



Proyecto final: Baumanómetro digital.

Brigido Rivera Rodrigo Rolando, Flores Lozornio Gustavo Jafet, Rojas Contreras Erick Alejandro
Laboratorio de Electrónica, Equipo 8,
Facultad de Ciencias, 2019-2
Universidad Nacional Autónoma de México

Resumen. Se construyo un baumanometro digital con el uso de una placa arduino UNO, un sensor MPX2050DP y un amplificador operacional. Donde los datos se analizaron con un método de pendientes para encontrar los picos máximos de pulsos en un intervalo de tiempo.

1. INTRODUCCIÓN

La medición de la presión arterial es un elemento imprescindible en la medicina, ya sea para estudios clínicos de determinadas enfermedades, para el control de la hipertensión, o para la evaluación del estado de pacientes dentro de las unidades de cuidados intensivos y salones de operaciones.

Existen dos formas fundamentales de realizar la medición de la presión arterial: La primera es mediante los Métodos intermitentes, los que brindan presiones puntuales como la sistólica(PS), diastólica(PD) y media(PM), calculándolas en un periodo de tiempo que cubre más de un latido del corazón. La segunda es mediante los Métodos continuos, los que brindan presiones puntuales latido a latido, o la forma de la onda continua de la presión arterial, o ambos.

La presión arterial puede medirse de forma invasiva (directa) y no invasiva (indirecta), siendo los métodos invasivos potencialmente más exactos, pero con la complejidad e inconveniencia en su aplicación hacen que los métodos no invasivos sean por el contrario los más extendidos. Los equipos de medición no invasiva son más seguros y fáciles de usar que los invasivos y pueden ser utilizados en situaciones en las que la medición invasiva no es absolutamente necesaria.

2. DESARROLLO EXPERIMENTAL.

Para medir la presión se uso un sensor de presión, este sensor te da una diferencia de potencial que es proporcional a la presión que mide, la diferencia de potencial que entrega es muy pequeña como para que el Arduino la detecte, debido a esto se tuvo que usar un amplificador operacional, para amplificar el voltaje entregado por el sensor, una vez que se tuvo esto, se conecto el voltaje de salida a la terminal A0. Para funcionar el sensor necesita una alimentación de al menos 5 volts, esta se obtuvo del puerto de 5 volts del Arduino.

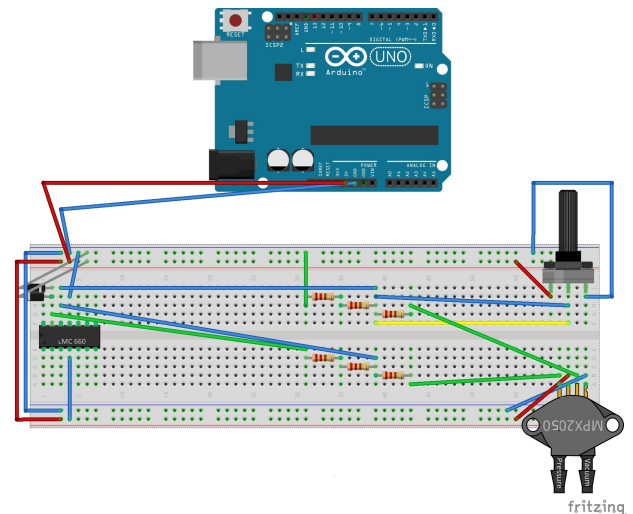


Fig. 1. Circuito usado para amplificar la señal.

El circuito y los materiales usados están en la siguiente imagen:

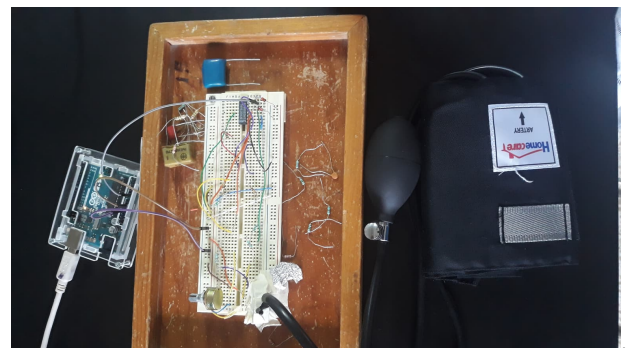


Fig. 2. Circuito y materiales usados.

El sensor de presión tiene dos entradas, una de ellas se inserta en una manguera que sale del brazalete que se coloca en el bíceps, la otra entrada del sensor queda expuesta a presión atmosférica, cuando se infla el brazalete, el sensor manda el voltaje que se amplifica. Este voltaje posteriormente se convertirá a una unidad de presión, mm de Hg para ser exactos.

Después de esto se prueba que el voltaje de salida del amplificador efectivamente cambie cuando la presión del brazalete varía, para esto se escribió un simple código que graficara los datos recogidos por el Arduino en la herramienta Serial Plotter, luego de verificarlo se procedió a realizar una prueba para ver si el sensor tenía la suficiente resolución como para que los pulsos de sangre se pudieran observar en el Serial Plotter del Arduino, se encontró que la forma en que se veía el pulso era en forma de picos.

Después de hacer todas las pruebas necesarias, ahora sí se procedió a elaborar el código que pudiera detectar los picos observados, que corresponden al pulso. El primer pico de presión se observa cuando la presión arterial es lo suficientemente grande como para ganarle a la presión ejercida por el brazalete, este primer pico corresponde a la llamada presión sistólica, de modo que el código debe detectar este primer pico y convertirlo a mm de mercurio, los picos posteriores corresponden a pulsos de sangre, la presión diastólica se supone debe corresponder al último pico de pulso, según los fabricantes de baumanómetros digitales esto no se toma así cuando se usa un sensor, la presión diastólica se toma cuando hay un pico en el cual su altura sea del 60 % del primer pico, esta se toma como la presión diastólica.

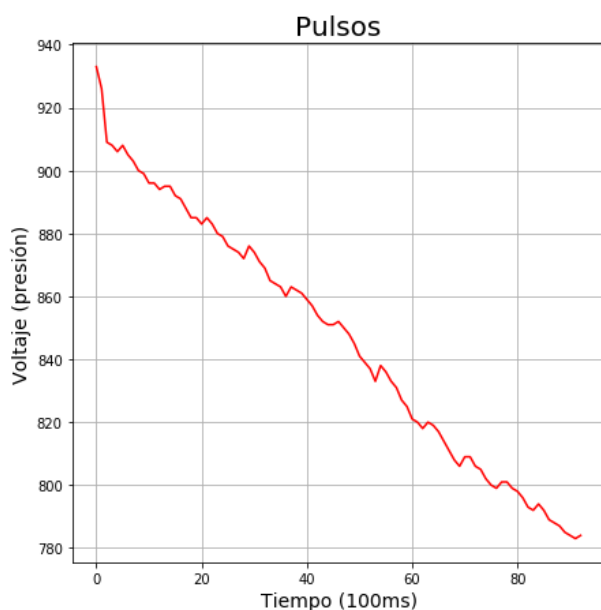


Fig. 3. Gráfica del voltaje vs el tiempo, los picos corresponden a los pulsos de presión debido a la presión arterial, son los que se necesitan detectar .

Para detectar esos picos se usa un método que consiste en detectar el cambio de signo en la pendientes, los datos que llegan al Arduino son enviados a python, una vez que llegan a python se convierten en unidades de presión (mm de Hg), esto se hace con una formula que se obtuvo calibrando el sensor con un manómetro de mercurio, se midió el voltaje que mandaba el sensor a diferentes presiones, se hizo varias veces y de ahí se obtuvo una estadística del valor del voltaje a cada presión. Las especificaciones del sensor dicen que su variación del voltaje respecto a la presión es de una forma lineal, así que teniendo esto en cuenta y los datos de la calibración, se hizo un ajuste a una recta, la recta que se obtuvo era del voltaje en función de la presión, despejando la presión se tiene la formula para convertir el voltaje a presión.

Con los datos en mm de Hg se procedió a analizarlos. Los datos se agregan a una lista, a estos datos pueden ser graficados, en la gráfica se puede observar como hay picos, cuando hay un pico el signo de la pendiente cambia, primero es positiva y luego se vuelve negativa, el programa básicamente detecta esto, cuando se detectan los picos hay un inconveniente, el programa detecta los cambios de signo, pero también entre pico y pico la pendiente cambia de signo de negativo a positivo, de modo que la siguiente parte del código se dedica a detectar si hay un máximo o un mínimo, nos interesan los máximos, de modo que estos los guardamos en una lista y de ahí se obtienen los valores de la presión diastólica y de la presión sistólica.

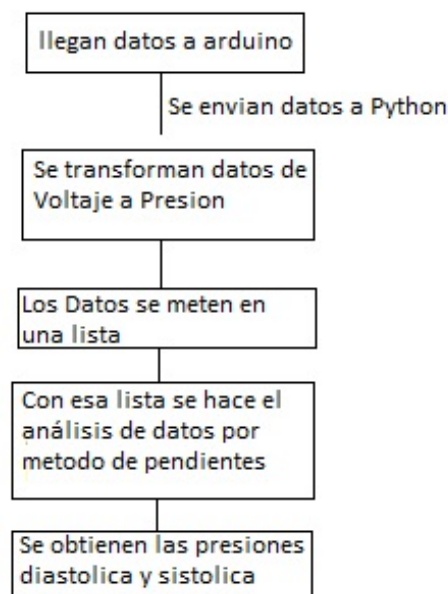


Fig. 4. Diagrama de flujo del proceso para obtener la presión .

3. RESULTADOS Y CONCLUSIONES.

En las diferentes pruebas que se realizaron se obtuvieron presiones dentro de un rango aceptable, los errores que se pudieron haber propagado en todo el proceso fueron debido a que el circuito con el que se trabajó tenía ruido que pudo haber llegado a afectar las mediciones, esto se trató de corregir, se pudo reducir el ruido considerablemente pero al final siempre estuvo ahí, este es el error que afectó más. Por otro cuando el sensor enviaba los datos, el Arduino los convertía a los en-

teros más próximos, no pudimos quitar este comportamiento, debido a esto también pudo existir algún error propagado. Así esto se podría mejorar si se construyera un circuito con un ruido aún menor y una calibración que tomará en cuenta los decimales que mandaba el sensor.

REFERENCIAS.

-
- [1] <https://www.luisllamas.es/controlar-arduino-con-python-y-la-libreria-pyserial/>