

Proyecto Control de Nivel de Agua con Sensor Capacitivo XKC-Y25-V

Daniel Enrique Perdomo Carvajal
u20192183270@usco.edu.co

Daniel Armando Torres Gaviria
u20191179884@usco.edu.co

Resumen— Por medio del presente documento se describe el funcionamiento de los sensores de presión diferencial de referencia: MPX5010dp & MPX5100DP y la implementación con una placa arduino para la medición de presión.

Términos de índice— Arduino, Sensores de Presión diferencial, Transductor.

I. INTRODUCCIÓN

La presión es en determinados procesos industriales una magnitud de vital importancia, ya sea en petroquímicas, producción de alimentos y bebidas, farmacéutica, metalúrgicas, entre muchas otras industrias, en donde su medición es necesaria para la mejora de dichos procesos por medio del control. Una medición defectuosa de esta magnitud física puede causar defectos en el producto terminado, daños en equipos por sobrepresión e incluso accidentes.

En el presente informe se encuentra el análisis y la implementación de un transmisor de presión, utilizando un sensor diferencial encargado de medir la diferencia de presión que hay en un sistema cerrado o abierto.

En esta práctica se realizará el montaje de un sistema que permite la medición de la presión del interior de un cilindro, dicho montaje consta de una jeringa que simula la variación de presión en un cilindro, cuyo émbolo será quien realice el cambio de la presión interna del mismo, dicha presión será medida por la entrada de alta del sensor mientras su entrada de bajo está expuesta libremente.

La diferencia entre las dos entradas genera una señal de salida en forma de Tensión, la cual tomaremos y convertiremos en una señal reconocible para Arduino implementando las fórmulas de conversión. Y como indicador visual se implementará una LCD que nos imprime el valor de presión existente en KPa (kilopascales) y el voltaje de salida que está generando el sensor.

El montaje físico incluye la conexión de 10 leds que indican la presión que se ejerce sobre el sensor.

II. OBJETIVO

- Medir la presión diferencia con el sensor MPX5010 y MPX5100, adecuar la señal con un microcontrolador para imprimirla en el lcd 16x2

III. MATERIALES, EQUIPOS Y/O SOFTWARE

- Elementos: Placa Arduino UNO, resistencias, LED's, jumpers macho-macho, jumpers hembra-macho, jeringa, lcd 16x2 , sensor MPX5010 y MPX5100

IV. MARCO TEÓRICO

A. Sensor de presión

Existen diferentes sensores de presión diferencial en el mercado de los microcontroladores conocidos como Sensores de presión tipo MPX, MPX 53 DP, MPX53GP, MPX2010DP, MPX2010GP, MPX2050DP, MPX10DP, MPX5010DP, etc, y básicamente varían en la capacidad de presión que es capaz de medir el sensor.

Los sensores de baja presión diferencial/manométrica basados en MEMS, como se muestra en la figura 1, miden la diferencia de presión entre las fuentes cuando se aplica presión a ambos lados del sensor (diferencial) y presión atmosférica (manométrica).



Fig. 1 . Sensor de Presión Diferencial/Manométrica MPX5010DP (0 ~ 10 kPa)

B. Sensor MPX5010

Es un sensor de presión de silicio monolítico de última generación diseñado para una amplia gama de aplicaciones, pero particularmente aquellas que emplean un microcontrolador o microprocesador con entradas ADC.

C. Características del Sensor MPX5010 y MPX5100

MPX5010 puede medir una presión de 10 kPa y MPX5100 puede

medir una presión de 100 kPa (14.5 psi). Da salida lineal sobre el rango como se muestra en la figura 2 extraída de la hoja de datos.

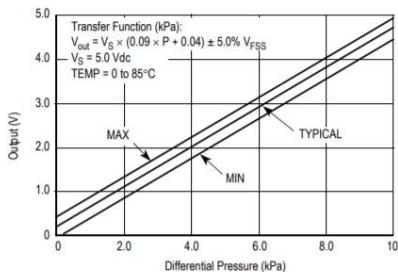


Figure 4. Output vs. Pressure Differential

Fig. 2 . voltaje de salida vs presión diferencial

el MPX5010DP que ya entrega en su salida un voltaje de 0v – 5v ideal para microcontroladores.

Como vemos en la figura 2, la ecuación para obtener la presión del sensor viene dada por:

$$p = \frac{V_{out} - 0.04 V_s \pm Tol}{0.09 V_s} \quad (1)$$

donde Vs es el voltaje de alimentación y Vout es el voltaje de salida del sensor, o sea, el que leemos con arduino en bytes y lo transformamos a voltaje) y Tol es la tolerancia, un ajuste que debemos hacerle al sensor para calibrar la medida.

Los pines de conexión para el MPX5100DP son los siguientes:

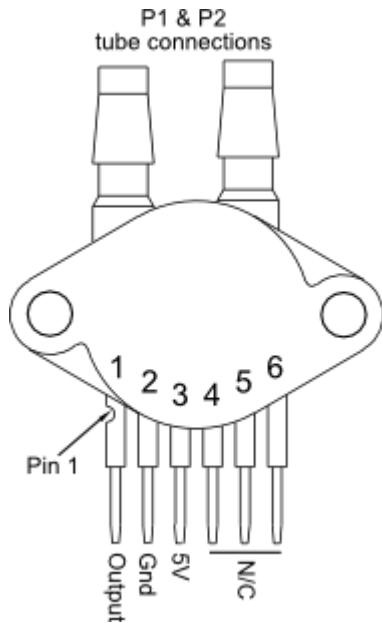


Fig. 3. diagrama de conexiones del un sensor MPX5100DP

V. PROCEDIMIENTO

En este caso vamos a valernos de la presión hidrostática presentada anteriormente para la medición del nivel al interior de una jeringa.

Según el datasheet, el sensor no puede entrar en contacto directo con un líquido, en este caso. no es necesario ya que estamos midiendo la presión de un gas.



Fig. 4 . topología de la jeringa

Se aplica presión sobre el émbolo de la jeringa que está conectado al sensor de presión en la entrada de Alta mediante una manguera.

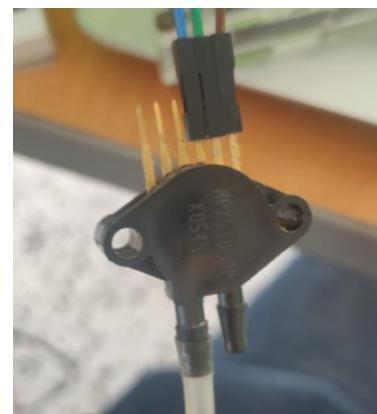


Fig. 5 . sensor de presión diferencial MPX5100DP

Dicho émbolo cada vez que es más introducido genera una presión mayor cada vez la cual medida y convertida en una señal eléctrica,



Fig. 6. implementación de Sensor de Presión Diferencial MPX5100DP con el Arduino

mediante la cual es recibida por el Arduino y transformamos dicha tensión en una señal digital que implementamos para poder obtener las variables que deseamos encontrar mediante la aplicación de fórmulas en el código del programa.

```
sensor_de_presion_diferencia_mpx5100_2 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

sensor_de_presion_diferencia_mpx5100_2
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
digitalWrite(9,LOW);
digitalWrite(10,LOW);
digitalWrite(11,LOW);

}

void loop()
{
aux=0;
aux2=0;
for(i=0;i<10;i++){
aux = aux + (float(analogRead(A0))*5.0/1023.0); //voltaje 0-5v
aux2 = aux2 + (float(analogRead(A0))*5000.0/1023.0); //voltaje milivoltios
delay(5);
}
Vout=(aux/10.0); //Voltaje del sensor
P=(aux2/10.0)/45; //Presión en Kpa

lcd.clear();
lcd.setCursor(0,0);lcd.print("La Presion es:");
lcd.setCursor(5,1);lcd.print("KPa");
lcd.setCursor(5,1);lcd.print("V:");
lcd.setCursor(0,1);lcd.print(P);
lcd.setCursor(12,1);lcd.print(Vout);
}

Fotografia de Tomado
```

Fig. 7. algoritmo implementado en el Arduino

Tras realizar dicha conversión el resultado obtenido será visible en la pantalla LCD que se implementa como indicador.



Fig. 8 . lectura del LCD, primera muestra

El código tiene como función hacer la toma de 10 muestras y hace un promedio para poder obtener un resultado menos variante o inestable.

```
aux=0;
aux2=0;
for(i=0;i<10;i++){
aux = aux + (float(analogRead(A0))*5.0/1023.0); //voltaje 0-5v
aux2 = aux2 + (float(analogRead(A0))*5000.0/1023.0); //voltaje milivoltios
delay(5);
}
Vout=(aux/10.0); //Voltaje del sensor
P=(aux2/10.0)/45; //Presión en Kpa
```

Fig. 9.

También están presentes leds indicadores que se iluminan según la cantidad de presión que se esté ejerciendo sobre el sensor.

en la siguientes figuras (10, 11, 12 ,13 y 14) tenemos las diferentes lecturas de presión, variando la compresión de un gas en el interior de un cilindro

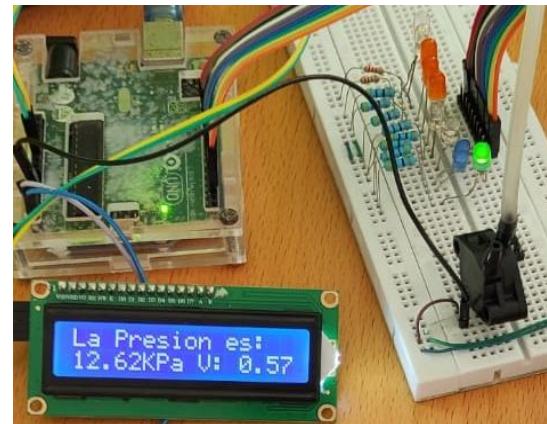


Fig. 10.

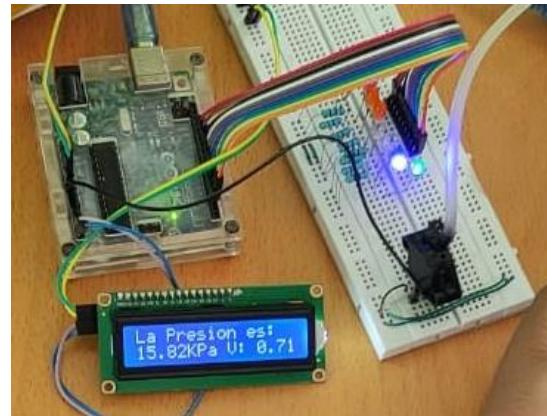


Fig. 11.

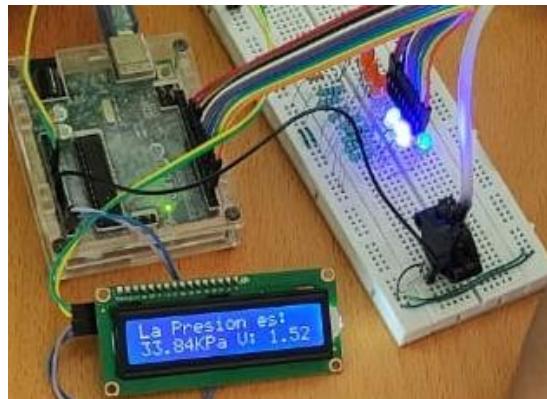


Fig. 12.

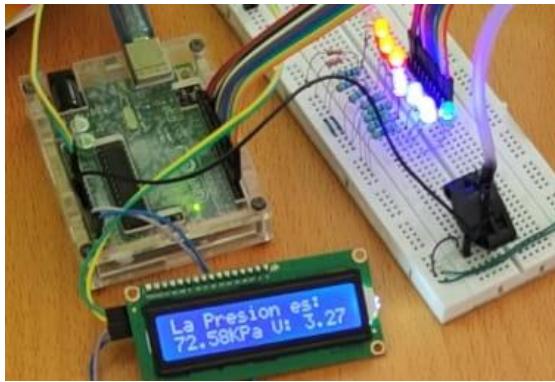


Fig. 13.

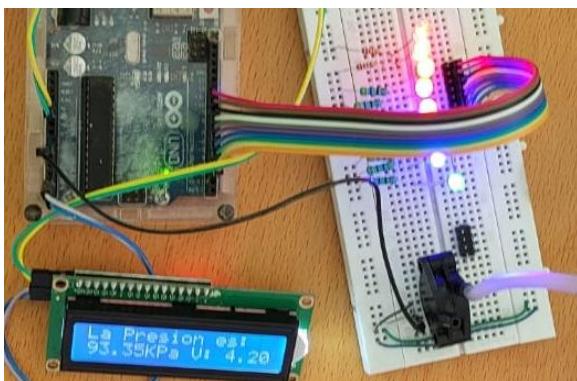


Fig. 14.

Se observó una proporcionalidad directa en cuanto al voltaje entregado por el transductor respecto a la presión aplicada y medida, como se observa en la figura 15 .

teniendo en cuenta las figuras (10, 11 12 ,13 y 14) construimos la tabla.1 para después construir una gráfica que relaciona la salida de voltaje del sensor con el valor que nos entrega el sensor con referencia a la presión y esta la podemos adecuar a la unidad que queramos o que nos convierta el microcontrolador

Vout (v)	Pa (KPa)	psi
0,24	5,3	0,76
0,71	15,82	2,29

1,52	33,84	4,9
3,27	72,58	10,52
4,2	93,35	13,53

Tabla. 1 . datos obtenido al variar el émbolo de la jeringa

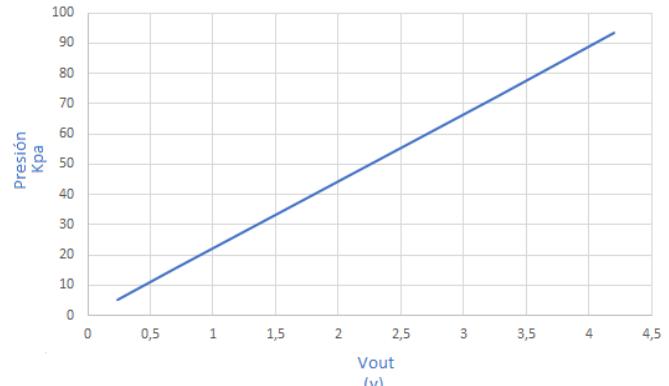


Fig. 15. salida del sensor vs presión diferencial

IX. CONCLUSIONES

- Se concluyó que el sensor es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud normalmente eléctrica, con la cual somos capaces de cuantificar y manipular.
- Se comprobó el correcto funcionamiento del sensor de presión diferencial, haciendo las mediciones con la diferencia de presión entre dos puntos las cuales eran plasmadas en un display para su óptima visualización.
- Se comprende la importancia del dispositivo en la automatización y control de procesos en los cuales se necesite medir caídas de presión, niveles de fluidos y caudales.

IIX. REFERENCIAS

[1]7 junio 2020. [En línea]. Available: <https://www.jsindustrial.com.pe/medidores-de-presion-industrial/#:~:text=Es%20conocida%20la%20importancia%20de,por%20sobrepresi%C3%B3n%20e%20incluso%20accidentes..> [Último acceso: 21 abril 2022].

[2] Charles. k. Alexander-Matthew Sadiku, ("capítulo 5") Amplificadores Operacionales - FUNDAMENTOS DE CIRCUITOS ELECTRÓNICOS. 3era edición

[3] Es.wikipedia.org. 2020. *Amplificador Operacional*. [online]

Available at:

<https://es.wikipedia.org/wiki/Amplificador_operacional#:~:text=Un%20amplificador%20operacional%2C%20a%20menudo,dos%20entradas%20y%20una%20salida.> [Accessed 3 December 2020].

[4]Coughlin, R. and Driscoll, F., 1999. *Amplificadores Operacionales Y Circuitos Integrados Lineales*. México: Pearson Educación.

[5]Franco, S. and Enríquez Brito, J., 2005. *Diseño Con Amplificadores Operacionales Y Circuitos Integrados Analógicos*. México, D.F.: McGraw-Hill Interamericana.

anexo:

El código implementado en el programa y la placa Arduino es el siguiente:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,12,2);

double Vout,P,Vs=5.0;
double aux, aux2;
int i;

void setup()
{
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  pinMode(2,OUTPUT); //pin digital 2, Salida, led verde.
  pinMode(3,OUTPUT); //pin digital 3, Salida, led azul.
  pinMode(4,OUTPUT); //pin digital 4, Salida, led azul.
  pinMode(5,OUTPUT); //pin digital 5, Salida, led azul.
  pinMode(6,OUTPUT); //pin digital 6, Salida, led azul.
  pinMode(7,OUTPUT); //pin digital 7, Salida, led naranja.
  pinMode(8,OUTPUT); //pin digital 8, Salida, led naranja.
  pinMode(9,OUTPUT); //pin digital 9, Salida, led naranja.
  pinMode(10,OUTPUT); //pin digital 10, Salida, led rojo.
  pinMode(11,OUTPUT); //pin digital 11, Salida, led rojo.
}

/*Subprocesos de los leds -----*/
void state_9()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,HIGH);
  digitalWrite(10,HIGH);
  digitalWrite(11,HIGH);
}

void state_8()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,HIGH);
  digitalWrite(10,HIGH);
  digitalWrite(11,HIGH);
}

void state_7()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,HIGH);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}

void state_6()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}

void state_5()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}

void state_4()
{
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,HIGH);
  digitalWrite(10,HIGH);
  digitalWrite(11,LOW);
}
```

```

{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}
void state_3()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}
void state_2()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}
void state_1()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
}

digitalWrite(10,LOW);
digitalWrite(11,LOW);
}
void state_0()
{
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
}
/*-----
void loop()
{
  aux=0;
  aux2=0;
  for(i=0;i<10;i++){
    aux = aux + (float(analogRead(A0))*5.0/1023.0); //Conversión de
    voltaje 0-5v
    aux2 = aux2 + (float(analogRead(A0))*5000.0/1023.0);
    //Conversión de voltaje milivoltios
    delay(5);
  }
  Vout=(aux/10.0); //Voltaje del sensor
  P=(aux2/10.0)/45; //Presión en Kpa

  lcd.clear();
  lcd.setCursor(0,0);lcd.print("La Presion es:");
  lcd.setCursor(5,1);lcd.print("KPa");
  lcd.setCursor(9,1);lcd.print("V:");
  lcd.setCursor(0,1);lcd.print(P);
  lcd.setCursor(12,1);lcd.print(Vout);

  /*Condiciones de los leds -----
  if (P<=14.0)          /*presiones bajas*/
  {
    state_0();
  }
  if ((P>=14.1)&&(P<=20.0))
  {
    state_1();
  }
  if ((P>=20.1)&&(P<=30.0))
  {
    state_0();
  }
}

```

```
{  
    state_2();  
}  
if ((P>=30.1)&&(P<=40.0))  
{  
    state_3();  
}  
if ((P>=40.1)&&(P<=50.0))      /*presiones medias*/  
{  
    state_4();  
}  
if ((P>=50.1)&&(P<=60.0))  
{  
    state_5();  
}  
if ((P>=60.1)&&(P<=70.0))  
{  
    state_6();  
}  
if ((P>=70.1)&&(P<=80.0))      /*presiones altas*/  
{  
    state_7();  
}  
if ((P>=80.1)&&(P<=90.0))  
{  
    state_8();  
}  
if (P>=90.1)  
{  
    state_9();      /*presiones muy altas*/  
}  
delay(500);  
}
```

