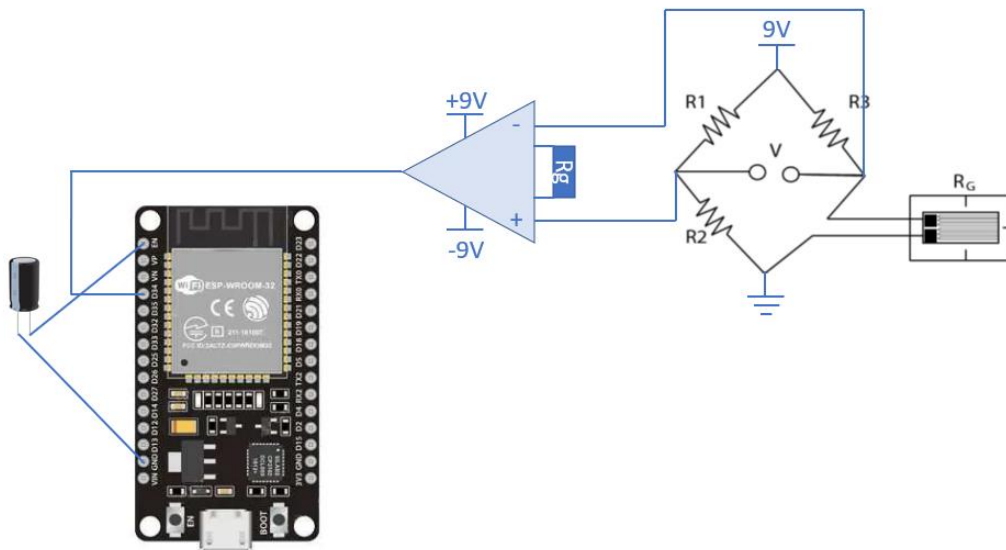


Metodología

Para el desarrollo del laboratorio nos basamos en el siguiente esquemático para nuestro circuito:



Utilizamos una placa de ESP32 para aprovechar su conexión inalámbrica por wifi. Agregamos un condensador de 10uF para poder correr los scripts realizados en el IDE de Arduino. Posteriormente utilizamos el pin 34 que es un pin de entrada análoga a digital. Para las resistencias del puente de Wheatstone y la resistencia de ganancia del amplificador, utilizamos resistencias iguales de 5Kohms; sin embargo, no encontramos resistencias de precisión. El amplificador que utilizamos fue el INA114 ya que el INA128 estaba imposible de conseguir. Para la fuente del puente y del amplificador utilizamos una pila de 9V, para evitar al máximo la saturación de señal del amplificador.

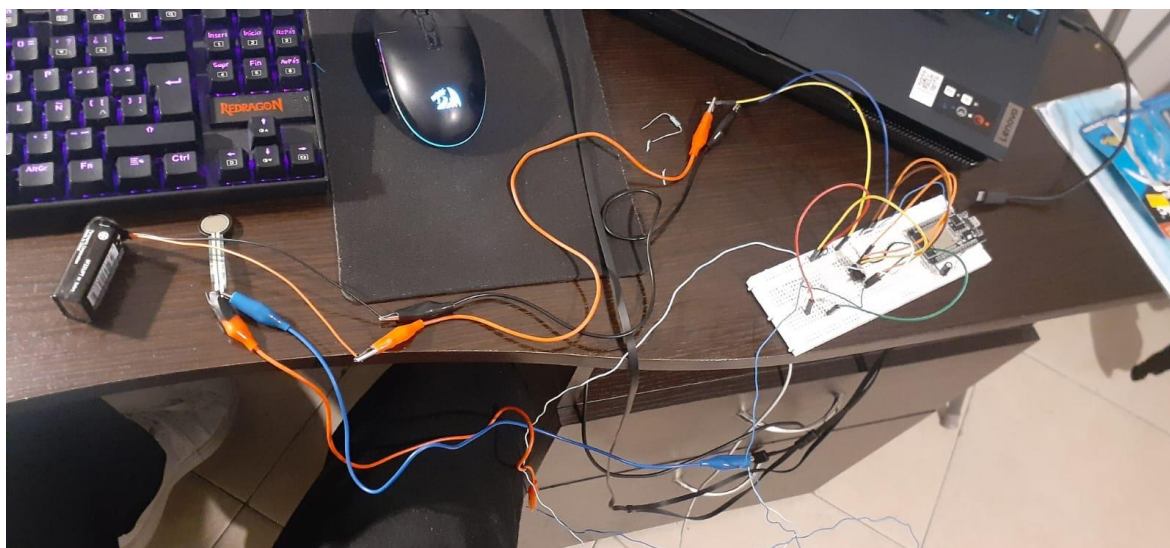
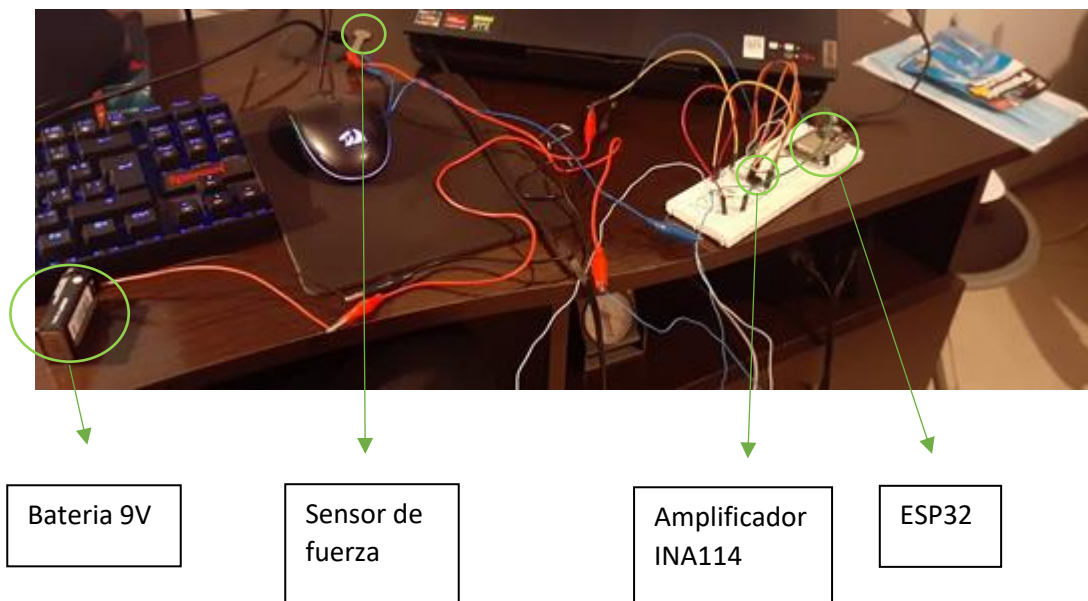
En cuanto al procesamiento de la señal, consultando en la literatura encontramos que para la conversión de una señal análoga a voltaje se debía utilizar el mayor numero que puede representar la tarjeta ESP32 que es de 4096 y el valor de mayor voltaje que la tarjeta puede detectar que en este caso es de 3.3V. De esta manera utilizamos la siguiente expresión para la conversión en voltaje:

$$\text{Voltaje} = \text{Lectura_sensor} * (3.3V/4096)$$

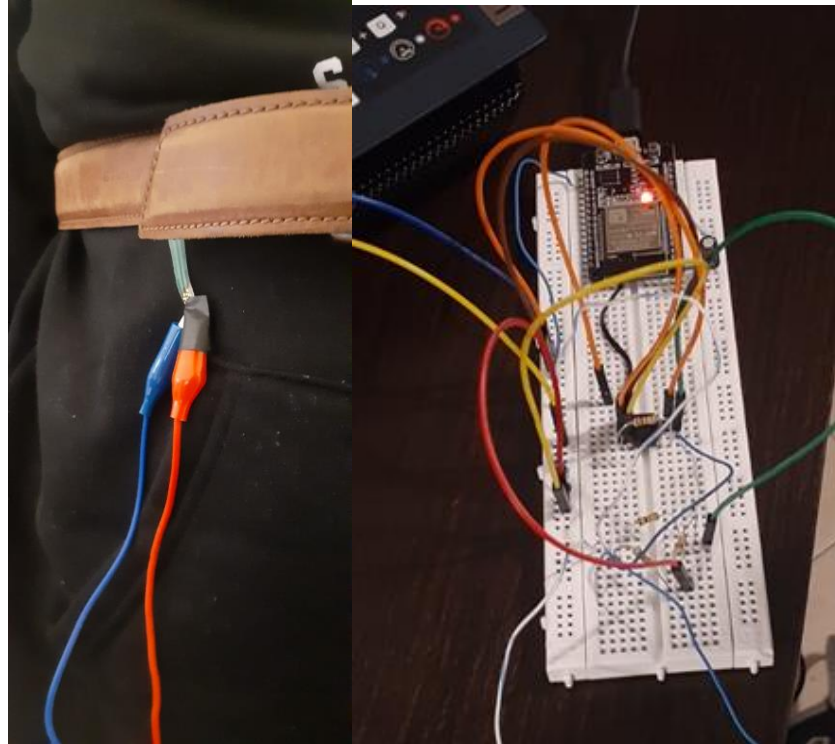
Para la transmisión de los datos utilizamos la conexión wifi de la tarjeta ESP32, estableciendo esta ultima como un servidor TCP/IP. Realizamos una pequeña aplicación en Python que actuó como cliente, recibiendo los datos recogidos por la tarjeta ESP32 por medio del sensor, convirtiendo el dato de lectura en voltios y graficándolos en tiempo real contra el tiempo en segundos.

Resultados

Este es el circuito realizado:



Conexiones en la protoboard y fijación del sensor:



El sensor fue fijado al cuerpo utilizando un cinturón. Posteriormente se graficaron los datos en tiempo real. En la gráfica se observan diferentes picos que representan las inspiraciones realizadas por el sujeto de prueba, mientras estas son registradas por medio de la aplicación que hicimos en Python. Los ejes de la gráfica corresponden al voltaje registrado por el ESP32 vs el tiempo en segundos.



Discusión y conclusiones

Consideramos que la escogencia del amplificador tiene que ver con la sensibilidad del dispositivo, ya que algunas veces las respiraciones muy suaves no fueron registradas. En la gráfica que presentamos se observan varios picos, en donde los mas altos corresponden a respiraciones moderadamente fuertes y los picos medianos a respiraciones normales, sin embargo, las respiraciones suaves en su mayoría no podían ser registradas. Creemos que con un sensor de fuerza mas grande o con más área de sensibilidad, nuestros resultados habrían sido mejores. También consideramos que la resistencia de ganancia R_g pudo haber sido de un valor menor a $5K\Omega$ ya que esto habría ayudado a captar las respiraciones más sutiles; sin embargo, se amplificarían también las respiraciones normales, saturando la señal en 3.3V lo que haría imposible diferenciar entre las señales de respiraciones suaves y fuertes ya que este voltaje es el máximo que la tarjeta ESP32 puede soportar por sus entradas análogas. Para solucionar esto se podrían realizar algunos ajustes en código para sobrepasar dicho limite, sin embargo, las implicaciones de realizar estos cambios a la tarjeta podrían afectar su desempeño y/o durabilidad. La utilización de servidores y clientes TCP/IP tiene una gran ventaja sobre el bluetooth ya que permite enviar los datos desde cualquier lugar con acceso a internet, sin tener en cuenta la distancia entre servidor y cliente.

Nota: El código de nuestro trabajo se encuentra adjunto a esta entrega