**Universidad Autónoma de Tamaulipas**

Facultad de ingeniería Arturo Narro Siller

Materia: Diseño Electrónico Basado en Sistemas Embebidos

Nombre: Gonzales Saldívar Luis Roberto

Guerrero Gamez Francisco Javier

Martínez Reyes Fernando

Sánchez Ramírez Alan Ariel

Villalobos de León Juan Carlos

Grupo: 8-I

Maestro: García Ruiz Alejandro Humberto

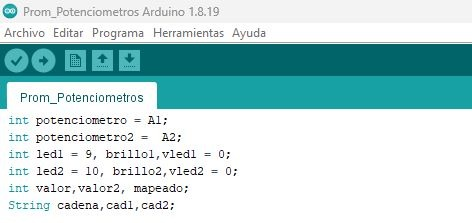
Documentación del Proyecto: medición de potenciómetro sacando la normalización, estandarización y complemento

**Introducción**

Crear un programa que simule un sistema embebido para determinar la posición a encender de un diodo emisor de luz, basándose en la lectura de los datos recibidos de un instrumento para medir las diferencias de potencial eléctrico (potenciómetro), aplicando formulas estadísticas tales como: Estandarización, Normalización y Complemento

**Código Arduino**

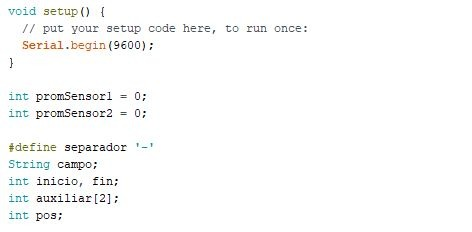
**Parte 1**



1.- Se declara los potenciómetros en el Arduino como A1 y A2.

Se declaran los leds en los pines del arduino 9 y 10 y se declaran las variables int, llamadas: Valor, Valor2 y Mapeado

De igual manera de declaran las variables String, llamadas: Cadena, Cad1 y Cad2



2.- Se declaran y se inicializan las variables PromSensor1 y PromSensor2 con valor 0 de tipo int.

Se declara la variable campo de tipo String,

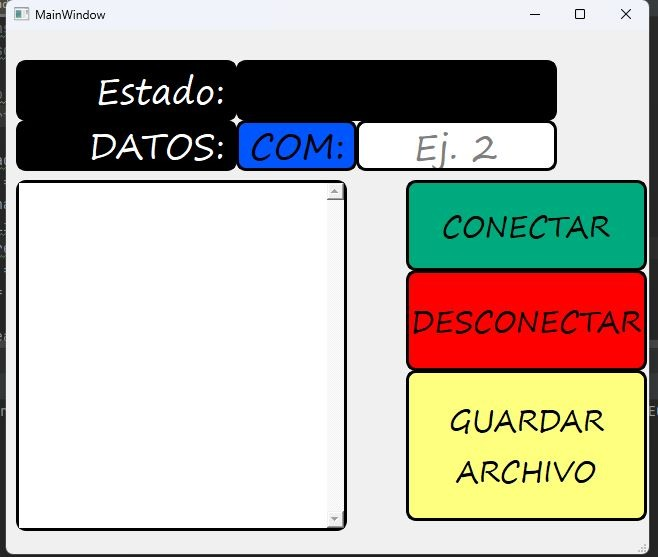
Se declaran la variable inicio y fin de tipo int

Y se declara la variable pos de tipo int



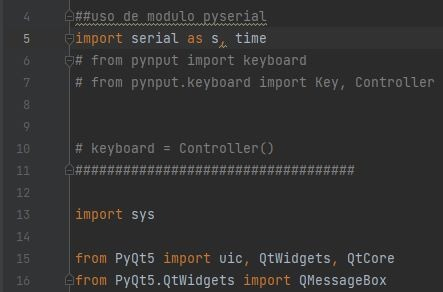
3.- Las variables valor y valor2 se les asigna el pin al que están asociado (potenciómetro) con al función serial analogRead

Se leen los valores de los potenciómetros 1 y 2 y se guardan en las variables Valor y Valor2. Posteriormente se imprimen los valores leídos con el promedio calculado de cada sensor y por ultimo los imprime con un delay de 10



4.- Con el programa QTdesigner se creo una interfaz preliminar en donde se ingresa el puerto COM y posteriormente se da inicio al botón conectar para generar datos, haciendo los visibles en la ventana de la interfaz y transferirlos a Python

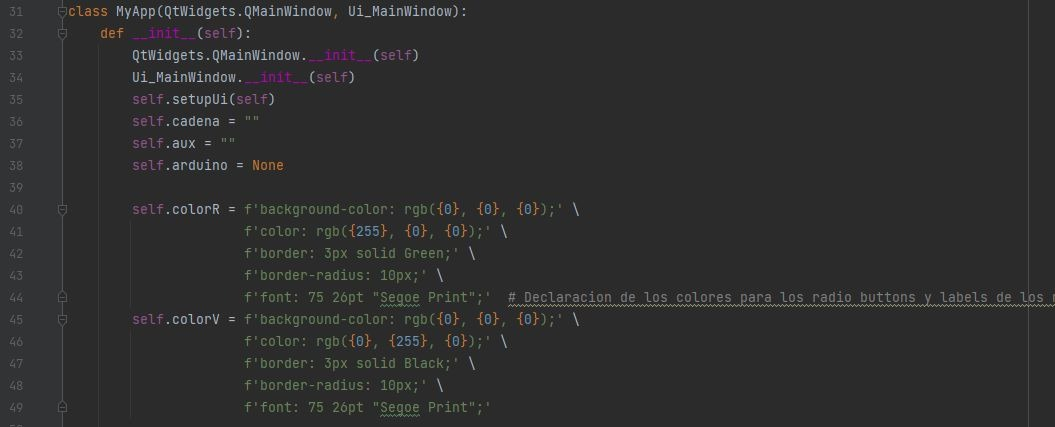
Con el botón guardar archivo, se generara un .TXT con los datos previamente mencionados



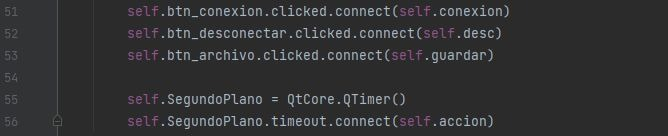
5.- Una vez en Python, se importa el módulo serial para conectar Arduino con Python, se importa el modulo PyQt5, uic, QtWidgets y QtCore para que se ejecute la interfaz grafica



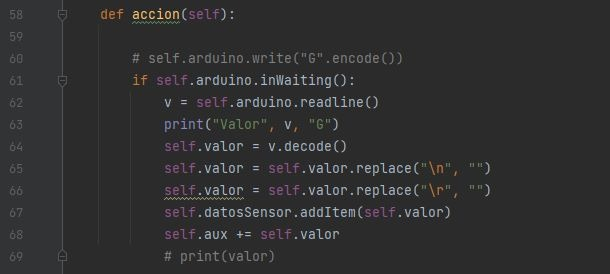
6.- Se declaran las variables y se inicializan algunas. Posteriormente se declara el nombre de la interfaz a ejecutar, realizada en QtDesigner



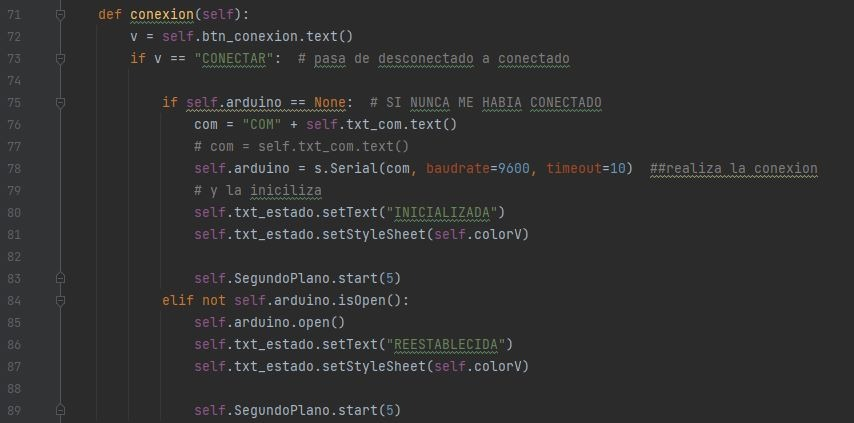
7.- Se declaran y se inicializan algunas variables con self para usarlos en todos los metodos del programa, además de asignarle color a radio buttona y labels de la interfaz



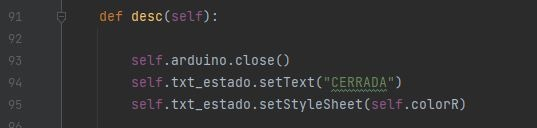
8.- Se conectan cada uno de los botones utilizados en la interfaz con un método en Python con la instrucción Clicked.connect. Se establece en segundo plano con Qtcore y timeout



9.- El método acción, se guardan datos que se encuentran en la placa de Arduino y se guardan en variable auxiliar



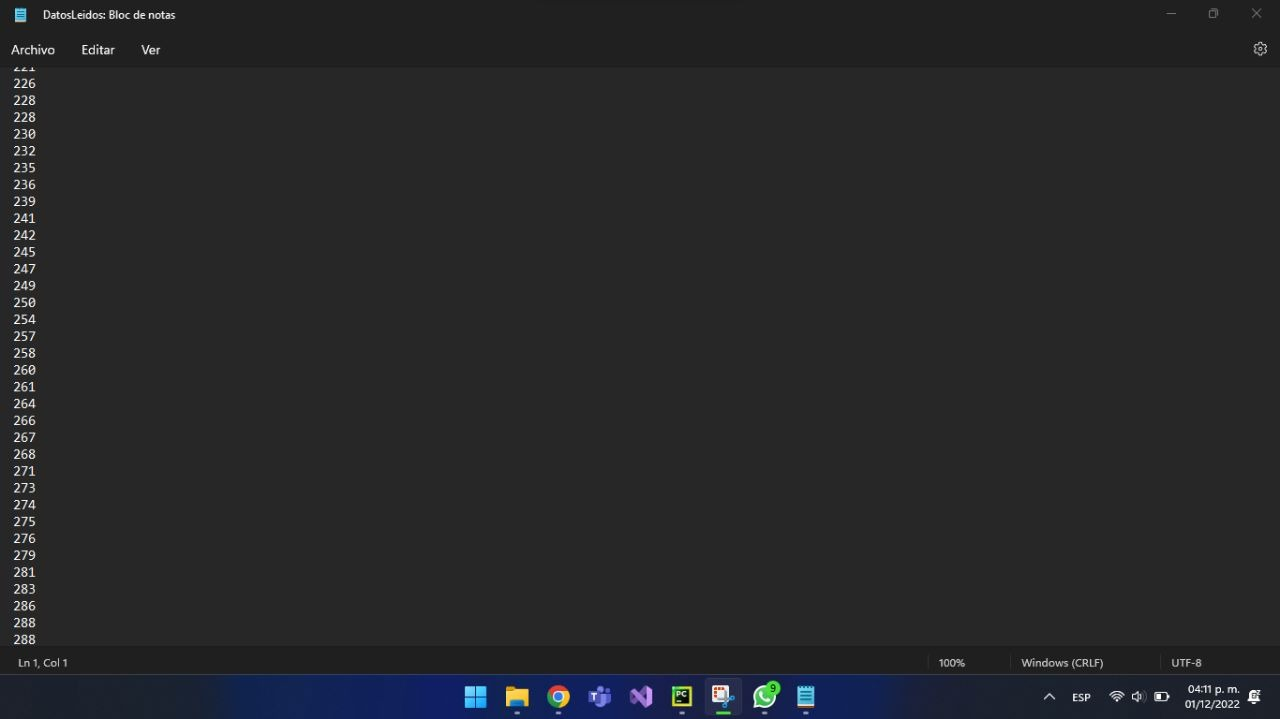
10.- En la variable v se almacena el texto, del botón conexión y de eso depende que las instrucciones se realizaran, y si v es igual a conectar establece la conexión con Arduino con s.Serial y se cambia el texto de estado a INICIALIZADO y para finalizar se inicia el segundo plano y como parámetro se le asigna el valor de 5.



11.- con el método desc se desconecta la conexión de Arduino con self.arduino.close(), y por ultimo se pone en el txt\_estado ”CERRADA” para ver en la interfaz que esta cerrada la conexión.

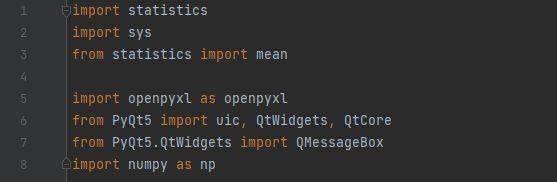


12.- En el método guardar con ayuda de un for recorra todo el contenido de aux dvidio por puntos con Split, en la variable f se asigna el nombre del txt y se abre para guardar los datos, en self.cadena se almacena en una columna las itreraciones y en la siguiente columna el valor de e, después con f.write se guarda en el txt lo que esta almacenado en self.cadena, y se cierra.

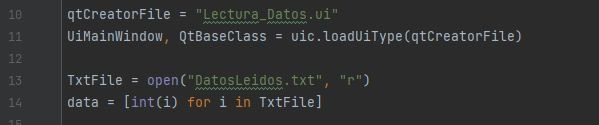


13.- Prueba de datos almacenados en el .TXT

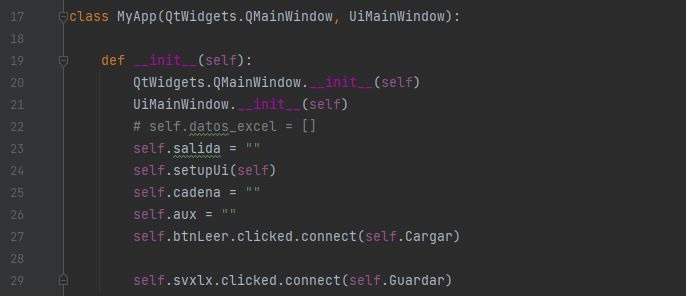
Parte 2



1.- Se importa el modulo mean de statistics, para realizar el promedio, y también se utiliza openpyxl para guardar los datos de pyhton en excel, además de importar PyQt5 para utilizar interfaz grafica con qt designer.



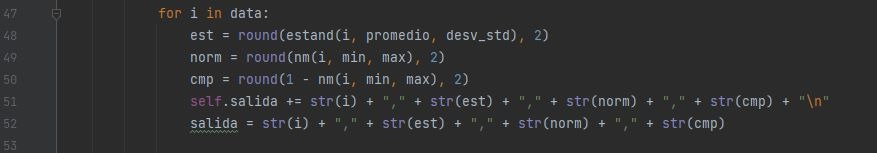
2.- se asigna el archivo de qt desginer llamado “Lectura\_datos.ui” para que se vea la interfaz realizada, posteriormente se le asigna a txtFile el valor de el text Datosleidos Y después en data se guarda todo el contenido de txtfile.



3.- En las líneas 23, 25 y 26 se definen las variables salida, cadena y auxiliar, con un valor de vacío, posteriormente se conecta el botón leer con el método cargar de la línea 27 y en la línea 29 se conecta el botón “svxlx” con el método guardar



4.- En este bloque se define el método cargar, para lo cual se tiene que calcular el valor máximo (en este caso) vinculado a la variable MAX, utilizando la función round que redonde el valor decimal a 2 dígitos, también calcula el valor mínimo, el promedio y el valor estándar. Posteriormente en el segmento de la línea 39 se guarda en la variable salida, se colocan los encabezados índice, estandarización, normalización y complemento, casteándolos a string



5.- Se utiliza un ciclo para recorrer el conjunto de datos y de esta forma calcular la estandarización, normalización y complemento, dichos métodos utilizan los cálculos anteriormente realizados con el segmento de código anterior



6.- Se define el método guardar, el cual se utilizara para almacenar los datos leídos y convertidos en formato “CSV”, en el cual en la variable datos se guardara el encabezado y se procede a usar un ciclo for para recorrer todos los datos del conjunto y finalmente se guardaran dichos datos



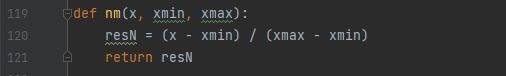
7.- Se define el método XMax el cual es para calcular el valor máximo del conjunto



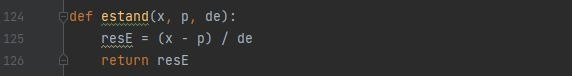
8.- Se define el método XMin el cual es para calcular el valor minimo del conjunto



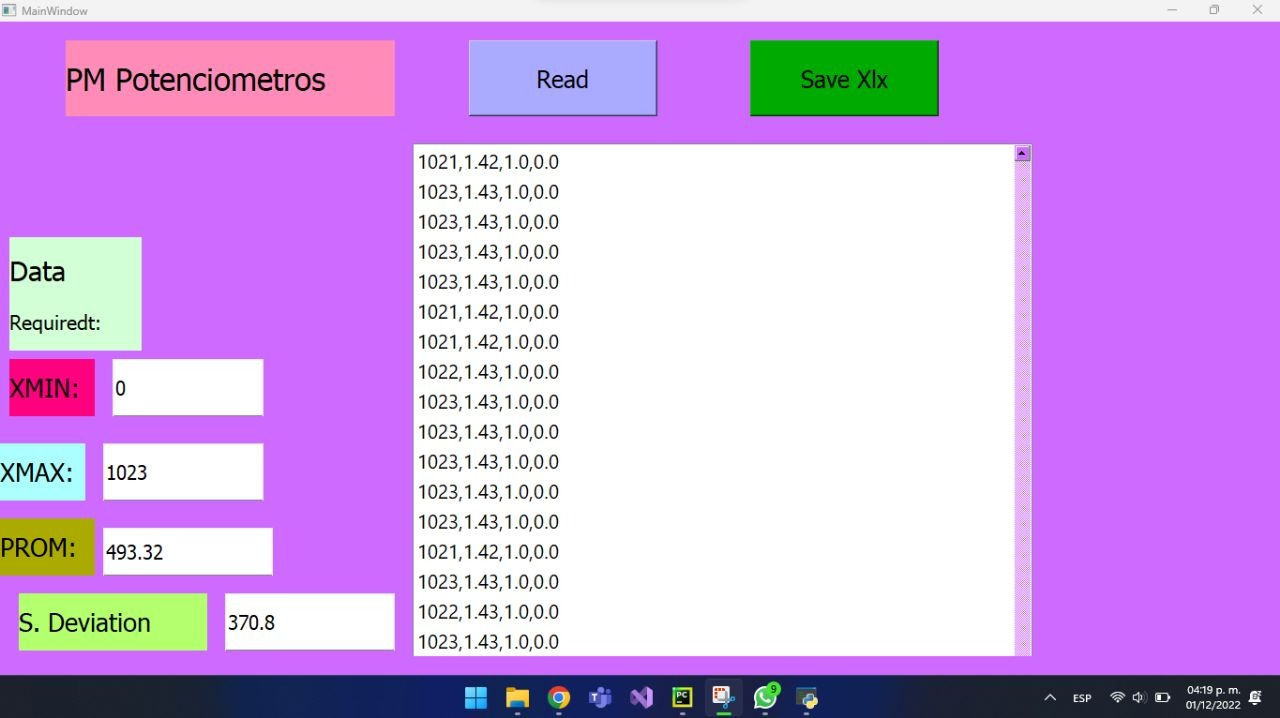
9.- Se define el método “mean”, el cual calcula el promedio del conjunto de datos



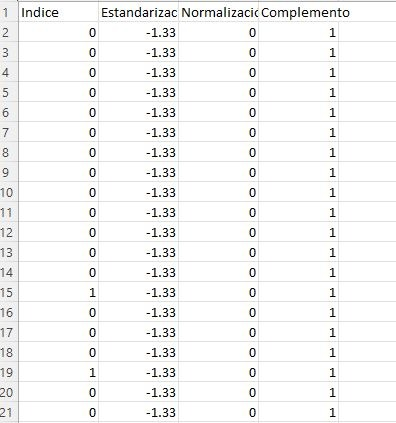
10.- Se define el método “nm”, para calcular la normalizacion



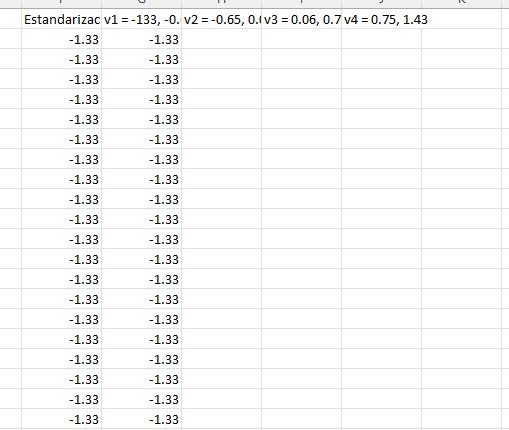
11.- Se define el método “estand”, para calcular la estandarizacion



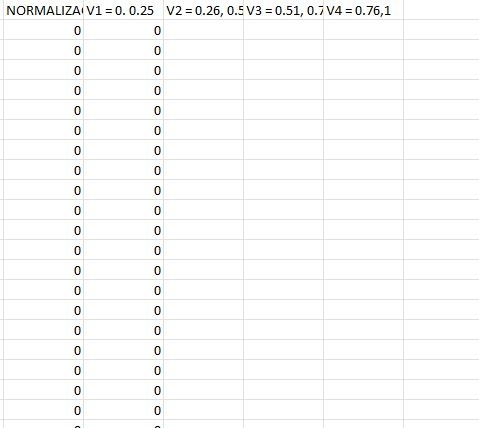
12.- Ahora teniendo una interfaz mas actualizada y eficiente, se procede a utilizar los componentes para la su ejecución. Con el botón Read, se encargara de leer los datos del archivo .txt. Utilizamos también el botón Save xlx, el cual guardara los datos convertidos en un archivo .TXT



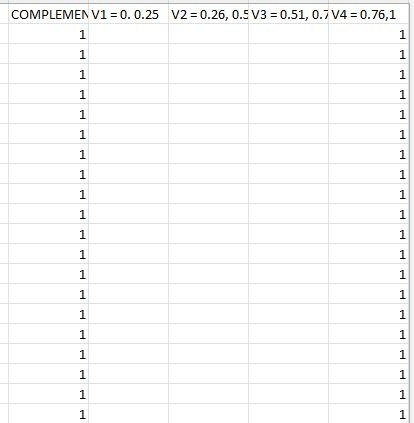
13.- Una vez teniendo los datos en el .TXT, se procede a cargar esta información en un archivo de Excel, con sus respectivos encabezados, posteriormente por cada medida estadística, se segmentara en 4 partes desde el valor mínimo al valor máximo.



14.- Se comienza por la estandarización y se divide en 4 partes, ejemplo: V1 = -133, -0.64, V2 = -0.65, 0.05, V3 = 0.06, 0.75 Y V4 = 0.75, 1.43



15.- Se comienza por la normalizacion y se divide en 4 partes, ejemplo: V1 = 0, 0.25, V2 = 0.26, 0.5, V3 = 0.51, 0.75 y V4 = 0.76, 1



16.- y por ultimo se comienza por el Complemento y se divide en 4 partes, ejemplo: V1 = 0, 0.25, V2 = 0.26, 0.5, V3 = 0.51, 0.75 y V4 = 0.76, 1

Una vez obteniendo estos datos con los segmentos correspondientes, se conectan 12 leds a la protoboard, 4 leds por cada medida estadística, y posteriormente se podrá leer nuevos datos desde el Arduino, aplicándoles cualquiera de una de estas tres medidas estadísticas y de esta forma, basándonos en los segmentos anteriormente mencionados se determina el “Led” a encender.