar a consecuencia las estructuras por las que fluyen.

2.4 Mediciones actuales.

Los componentes del sistema de lectura son los siguientes:

- Sonda óptica de doble punta
- Módulo optoelectrónico
- Tarjeta de adquisición de datos
- PC

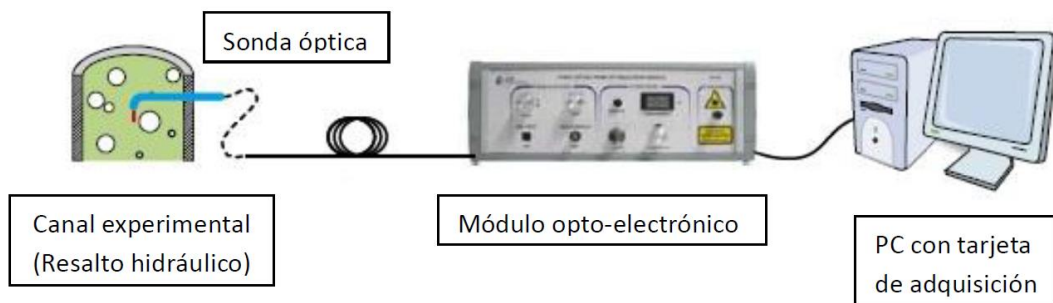


Figura 2.6 Esquema de instrumentación

2.4.1 Sonda óptica.

Para el estudio de la dinámica del agua se hacen mediciones con una sonda de fibra óptica parcialmente sumergida en el canal de pruebas antes descrito (Figura 2.2) y se leen en un ordenador a través de una tarjeta de adquisición de datos.

Para no interferir en las mediciones es importante que la sonda sea lo más pequeña posible. La distancia entre las puntas debe ser menor del tamaño de las burbujas.

La sonda consta de dos filamentos de fibra óptica por donde se emiten pulsos de luz y se interpreta la respuesta a través del módulo optoelectrónico.



Figura 2.7 Sonda óptica de doble punta

El funcionamiento se debe a la reflexión y [refracción de la luz]. Cuando hay un cambio de fase también hay un cambio en el índice de refracción de la luz.

Este tipo de sistema es capaz de diferenciar el tipo de medio en el que se encuentra la sonda debido al tipo de refracción que recibe es capaz de diferenciar agua, aire, CO₂, nitratos en las aguas, etc.

2.4.2 *Módulo optoelectrónico.*

Este módulo manda pulsos de luz a uno de los filamentos de fibra óptica y recibe los pulsos de luz que recibe el otro filamento. Esos pulsos de luz recibidos los convierte en señal eléctrica amplificada y a través de una tarjeta de adquisición de datos de alta frecuencia es capaz de registrar los cambios de flujo en el extremo de la sonda.



Figura 2.8 Módulo optoelectrónico

2.5 Problemática actual.

El uso más común de este sistema es medir la cantidad de aireación de un flujo de agua para poder estudiar las zonas donde podría golpear con más fuerza. Debido a esto, la sonda debe sumergirse en fuertes corrientes de agua donde es golpeada no solo con la fuerza del agua sino también con los sedimentos que ésta pueda arrastrar. Esto genera que la sonda de fibra óptica sufra averías de vez en cuando de elevado coste de reparación, además del coste de adquisición del sistema.

Capítulo 3

Descripción de la arquitectura Hardware y Software desarrollada

3.1 Introducción.

En este capítulo describiremos todo lo relacionado con el funcionamiento del sistema, desde la sonda, el microcontrolador, el circuito de acondicionamiento hasta la comunicación WLAN tanto del microcontrolador como del programa desarrollado en el PC.

3.2 Hardware.

En este apartado se expondrán los componentes utilizados y todo el trabajo realizado de hardware.

Es importante recordar que todo el hardware estará instalado dentro de una caja estanca de PVC.

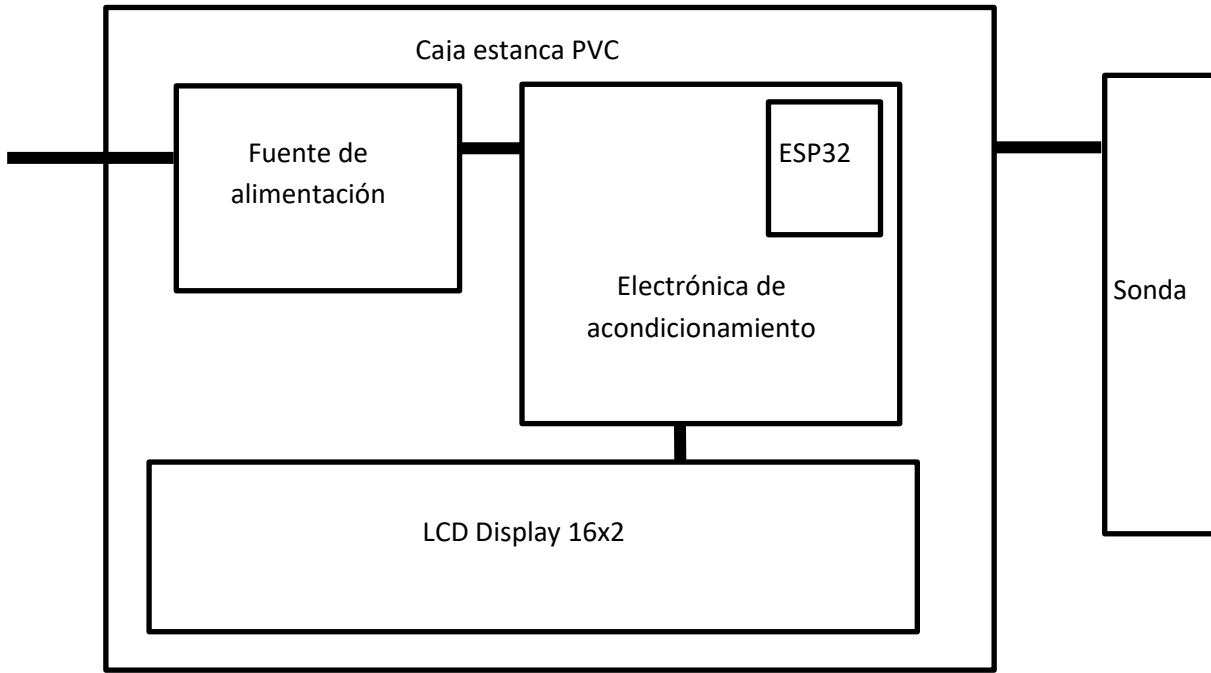


Figura 3.1 Esquema de bloques del Hardware.

3.2.1 Fuente de alimentación.

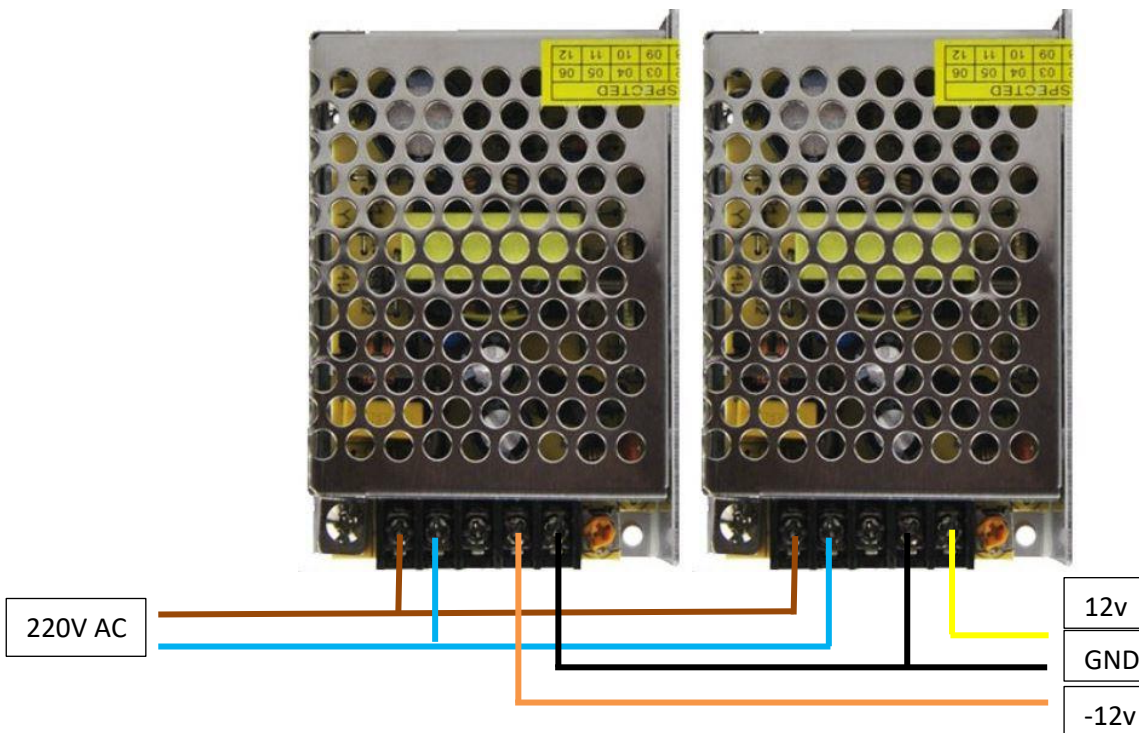


Figura 3.2 Fuente simétrica $\pm 12V$ 1A.

Necesitamos disponer de una fuente de tensión simétrica de $\pm 12V$ tanto para generar una señal cuadrada como para alimentar los amplificadores operaciones.

Esto lo vamos a conseguir mediante dos fuentes de tensión industriales de 12V conectadas en serie como se puede ver en la Figura 3.2.

3.2.2 Sonda de resistividad.

Como se introdujo en el Capítulo 1, se pretende sustituir el equipo actual de fibra óptica por un equipo capaz de tomar datos mediante conductividad eléctrica para reducir los costes de reparaciones de la sonda, ya que se introduce en fuertes corrientes de agua donde los sedimentos que ésta arrastra suelen estropear los filamentos con cierta regularidad.

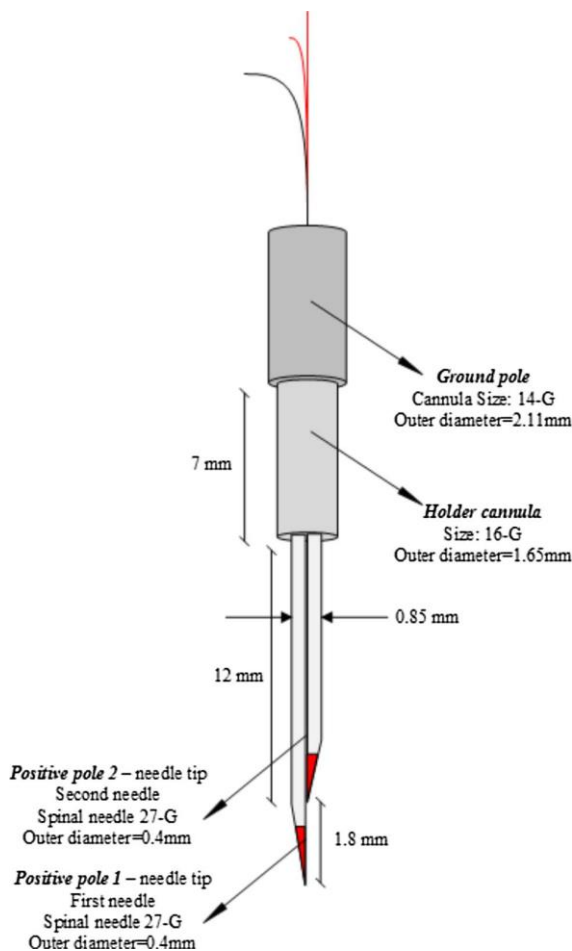


Figura 3.3 Ejemplo de [geometría de la sonda].

Figura 3.4 Sonda de resistividad.

Dicha sonda se ha fabricado de forma artesanal. Consta de dos polos fabricados con alambre de platino de 0,2 mm de sección empalmado con hilo de cobre barnizado por el interior de un perfil redondo de aluminio de 6 mm de diámetro exterior y 4 mm de diámetro interior sellado con resina Epoxi.

3.2.3 *ESP32.*

[ESP32] es la denominación de una familia de chips que logra un consumo de energía ultra bajo y bajo coste creado por Espressif Systems.

El SoC (System On a Chip) es el denominado ESP-WROOM-32 es un potente microcontrolador con arquitectura de 32 bits, conectividad Bluetooth y wifi. El SoM (System on Module) integra en un módulo el SoC ESP32, memoria FLASH, cristal oscilador y antena Wifi en PCB.

La plataforma ESP32 se puede programar en multitud de lenguajes, como C++, Python, Javascript o basic. En este caso ha sido programada mediante el entorno de programación Arduino (en lenguaje C++), ya que tiene un lenguaje de programación estudiado durante la carrera y posee un IDE muy sencillo de utilizar que además tiene detrás una gran cantidad de librerías y una comunidad de usuarios muy activa.

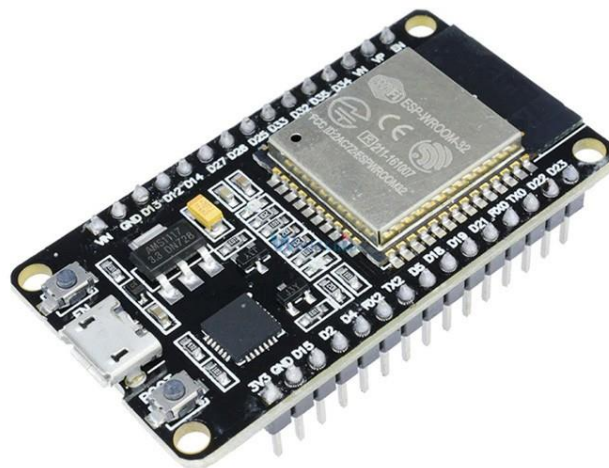


Figura 3.5 SoM ESP32.

Las especificaciones técnicas más relevantes son:

- Voltaje de Alimentación: 3.3V
- Voltaje lógico entradas/salidas (GPIO): 3.3V
- CPU: Dual core Tensilica Xtensa LX6 (32 bit)
- Frecuencia del reloj: 240MHz
- SRAM: 520KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO: 34
- UART: 2
- SPI: 3
- I2C: 2
- Wi-Fi, Protocolo 802.11 b/g/n/e/i (802.11n up to 150 Mbps)
- Wi-Fi, certificación RF: FCC/CE/IC/TELEC/KCC/SRRC/NCC
- Wi-Fi Security WPA/WPA2/WPA2-Enterprise/WPS
- Stack de Protocolo TCP/IP integrado

3.2.4 LCD 1602 I2C.

Esta [pantalla LCD] tiene un display de 16 caracteres y 2 filas. La aplicación principal de esta pantalla en el sistema es proporcionarnos la IP del ESP32 a la cual debemos conectarnos para solicitar los datos.



Figura 3.6 LCD 1602 16x2 I2C.

Características del LCD 1602:

- Tensión de alimentación: 5V
- Comunicación mediante I2C (mediante el módulo PCF8574T incorporado)
- Pantalla LCD, 16 caracteres y 2 líneas
- Fondo azul y letras en blanco
- Dimensiones: 80 x 36 x 19 (mm)

3.2.5 Electrónica de acondicionamiento.

La electrónica integra el microcontrolador ESP32, el oscilador para generar la onda cuadrada regulable en frecuencia y el circuito de acondicionamiento mediante amplificadores operacionales para adaptar la señal al microcontrolador.

Un aspecto importante en circuito de acondicionamiento es el hecho de que la sonda no se alimenta con tensión continua, sino con una señal cuadrada de entre 500 Hz y 1 kHz de frecuencia para evitar la corrosión en el extremo de la sonda.

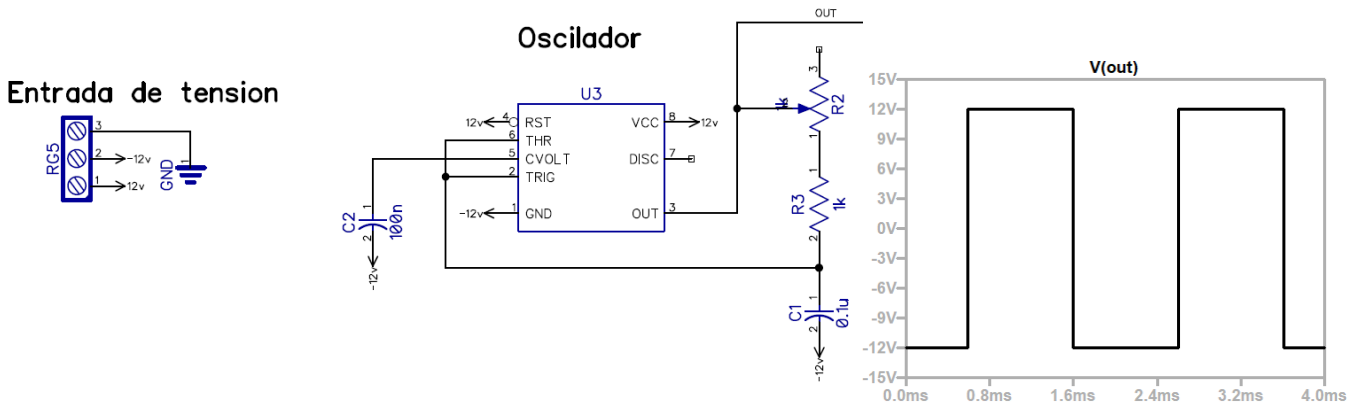


Figura 3.7 Oscilador

Esta señal es generada con un circuito integrado NE555 configurado con unos componentes pasivos en modo astable.

Esta señal cuadrada pasa por un divisor resistivo (Figura 3.8), donde la resistencia nombrada como R1 tiene un valor de 50k y R2 sería el valor resistivo de un filamento de la sonda y la masa en paralelo con una resistencia de gran valor 1Meg. De este modo, cuando la sonda está sumergida la resistencia del canal 1 de la sonda tiene un valor tan bajo que es casi nulo y el conjunto de resistencias en paralelo tendrá un valor muy bajo y por lo tanto la tensión de salida será prácticamente cero.

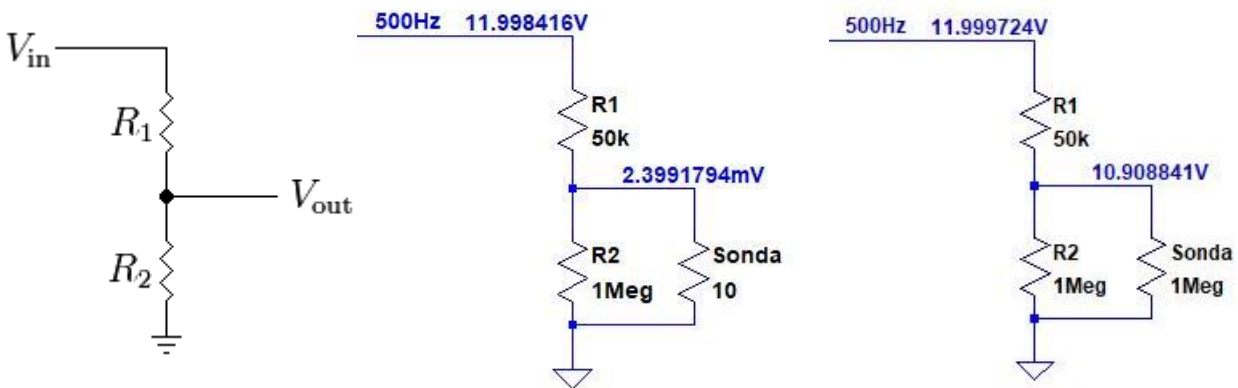


Figura 3.8 Ejemplo divisor resistivo. Sonda sumergida en agua. Sonda no sumergida

A continuación, la señal pasa por un amplificador operacional OPA445AP configurado como seguidor de tensión, para que la medición sea lo más precisa posible. La señal se compara con una tensión continua de 5V con otro amplificador operacional LM358N configurado como comparador. De esta forma conseguimos una señal binaria mucho más clara. A continuación, pasa por otro seguidor de tensión como paso previo al microcontrolador para impedir que la medición afecte a la medida.

Al salir del seguidor de tensión, mediante un divisor resistivo con un potenciómetro de precisión el cual debemos calibrar para que en ningún caso la tensión a la salida supere los 3V.

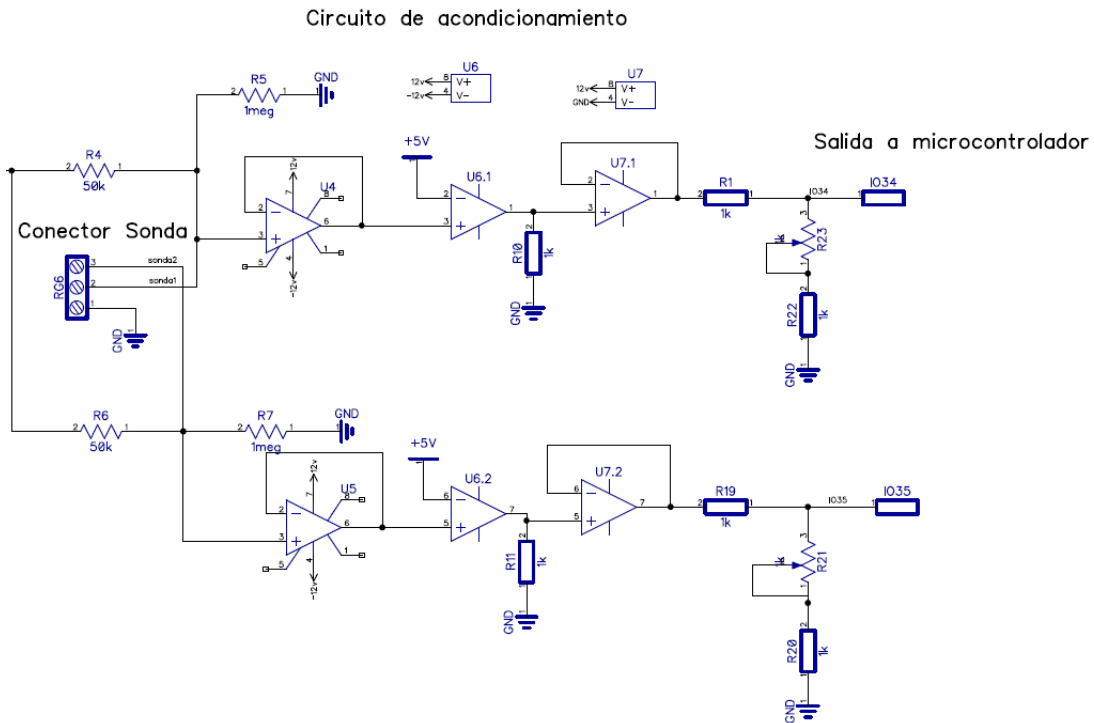


Figura 3.9 Acondicionamiento de señal

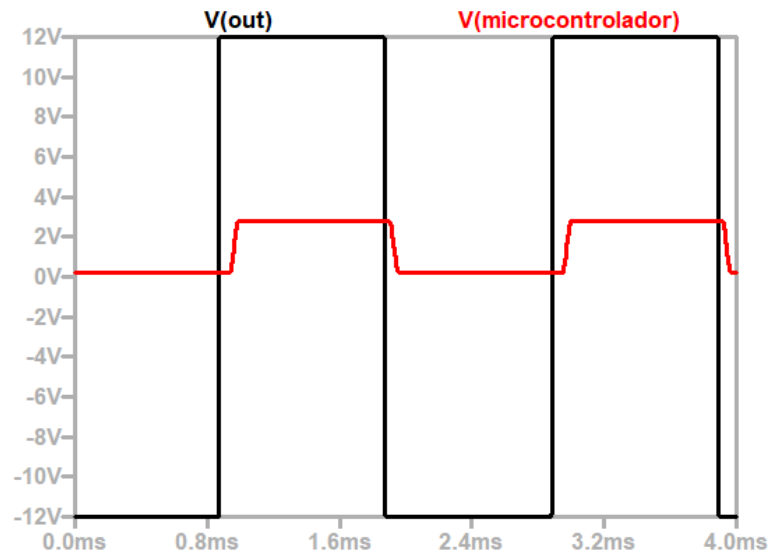


Figura 3.10 Señal cuadrada y señal de entrada del microcontrolador.

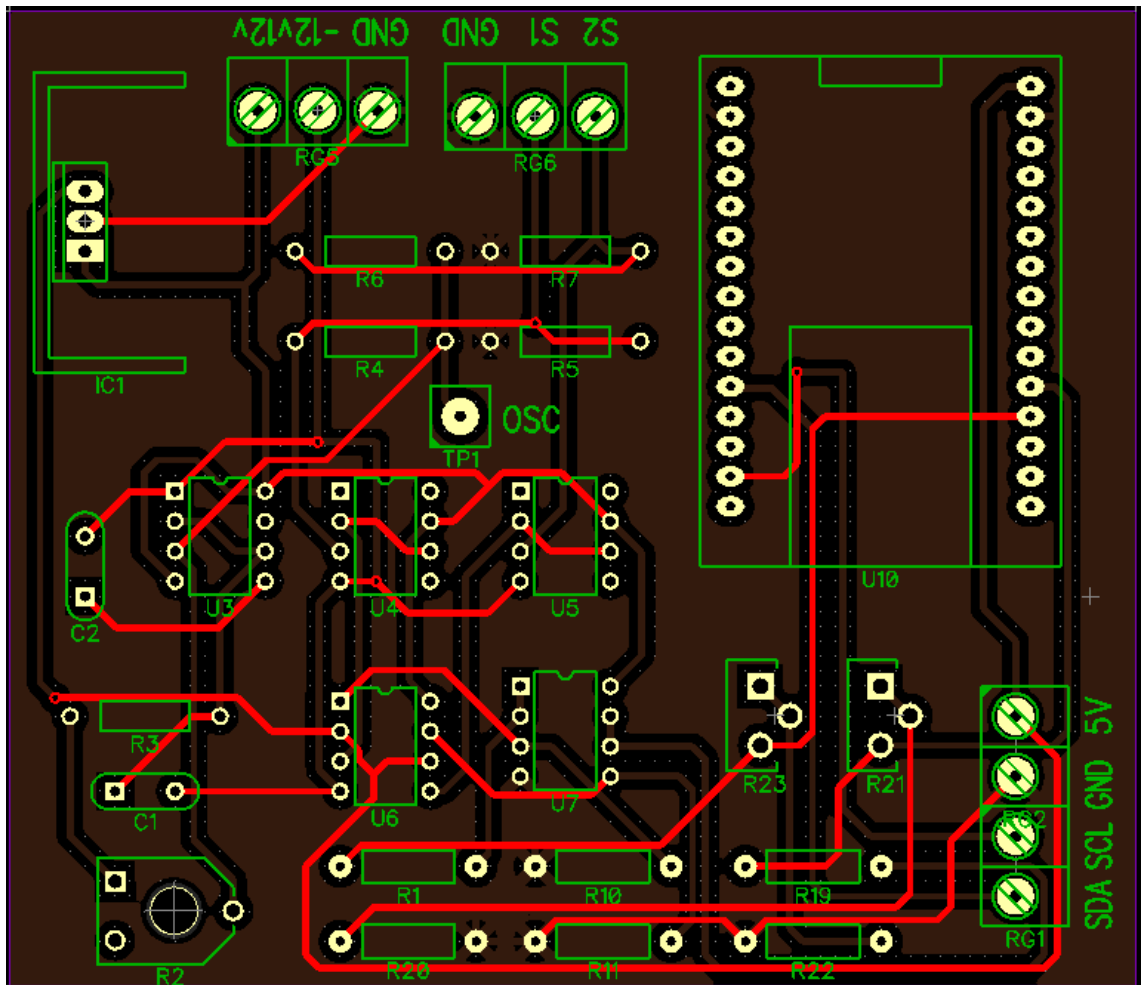


Figura 3.11 PCB.

3.3 Software.

3.3.1 ESP32.

En el punto anterior hemos mencionado todo el proceso de generar y acondicionar la señal para que el microcontrolador pueda leerla.

Los ensayos se realizaron en un canal de pruebas, pero está diseñado para poder medir en cascadas de agua más grandes por lo tanto toda la electrónica se encuentra dentro de una caja estanca alimentada a 220V mediante un conector IP68 y los datos los transmite por wifi. Por ese motivo hemos decidido utilizar la placa ESP32, ya que cuenta con tecnología wifi.

Descartamos la comunicación Bluetooth ya que los datos deberán ser recibidos por un PC que estará a resguardo y en algunas ocasiones la distancia entre el PC y la electrónica superará los 10 metros.

Al encender la electrónica, la placa ESP32 se conectará a una red wifi llamada "DESKTOP-MI9908O 8910" generada desde el propio PC con una contraseña "k810V73". Al conectarse mostrará a través de un LCD la ip a la cual debemos conectarnos desde el PC.

En el capítulo siguiente explicaremos detalladamente el código. Pero básicamente desde el ordenador hacemos una petición de medida y el microcontrolador al recibir esa petición inicia la medición de 10 segundos a una frecuencia de 1kHz. Una vez medido empieza el bucle de comunicación donde envía los datos en paquetes y son procesados por el programa del PC y se queda a la espera de recibir otra medida.

A continuación, vamos a comentar el código íntegro del servidor cargado en la placa ESP32. En primer lugar, definimos las librerías necesarias y los datos de frecuencia de lectura y número total de datos a medir.

Debemos definir también el nombre de la red wifi y la contraseña, la cual conocemos ya que el método más sencillo para que siempre acceda a la misma red y no tener que modificar este dato es generar una red a través del ordenador.

Las variables más importantes son los vectores donde almacenaremos el tiempo de medida y las lecturas de los dos sensores.

```
#include <WiFi.h>
#include <LiquidCrystal_I2C.h>

#define frecuencia 1000
#define datos 10000

char ssid[] = "DESKTOP-MI99080 8910";
char pass[] = "k810V73";

int status = WL_IDLE_STATUS;
int sensor1 = 34;
int sensor2 = 35;

LiquidCrystal_I2C lcd(0x27, 16, 2);

unsigned long ahora = 0;
unsigned long antes = 0;
double MuestreoEnMicro = 1000000 / frecuencia;

long Medida[datos];
long Medida2[datos];
float tiempo[datos];

WiFiServer server(80);
WiFiClient client = server.available();

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  while ( status != WL_CONNECTED) {
    Serial.print("Intentando conectar a: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(5000);
  }
  server.begin();
  printWifiStatus();
}
```

En el setup se inicializa el LCD, las patillas que utilizaremos como entrada analógica deberemos nombrarlas como entrada. Después de esto intentará a conectarse a la red wifi y una vez lo haga lanzará la función “printWifiStatus()”. Esta función nos mostrará en la LCD la ip que se le ha asignado.

```
void printWifiStatus() {
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  IPAddress ip = WiFi.localIP();

  Serial.print("Direccion IP: ");
  Serial.println(ip);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Direccion IP: ");
  lcd.setCursor(0,1);
  lcd.print(ip);

  long rssi = WiFi.RSSI();
  Serial.print("Intensidad de señal (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

En el bucle principal del programa inicializamos el servidor y una vez que se conecte un cliente recibimos un dato que corresponderá con una petición.

```
void loop() {
  WiFiClient client = server.available();
  String recibidos = "";
  if (client) {
    Serial.println("Nuevo cliente");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        recibidos += c;
        if (c == '\n' && currentLineIsBlank) {
          if (recibidos.startsWith(":DCBA")) {
            Serial.println("Petición de medida");
            medir();
          }
        }
      }
    }
  }
}
```

Si recibimos el comando “:DCBA” iniciamos la función de lectura de datos.

```
void medir() {
  Serial.println("Midiendo");
  ahora = micros();
  int iteracion = 0;
  while ((micros() - ahora) > MuestreoEnMicro && iteracion < datos) {
    ahora = micros();
    if (iteracion == 0) antes = ahora;
    tiempo[iteracion] = (ahora - antes) / 1000;
    Medida[iteracion]=analogRead(sensor1);
    Medida2[iteracion]=analogRead(sensor2);
    iteracion++;
  }
  Serial.print("Medido en: ");
  Serial.print((ahora - antes) / 1000000);
  Serial.println(" segundos");
}
```

La sincronización de las mediciones de esta función se basa en la lectura de los microsegundos con la función “micros()”. Se lee y almacena los microsegundos que han transcurrido antes de iniciar las lecturas. Ese dato se irá restando a cada lectura de microsegundos para obtener el valor relativo. El bucle while evaluará si hemos llegado al microsegundo correspondiente al siguiente valor, dependiendo de la frecuencia de lectura, siempre y cuando no hayamos llegado al valor máximo de datos a leer.

Es importante que haya la menor cantidad de sentencias en este bucle para que no haya pérdidas de datos, ya que estamos trabajando a nivel de microsegundos. Posteriormente en el loop trataremos estos datos para enviarlos.

Mediante una serie de else if les daremos forma a los datos para que ocupen el mismo número de caracteres y los iremos concatenando en modo texto en el formato antes explicado. Una vez tengamos el paquete completo será enviado y quedará a la espera de recibir confirmación. Al acabar los datos enviara un comando “:\$.” con el fin de desconectar el cliente.

```
int iteracion = 0;
while (iteracion < datos) {
    unsigned long t = tiempo[iteracion];
    String trama = ":";
    if (t < 10) trama += "00000";
    else if (t < 100) trama += "0000";
    else if (t < 1000) trama += "000";
    else if (t < 10000) trama += "00";
    else if (t < 100000) trama += "0";
    trama += String(t);

    unsigned int d = Medida[iteracion];
    if (d < 10) trama += "000";
    else if (d < 100) trama += "00";
    else if (d < 1000) trama += "0";
    trama += String(d);

    unsigned int d2 = Medida2[iteracion];
    if (d2 < 10) trama += "000";
    else if (d2 < 100) trama += "00";
    else if (d2 < 1000) trama += "0";
    trama += String(d2);

    client.print(trama);
    char espera;
    while (espera != ':') {
        espera = client.read();
        delay(5);
    }
    espera = ' ';
    iteracion++;
}
client.print(":$:");
break;
}

if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
delay(1);
client.stop();
Serial.println("cliente desconectado");
}
}
```

3.3.2 Software WLAN PC.

El programa que se ejecuta en el PC ha sido desarrollado enteramente utilizando el entorno de desarrollo Visual Studio.

Lo primero que debemos es crear un proyecto nuevo del tipo aplicación de Windows Forms (.NET Framework).

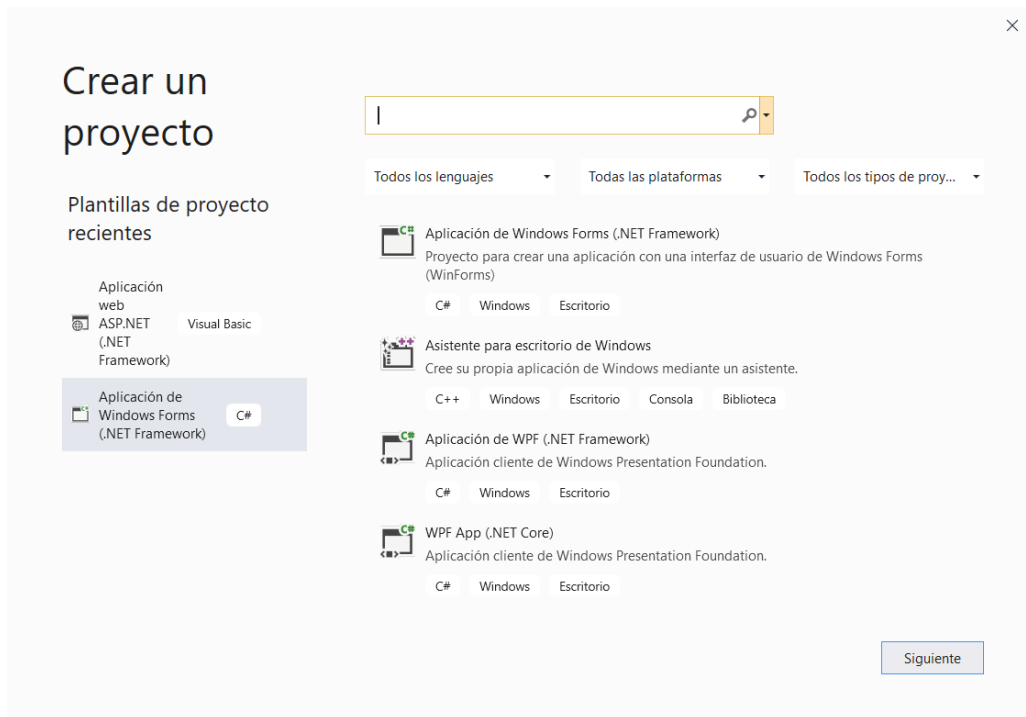


Figura 3.12 Crear nuevo proyecto.

Creamos un Form básico para trabajar en el que tenemos un cuadro de texto para introducir datos, un cuadro de texto para reflejar información, un botón para lanzar la petición de lectura y un elemento de tipo chart que se corresponde con una gráfica donde se mostraran los datos recibidos.

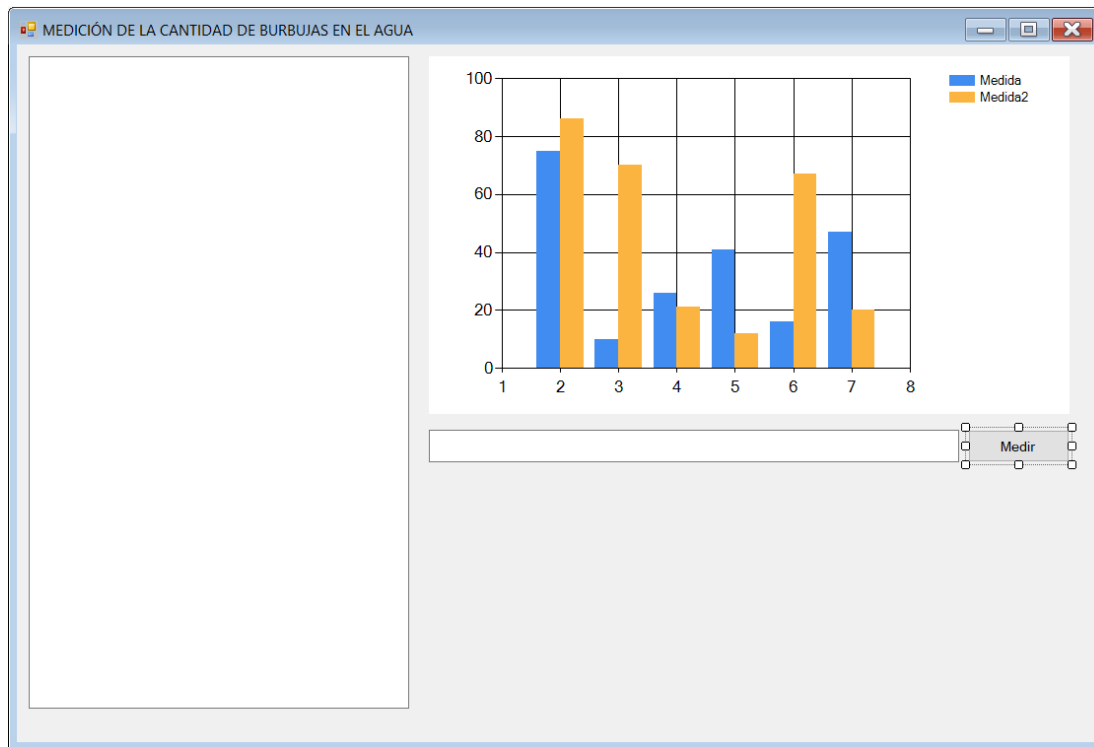


Figura 3.13 Interfaz del programa.

Procedemos a inicializar componentes en el archivo “NombreProyecto\Form1.Designer.cs”, la función “private void InitializeComponent()”. En esta función debemos inicializar los componentes creados y las series necesarias para la gráfica.

```
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea1 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();
    System.Windows.Forms.DataVisualization.Charting.Legend legend1 = new System.Windows.Forms.DataVisualization.Charting.Legend();
    System.Windows.Forms.DataVisualization.Charting.Series series1 = new System.Windows.Forms.DataVisualization.Charting.Series();
    System.Windows.Forms.DataVisualization.Charting.Series series2 = new System.Windows.Forms.DataVisualization.Charting.Series();
    System.Windows.Forms.DataVisualization.Charting.Series series3 = new System.Windows.Forms.DataVisualization.Charting.Series();
    this.chart1 = new System.Windows.Forms.DataVisualization.Charting.Chart();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.button1 = new System.Windows.Forms.Button();
    this.textBox2 = new System.Windows.Forms.TextBox();
    this.contextMenuStrip1 = new System.Windows.Forms.ContextMenuStrip(this.components);
    ((System.ComponentModel.ISupportInitialize)(this.chart1)).BeginInit();
    this.SuspendLayout();
}
```

```
//
// chart1
//
chartArea1.Name = "ChartArea1";
this.chart1.ChartAreas.Add(chartArea1);
legend1.Name = "Legend1";
this.chart1.Legends.Add(legend1);
this.chart1.Location = new System.Drawing.Point(418, 12);
this.chart1.Margin = new System.Windows.Forms.Padding(3, 3, 3, 2);
this.chart1.Name = "chart1";
series1.ChartArea = "ChartArea1";
series1.Legend = "Legend1";
series1.Name = "Tiempo";
series2.ChartArea = "ChartArea1";
series2.Legend = "Legend1";
series2.Name = "Medida";
series3.ChartArea = "ChartArea1";
series3.Legend = "Legend1";
series3.Name = "Medida2";
//this.chart1.Series.Add(series1);
this.chart1.Series.Add(series2);
this.chart1.Series.Add(series3);
this.chart1.Size = new System.Drawing.Size(650, 363);
this.chart1.TabIndex = 5;
this.chart1.Text = "chart1";
this.chart1.Click += new System.EventHandler(this.chart1_Click);
```

En la inicialización de la gráfica debemos añadir la leyenda con las series antes creadas.

```
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(418, 391);
this.textBox1.Multiline = true;
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(538, 33);
this.textBox1.TabIndex = 6;
//
// button1
//
this.button1.Location = new System.Drawing.Point(965, 391);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(103, 33);
this.button1.TabIndex = 7;
this.button1.Text = "Medir";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// textBox2
//
this.textBox2.Location = new System.Drawing.Point(12, 12);
this.textBox2.Multiline = true;
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(386, 662);
this.textBox2.TabIndex = 8;
```

En la declaración del botón y los cuadros de texto solo deberemos modificar el tamaño, nombre y añadir la opción de multilínea que por defecto no la crea.

```
//
// contextMenuStrip1
//
this.contextMenuStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);
this.contextMenuStrip1.Name = "contextMenuStrip1";
this.contextMenuStrip1.Size = new System.Drawing.Size(61, 4);
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(1094, 708);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.button1);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.chart1);
this.Name = "Form1";
this.Text = "MEDICIÓN DE LA CANTIDAD DE BURBUJAS EN EL AGUA";
((System.ComponentModel.ISupportInitialize)(this.chart1)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();
}
```

A continuación, vamos a comentar la función que establecerá conexión con el ESP32.

```
private void button1_Click(object sender, EventArgs e)
{
    string NombreArchivo = "medida.xlsx";
    File.Delete(NombreArchivo);
    FileInfo InformacionHojaCalculo = new FileInfo(NombreArchivo);
    ExcelPackage ihc = new ExcelPackage(InformacionHojaCalculo);
    var IntroducirDato = ihc.Workbook.Worksheets.Add("medida");
    IntroducirDato.Cells["A1"].Value = "Tiempo";
    IntroducirDato.Cells["B1"].Value = "Sonda1";
    IntroducirDato.Cells["C1"].Value = "Sonda2";
    IntroducirDato.Cells["A1:C1"].Style.Font.Bold = true;
    int linea = 2;

    Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    IPEndPoint direccion = new IPEndPoint(IPAddress.Parse(textBox1.Text), 80);
    try
    {
        byte[] datos = { 58, 68, 67, 66, 65, 13, 10, 13, 10 };
        socket.Connect(direccion);
        textBox2.AppendText("Conectado\r\n");
        socket.Send(datos);
        textBox2.Multiline = true;

        byte[] buffer = new byte[1054];
        int pos = 0;
        string sdatos = "";
    }
}
```

Lo primero que podemos ver es la creación de un archivo .xlsx donde almacenará los datos y posteriormente podremos extraer y comparar con otras mediciones. En el archivo introducimos el título de tres columnas con el nombre “Tiempo”, “Sonda1” y “Sonda2”. Creamos una variable llamada “línea” para ir desplazando el cursor e ir introduciendo datos secuencialmente línea por línea.

La clase socket es la encargada de establecer conexión a través de la red wifi con un servidor, solo necesitará que le introduzcamos en el cuadro de texto 1 la ip del servidor.

La variable llamada “datos” almacena en ella los caracteres ascii “:DCBA” que el servidor tendrá que interpretar y posteriormente creamos el buffer donde recibiremos los datos.

```
while ((pos = socket.Receive(buffer)) != 0)
{
    sdatos = Encoding.UTF8.GetString(buffer, 0, pos);
    //textBox2.AppendText("\r\n");
    //textBox2.AppendText(sdatos);
    //textBox2.AppendText("\r\n");

    if (sdatos == ":$:")
    {
        break;
    }

    long tiempo = Convert.ToInt32(sdatos.Substring(1, 6));
    long medida1 = Convert.ToInt32(sdatos.Substring(7, 4));
    long medida2 = Convert.ToInt32(sdatos.Substring(11, 4));

    IntroducirDato.Cells["A" + linea.ToString()].Value = tiempo;
    IntroducirDato.Cells["B" + linea.ToString()].Value = medida1;
    IntroducirDato.Cells["C" + linea.ToString()].Value = medida2;

    linea++;

    chart1.Series["Medida"].Points.AddXY(tiempo, medida1);
    chart1.Series["Medida2"].Points.AddXY(tiempo, medida2);

    byte[] a = { 58 };
    socket.Send(a);
}
```

El bucle while se ejecutará tantas veces como sea necesario mientras esté recibiendo datos. Leemos los datos del buffer, los convertimos a tipo String y los almacenamos en la variable “sdatos”.

Si en algún momento recibimos la cadena de caracteres “:\$.” el programa deberá salir del bucle ya que ha recibido el comando de final de trama.

Como anticipamos en el capítulo anterior, el valor que se almacena en “sdatos” tendrá la estructura: “:TTTTT*XXXXYYYY” de la cual extraeremos los datos del tiempo de la lectura, el valor de la medida de la sonda1 y el valor de la sonda2 y los almacenamos en tres variables de tipo long, las cuales utilizaremos para graficar.

Con el fin de que se sincronicen a la perfección dato a dato el servidor y el cliente cada lectura el cliente (PC) enviará un comando “:” que el servidor (ESP32) deberá esperar para enviar el siguiente dato.

```
        textBox2.AppendText("\r\n");
        textBox2.AppendText("Recibido con éxito");
        textBox2.AppendText("\r\n");
        socket.Disconnect(true);
    }
    catch (Exception error)
    {
        textBox2.AppendText("Error: {0}" + error.ToString());
    }
    IntroducirDato.View.FreezePanes(2, 1);
    ihc.Save();
    MessageBox.Show("Excel exportado con éxito");
}
```

Una vez recibidos todos los datos, lanza una frase de confirmación en el cuadro de texto 2 y un mensaje en una ventana emergente avisándonos de que ha exportado el Excel correctamente.

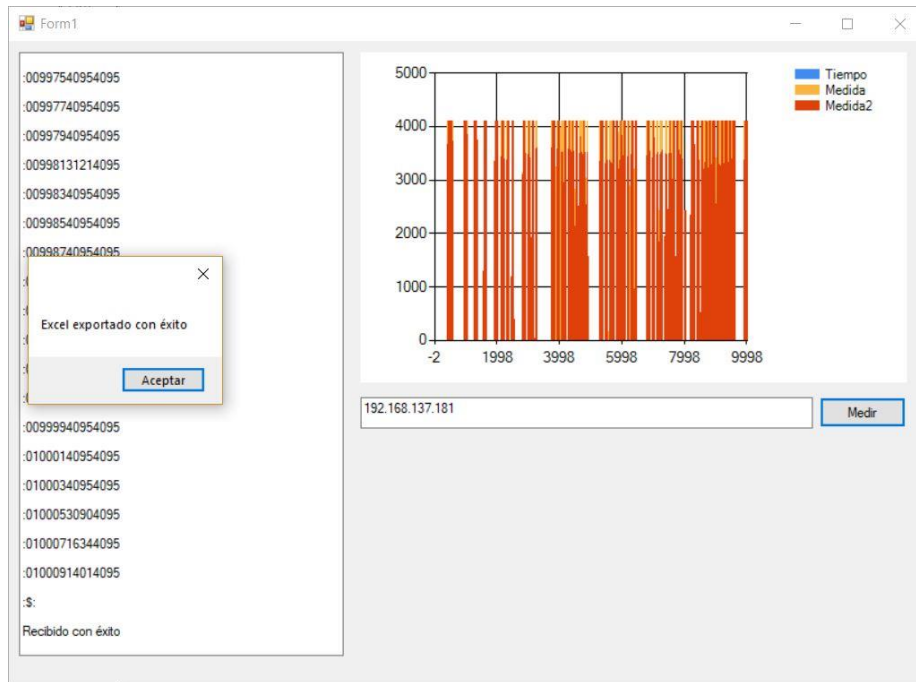


Figura 3.14. Interfaz de PC tras recibir los datos

	A	B	C
1	Tiempo	Sonda1	Sonda2
197	195	0	272
198	196	0	1286
199	197	0	1233
200	198	0	526
201	199	0	0
202	200	0	0
203	201	0	0
204	202	0	0
205	203	408	237
206	204	2263	1725
207	205	4095	2971
208	206	4095	3655
209	207	3810	2929
210	208	2499	1869
211	209	2	0
212	210	0	0
213	211	0	0
214	212	0	0
215	213	438	274
216	214	2329	1739
217	215	4095	4095
218	216	4095	4095
219	217	4095	4095
220	218	4095	4095
221	219	4095	4095
222	220	4095	4095
223	221	4095	4095
224	222	4095	4095
225	223	4095	4095
226	224	4095	4095

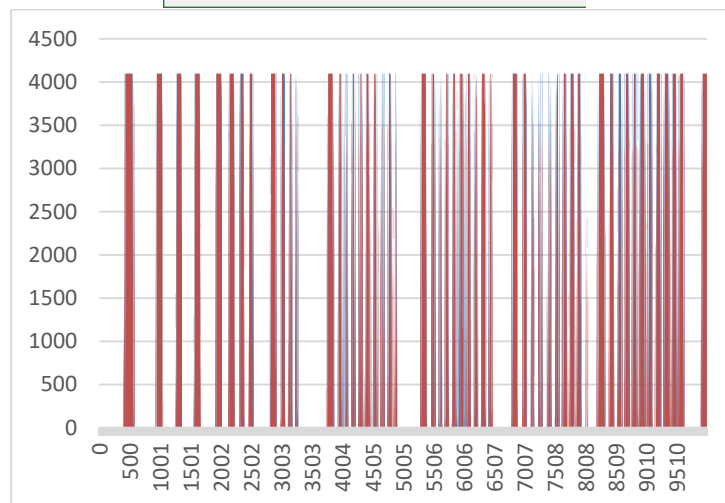


Figura 3.15. Excel exportado

Capítulo 4

Resultados, conclusiones y trabajo futuro

4.1 Resultados.

En este capítulo muestro los datos obtenidos comparados con el sistema que se plantea sustituir. En la gráfica se muestra la distancia a la que se coloca la sonda de donde empieza el flujo turbulento que ocasiona las burbujas y la C (%) que define la concentración de líquido en ese flujo multifásico, en un flujo laminar donde no hay burbujas se presupone el 100%.

Los datos obtenidos son bastante prometedores ya que viendo los resultados de la gráfica se comporta de una forma bastante similar las dos lecturas.

FIBRA		Aluminio
y (m)	C (%)	C (%)
0,01	4,19	3,23
0,02	7,6	9,75
0,03	13,54	13,97
0,04	16,76	16,36
0,05	16,28	13,97
0,06	16,77	17,99
0,07	19,57	35,27
0,08	39,41	65,99
0,09	69	81,83
0,1	80,86	93,42
0,11	94,17	98,35

Figura 4.1 Tabla de datos comparados.

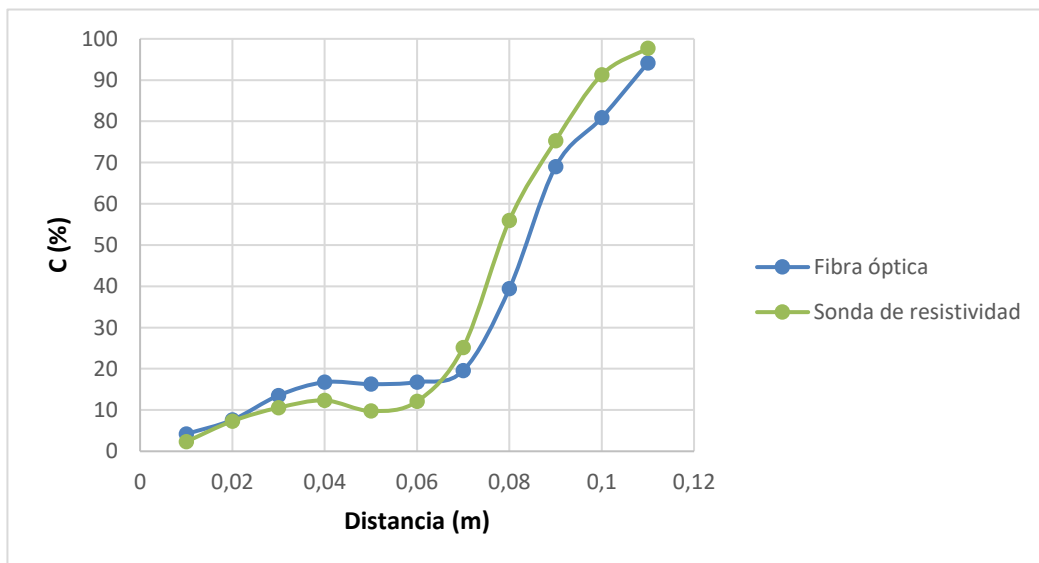


Figura 4.2 Gráfica de datos comparados.

Se puede observar que cuando la sonda está muy cerca y hay una alta concentración de burbujas toma una medida inferior. Puede deberse a que la velocidad de lectura de la fibra óptica es mucho más elevada que la planteada en este sistema.

Una vez nos alejamos y la concentración de burbujas baja, la lectura de la sonda de resistividad es un poco superior a la lectura de la fibra óptica.

4.2 Conclusiones.

En conclusión, se puede decir que el funcionamiento de este sistema es el esperado y que cumple con los requisitos demandados. Para la aplicación que nos compete podemos decir que los datos obtenidos son suficientes para realizar estudios de dinámicas de los flujos multifásicos a un coste muy inferior del sistema que se plantea sustituir.

La fibra óptica quizás para esta aplicación no es la opción más adecuada ya que es mucho más sensible y no necesitamos conocer el tipo de medio en el que está sumergido puesto que ya lo conocemos y es un equipo muy sensible para esta aplicación.

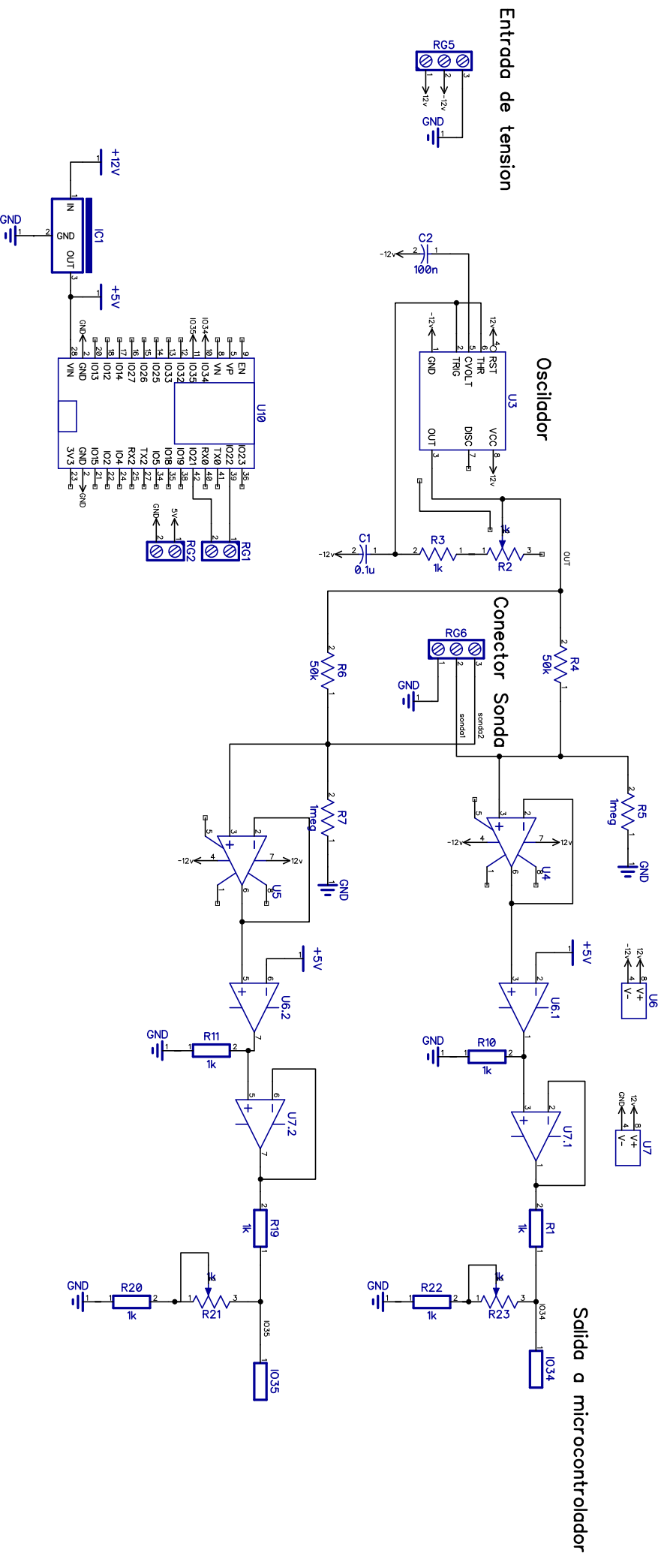
4.3 Trabajo futuro.

En una futura modificación sería muy interesante que se pudieran ver lecturas en tiempo real antes de iniciar una medición, como en una tarjeta de adquisición de datos. También se podría programar la opción de que el cliente le introdujera los datos al servidor de tiempo y de frecuencia de lectura en vez de estar preestablecidas.

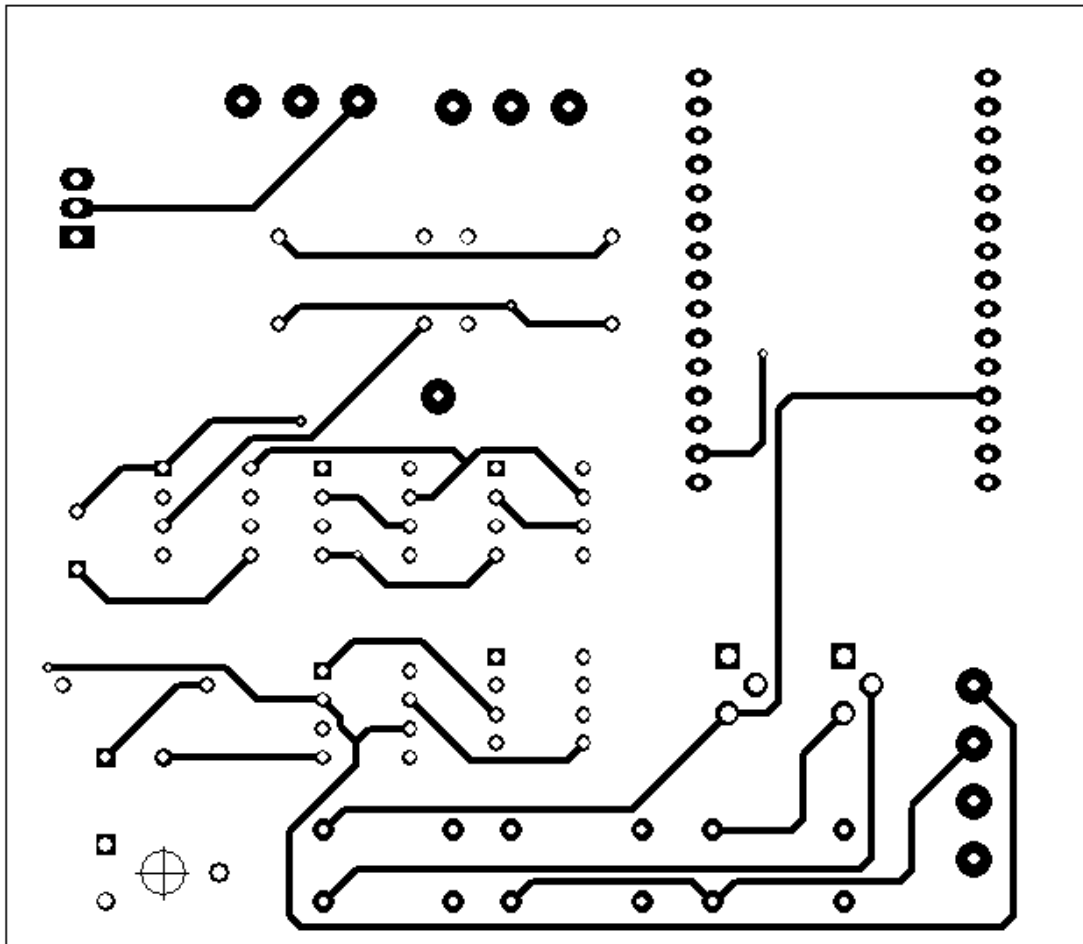
Bibliografía

- [energía hidráulica] <https://www.energyavm.es/energia-hidraulica/#:~:text=La%20energ%C3%ADa%20hidr%C3%A1ulica%20es%20la,monta%C3%B1as%2C%20crea%20r%C3%ADos%20y%20arroyos.&text=A%C3%BAas%20as%C3%AD%2C%20la%20energ%C3%ADa%20hidr%C3%A1ulica,cuanto%20a%20generaci%C3%B3n%20de%20electricidad.>
- [HM 162] <https://www.gunt.de/es/productos/hidraulica-para-ingenieros-civiles/ingenieria-hidraulica/flujo-en-canales-abiertos/canal-de-ensayo-309x450mm/070.16200/hm162/glct-1;pa-150;ca-179;pr-675.>
- [flujo multifásico] https://es.qaz.wiki/wiki/Multiphase_flow.
- [resalto hidráulico] <https://www.cuevadelcivil.com/2011/03/resalto-hidraulico.html#:~:text=Ocurre%20en%20el%20paso%20brusco,figura%20siguiente%20muestra%20este%20fen%C3%B3meno.>
- [refracción de la luz] https://es.wikipedia.org/wiki/%C3%8Dndice_de_refracci%C3%B3n.
- [geometría de la sonda] Niloufar Jamshidi, Navid Mostoufi (2018) Investigating bubble dynamics in a bubble column containing shear thinning liquid using a dual-tip probe.
- [ESP32] <https://www.naylampmechatronics.com/espressif-esp/382-modulo-esp-wroom-32-esp32-wifi.html.>
- [pantalla LCD] <https://medium.com/@elisabethlevana/playing-with-i2c-lcd-on-esp32-6b070e497eb.>

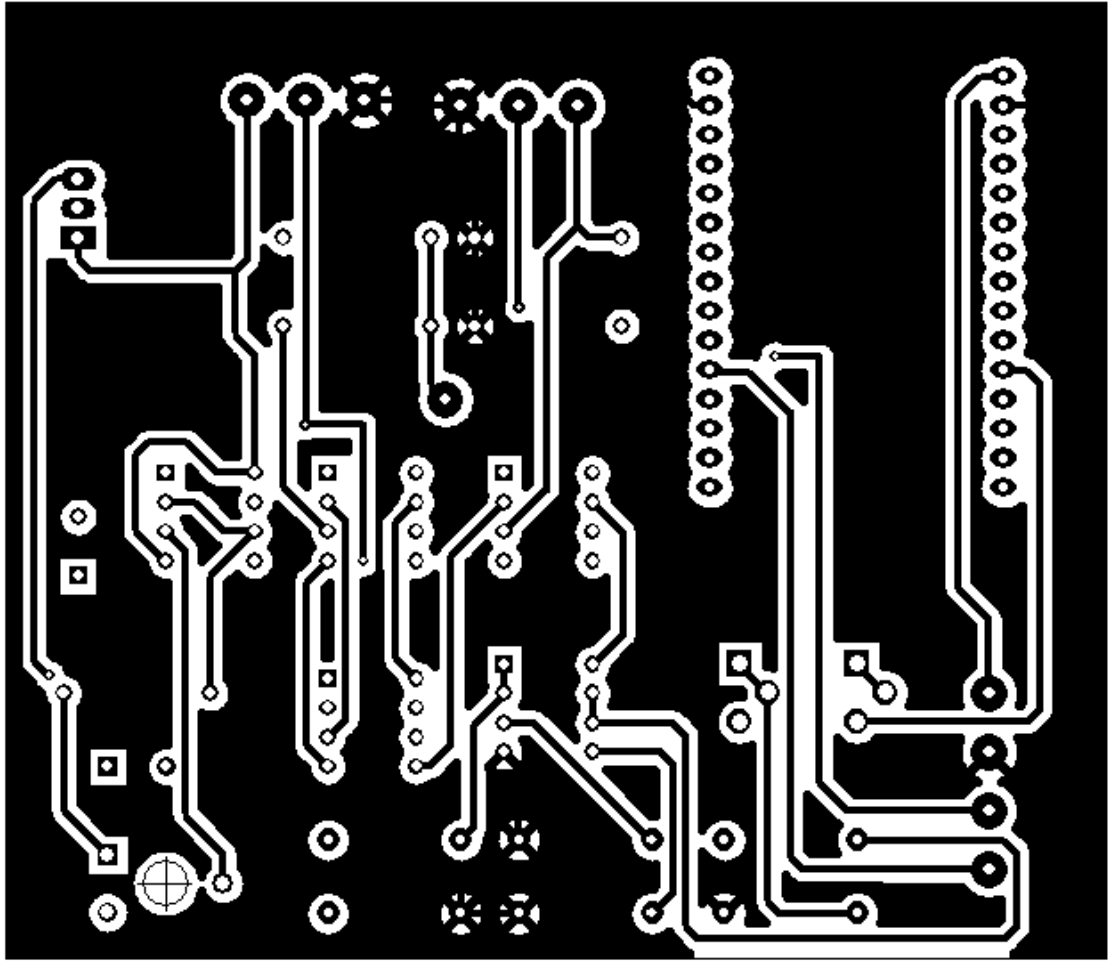
Circuito de acondicionamiento



Top



Bottom



Serigrafía

