

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

***ING. EN SISTEMAS COMPUTACIONALES***



## Proyecto Final

Sensor de humedad, Circuito de Acondicionamiento de Señal (CAS),  
Convertidor analógico a Digital (ADC) & Microcontrolador

**ASIGNATURA:** Instrumentación y Control

**PROFESOR:** Cervantes De Anda Ismael

**GRUPO:** 5CV1

**ALUMNOS:**

- García Quiroz Gustavo Ivan
- Ortiz González Alan
- Romero Hernández Oscar David

**Fecha de entrega: 21/01/2024**

## INDICE

OBJETIVO .....	4
MATERIAL EMPLEADO.....	5
Instrumentos y Herramientas:.....	5
Componentes Electrónicos: .....	5
INVESTIGACIÓN TEORICA.....	6
Convertidor Analógico-Digita (ADC)l.....	7
Protocolo RS232.....	9
Estándares de RS232 .....	10
Niveles de Voltaje RS232 .....	10
Velocidad en Baudios .....	11
Modo de Operación .....	11
Formato de Protocolo .....	12
Circuito impreso, tarjeta de circuito impreso o PCB.....	12
Composición.....	13
Métodos de impresión de un circuito .....	14
DESARROLLO DEL PROYECTO.....	15
DISEÑO DEL CIRCUITO DE ACONDICIONAMIENTO DE SEÑAL (CAS) .....	16
Cálculos .....	16
DISEÑO DEL CONVERTIDOR ANALOGICO DIGITAL DE 8 BITS (ADC) .....	19
Cálculos .....	21
DISEÑO DEL CIRCUITO GENERADOR DE SEÑAL BAJO PROTOCOLO RS-232....	23
Método de planchado para imprimir circuitos.....	31
Paso 1. Impresión del circuito .....	31

Paso 2. Transferencia a la placa de cobre .....	33
Paso 3. Eliminación del cobre sobrante.....	34
Paso 4. Acabado final .....	35
Paso 5. PCB con los componentes.....	35
Programa en pc .....	37
Explicación .....	37
Script 'humedad.py' .....	37
Script 'humedad_interfaz.py' .....	39
Cálculos .....	42
CAS .....	42
ADC.....	43
Divisor de voltaje .....	43
TABLAS DE MEDICIONES.....	45
PRUEBAS Y TOMA DE MEDICIONES DEL CIRCUITO ARMADO .....	46
CONCLUSIONES.....	51
Conclusiones generales.....	51
Conclusiones específicas .....	51
Anexo Código.....	53
Anexo Código para programar al microcontrolador.....	53
Anexo Código para el proyecto .....	56
Script 'humedad.py' .....	56
Script 'humedad_interfaz.py' .....	57
BIBLIOGRAFÍA .....	60

## OBJETIVO

El propósito fundamental de este proyecto es poner en práctica los conocimientos adquiridos sobre el Circuito de Acondicionamiento de Señal (CAS) en relación con un sensor de humedad. Se busca lograr que el sensor proporcione una salida de voltaje, aplicando los principios aprendidos sobre el funcionamiento del CAS.

Asimismo, se aplicará el funcionamiento del Convertidor Analógico a Digital (ADC) para transformar el valor de voltaje de entrada proveniente del CAS en un código binario. Este paso es crucial para la posterior interpretación y procesamiento de la información por parte del microcontrolador.

El proyecto también tiene como objetivo aplicar los conocimientos sobre el microcontrolador PIC16F628A para establecer la conexión del circuito con la computadora. Esto permitirá expresar los valores binarios obtenidos a través del ADC y el CAS en una aplicación de fácil acceso y comprensión para el usuario final.

Y por último se realizará una aplicación para la computadora que traduzca los valores binarios a decimales mediante la implementación de programación y fórmulas correspondientes al ADC y el CAS. Además, la aplicación mostrará los porcentajes y valores de humedad relativa de manera clara y concisa.

## MATERIAL EMPLEADO

### Instrumentos y Herramientas:

- Multímetro
- Protoboard
- Fuente de VCD variable
- 2 Puntas Caimán – Caimán
- 6 Puntas Banana – Caimán
- Placa fenólica (lisa)
- Impresora láser
- Cúter
- Regla metálica
- Hojas de papel couche
- Tabla
- Franela
- Masking tape
- Plumón de tinta permanente (fino y grueso)
- Goma (borrador de tinta)
- Plancha
- Bandeja de plástico
- Cloruro férrico
- Flux para limpiar impresos

### Componentes Electrónicos:

- 2 Capacitores de 0.1  $\mu$ F
- 1 ADC0804
- 1 PIC16F628A
- 1 Push button
- 1 Capacitor de 150 pF
- 1 Capacitor de 10  $\mu$ F
- 10 Resistencias de 390  $\Omega$
- 10 Diodos LEDs
- 1 Resistencia de 10k  $\Omega$
- 1 Resistencia de 100k  $\Omega$
- 1 Resistencia de 22k  $\Omega$
- 1 Resistencia de 4.7k  $\Omega$
- 1 Resistencia de 150k  $\Omega$
- 2 Amplificadores operacionales LM358
- 1 Sensor HMZ-433A
- Módulo Convertidor USB a Serial RS232 TTL (PL2303)

Con este conjunto de herramientas y componentes, se llevará a cabo la implementación y ensamblaje del sistema que permitirá la medición de humedad y su posterior visualización en una interfaz de usuario amigable.

## INVESTIGACIÓN TEORICA

Para el desarrollo del proyecto vamos a hacer uso del medidor de humedad (HMZ-433A) que nos va a brindar facilidades con el desarrollo del proyecto requerido, en este caso, los sensores son dispositivos utilizados en instrumentación para medir diversas magnitudes físicas, como temperatura, presión, humedad, luz, fuerza, posición, velocidad, entre otras. Estos sensores son esenciales en una amplia gama de aplicaciones industriales, científicas y de consumo.

Para comprender completamente a este instrumento, debemos saber que hay varios tipos de sensores como:

- *Sensores de temperatura:* Miden la temperatura en grados Celsius o Fahrenheit.
- *Sensores de presión:* Se utilizan para medir la presión de fluidos y gases.
- *Sensores de humedad:* Detectan la cantidad de humedad en el aire o en una sustancia.
- *Sensores de luz:* Detectan niveles de luz y se utilizan en aplicaciones como fotografía, automatización industrial y sensores de movimiento.
- *Sensores de fuerza y carga:* Miden fuerzas y cargas.
- *Sensores de posición y velocidad:* Detectan la posición o la velocidad de un objeto en movimiento.
- *Sensores de nivel:* Se emplean para medir el nivel de líquidos o sólidos en tanques o recipientes.

Son sensores que generan una señal eléctrica a partir de la magnitud medida, sin alimentación. Se basan en efectos físicos reversibles, por lo que puede encontrarse el accionador “dual “. Estos efectos, si se producen de forma involuntaria, se convierten en posibles interferencias.

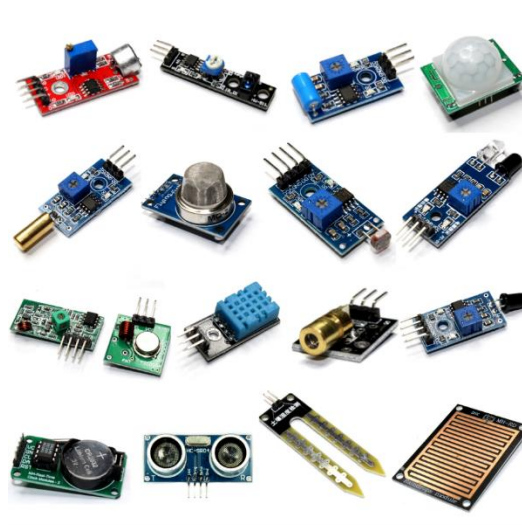
Esto es una alternativa para medir muchas de las magnitudes ordinarias como, por ejemplo: temperatura, fuerza, presión, y otras magnitudes afines.

En esta parte se toma en cuenta las fuentes de Interferencia cuyos efectos pueden producirse inadvertidamente en los circuitos. Ejemplos:

- Ruidos (Mecánicos, Magnéticos, Eléctricos, Térmicos, Electrónicos).
- Fuerzas Electromotrices.
- Vibraciones.
- Presencia de Dieléctricos o Potenciales galvánicos.

Los Medidores de Humedad son instrumentos de medición que utilizamos para medir el porcentaje de humedad presente en materiales.

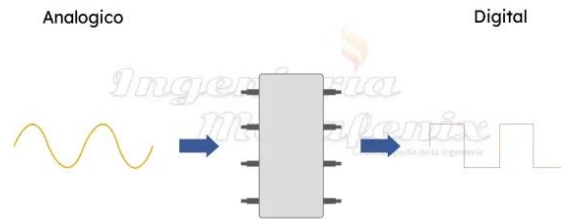
Estos instrumentos varían según la aplicación que se necesite como por ejemplo en madera, papel, cartón, granos, PH, tierra, balanzas de humedad, etc.



*Imagen 1 / Diferentes Tipos De Sensores*

## Convertidor Analógico-Digital (ADC)

En el proyecto, vamos a hacer uso del dispositivo encargado de realizar esta conversión, conocido como *Convertidor Analógico-Digital (ADC)*; el cual es aquel que permite a los sistemas electrónicos capturar, procesar y almacenar datos provenientes de sensores, dispositivos de medida y otras fuentes analógicas.



*Imagen 2 | Conversión analógico a digital*

Estas señales variables o mejor conocidas como señales analógicas, las podemos captar en toda la naturaleza a través de lo que conocemos como magnitudes físicas, por ejemplo; la luz, el sonido, la temperatura, el viento, etc.



*Imagen 3 | Diferentes magnitudes*

Con todo esto, se nos aclara que el ADC es un componente crítico en la intersección de la electrónica desempeñando un papel vital en la adquisición y procesamiento de datos en nuestra materia, por ende, en cada circuito que lleguemos a realizar.

Estos convertidores funcionan a través de un proceso llamado modulación por impulsos codificados, este sistema consta de cuatro partes esenciales que describen como es el proceso de pasar de una señal variable a una discreta.

1. **Muestreo:** En esta parte como su nombre lo indica se toman muestras de la señal analógica, que sirven como referencia para los siguientes pasos.



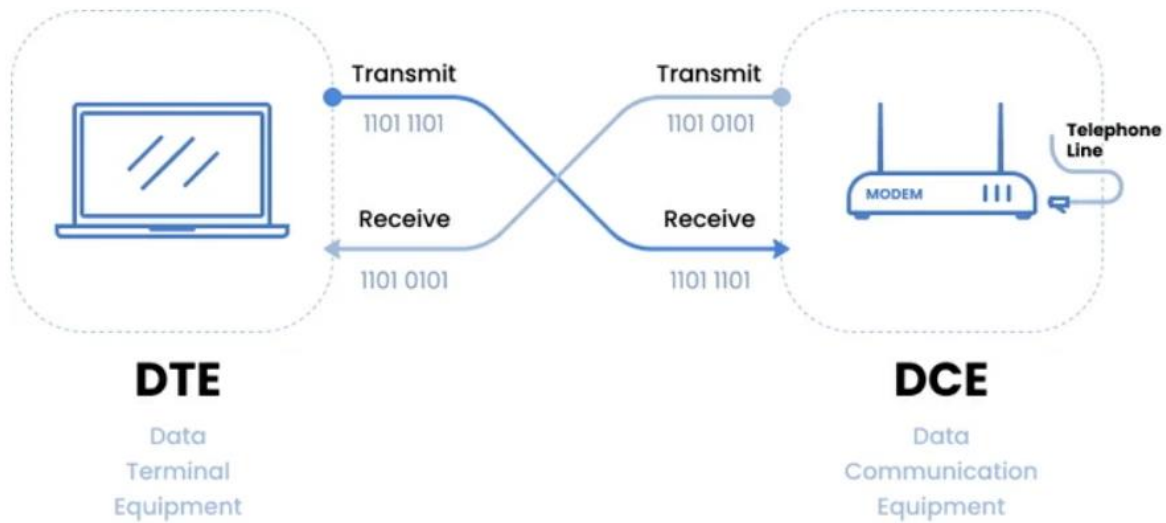
2. **Cuantificación:** La cuantificación se utiliza para asignarle un valor a todas las muestras tomadas, dependiendo de la precisión que se tenga el valor será más cercano al real.
3. **Codificación:** Una vez que los datos tienen su valor numérico el sistema los convierte en números binarios que representan las variaciones de la señal original.
4. **Compresión:** Debido a que la cantidad de datos que se guardan de una señal es bastante grande, y en ocasiones tiene que ser transmitida hacia un dispositivo exterior, y este tiene una tasa de lectura baja, es necesario comprimir la información.

El ADC es una piedra angular en la interfaz entre el mundo analógico y el mundo digital, permitiendo a los sistemas electrónicos capturar, procesar y almacenar datos provenientes de sensores, dispositivos de medida y otras fuentes analógicas.

Se presento aquí la prueba de laboratorio de un circuito realizado específicamente para experimentar, de manera básica, con la comunicación entre un microcontrolador PIC 16F628A y el puerto serie de una PC (estándar RS-232).

## **Protocolo RS232**

La especificación formal del protocolo RS232 lo define como una interfaz de transmisión de datos binarios en serie entre el equipo DTE y el equipo DCE. Un DTE o equipo de terminal de datos, como un ordenador, se encuentra en un extremo de la conexión serie RS232. El equipo de comunicación de datos (DCE), como un módem, se encuentra en el otro extremo de la conexión.



*Imagen 2 transmisión de datos binarios*

Este diagrama muestra la conexión entre un DTE RS232 (ordenador) y un DCE RS232 (módem). En el ejemplo, el DTE envía datos binarios "11011101" al DCE y el DCE transmite la secuencia binaria "11010101" al dispositivo DTE.

RS232 define los estándares eléctricos, los modos operativos, los niveles de voltaje comunes y la cantidad de bits que se transfieren entre un DTE y un DCE. Es el protocolo de transmisión estándar utilizado en líneas telefónicas terrestres.

### **Estándares de RS232**

Las especificaciones eléctricas de la interfaz RS232 se definieron en 1969. Describen los voltajes eléctricos, la velocidad en baudios, los modos operativos, la impedancia de línea y la velocidad de respuesta utilizada por el protocolo.

### **Niveles de Voltaje RS232**

Los voltajes de línea del RS232 varían de -25V a +25V. Definidos como voltajes de señal y voltajes de control.

Signal Voltage Levels	Logical State	Control Signal Voltage Levels (volts)	Logical State
-3 to -25	OFF (0)	-3 to -25	OFF (0)
+3 to +25	ON (1)	+3 to +25	ON (1)

*Imagen 3 Niveles de Voltaje RS232*

Un voltaje de señal entre -3V y -25V representa un '1' lógico y los voltajes entre +3V a +25V representan un '0' lógico. Las señales de voltaje de control usan la lógica negativa donde el "1" lógico indica -3V a -25V y un "0" lógico indica +3V a +25V. Una lectura de voltaje entre -3V y +3V se considera indeterminada.

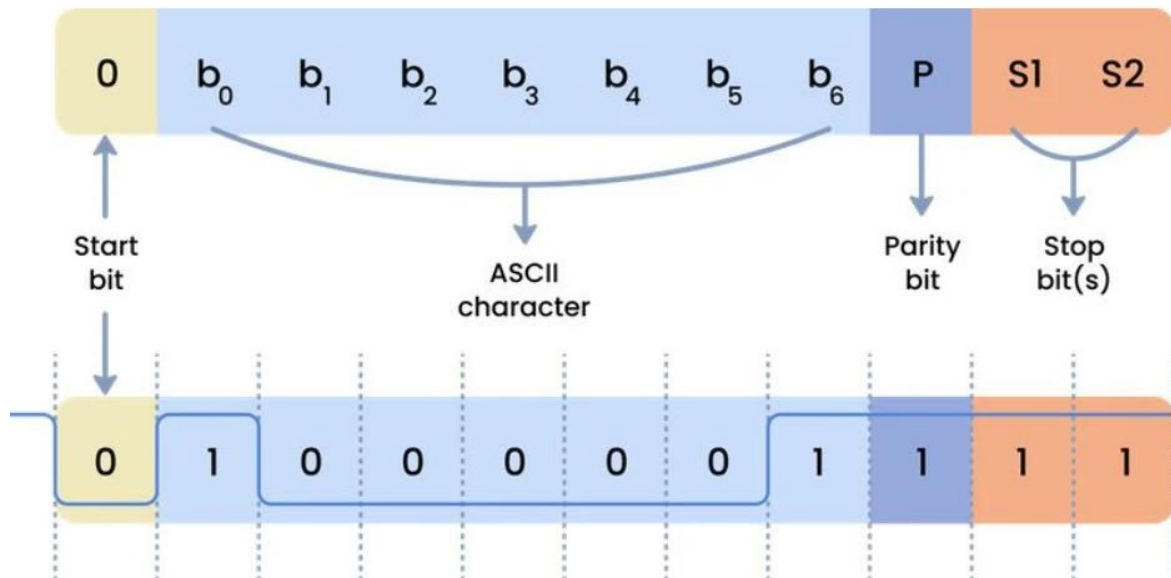
### **Velocidad en Baudios**

La velocidad en baudios describe el número de bits binarios que se transmiten por segundo. En el protocolo RS232, se admiten velocidades de transmisión de 110 a 230400. Las velocidades en baudios 1200, 4800, 9600 y 115200 son las más usadas. La velocidad en baudios determina la velocidad a la que se produce la transmisión y debe ser la misma en ambos lados de la comunicación.

### **Modo de Operación**

Los dispositivos RS232 utilizan señalización de un solo extremo (dos hilos) para realizar la transmisión de los datos. En este tipo de cableado, un cable está conectado a tierra mientras que el otro se usa para transmitir un voltaje variable. Pueden ser afectados por el ruido producido por las diferencias en el voltaje de tierra de los circuitos del controlador y el receptor. Una ventaja del método de un solo extremo es que se necesitan menos cables la comunicación.

## Formato de Protocolo



*Imagen 4 Formato de Protocolo*

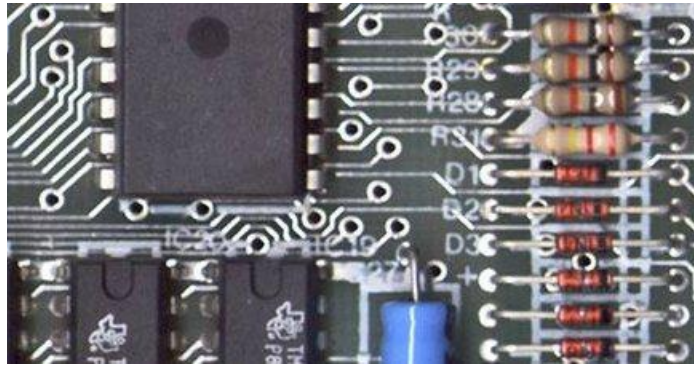
Un mensaje entregado a través del protocolo RS232 comienza enviando un bit de inicio "0". Seguido por siete bits de datos ASCII con un bit de paridad agregado para realizar la verificación. Los bits de paridad determinan la validez del mensaje. La transmisión finaliza con un bit de parada binario "1". Generalmente se envían uno o dos bits de parada.

En el diagrama anterior, el carácter ASCII "A" se transmite con un flujo binario en serie de "1" y "0". Hay un retraso predeterminado entre la transmisión de cada bit cuando la línea se considera inactiva.

## **Circuito impreso, tarjeta de circuito impreso o PCB**

Para un realizar un circuito impreso existen varias técnicas y para implementar este procedimiento en este proyecto se usara el metodo del planchado.

En electrónica, un circuito impreso, tarjeta de circuito impreso o PCB (sprinted circuit board), es una superficie constituida por caminos o pistas de material conductor laminadas sobre un sustrato no conductor.



*Imagen 5 tarjeta de circuito impreso o PCB*

El circuito impreso se utiliza para conectar eléctricamente, a través de los caminos conductores, y sostener mecánicamente, por medio del sustrato, un conjunto de componentes electrónicos. Los caminos son generalmente de cobre mientras que el sustrato se fabrica de resinas de fibra de vidrio reforzada, entre los mas usuales se encuentran:

- cerámica
- plástico
- teflón
- polímeros (baquelita)

La producción de los PCB y el montaje de los componentes puede ser automatizada. Esto permite que en ambientes de producción en masa, sean más económicos y confiables que otras alternativas de montaje.



*Imagen 6 polímeros*

## **Composición**

La mayoría de los circuitos impresos están compuestos aproximadamente de dieciséis capas conductoras, separadas y soportadas por capas de material aislante laminadas entre sí. Las capas pueden conectarse a través de orificios, llamados vías. Los orificios pueden ser electo recubiertos, o se pueden utilizar pequeños remaches.

### **Métodos de impresión de un circuito**

La gran mayoría de las tarjetas para circuitos impresos se hacen adhiriendo una capa de cobre sobre todo el sustrato, posteriormente se retira el cobre no deseado después de aplicar una máscara temporal, dejando sólo las pistas de cobre deseado. Existen varios métodos para la producción de circuitos impresos:

- **Impresión por planchado**

Utiliza la impresión del circuito mediante el calcado de tinta utilizando una plancha, la plancha es la encargada de pasar la imagen del circuito a la placa de cobre con el calor generado en el electrodoméstico.

- **Impresión serigráfica**

Utiliza tintas resistentes al grabado para proteger la capa de cobre. Los grabados posteriores retiran el cobre no deseado.

- **Fotograbado**

Utiliza una fotomecánica y grabado químico para eliminar la capa de cobre del sustrato. La fotomecánica usualmete se prepara con un fotoplotter

- **Fresado**

Utiliza una fresa mecánica de 2 o 3 ejes para quitar el cobre del sustrato. Una fresa para circuitos impresos funciona en forma similar a un plotter, recibiendo comandos desde un programa que controla el cabezal de la fresa los ejes x, y, z.

Para este proyecto se usara el método de planchado ya que es el procedimiento practicado en laboratorio.

## DESARROLLO DEL PROYECTO

El proyecto se compone de 3 partes/fases, en este caso se enfocará terminara con la tercera fase, la cual consiste en realizar la conexión del circuito con la computadora haciendo uso del el microcontrolador PIC16F628A. Esto permitirá expresar los valores binarios obtenidos a través del ADC y el CAS en una aplicación de fácil acceso y comprensión para el usuario final.

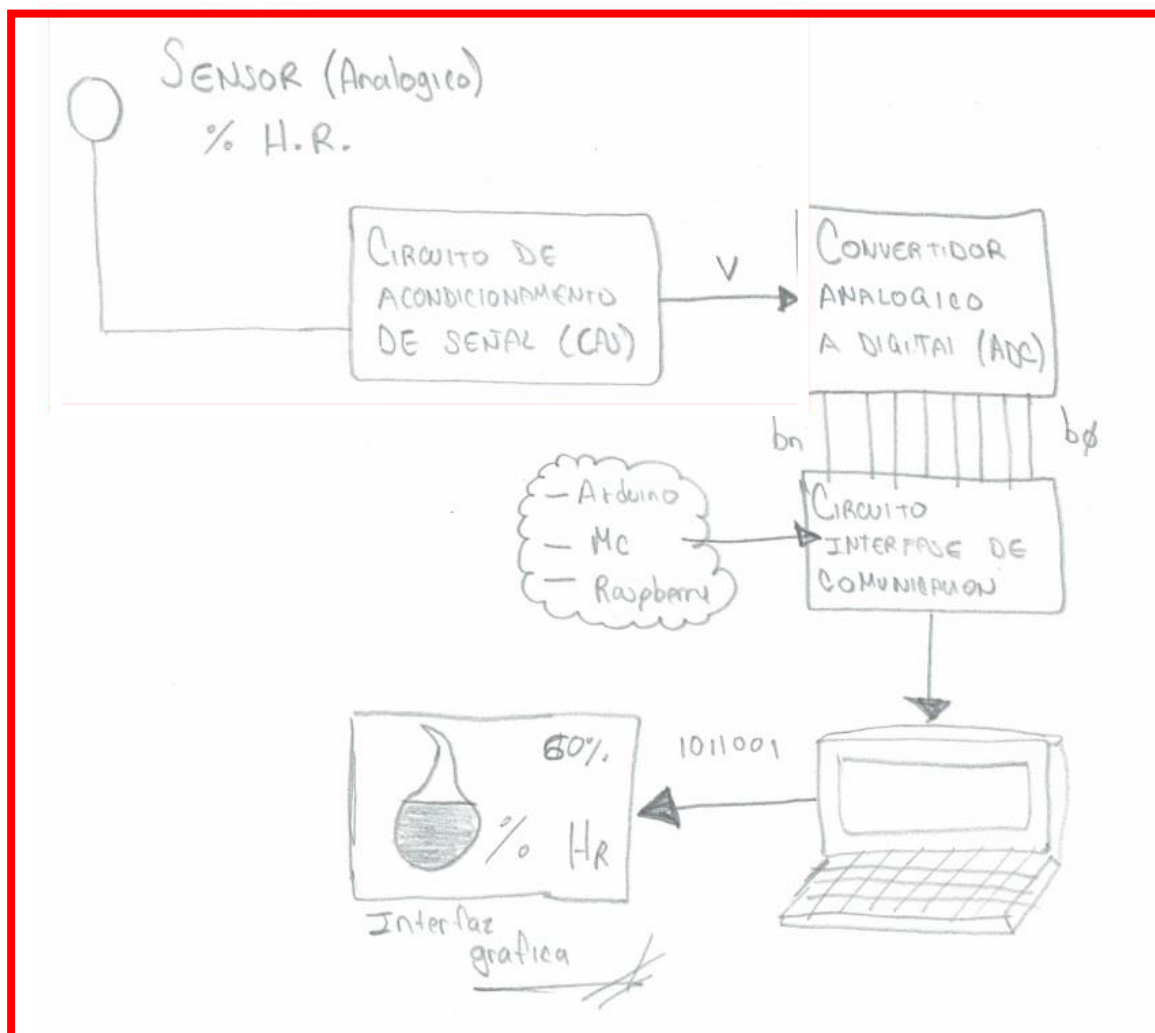


Diagrama 1 / Proyecto Completo

## DISEÑO DEL CIRCUITO DE ACONDICIONAMIENTO DE SEÑAL (CAS)

Para el proyecto se decidió diseñar el CAS de la primera entrega del proyecto, esto para facilitar la conexión y reducir la cantidad de cables en la protoboard. Donde originalmente teníamos el siguiente CAS, por lo que utilizo un Amplificador no inversor y se realizaron los cálculos para saber los valores de las resistencias

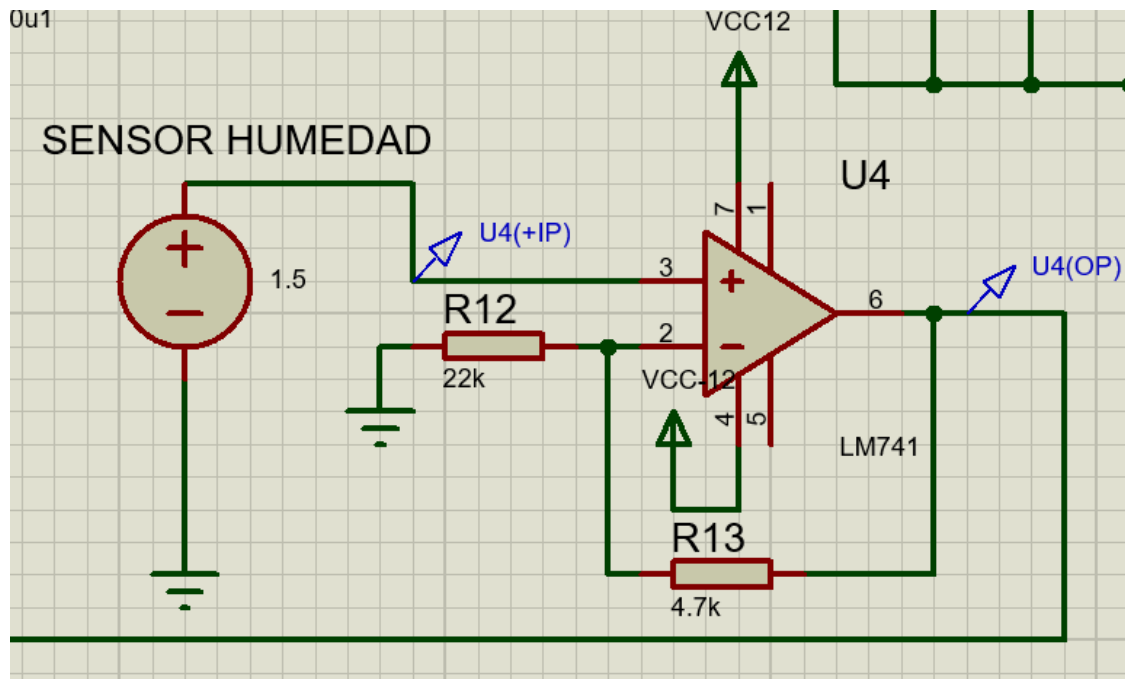


Figura 4 Proyecto diseño del circuito de acondicionamiento de señal

### Cálculos

Se respetó los rangos de amplificación de nuestro diseño anterior del CAS, el cual es:

Se requiere amplificar a 0 V – 4 V

**Sensor: 0v – 3.3v    ->    CAS: 0v – 4.0v**

$$V_{sal} = V_{sen} \left( 1 + \frac{R_f}{R_a} \right)$$

Donde

$$A_v = 1 + \frac{R_f}{R_a}$$



$$R_a = \frac{R_f}{A_v - 1}$$

$$V_{sen}(Alcance) = V_{\max} - V_{\min} = 3.3\text{ V} - 0\text{ V} = 3.3\text{ V}$$

$$V_{sal}(Alcance) = V_{\max} - V_{\min} = 4.0\text{ V} - 0\text{ V} = 4.0\text{ V}$$

$$A_v = \frac{V_{sal}}{V_{sen}} = \frac{4.0\text{ v}}{3.3\text{ v}} \approx 1.21$$

$$\text{Sea } R_f = 4.7\text{ k } \Omega$$

$$A_v = 1 + \frac{R_f}{R_a} \Rightarrow \frac{40}{33} = 1 + \frac{4.7\text{ k } \Omega}{R_a} \Rightarrow \frac{7}{33} = \frac{4.7\text{ k } \Omega}{R_a} \Rightarrow R_a = \frac{4.7\text{ k } \Omega}{7/33}$$

$$R_a = \frac{4.7\text{ k } \Omega}{7/33} = \frac{155100\text{ } \Omega}{7} = 22,157.14286\text{ } \Omega$$

Donde el valor comercial más cercano es

$$R_a = 22\text{ k } \Omega$$

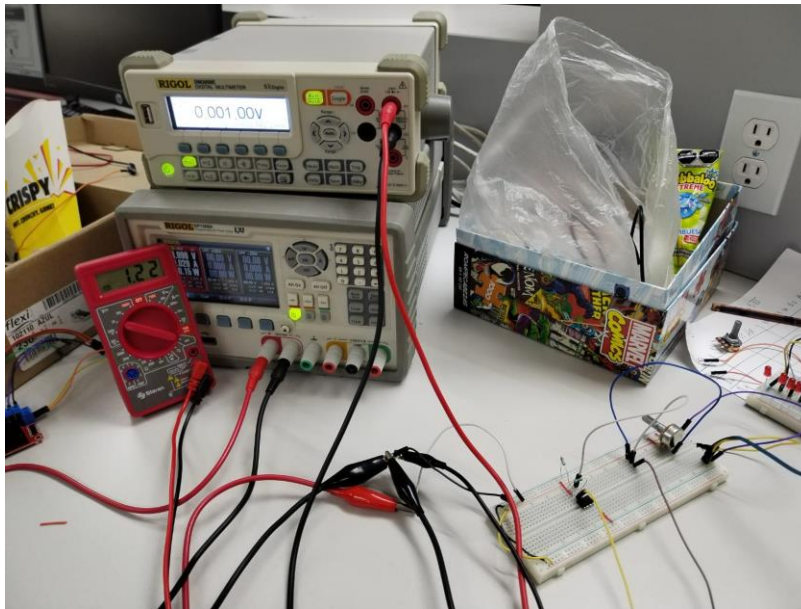


Figura 5 Medición del CAS

El circuito de acondicionamiento de señal resultante queda de la siguiente forma mostrada en la Figura 1, en donde se cambió la fuente de voltaje mostrada en la Figura 4, por un sensor de humedad, ya que inicialmente se había hecho un prototipo para que después se empleara el sensor de humedad relativa el cual si va a dar un valor de voltaje variable según la humedad del sensor.

El amplificador operacional LM741 se cambio a un equivalente que es el LM358 ya que para usos prácticos de la etapa final del proyecto, se decidió usar este amplificador operacional y para el sensor de Humedad Relativa en el aire se usará un HMZ-433A.

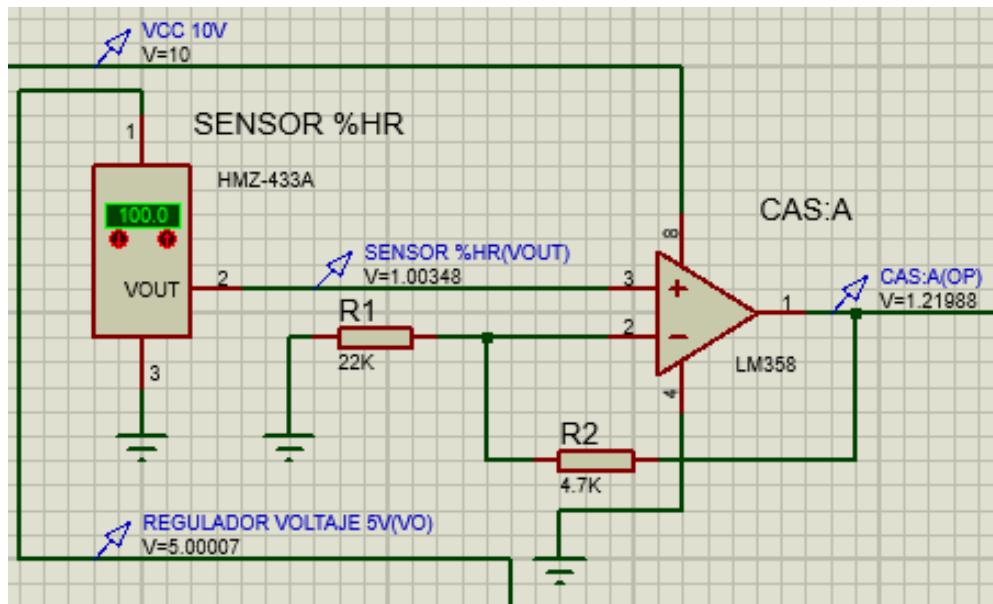


Figura 2 diseño del circuito de acondicionamiento de señal modificado

## DISEÑO DEL CONVERTIDOR ANALOGICO DIGITAL DE 8 BITS (ADC)

Como el sensor es un sensor generador y da como salida voltaje, para el CAS simplemente hay que amplificar la salida del voltaje para tener un mejor manejo. Para el sensor de Humedad Relativa en el aire se usará un **HMZ-433A**, el cual contiene 4 pines y para el convertidor Analógico Digital se usará un **ADC0804** donde checando el Datasheet de ambos tenemos la siguiente información de como conectar el sensor

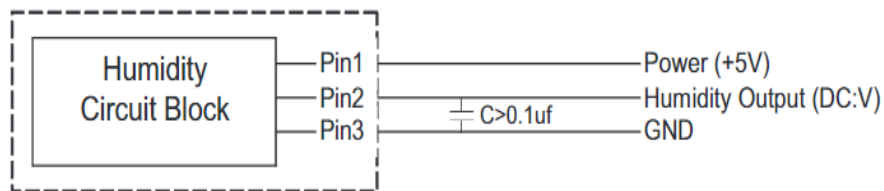


Imagen 7 | Pinout del sensor HMZ-433A

Podemos observar que para conectar el sensor necesitamos usar 3 pines de los 4, donde el sensor es alimentado por 5V.

### Convertidor ADC0804

#### Pinout

ADC0802, ADC0803, ADC0804  
(PDIP, Cerdip)  
TOP VIEW



CS	1	20	V+ OR V <sub>REF</sub>
RD	2	19	CLK R
WR	3	18	DB <sub>0</sub> (LSB)
CLK IN	4	17	DB <sub>1</sub>
INTR	5	16	DB <sub>2</sub>
V <sub>IN</sub> (+)	6	15	DB <sub>3</sub>
V <sub>IN</sub> (-)	7	14	DB <sub>4</sub>
AGND	8	13	DB <sub>5</sub>
V <sub>REF</sub> /2	9	12	DB <sub>6</sub>
DGND	10	11	DB <sub>7</sub> (MSB)

Imagen 8 | Convertidor ADC0804

Para conectar el siguiente circuito se uso la conexión de la practica realizada con anterioridad en el laboratorio, y el resultado es el que esta e:

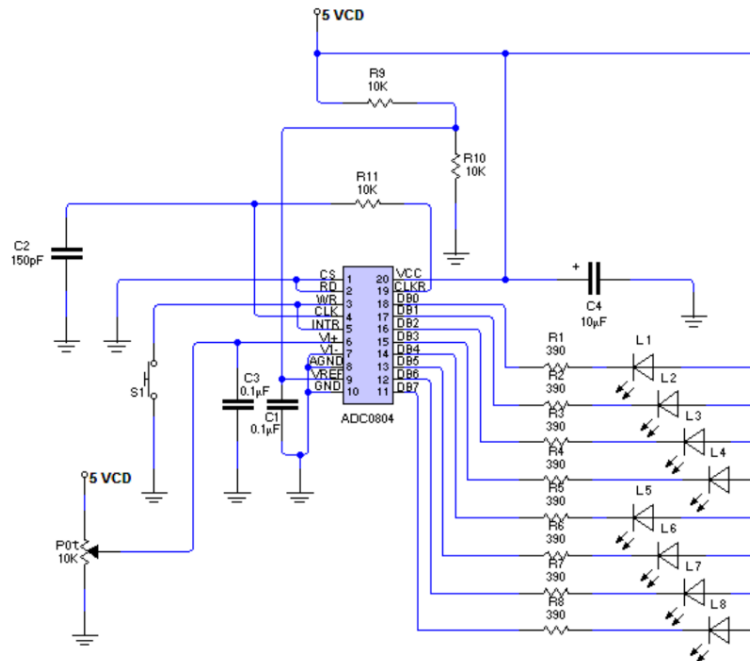


Imagen 9 | Circuito Convertidor ADC

Pero antes de implementar el circuito completo primero se calculo las resistencias que se iba a usar en el divisor de voltaje para que de 0 – 5 V salga 2V que es el voltaje de referencia que es la mitad de este para sumar los 4v que es el rango que operamos con el CAS.

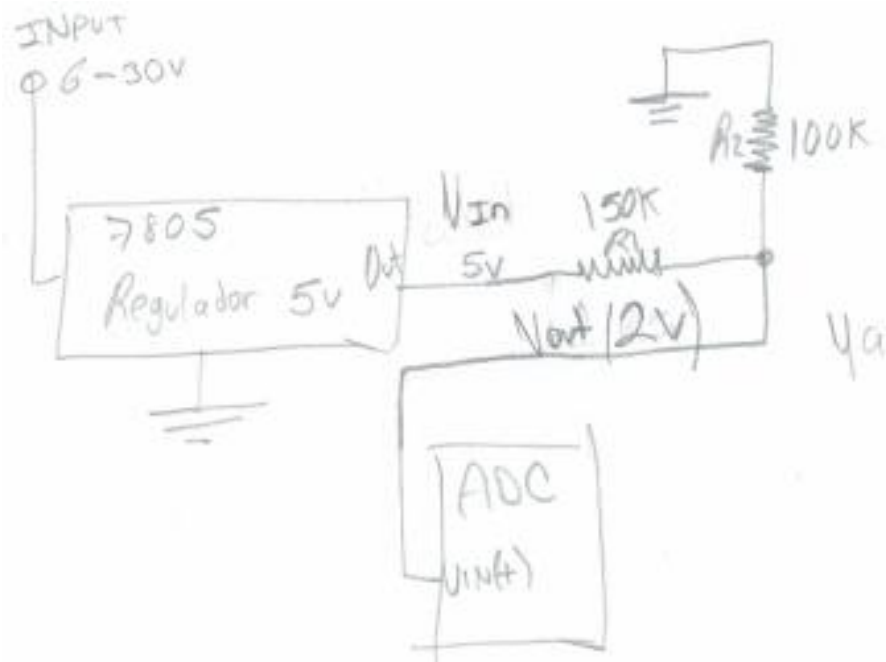


Imagen 10 Resistencias para el divisor de voltaje

### Cálculos

$$V_{out} = V_{ref} \left( \frac{R_2}{R_1 + R_2} \right) \quad \text{donde: } V_{out} = 2.0v \quad V_{in} = 5.0v$$

Proponiendo

$$R_2 = 100k \, \Omega$$

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} \Rightarrow \frac{V_{out}}{V_{in}} (R_1 + R_2) = R_2 \Rightarrow \frac{2v}{5v} (R_1 + 100k) = 100k\Omega$$

$$0.4R_1 + 40k = 100k \Rightarrow \left( \frac{2}{5} \right) R_1 = 60k\Omega \Rightarrow R_1 = \frac{60k\Omega}{0.4} \Rightarrow R_1 = 150k\Omega$$

$$\mathbf{R_1 = 150k \, \Omega \quad R_2 = 100k \, \Omega}$$

Una vez hecho los cálculos se procedió a hacer el circuito, uniendo el sensor, el CAS y el ADC.

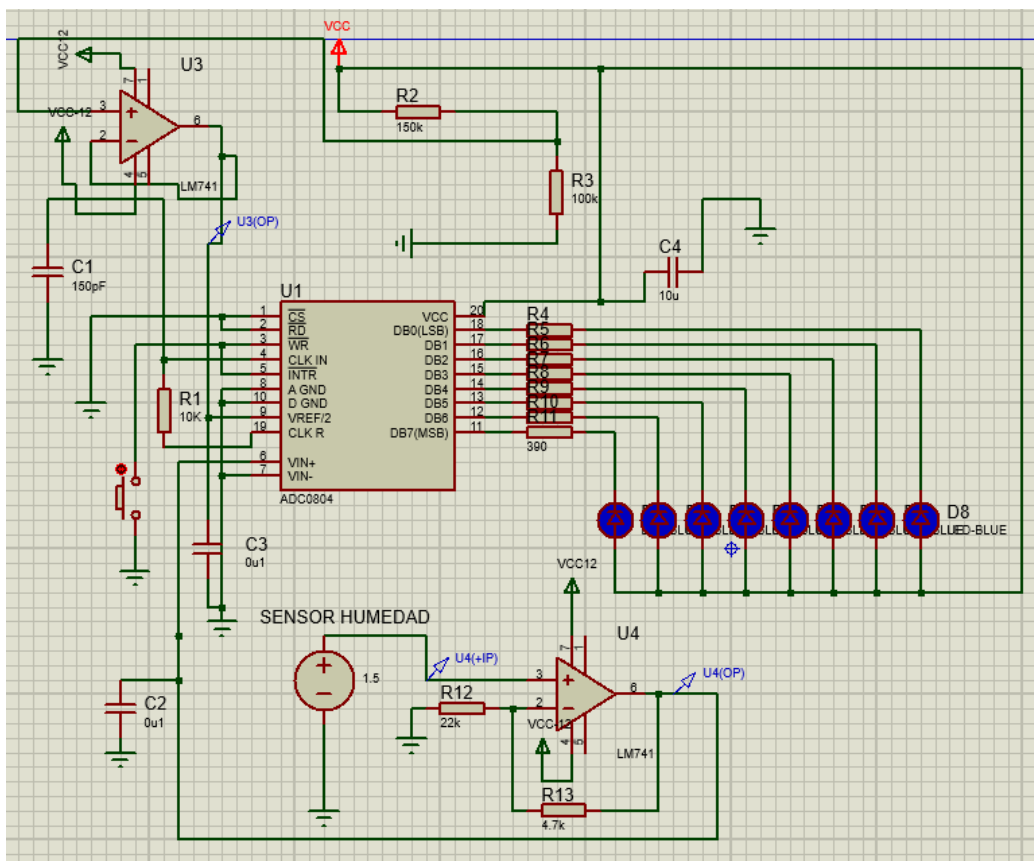


Imagen 11 Diseño del convertidor analógico digital de 8 bits (ADC)

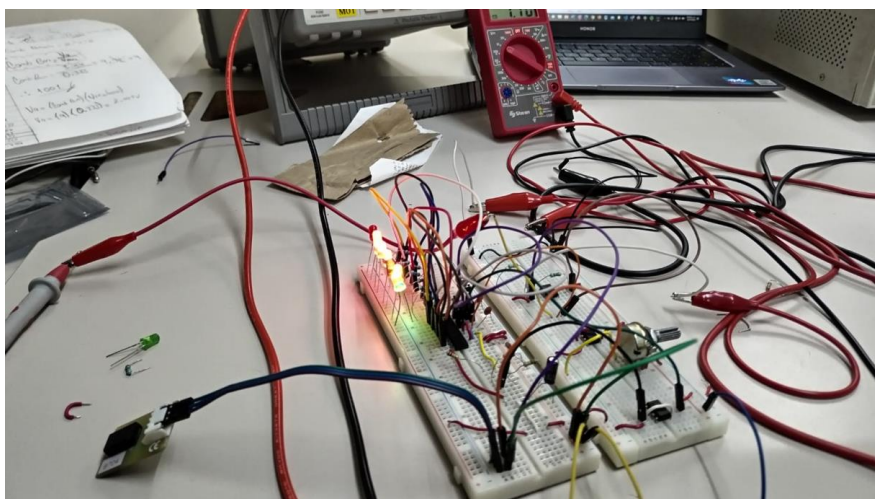


Imagen 12 Medición del ADC

## DISEÑO DEL CIRCUITO GENERADOR DE SEÑAL BAJO PROTOCOLO RS-232.

Completado circuito para el sensor es un sensor generador después se van transmitir las señales binarias a la computadora para que sea posible usar estos valores en nuestra programa de pc.

Conectar el circuito a la computadora es un procedimiento que se realiza conectando los 8 bits de salida del ADC al microcontrolador PIC16F84A y luego se conecta el PIC16F84A al Modulo Convertidor Usb A Serial RS 232.

Para el convertidor Analógico Digital se usará un **ADC0804**, y para conectar el ADC al microcontrolador PIC16F84A se usarán los pines de salida del ADC  $DB_1, DB_2, DB_3, \dots, DB_7$  mostrados en el datasheet de la Imagen 13.

### Convertidor ADC0804

#### Pinout

ADC0802, ADC0803, ADC0804  
(PDIP, Cerdip)  
TOP VIEW



CS	1	20	V+ OR V <sub>REF</sub>
RD	2	19	CLK R
WR	3	18	DB <sub>0</sub> (LSB)
CLK IN	4	17	DB <sub>1</sub>
INTR	5	16	DB <sub>2</sub>
V <sub>IN</sub> (+)	6	15	DB <sub>3</sub>
V <sub>IN</sub> (-)	7	14	DB <sub>4</sub>
AGND	8	13	DB <sub>5</sub>
V <sub>REF</sub> /2	9	12	DB <sub>6</sub>
DGND	10	11	DB <sub>7</sub> (MSB)

Imagen 14 | Convertidor ADC0804

Se usará el PIC16F84A para conectar al ADC mediante los pines RA0, RA1, RA2,..., RA7 mostrados en el datasheet de la Imagen 1z.

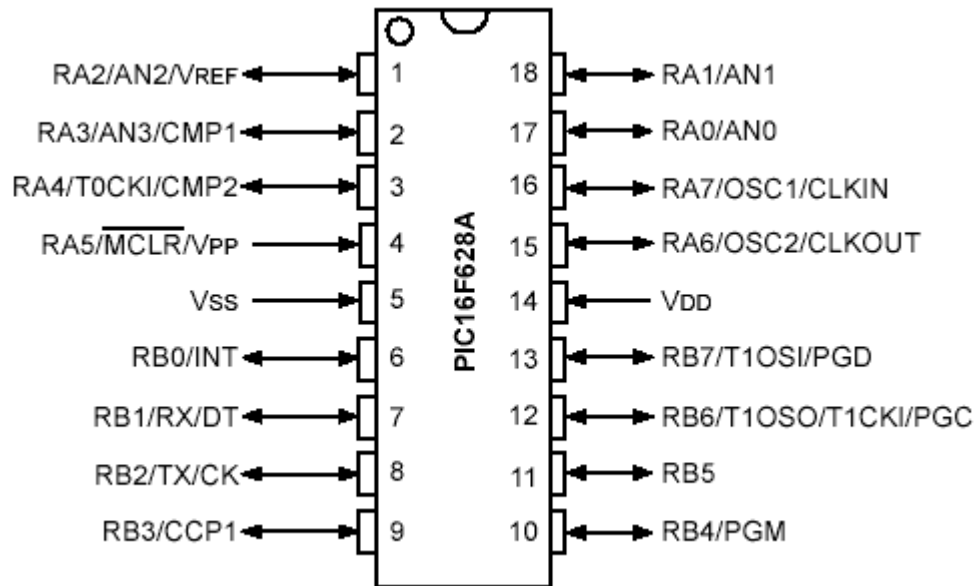


Imagen 15 microcontrolador PIC16F84A

Para conectar el siguiente circuito se uso la conexión de la practica realizada con anterioridad en el laboratorio:

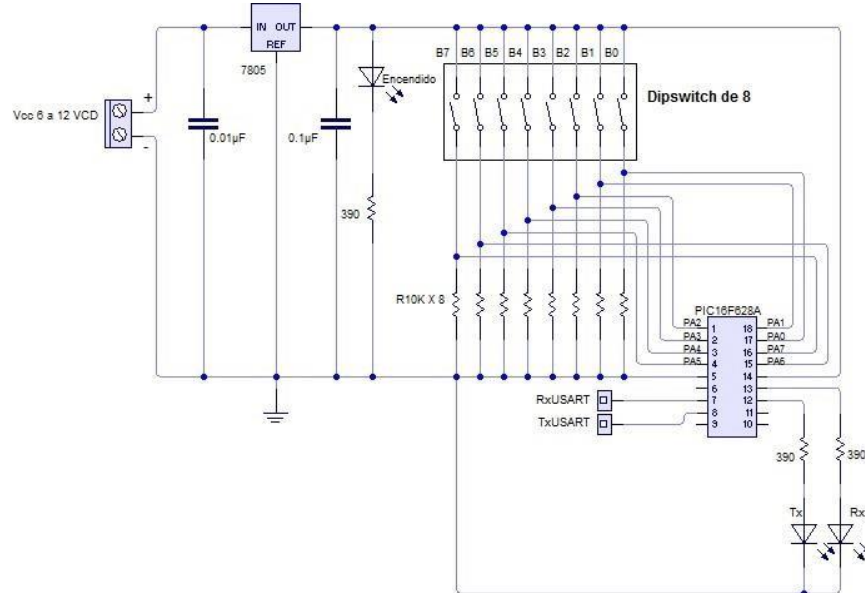


Figura 3 Circuito generador de señal bajo protocolo RS-232

Al microcontrolador se le tiene que grabar el código (el código del programa se encuentra en al final del presente documento) previamente ensamblado en el cual, las terminales del puerto A, deben configurarse como entradas, para que lean el estado lógico que será



generado por el sensor de humedad. Posteriormente, el microcontrolador después de leer las terminales del puerto A, enviara la información de cada uno de estos bits, por medio de la salida Tx que trabajara mediante protocolo RS-232. El siguiente diagrama mostrado en la Imagen 16 el cual es la implementación en el proyecto.

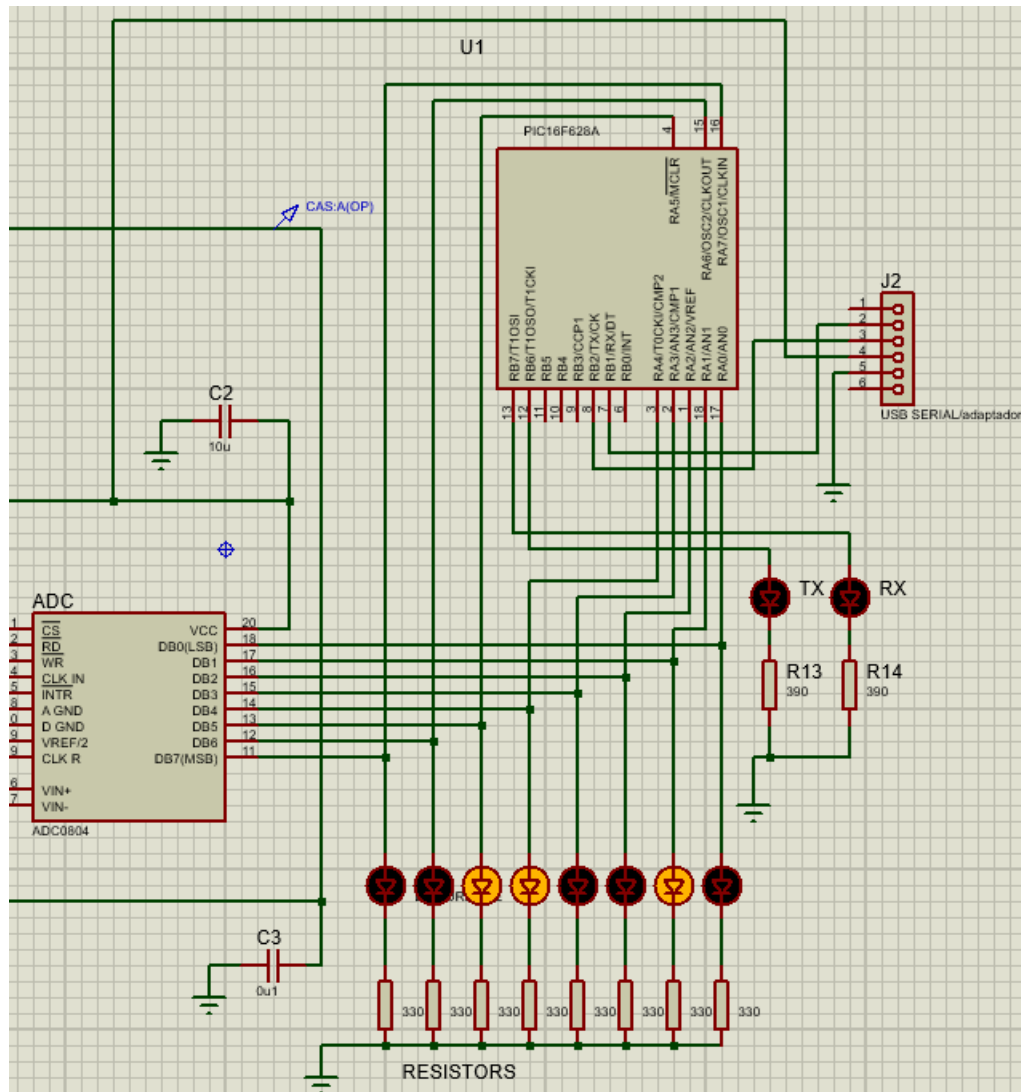


Imagen 17 Implementación del protocolo RS-232

Se tiene que conectar el microcontrolador al adaptador usb que servirá para ver los valores binarios en la computadora. El Módulo Convertidor USB a Serial RS232 TTL tiene las siguientes entradas que se muestra en la Imagen 18

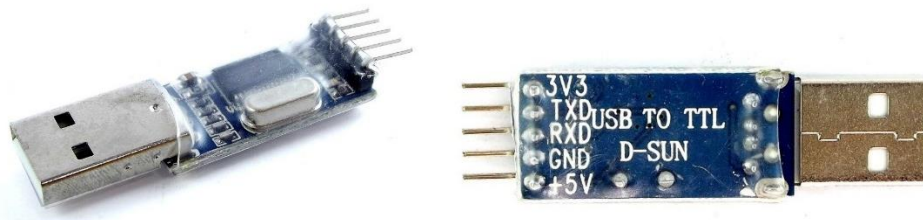


Imagen 19 Módulo Convertidor USB a Serial RS232 TTL

Las entradas TX y RX corresponden a la señales de salida del microcontrolador y se debe poner la tierra o GND, además se necesita 5V de energía.

Ya habiendo puesto el modulo al circuito debe quedar como se muestra em el siguiente diagrama:

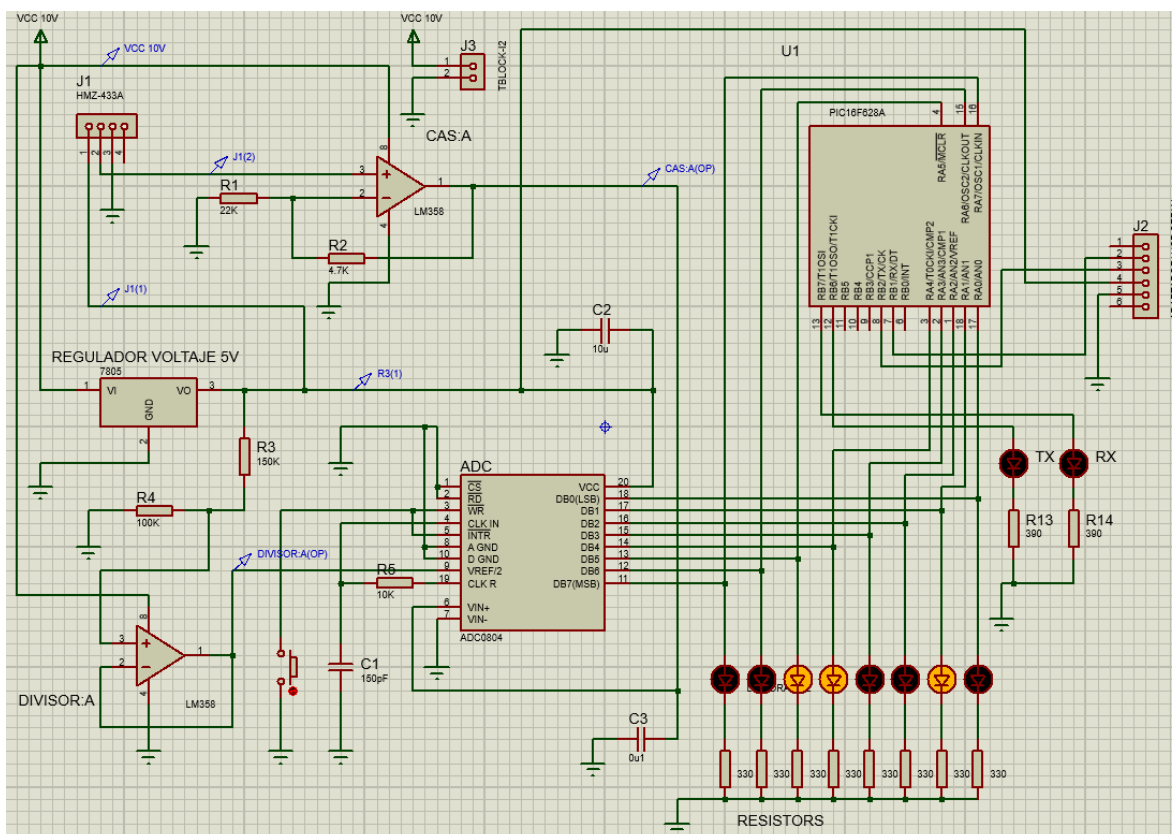
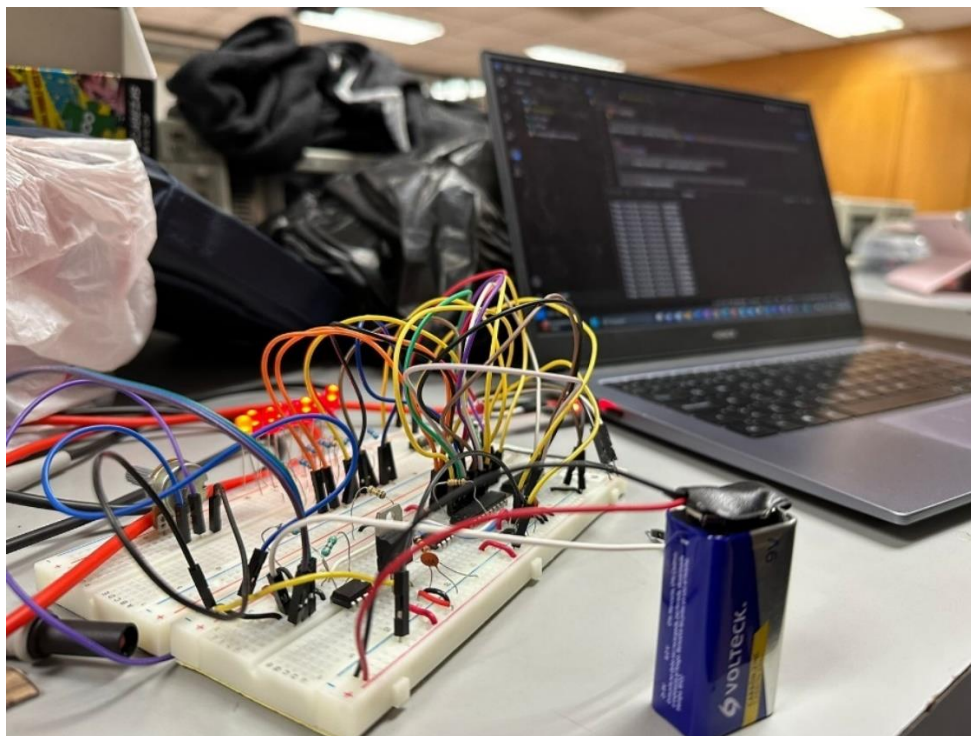


Imagen 20 Diagrama del circuito generador de señal bajo protocolo RS-232.



*Imagen 21 Circuito generador de señal bajo protocolo RS-232.*

Antes de usar la computadora se tienen que instalar drivers para el Módulo Convertidor USB a Serial RS232 TTL para ello se tiene que ingresar a la página de <https://www.miklor.com/COM/software/P3200.exe> y descargar e instalar los drivers.

Con esta configuración, el circuito se encuentra listo para transmitir los valores de humedad a la computadora, donde serán procesados y presentados mediante la aplicación desarrollada. La implementación exitosa de esta fase asegura una comunicación efectiva entre el sistema de medición de humedad y la interfaz de usuario en la PC.

Para observar que sean recibidas las tramas hemos corrido el siguiente programa base “humedad.py” en Python que muestra las señales que esta capturando:

```
import serial

# Configuración del puerto serial
puerto_serial = serial.Serial('COM3', 9600, timeout=1) # Reemplaza 'COMX' con el nombre correcto del puerto

try:
    while True:
        # Lee una trama de 8 bytes desde el puerto serial
```

```

trama_recibida = puerto_serial.read(8)
# Muestra los datos de la trama recibida en formato binario
if trama_recibida:
    datos_binarios = ''.join(format(byte, '08b') for byte in trama_recibida)
    print(f'Trama recibida (binario): {datos_binarios}')
except KeyboardInterrupt:
    print("Programa interrumpido por el usuario")
finally:
    # Cierra el puerto serial al finalizar
    puerto_serial.close()

```

Este código es un ejemplo de cómo se puede leer datos de un dispositivo conectado a través de un puerto serial en Python. La biblioteca serial se utiliza para establecer una conexión con el dispositivo y configurar el puerto serial. En este caso, se establece una conexión con el puerto COM3 a una velocidad de transmisión de 9600 baudios.

El código luego entra en un bucle infinito que lee una trama de 8 bytes desde el puerto serial y muestra los datos de la trama recibida en formato binario. Si no se recibe ninguna trama, el programa continúa esperando. El bucle se interrumpe si el usuario presiona Ctrl + C.

Finalmente, el puerto serial se cierra al finalizar el programa

Este script de Python se ejecuta en la línea de comandos. Aquí hay un ejemplo de cómo podría verse la ejecución y la salida podría ser algo como esto:

```

Trama recibida (binario): 01100001
Trama recibida (binario): 01100010
Trama recibida (binario): 01100011
Trama recibida (binario): 01100100
Trama recibida (binario): 01100101

```

...

Esto asume que hay un dispositivo conectado al puerto serial 'COM3' que está enviando datos. Los datos exactos que se muestran dependerán de lo que esté enviando el dispositivo. Si se presiona Ctrl+C para interrumpir el programa, se verá algo como esto:

Programa interrumpido por el usuario

Y luego el programa se cerrará, liberando el puerto serial.

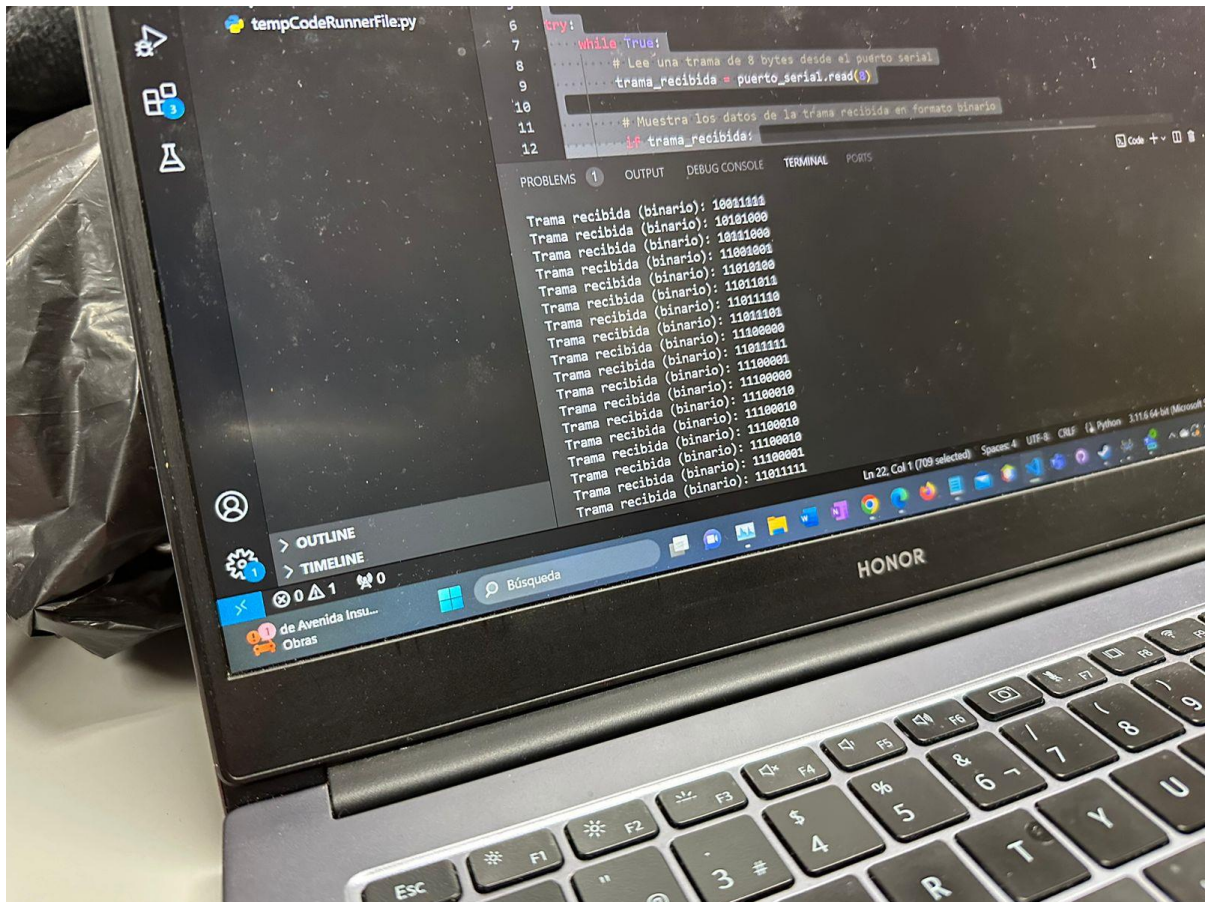
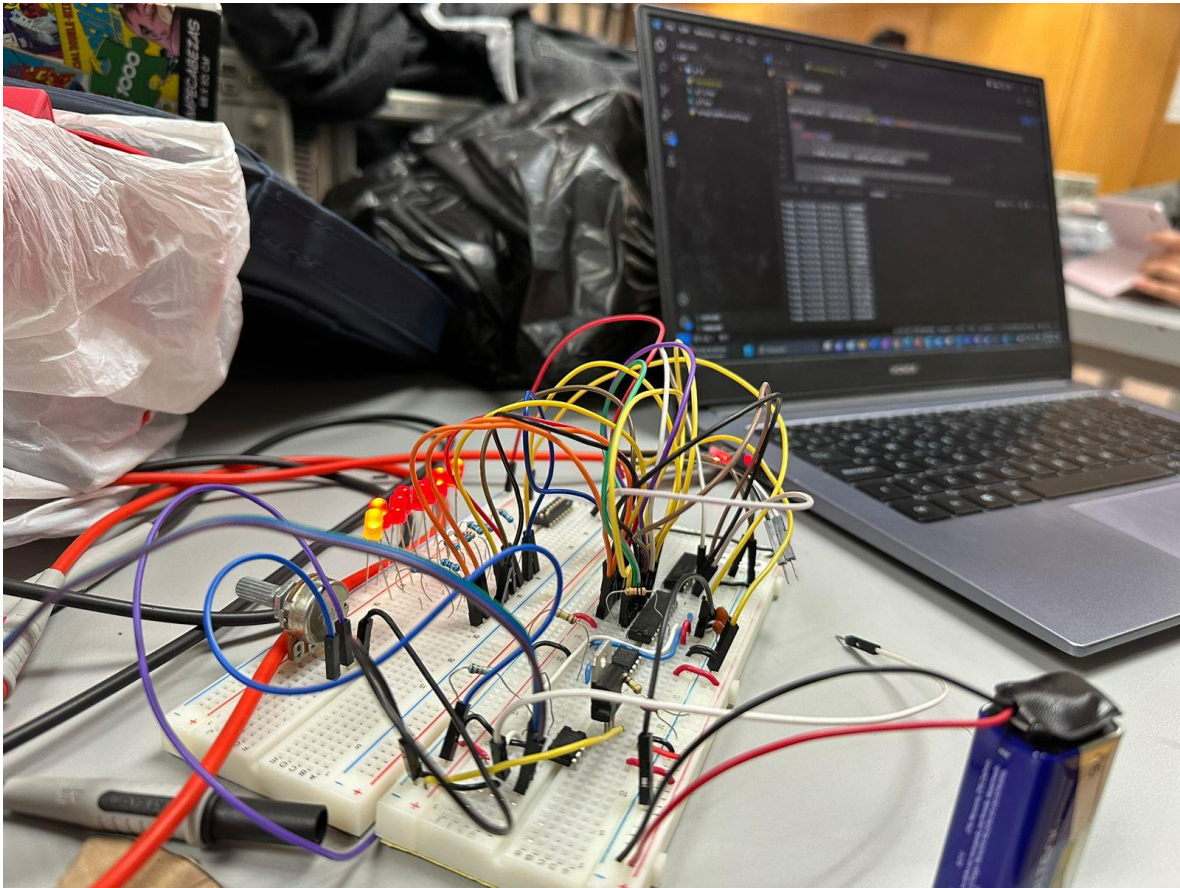


Imagen 22 Programa “humedad.py”





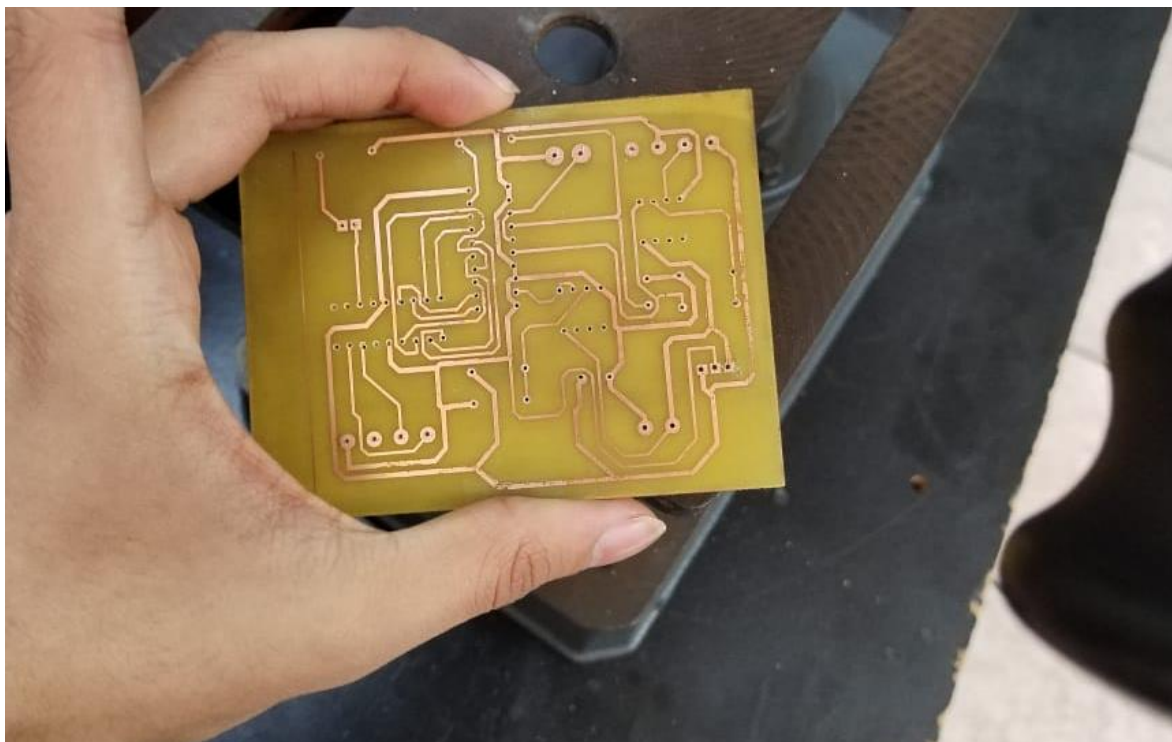
*Imagen 23 Circuito generador de señal bajo protocolo RS-232.*

El diagrama del circuito se va a diseñar en un prototipo de PCB ya que se va a crear circuito impreso como propósito particular para este proyecto.

## Método de planchado para imprimir circuitos.

El planchado de circuitos impresos se asocia comúnmente con la técnica de transferencia de toner para la creación de prototipos de PCB en entornos caseros o de laboratorio. Este proceso implica la impresión de un diseño de PCB en papel especial con tóner, que luego se transfiere a una placa de cobre mediante calor y presión.

El planchado de circuitos impresos es una técnica de transferencia térmica del tóner de un acetato impreso (circuito impreso en papel couche) a una placa de cobre



*Imagen 24 Circuito impreso en la placa*

Este método consta de los siguientes pasos:

### **Paso 1. Impresión del circuito**

Teniendo hecho el diagrama del circuito para PCB en el computador, lo imprimimos en alta resolución sobre el papel termo transferible, usando una impresora láser. Se imprime sobre cualquier cara del papel, ya que las dos caras son iguales.

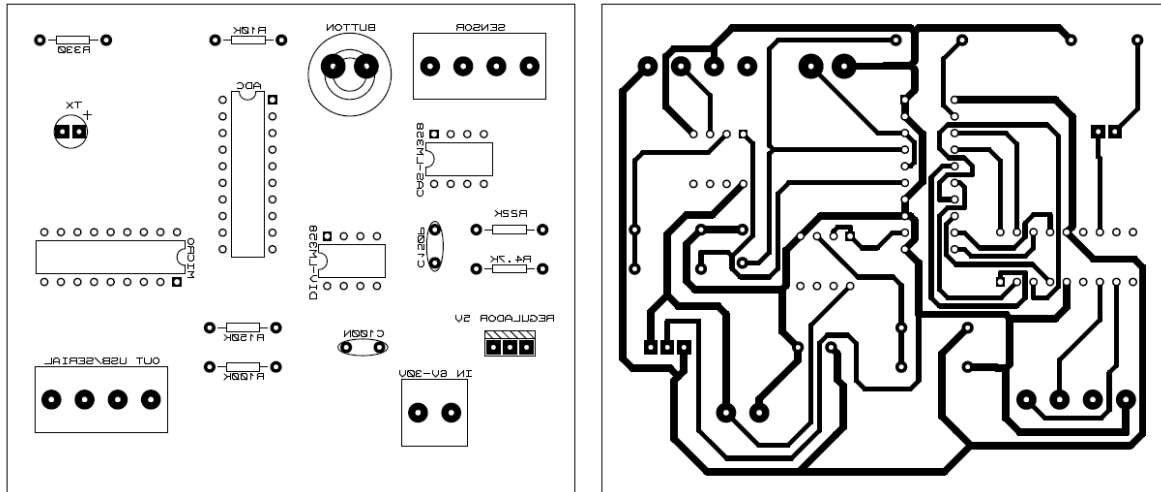


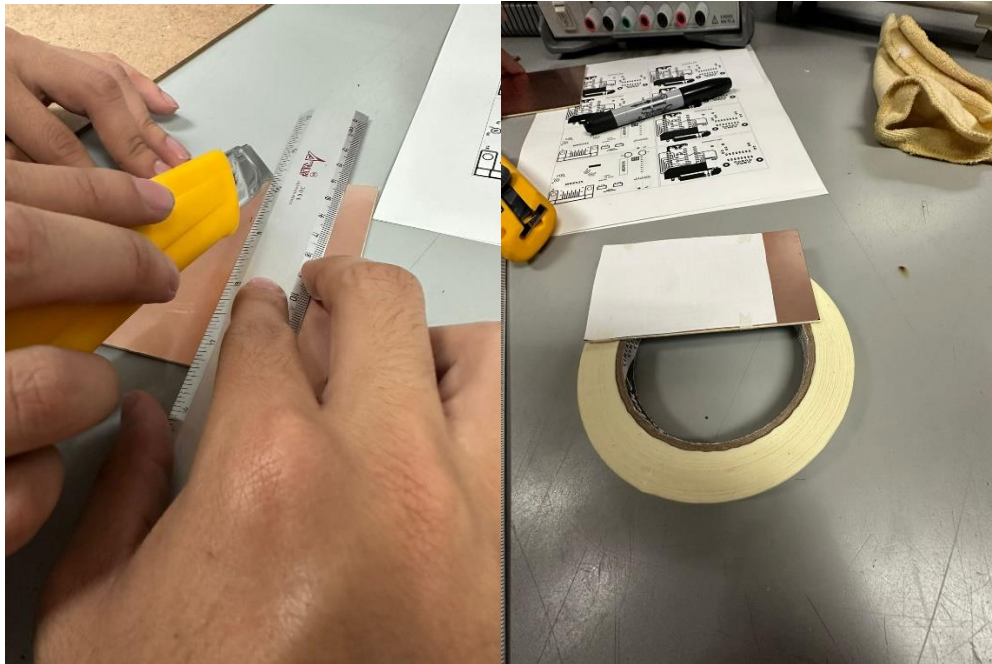
Imagen 25 diagrama del circuito para PCB

Una vez tengamos el diseño del circuito impreso sobre el papel couche, lo recortamos usando unas tijeras, dejando una margen que nos permita manipularlo. El papel termotransferible restante lo podremos guardar para la elaboración de futuros circuitos impresos.

Ahora se debe cortar la placa fenólica a la medida del circuito impreso. Se debe medir el tamaño del circuito que esta en la hoja de papel couche y se mide estas medidas sobre la placa fenólica usando la regla de metal.

Se coloca el pedazo de papel que tiene el circuito en posición boca abajo sobre la placa fenólica y se fija usando maskin tape.





*Imagen 26 Se coloca la placa fenólica y el papel impreso.*

## **Paso 2. Transferencia a la placa de cobre**

En esta etapa del proyecto, debemos transferir el tóner del papel a la placa de cobre, para lo cual utilizaremos el calor de LA PLANCHA.

Debemos cortar la placa virgen a las medidas de nuestro PCB. Seguido de una limpieza concienzuda de la placa de circuito impreso virgen, para que quede libre de suciedad, grasa, etc. Utilizaremos para ello el polvo limpiador y la lana de acero, que debe ser lo más fina posible para que no queden rayas.



*Imagen 27 Transferencia de tóner a la placa de cobre*

En este caso se tuvo que remarcar con plumón las partes que no se adhirieron a la placa hasta que se complete el diagrama y para asegurar que sirviera la placa se tuvieron que hacer 3 veces el procedimiento igual hasta esta parte del proyecto.



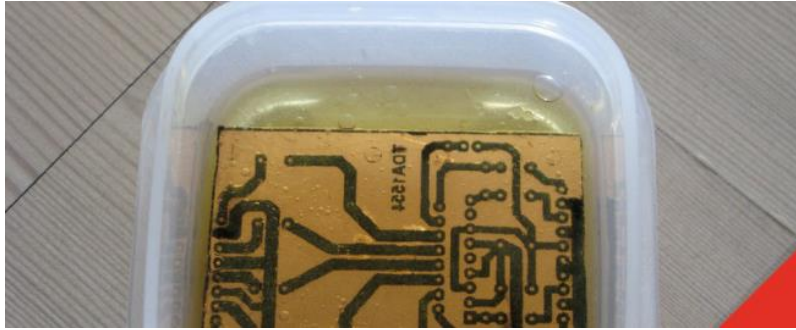
*Imagen 28 Transferencia de tóner a la placa de cobre*

### **Paso 3. Eliminación del cobre sobrante**

En este paso, se va a eliminar todas las zonas de cobre "visibles" que sobran de nuestra placa virgen, es decir, las que no están cubiertas por el tóner.

Se sumerge la placa en la solución de cloruro férrico aproximadamente 30 min. Si el cobre está disminuyendo pasará de un color cobrizo a uno rosado, Si el cobre se ha disuelto sacamos la placa, pero aún no vemos un avance significativo, entonces seguimos agitando en un sentido y luego en otro.

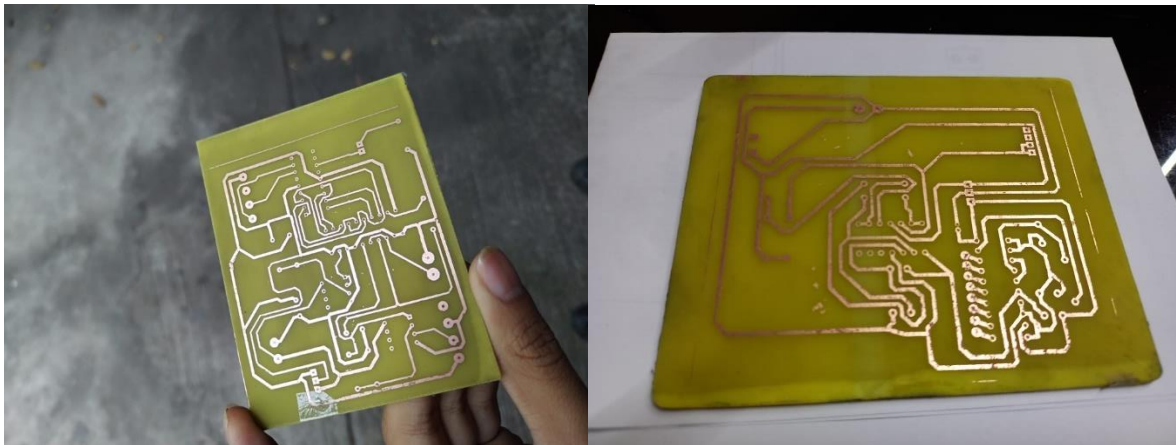
Una vez que se haya caído el cobre, saque cuidadosamente la placa, enjuáguela con agua corriente y séquela a temperatura ambiente.



*Imagen 29 Eliminado el cobre sobrante*

#### **Paso 4. Acabado final**

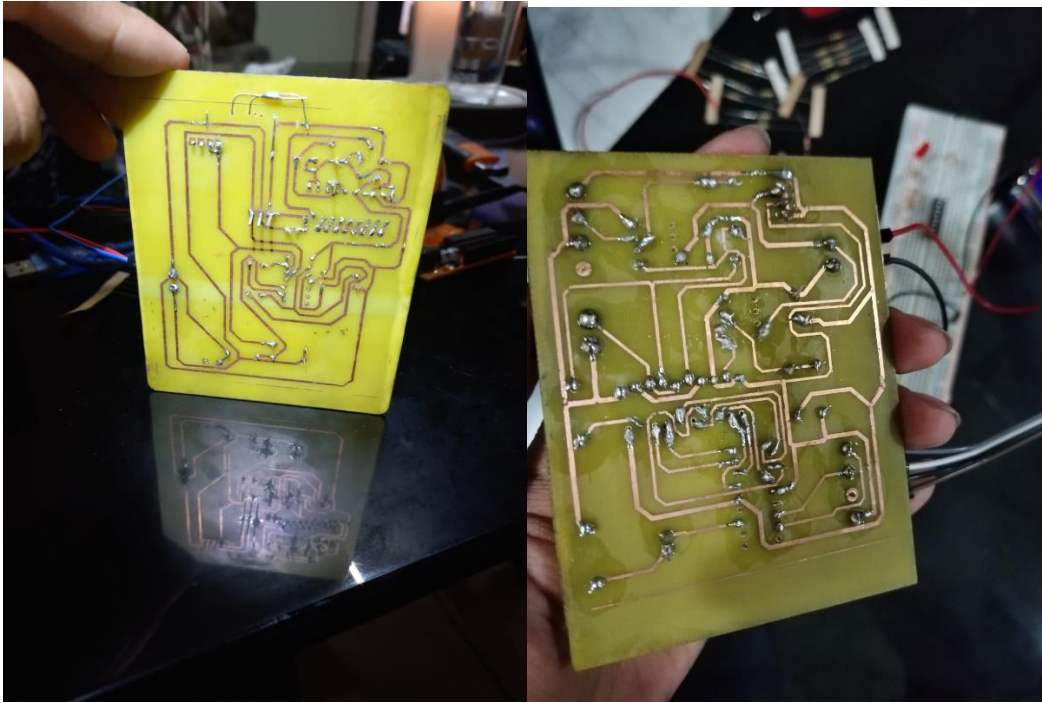
Con la placa ya libre de percloruro, utilizamos nuevamente la lana de acero con el polvo limpiador para remover todos los restos de t  n que hay sobre el PCB, y ya deber  amos tener nuestro PCB casi listo, restando solamente efectuar los agujeros para los componentes de ser necesario.



*Imagen 30 PCB sin los componentes*

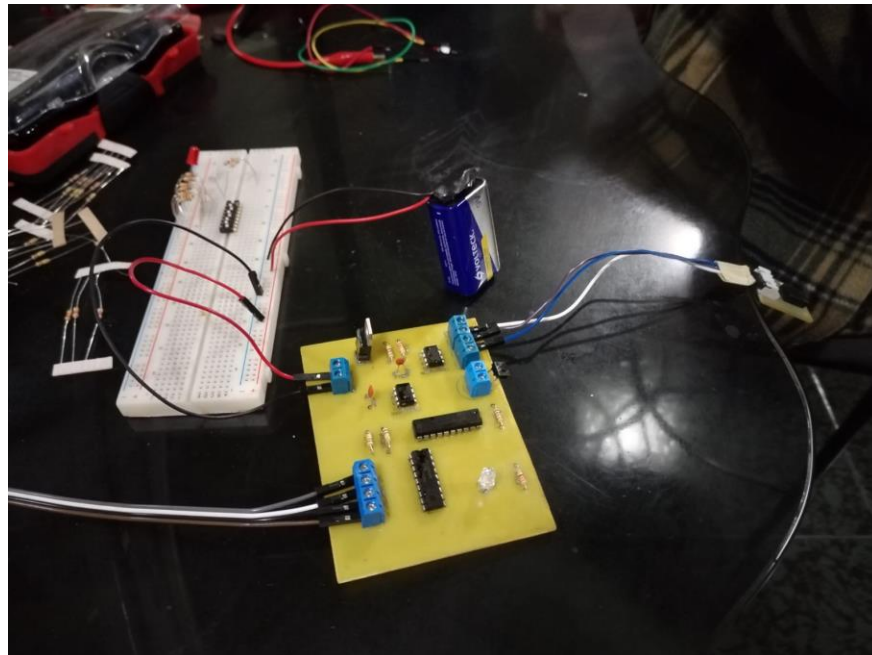
#### **Paso 5. PCB con los componentes**

Se colocan los componentes usados en la protoboard y se debe soldar usando un Caut  n y la pasta de soldar y queda el resultado siguiente:



*Imagen 31 Soldar sobre la placa*

Por ultimo se debe conectar todos los cables los cuales son la energía de 5 volts y el modulo que va a usarse para adaptarse a la computadora



*Imagen 32 Circuito para PCB*



## Programa en pc

Se programo una aplicación para visualizar las cantidades obtenidas del sensor de humedad que pueda ser de fácil uso para el usuario, y el funcionamiento del programa será explicado a continuación.

## Explicación

Se usaron 2 scripts 'humedad.py' y 'humedad\_interfaz.py' los cuales se puede ver al final de este reporte en el Anexo Código para proyecto.

### Script 'humedad.py'

Este script se encarga de la lectura de datos provenientes del sensor de humedad a través de un puerto serial. Aquí se utiliza la biblioteca **serial** para la comunicación serial. A continuación, se presenta un resumen de su funcionalidad:

```
import serial

V_RES = 4 / 255
AV = 40 / 33
V_MAX_SEN = 3.3

# Configuración del puerto serial
puerto_serial = serial.Serial('COM3', 9600, timeout=1) # Reemplaza 'COMX' con el nombre correcto del puerto

try:
    while True:
        # Lee una trama de 8 bytes desde el puerto serial
        trama_recibida = puerto_serial.read(8)

        print(f"Trama{trama_recibida}")

        # Muestra los datos de la trama recibida en formato binario y decimal
        if trama_recibida:
            datos_binarios = ''.join(format(byte, '08b') for byte in trama_recibida)
            datos_decimales = int.from_bytes(trama_recibida, byteorder='big')
```

```

    vcas = datos_decimales * V_RES
    vsen = vcas / AV
    hr = (vsen * 100) / V_MAX_SEN

    print(f'Trama recibida: \n {datos_binarios} | {datos_decimales} | {hr:.2f} % | Vsen = {vsen:.4f} | Vcas {vcas:.4f}')

except KeyboardInterrupt:
    print("Programa interrumpido por el usuario")

finally:
    # Cierra el puerto serial al finalizar
    puerto_serial.close()

```

Este es un script de Python que lee datos de un sensor de humedad a través de un puerto serial. Aquí está el desglose:

1. Importa el módulo `serial` para la comunicación serial.
2. Define algunas constantes: `V\_RES`, `AV`, `V\_MAX\_SEN` que se utilizan para calcular la humedad relativa.
3. Configura el puerto serial con el nombre del puerto y la velocidad de transmisión (baud rate).
4. Entra en un bucle infinito donde lee una trama de 8 bytes desde el puerto serial.
5. Si se reciben datos, los convierte a formato binario y decimal.
6. Calcula `vcas`, `vsen` y `hr` (humedad relativa) a partir de los datos recibidos.
7. Imprime los datos recibidos y los valores calculados.
8. Si se interrumpe el programa (por ejemplo, con Ctrl+C), imprime un mensaje y cierra el puerto serial.

Este script se utiliza típicamente para leer y procesar datos de un sensor de humedad conectado a un puerto serial de la computadora.

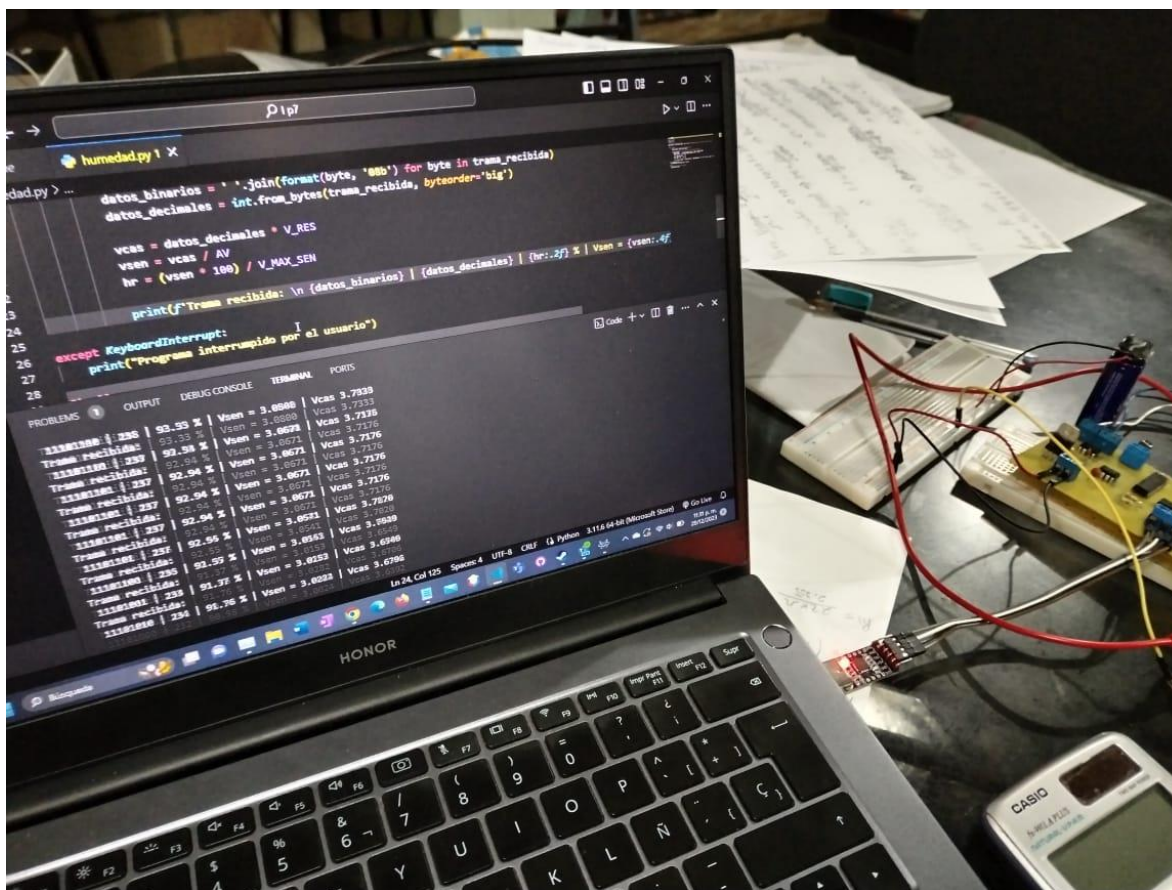


Imagen 33 Datos de sensor de humedad

### Script 'humedad\_interfaz.py'

Este script implementa una interfaz gráfica de usuario (GUI) utilizando la biblioteca **tkinter** y presenta una representación visual de los datos de humedad obtenidos del sensor. A continuación, se resumen las principales características del script:

- Se define la clase **HumidityMonitorApp** que contiene la configuración de la interfaz, etiquetas, lienzo para el voltímetro y botones.

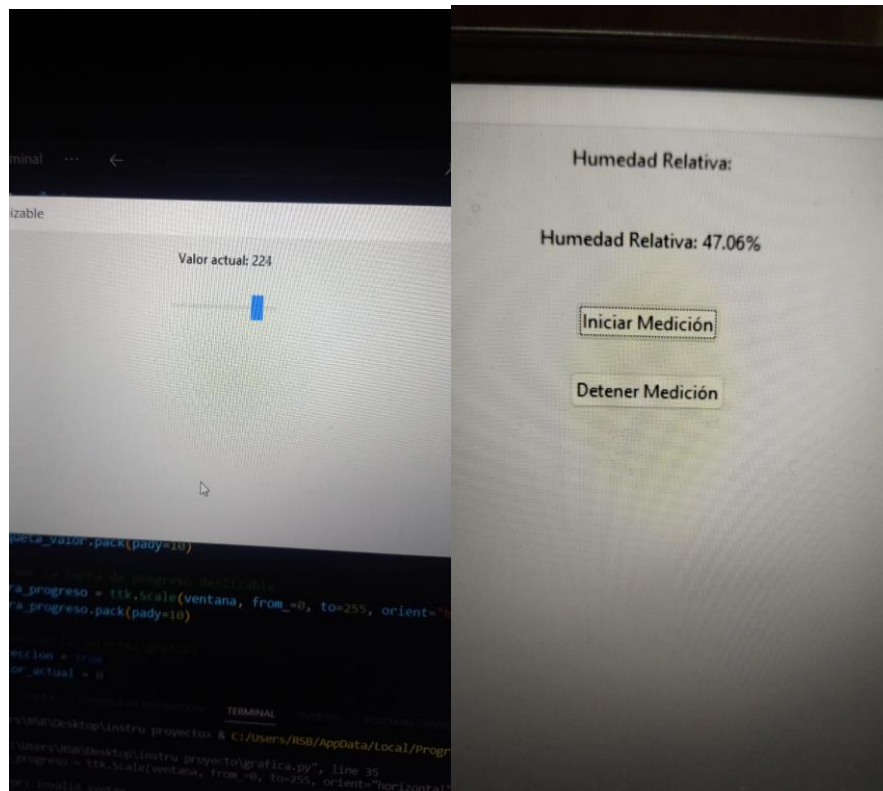


Imagen 34 configuración de la interfaz, etiquetas

- Se carga y muestra imágenes en la interfaz para representar visualmente la humedad (lluvia, gota y sol) mediante los archivos 'lluvia.png', 'gota2.png' y 'sol.png'.
- Se establece la comunicación serial a través del puerto 'COM3'.
- Se implementan métodos para iniciar y detener la medición, así como para actualizar la visualización de la humedad en la GUI.
- Se utiliza un bucle **while** para leer continuamente datos del puerto serial mientras la medición está en curso.
- La interfaz muestra el porcentaje de humedad relativa, actualiza un voltímetro y cambia la imagen según el nivel de humedad.
- El programa puede ser interrumpido por el usuario y el puerto serial se cierra de manera adecuada al finalizar.

En resumen, este conjunto de scripts proporciona una solución completa para la adquisición y visualización de datos de humedad a través de una interfaz gráfica de usuario en una PC.



Los videos demostrativos del funcionamiento de este programa se encuentran en el siguiente enlace y de igual manera de encuentra en la bibliografía del presente documento: [https://drive.google.com/drive/folders/1ep3VQIDnU1olrGRG\\_TZfMjBiYtbaQv7w?usp=drive\\_link](https://drive.google.com/drive/folders/1ep3VQIDnU1olrGRG_TZfMjBiYtbaQv7w?usp=drive_link)

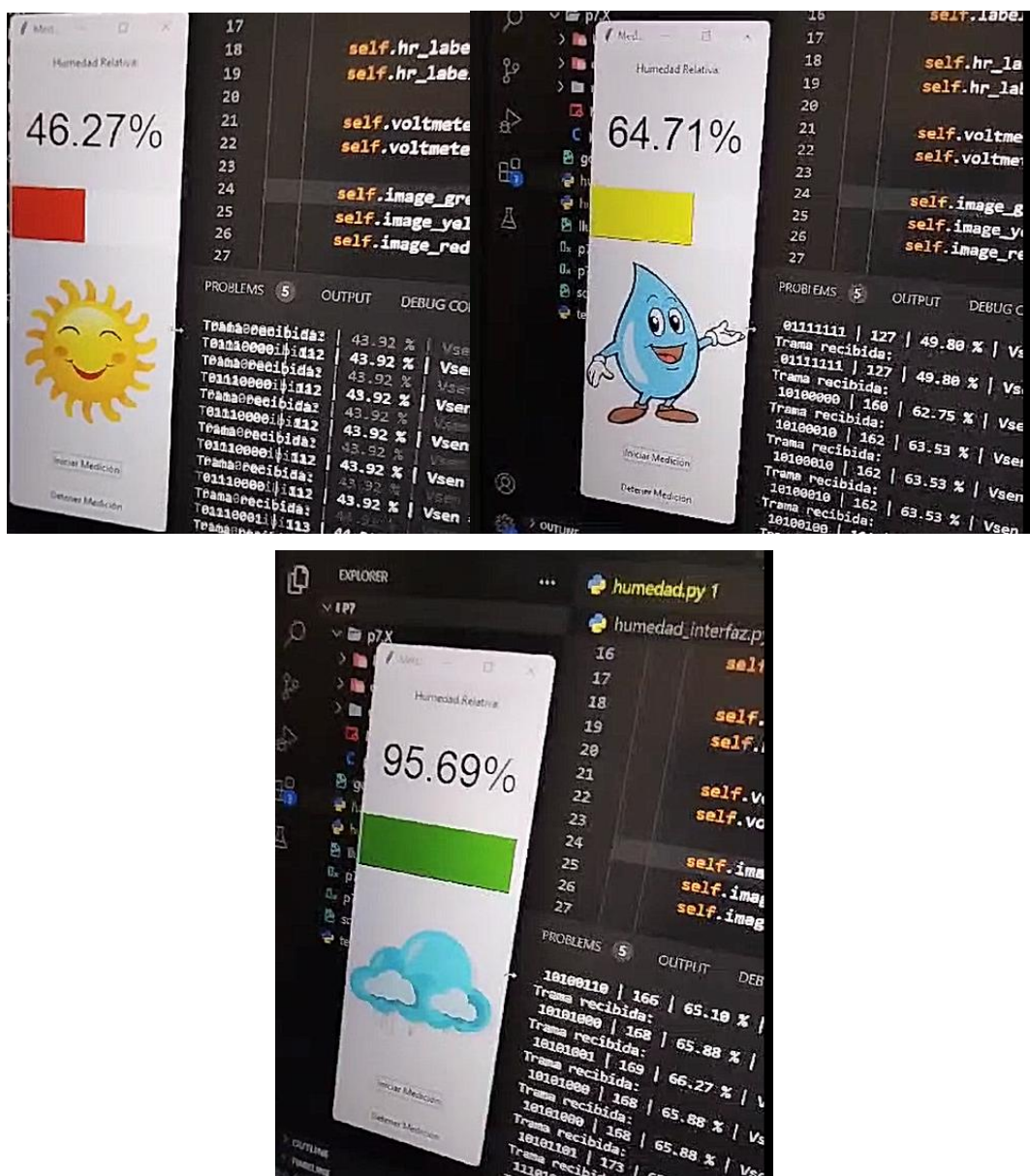


Imagen 35 interfaz para representar visualmente la humedad (lluvia, gota y sol)

## Cálculos

### CAS

Se respetó los rangos de amplificación de nuestro diseño anterior del CAS, el cual es:

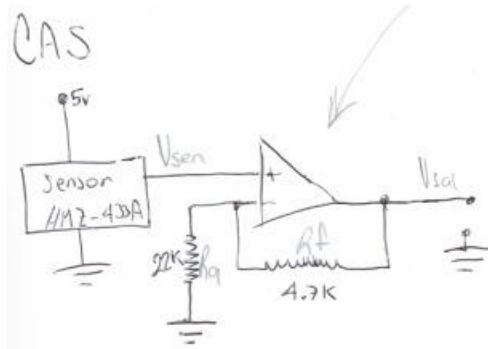


Imagen 36 Diseño del CAS

	HMZ-433A (0 V - 3.3 V)							
%HR	%30	%40	%50	%60	%70	%80	%90	%100
$V_o(V)$	1 V	1.32 V	1.65 V	1.98 V	2.31 V	2.64 V	2.97 V	3.3 V

Tabla 1 Tabla de valores del sensor de humedad

Se requiere amplificar a 0 V – 4 V

**Sensor: 0v – 3.3v    ->    CAS: 0v – 4.0v**

$$V_{sal} = V_{sen} \left( 1 + \frac{R_f}{R_a} \right)$$

Donde

$$A_v = 1 + \frac{R_f}{R_a}$$

$$R_a = \frac{R_f}{A_v - 1}$$

$$V_{sen}(Alcance) = V_{\max} - V_{\min} = 3.3 V - 0V = 3.3 V$$

$$V_{sal}(Alcance) = V_{\max} - V_{\min} = 4.0 V - 0V = 4.0 V$$

$$A_v = \frac{V_{sal}}{V_{sen}} = \frac{4.0v}{3.3v} \approx 1.21$$

$$Sea R_f = 4.7k \Omega$$

$$A_v = 1 + \frac{R_f}{R_a} \Rightarrow \frac{40}{33} = 1 + \frac{4.7k\Omega}{R_a} \Rightarrow \frac{7}{33} = \frac{4.7k\Omega}{R_a} \Rightarrow R_a = \frac{4.7k\Omega}{7/33}$$

$$R_a = \frac{4.7k\Omega}{7/33} = \frac{155100\Omega}{7} = 22,157.14286\Omega$$

Donde el valor comercial más cercano es

$$R_a = 22k\Omega$$

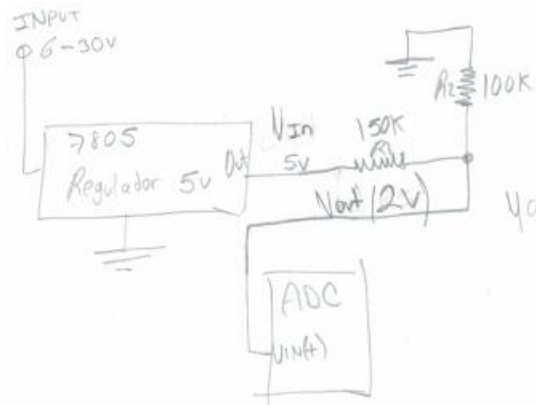
## ADC

### Divisor de voltaje

Se calculo las resistencias que se iba a usar en el divisor de voltaje para que de 0 – 5 V salga 2V que es el voltaje de referencia que es la mitad de este para sumar los 4v que es el rango que operamos con el CAS.

Voltaje: 0 V – 5 V a 0 V – 2 V

Ya que nuestro  $V_{ref}$  es de 0 V a 4 V y el ADC trabaja a  $\frac{1}{2}V_{ref}$ .



$$V_{out} = V_{ref} \left( \frac{R_2}{R_1 + R_2} \right) \quad \text{donde: } V_{out} = 2.0v \quad V_{in} = 5.0v$$

Proponiendo

$$R_2 = 100k\Omega$$

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} \Rightarrow \frac{V_{out}}{V_{in}} (R_1 + R_2) = R_2 \Rightarrow \frac{2v}{5v} (R_1 + 100k) = 100k\Omega$$

$$0.4R_1 + 40k = 100k \Rightarrow \left(\frac{2}{5}\right)R_1 = 60k\Omega \Rightarrow R_1 = \frac{60k\Omega}{2/5} \Rightarrow R_1 = 150k\Omega$$

$$R1 = 150k \Omega \quad R2 = 100k \Omega$$

%HR	Vsen	VCAS	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	DECIMAL
			128	64	32	16	8	4	2	1	
0%	0 V	0 V	0	0	0	0	0	0	0	0	0
100%	3.3 V	4.0 V	1	1	1	1	1	1	1	1	255
44.7058%	1.4752 V	1.7882 V	0	1	1	1	0	0	1	0	114

$$\text{Combinación binaria} = \frac{V_{sal} - V_{ref}}{V_{res}}$$

$$\text{Para } 0\% \Rightarrow \frac{0V}{\frac{4}{255}V} = 0$$

$$\text{Para } 100\% \Rightarrow \frac{4V}{\frac{4}{255}V} = 255$$

$$V_{res} = \frac{V_{rango}}{2^n - 1} = \frac{4V}{2^8 - 1} = \frac{4V}{255} \approx 0.0156 V$$

Para calcular el %HR en Python

$$\text{Convertir a decimal} \Rightarrow 01110010 \Rightarrow 114$$

$$VCAS = \text{DECIMAL}(V_{res})$$

$$VCAS = 114 \left( \frac{4}{255} V \right) = \frac{152}{85} \approx 1.7882 V$$

$$V_{sen} = \frac{VCAS}{A_v} = \frac{152/85}{40/33} = \frac{627}{425} V = 1.4752 V$$

Si

$$100\%HR = 3.3 V$$

$$\%HR = V_{sen}$$

$$\%HR = \frac{V_{sen} * 100}{3.3} = \frac{627}{425} * 100 = \frac{760}{17} \% = 44.7058\%HR$$

## TABLAS DE MEDICIONES

Tabla de mediciones del sensor de humedad relativa %HR (MEDIDO)

%HR	$V_{sen}$ (V)	$V_{sat}$ (V)	Combinación binaria								Decimal
			b7	b6	b5	b4	b3	b2	b1	b0	
			1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256	
0%	0.0 V	0 V	0	0	0	0	0	0	0	0	0
9%	0.3 V	0.3656 V	0	0	0	1	0	1	1	1	23
18%	0.6 V	0.7334 V	0	0	1	0	1	1	1	0	46
27%	0.9 V	1.0952 V	0	1	0	0	0	1	1	0	70
36%	1.2 V	1.4574 V	0	1	0	1	1	1	0	1	93
45%	1.5 V	1.8222 V	0	1	1	1	0	1	0	0	116
54%	1.8 V	2.1859 V	1	0	0	0	1	1	0	0	140
63%	2.1 V	2.5462 V	1	0	1	0	1	0	0	0	168
72%	2.4 V	2.9139 V	1	0	1	1	1	0	0	0	184
81%	2.7 V	3.2683 V	1	1	0	0	0	0	0	0	192
90%	3.0 V	3.6354 V	1	1	1	0	0	1	1	0	230
100%	3.3 V	4.1055 V	1	1	1	1	1	1	1	1	255

Tabla 2 mediciones del sensor de humedad relativa %HR (MEDIDO)

Tabla de mediciones del sensor de humedad relativa %HR (SIMULADO)

%HR	$V_{sen}$ (V)	$V_{sat}$ (V)	Combinación binaria								Decimal
			b7	b6	b5	b4	b3	b2	b1	b0	
			1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256	
0%	0.0 V	0.0 V	0	0	0	0	0	0	0	0	0
9%	0.3 V	0.3636 V	0	0	0	1	0	1	1	1	23
18%	0.6 V	0.7302 V	0	0	1	0	1	1	1	0	46
27%	0.9 V	1.0924 V	0	1	0	0	0	1	1	0	70
36%	1.2 V	1.4579 V	0	1	0	1	1	1	0	1	93
45%	1.5 V	1.8220 V	0	1	1	1	0	1	0	0	116
54%	1.8 V	2.2181 V	1	0	0	0	1	1	0	0	140
63%	2.1 V	2.5202 V	1	0	1	0	0	0	1	1	163
72%	2.4 V	2.9143 V	1	0	1	1	1	0	1	0	186
81%	2.7 V	3.2784 V	1	1	0	1	0	0	0	1	209
90%	3.0 V	3.6425 V	1	1	1	0	1	0	0	1	233
100%	3.3 V	4.0066 V	1	1	1	1	1	1	1	1	255

Tabla 3 Mediciones del sensor de humedad relativa %HR (SIMULADO)

## PRUEBAS Y TOMA DE MEDICIONES DEL CIRCUITO ARMADO

Probando el sensor con la humedad del aire ambiente (Lab. Electrónica) el cual resulto aproximado a 0.30 V(multímetro de color rojo) luego se observó que el voltaje de salida es 365.75 mV, y después se ve que equivale según lo indicado por los LED's a número binario 00010111(23 en decimal), siendo que 0 es cuando se apaga el LED y 1 si se prende así como muestra la Imagen 3.

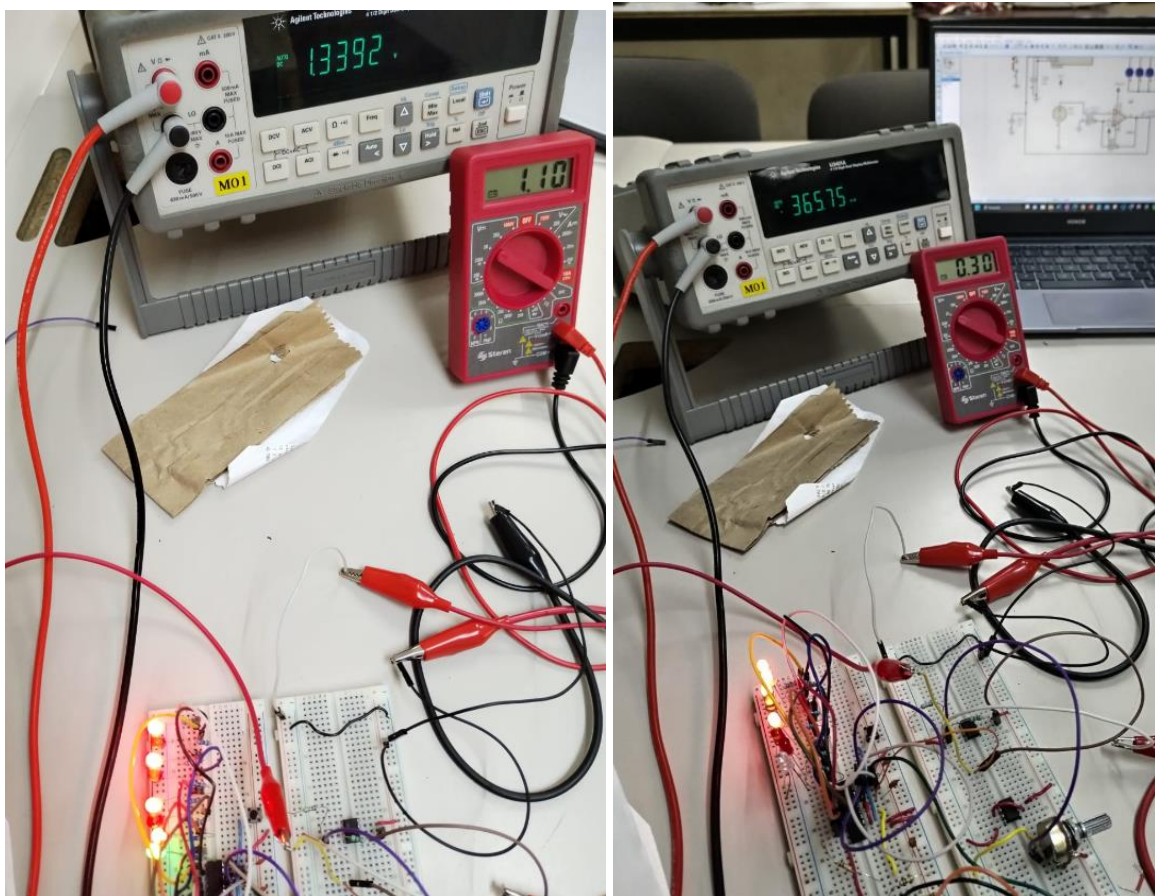
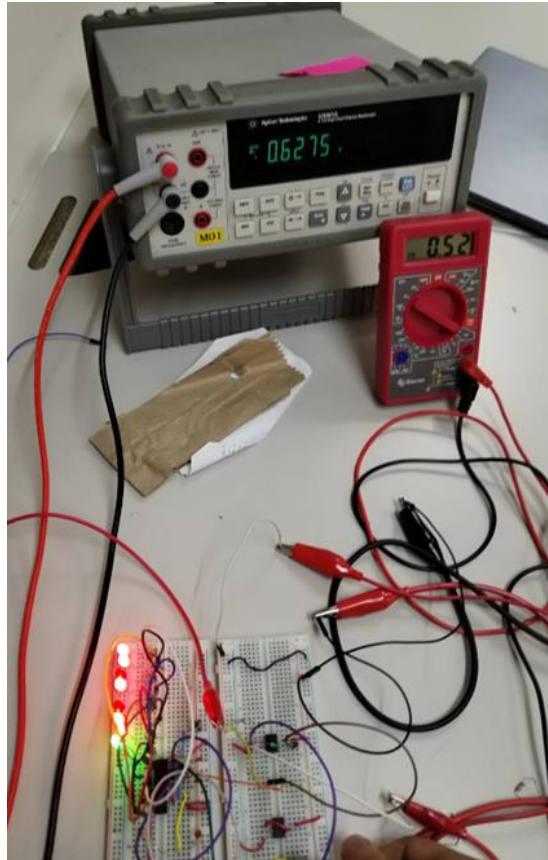


Imagen 37 | Probando Sensor Con Humedad del Ambiente (Lab. Electrónica)

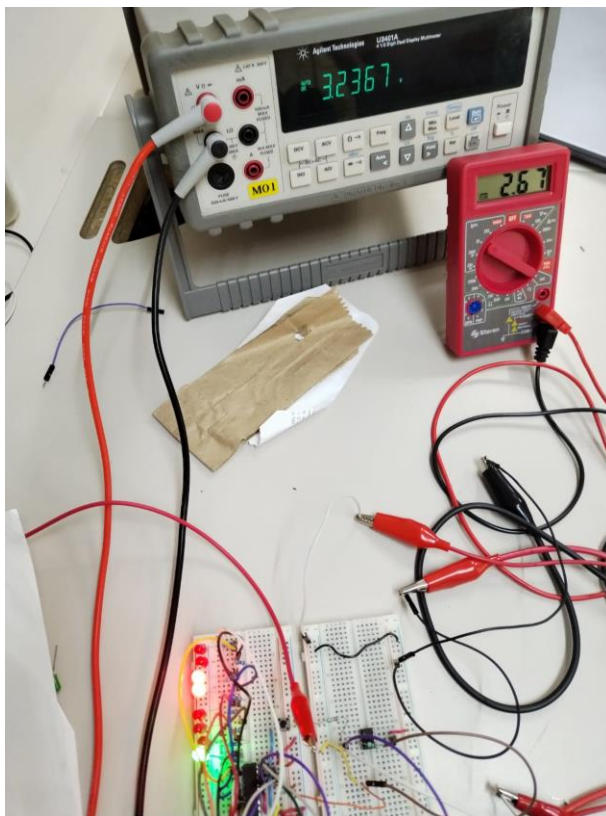


Usando la humedad de la boca, el sensor indicó aproximadamente 0.52V luego se observó que el voltaje de salida es 0.6275 V.



*Imagen 38 | Probando Sensor Con Humedad de la boca*

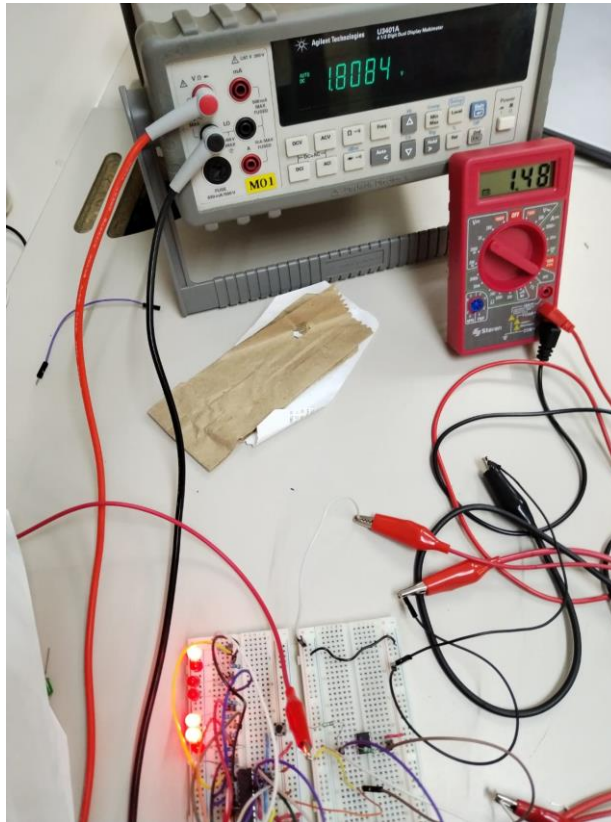
Usando el rociador para cabello, la humedad del sensor indicó aproximadamente 2.67V luego se observó que el voltaje de salida es 3.2367 V, y después se observa según lo indicado por los LED's el número binario 01110100 (192 en decimal).



*Imagen 39 | Probando Sensor Con Humedad del rociador*

Usando todavía el rociador para cabello, pero aplicando menos humedad, la humedad del sensor indicó aproximadamente 1.48V luego se observó que el voltaje de salida es 1.8084V, y después se observa según lo indicado por los LED's el número binario 11000000 (116 en decimal).

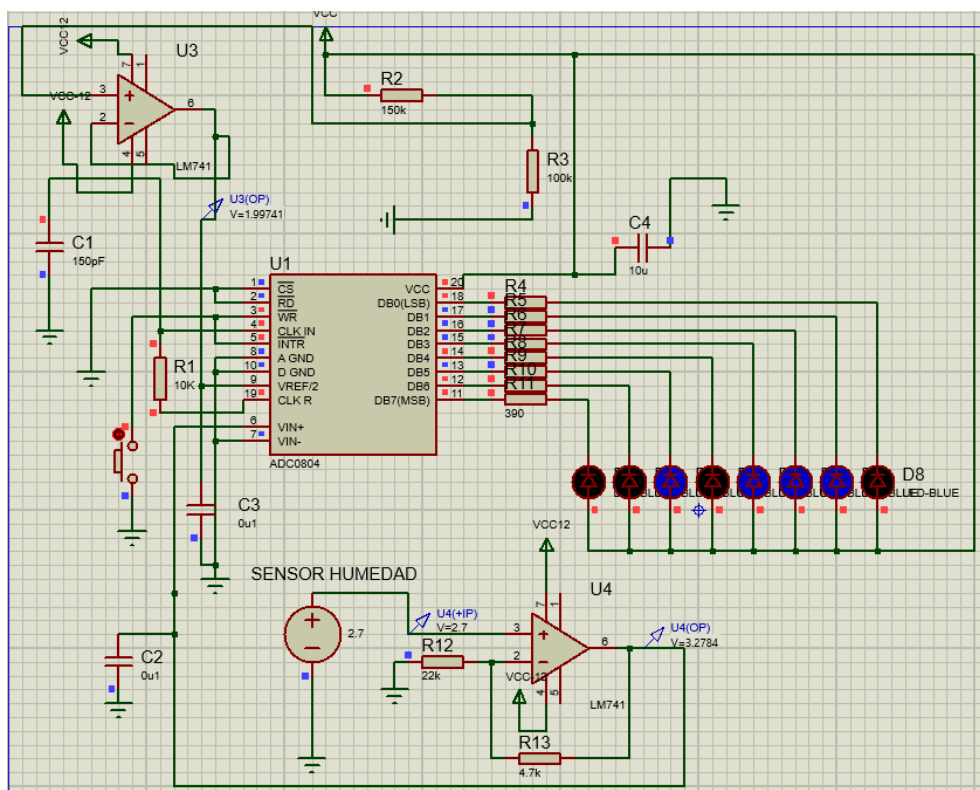
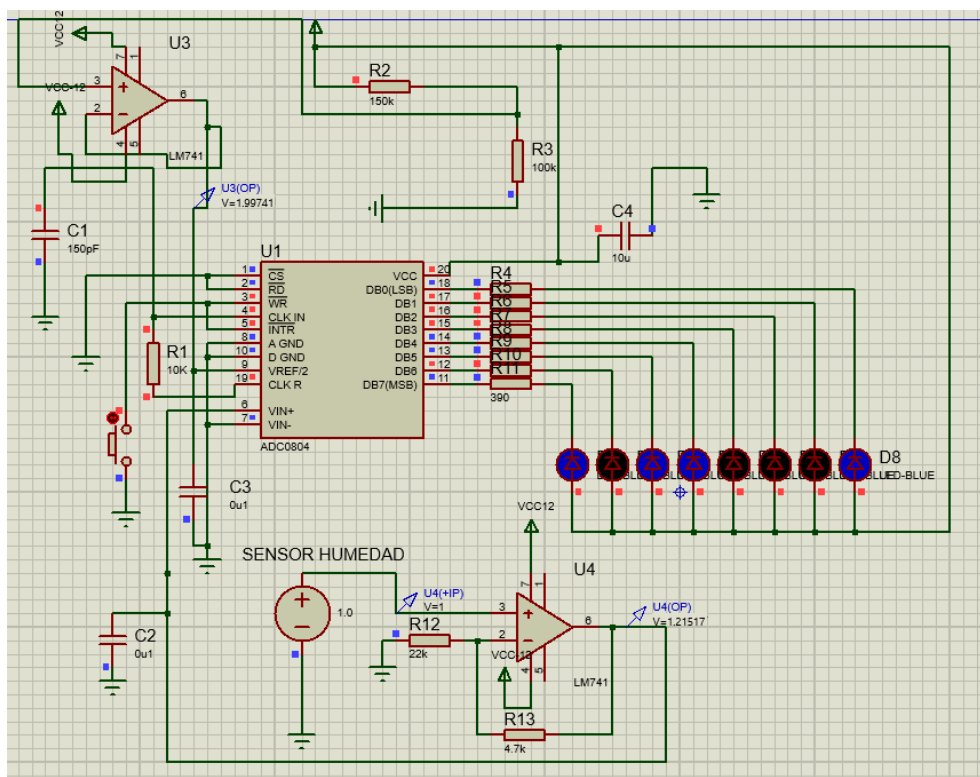




*Imagen 40 | Probando Sensor Con Humedad del rociador*

Se continuaron las mediciones y se registraron en la tabla de mediciones, así como sus simulaciones, y así queda nuestro circuito completo.

Probando Circuito Armado Completo en simulador:



# CONCLUSIONES

## Conclusiones generales

La realización de este proyecto, centrado en el desarrollo de un sistema de medición de humedad mediante un sensor HMZ-433A, ha proporcionado aplicar de manera práctica los conocimientos teóricos adquiridos sobre el Circuito de Acondicionamiento de Señal (CAS), el Convertidor Analógico a Digital (ADC) y el microcontrolador PIC16F628A. Esta implementación permitió la conexión efectiva del sensor de humedad al sistema.

El ADC se destacó como un componente esencial ya que su función de convertir las señales analógicas del sensor en datos digitales facilitó la posterior interpretación y procesamiento por parte del microcontrolador.

Se profundizó en la comprensión y aplicación del protocolo RS-232 para la comunicación entre el microcontrolador y la computadora.

El sensor HMZ-433A se destacó como un componente fundamental para medir la humedad en diferentes materiales. Su aplicación práctica en el proyecto destaca su utilidad en unas muchas aplicaciones industriales y de consumo.

La comprensión de los métodos de impresión de circuitos impresos, en particular el método de planchado ofrece una perspectiva valiosa sobre la producción de circuitos en entornos de laboratorio y proyectos prácticos.

## Conclusiones específicas

- **García Quiroz Gustavo Ivan**

El proyecto integral de diseño e implementación de un sistema de medición de humedad, abarcando desde el Circuito de Acondicionamiento de Señal (CAS) hasta la conexión con la computadora mediante el protocolo RS-232, ha proporcionado valiosas lecciones y resultados significativos.

Se observó que el protocolo RS-232, al utilizar señalización de un solo extremo, mostró sensibilidad a interferencias y ruido, especialmente en entornos donde las diferencias en el

voltaje de tierra pueden afectar la calidad de la comunicación. Esto puede generar lecturas indeterminadas y requerir medidas adicionales para mitigar posibles problemas.

La estructura del formato de protocolo RS-232, con bits de inicio, datos, paridad y parada, si bien es estándar, puede considerarse compleja para aplicaciones más simples. La necesidad de seguir un formato específico puede aumentar la complejidad del diseño y la programación, especialmente para usuarios menos experimentados.

El proyecto logró los objetivos técnicos establecidos, y proporcionó una plataforma valiosa para el desarrollo de conocimientos del acondicionamiento de señal, el convertidor analógico a digital y el microcontrolador.

- **Ortiz González Alan**

Por medio de este proyecto hemos sido capaces de comprender y saber utilizar los diversos temas brindados por el docente a lo largo del curso, fuimos comprendiendo desde el como funciona el sensor hasta el punto de poder manipularlo para recibir la información que requerimos de forma binaria; para esta tercera entrega hicimos uso de las placas, por la cual, mediante la perforación, impresión y procesada, fuimos capaces de ahí meter nuestro circuito realizado previamente en protoboard.

Ahora, con esto dicho, se realizó un programa para cada porcentaje de humedad que nos brinde el mismo sensor mediante la transmisión de datos hacia nuestro equipo de cómputo.

- **Romero Hernández Oscar David**

El proyecto de diseño e implementación de un sistema de medición de humedad, centrado en el sensor HMZ-433A, permitió aplicar conocimientos teóricos en acondicionamiento de señal, Convertidor Analógico a Digital (ADC) y microcontrolador PIC16F628A. Aunque el protocolo RS-232 mostró sensibilidad a interferencias, se lograron los objetivos técnicos, destacando la importancia del ADC y del sensor en diversas aplicaciones. Resolvimos el desafío del proyecto y demostramos habilidades en la producción de circuitos impresos. En conjunto, el proyecto proporcionó una valiosa experiencia práctica, resaltando la aplicación efectiva de conocimientos en ingeniería electrónica combinada con sistemas.

## Anexo Código

### Anexo Código para programar al microcontrolador

A continuación, esta expresado el código para programar al microcontrolador, mismo que puede copiar al editor en lenguaje C, para generar el archivo hexadecimal (.hex).

```
// CONFIG
#pragma config FOSC = INTOSCIO // Oscillator Selection bits (INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)
#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = ON // Power-up Timer Enable bit (PWRT enabled)
#pragma config MCLRE = OFF // RA5/MCLR/VPP Pin Function Select bit (RA5/MCLR/VPP pin function is digital input, MCLR internally tied to VDD)
#pragma config BOREN = OFF // Brown-out Detect Enable bit (BOD disabled)
#pragma config LVP = OFF // Low-Voltage Programming Enable bit (RB4/PGM pin has digital I/O function, HV on MCLR must be used for programming)
#pragma config CPD = OFF // Data EE Memory Code Protection bit (Data memory code protection off)
#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.
#include <xc.h>
#include <pic16f628a.h>
#define _XTAL_FREQ 4000000
#define Bit7 PORTAbits.RA7
#define Bit6 PORTAbits.RA6
#define Bit5 PORTAbits.RA5
#define Bit4 PORTAbits.RA4
#define Bit3 PORTAbits.RA3
#define Bit2 PORTAbits.RA2
#define Bit1 PORTAbits.RA1
#define Bit0 PORTAbits.RA0
#define RxUART PORTBbits.RB1
#define TxUART PORTBbits.RB2
#define Led_Tx PORTBbits.RB6
#define Led_Rx PORTBbits.RB7
#define GIE INTCONbits.GIE
#define TOIE INTCONbits.T0IE
#define TOIF INTCONbits.T0IF
#define PEIE INTCONbits.PEIE
#define SPEN RCSTAbits.SPEN
#define CREN RCSTAbits.CREN
```

```

#define TRMT TXSTAbits.TRMT
#define RCIE PIE1bits.RCIE
#define TXIE PIE1bits.TXIE
#define TXIF PIR1bits.TXIF
#define RCIF PIR1bits.RCIF
char ByteRx;
char TxUSART;
void __interrupt() VectorInterrupcion(void) // Funciones de interrupción.
{
if (RCIF == 1) //Interrupción por uso de UART (RS-232)
{ GIE = 0; //Desactivación general de interrupciones
ByteRx = RCREG;
RCIF = 0; //Limpia la bandera de interrupción por recepción
GIE = 1; //Activa las interrupciones
}
}
void ConfigPIC(void)
{
TRISA = 0B11111111; //Puerto A como entradas
TRISB = 0B00000010; //Puerto B como salidas solo bit 1 como entrada
CMCON = 0X07; //Terminales del puerto A como I/O digitales
PORTA = 0;
PORTB = 0;
}
void ConfigUSART(void) //Configuracion_USART 9600 BPS, sin bit de paridad, 1 bit stop
{
TXIE = 0; //Desactiva interrupción por fin de transmisión por usart.
TXSTA = 0B00100110;
SPBRG = 25;
RCSTA = 0B10010000;
//Este bit es parte del registro RCSTA SPEN = 1; //Habilitación del puerto de comunicación serial
//Este bit es parte del registro RCSTA CREN = 1; //Activa la recepción continua
RCIE = 1; //Activar interrupción por fin de recepción por usart.
TXIF = 0;
RCIF = 0;
GIE = 1; //Activar habilitador general de interrupciones.
PEIE = 1; //Activar habilitador general de interrupciones por periféricos.
RCREG = 0;
}
void Transmite(void) //Transmite por RS-232 9600 BPS, sin bit de paridad, 1 bit stop
{
TXSTA = 0B00100110;
SPBRG = 25;

```

```
TXREG = TxUSART;
while(!TXIF);
while(!TRMT); //Esta bandera es la que funciona para detectar cuando se vacía el buffer de transmisión
}
void main(void)
{
    ConfigPIC();
    ConfigUSART();
    while (1)
    { TxUSART = PORTA;
      Led_Tx = 1;
      Transmite();
      __delay_us(100);
      Led_Tx = 0;
      __delay_ms(1000);
    }
    return;
}
```

## Anexo Código para el proyecto

### Script 'humedad.py'

Este script se utiliza típicamente para leer y procesar datos de un sensor de humedad conectado a un puerto serial de la computadora.

```
import serial

V_RES = 4 / 255
AV = 40 / 33
V_MAX_SEN = 3.3

# Configuración del puerto serial
puerto_serial = serial.Serial('COM3', 9600, timeout=1) # Reemplaza 'COMX' con el nombre correcto del puerto

try:
    while True:
        # Lee una trama de 8 bytes desde el puerto serial
        trama_recibida = puerto_serial.read(8)

        print(f"Trama{trama_recibida}")

        # Muestra los datos de la trama recibida en formato binario y decimal
        if trama_recibida:
            datos_binarios = ''.join(format(byte, '08b') for byte in trama_recibida)
            datos_decimales = int.from_bytes(trama_recibida, byteorder='big')

            vcas = datos_decimales * V_RES
            vsen = vcas / AV
            hr = (vsen * 100) / V_MAX_SEN

            print(f"Trama recibida: \n {datos_binarios} | {datos_decimales} | {hr:.2f} % | Vsen = {vsen:.4f} | Vcas {vcas:.4f}")
except KeyboardInterrupt:
    print("Programa interrumpido por el usuario")

finally:
    # Cierra el puerto serial al finalizar
    puerto_serial.close()
```



## Script 'humedad\_interfaz.py'

Este script proporciona una solución completa para la adquisición y visualización de datos de humedad a través de una interfaz gráfica de usuario en una PC.

```
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk
import serial

V_RES = 4 / 255
AV = 40 / 33
V_MAX_SEN = 3.3

class HumidityMonitorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Medición de Humedad Relativa")

        self.label = ttk.Label(root, text="Humedad Relativa:")
        self.label.pack(pady=10)

        self.hr_label = tk.Label(root, text="", font=("Helvetica", 36))
        self.hr_label.pack(pady=20)

        self.voltmeter_canvas = tk.Canvas(root, width=200, height=60, bg="white")
        self.voltmeter_canvas.pack(pady=10)

        self.image_green = self.load_resized_image("lluvia.png", (200, 200))
        self.image_yellow = self.load_resized_image("gota2.png", (200, 200))
        self.image_red = self.load_resized_image("sol.png", (200, 200))

        self.image_label = tk.Label(root, image=self.image_green)
        self.image_label.pack(pady=10)

        self.start_button = ttk.Button(root, text="Iniciar Medición", command=self.start_measurement)
        self.start_button.pack(pady=10)

        self.stop_button = ttk.Button(root, text="Detener Medición", command=self.stop_measurement)
```

```

self.stop_button.pack(pady=10)

self.puerto_serial = serial.Serial('COM3', 9600, timeout=1)
self.is_measuring = False

def start_measurement(self):
    self.is_measuring = True
    self.read_serial_data()

def stop_measurement(self):
    self.is_measuring = False

def update_voltmeter(self, humidity):
    voltage_ratio = humidity / 100
    fill_color = "green" if voltage_ratio > 0.8 else "yellow" if voltage_ratio > 0.5 else "red"
    self.voltmeter_canvas.delete("all")
    self.voltmeter_canvas.create_rectangle(0, 0, 200 * voltage_ratio, 60, fill=fill_color)

def update_image(self, humidity):
    humidity = humidity / 100
    if humidity > 0.8:
        self.image_label.config(image=self.image_green)
    elif humidity > 0.5:
        self.image_label.config(image=self.image_yellow)
    else:
        self.image_label.config(image=self.image_red)

def read_serial_data(self):
    try:
        while self.is_measuring:
            trama_recibida = self.puerto_serial.read(8)

            if trama_recibida:
                datos_binarios = ''.join(format(byte, '08b') for byte in trama_recibida)
                datos_decimales = int.from_bytes(trama_recibida, byteorder='big')
                vcas = datos_decimales * V_RES
                vsen = vcas / AV
                hr = (vsen * 100) / V_MAX_SEN

                print(f'Trama recibida: \n {datos_binarios} | {datos_decimales} | {hr:.2f} % | Vsen =
{vsen:.4f} | Vcas {vcas:.4f}')

                self.hr_label.config(text=f'{hr:.2f} %')

```

```

        self.update_voltmeter(hr)
        self.update_image(hr)

        self.root.update_idletasks()
        self.root.update()

    except KeyboardInterrupt:
        print("Programa interrumpido por el usuario")

    finally:
        self.puerto_serial.close()

def load_resized_image(self, path, size):
    original_image = Image.open(path)
    resized_image = original_image.resize(size, Image.ANTIALIAS)
    return ImageTk.PhotoImage(resized_image)

if __name__ == "__main__":
    root = tk.Tk()
    app = HumidityMonitorApp(root)
    root.mainloop()

```

## BIBLIOGRAFÍA

[01] HMZ-433A Datasheet (PDF) - RFE international. (s/f). Alldatasheet.es. Recuperado el 4 de octubre de 2023, de <https://www.alldatasheet.es/datasheet-pdf/pdf/334403/RFE/HMZ-433A.html>

[02] Medidores de Humedad. (s/f). Proconsa México. Recuperado el 4 de octubre de 2023, de <https://proconsamexico.com/categoria-de-producto/medidores-de-humedad/>

[03] (S/f). Cartagena99.com. Recuperado el 4 de octubre de 2023, de <https://www.cartagena99.com/recursos/alumnos/apuntes/tema-4-sensores-generadores.pdf>

[04] HMZ-433A Datasheet (PDF) - RFE international. (s/f). Alldatasheet.es. Recuperado el 4 de octubre de 2023, de <https://www.alldatasheet.es/datasheet-pdf/pdf/334403/RFE/HMZ-433A.html>

[05] Medidores de Humedad. (s/f). Proconsa México. Recuperado el 4 de octubre de 2023, de <https://proconsamexico.com/categoria-de-producto/medidores-de-humedad/>

[06] (S/f). Cartagena99.com. Recuperado el 4 de octubre de 2023, de <https://www.cartagena99.com/recursos/alumnos/apuntes/tema-4-sensores-generadores.pdf>

[07] Como funciona el convertidor analógico a digital. (s.f.). Ingeniería Mecafenix. <https://www.ingmecafenix.com/electronica/componentes/convertidor-analogico-a-digital/>

[08] Vazquez Gallego, F. (s.f). [https://openaccess.uoc.edu/bitstream/10609/141046/10/PLA3\\_Conversi%C3%B3n%20anal%C3%B3gico-digital.pdf](https://openaccess.uoc.edu/bitstream/10609/141046/10/PLA3_Conversi%C3%B3n%20anal%C3%B3gico-digital.pdf). Repositori Institucional (O2): Página de inicio. [https://openaccess.uoc.edu/bitstream/10609/141046/10/PLA3\\_Conversión%20analógico-digital.pdf](https://openaccess.uoc.edu/bitstream/10609/141046/10/PLA3_Conversión%20analógico-digital.pdf)

[09] Conversor analógico-digital. (s.f.). ingeniatic.  
<https://www.etsist.upm.es/estaticos/ingeniatic/index.php/tecnologias/item/425-conversor-analogico-digital.html>

[10] Módulo PL2303 Convertidor USB a serial TTL. (s/f). Naylamp Mechatronics - Perú. Recuperado el 29 de diciembre de 2023, de <https://naylampmechatronics.com/conversores-ttl/40-modulo-pl2303-conversor-usb-a-serial-ttl.html>

[11] Weis, O. (2020, abril 1). Última revisión sobre el protocolo RS232. Serial Port Monitor. <https://www.serial-port-monitor.org/es/articles/serial-communication/rs232-interface/>

#### **videos del programa**

[12] Google Drive. (s.f.). Carpeta de Google Drive. Recuperado de [https://drive.google.com/drive/folders/1ep3VQIDnU1olrGRG\\_TZfMjBiYtbaQv7w?usp=drive\\_link](https://drive.google.com/drive/folders/1ep3VQIDnU1olrGRG_TZfMjBiYtbaQv7w?usp=drive_link)