

MATERIA DE SISTEMAS EMBEBIDOS-PROYECTO 2B

SISTEMA DETECCIÓN DE POSTURA EN SILLA DE RUEDAS

Edwin A. Solano Arciniegas

26 de enero de 2021

1. Introducción

ADQUISICION DE DATOS Y ACONDICIONAMIENTO DE SEÑALES

en la actualidad hay miles de dispositivos los cuales convierten la información analógica en señales las cuales sea posible manipular, estos sistemas inteligentes de toma de datos son los famosos sensores o transductores que son capaces de traducir un fenómeno físico a señales eléctricas y hacer posible el uso de esos datos para un respectivo análisis, estos sistemas han ido evolucionando de acuerdo al desarrollo tecnológico y con el avance de microchips son mucho mejores y la información es más detallada en tal novedad la recolección y análisis es mucho más eficiente.

Para poder obtener los datos y que estos tengan la mayor eficiencia posible es necesario realizar la medición conforme indica el fabricante con la elección de materiales correctos y los elementos que hacen que el sistema funcione adecuadamente con el propósito del tema.

Un tema importante a tener en cuenta son los conversores análogos-digitales; estos, transforman una señal eléctrica a un número binario ahí es donde una computadora puede analizar los datos. Entre los varios conversores están ADC de aproximación sucesiva, conformado por un comparador y una lógica de control y registros para los datos que ingresan en el conversor. ADC de voltaje a frecuencia, ADC sigma delta rechazo de ruido, entre otros. Estos ADC deben tener dos factores importantes la precisión y la resolución esto sirve para aplicaciones de prueba y medición, el uno hace reducir a lo mínimo posible el error producido por factores externos una mala lectura representa una pérdida de datos.

La resolución o calibración se refiere al ajuste de ganancia y el desplazamiento de un canal ADC estos se calibran en fabrica, pero cuando el uso es continuo estos factores puede cambiar. Los fabricantes de ADC se vacan en histogramas de ruido para observar la presión del dispositivo ya que el ruido eléctrico puede afectar

considerablemente el resultado.

ACONDICIONAMIENTO DE SEÑALES FUNDAMENTALES

Para la adquisición de un dato se necesita un ADC el cual consta de un multiplexor, seguido de un amplificador de instrumentación, existen señales que son muy bajas y para poder aplicarlas se necesita de un sistema que amplifique estas señales este fenómeno lo usa la mayoría de sistemas y los más usados son los amplificadores operacionales que se configuran fácilmente para poder amplificar o disminuir el tamaño de una señal, el amplificador inversor acepta una señal de entrada, la amplifica e invierte la polaridad de los terminales de salida, el amplificador inversor hace lo mismo con la condición que las polaridades de entrada son las mismas de salida, los amplificadores diferenciales combinan los inversores y no inversores la señal de salida solo responde al voltaje diferencial de las dos terminales de entrada, amplificadores de ganancia programable son amplificadores operacionales no inversores el cual se puede controlar un interruptor analógico, amplificadores de instrumentación estos son para aplicaciones especiales sirven para aumentar la precisión y la estabilidad.

Existen también los llamados filtros entre los cuales se encuentran el filtro chebyshev, buttherworth, bessel, que son filtros los cuales hacen pasar cierta cantidad de datos o de voltaje, los filtros pasivos y activos son los que maneja los condensadores, resistencias, inductores y los activos son amplificadores operacionales. El aislamiento de los amplificadores se divide en secciones de entrada y salida, airadas entre sí, también existe el aislamiento digital el cual es similar al de los amplificadores analógicos, estos son prográmales para transmitir datos en cualquier dirección.

Al momento de diseñar un sistema se tiene varias restricciones, uno de ellos es la funcionalidad donde los microprocesadores tienen muchas limitantes, el rendimiento de estos, la energía que se necesita y por último

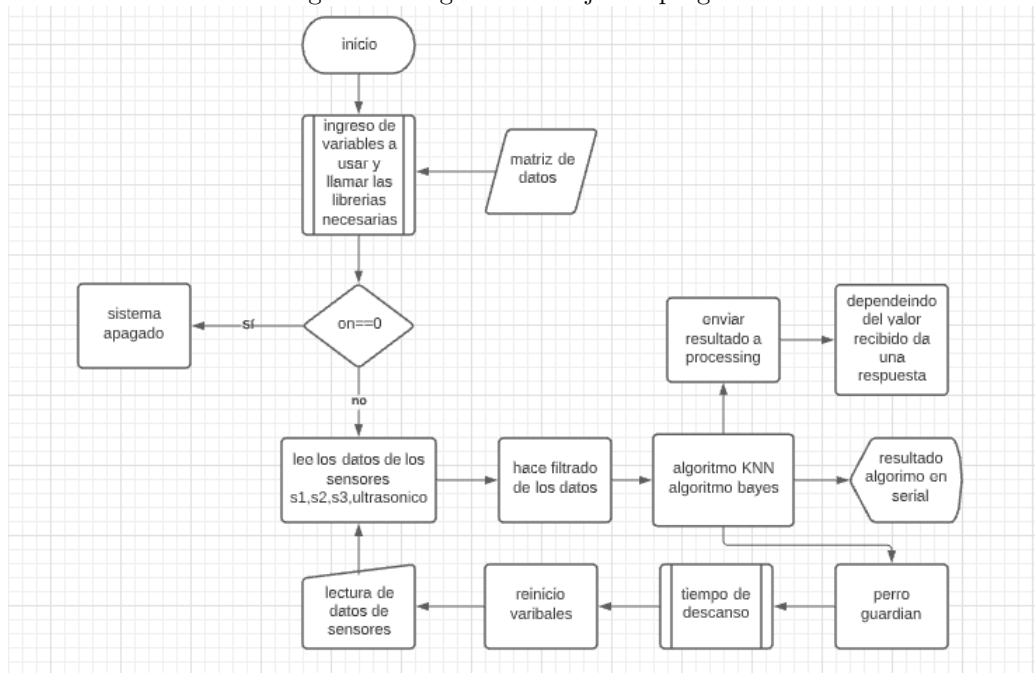
el costo ya que si es un diseño muy robusto se va a necesitar mayor inversión y por ende se aumenta el precio.

2. Diseño del Sistema

2.1. Diagrama de Flujo

El diagrama de flujo explica como esta desarrollado el codigo en arduino para la toma de decisiones y muestra de resultado:

Figura 1: Diagrama de flujo del programa



2.2. Diagrama de Bloques

En el diagrama de bloques se identifica que componentes están conectados al Arduino como lo son los puertos COM y la virtual terminal, los sensores:

Figura 2: Diagrama de bloques

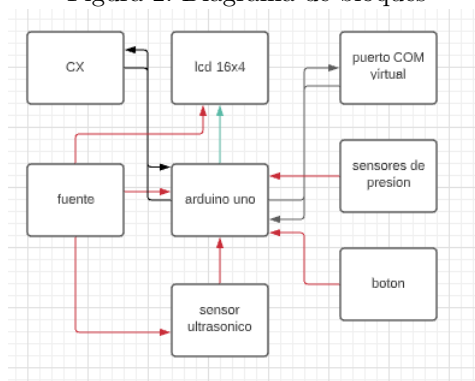
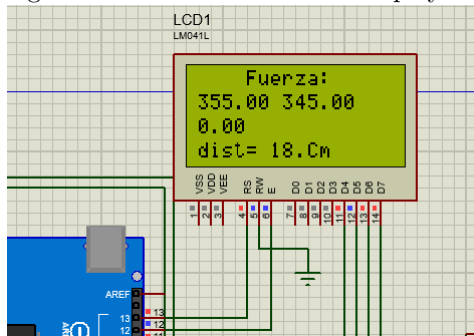
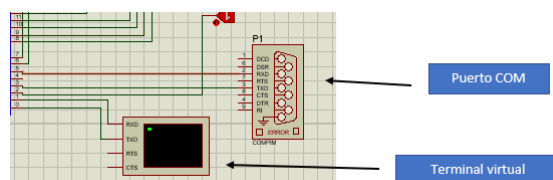


Figura 6: muestra de datos en display lcd



Conexión del puerto COM virtualizado con el software VSPE y la terminal virtual el cual uno muestra los datos de manera serial y el otro envía los datos a procesing y muestra el resultado.

Figura 7: concexion del puerto COM virtual y la terminal CX



3.1.2. Simulación Processing

Se realiza una simulación en Processing de tal manera que muestra en pantalla la posición en al que se encuentra sentado y le da una imagen de como esta sentado.

Figura 8: simulacion postura correcta postura 1 de 4

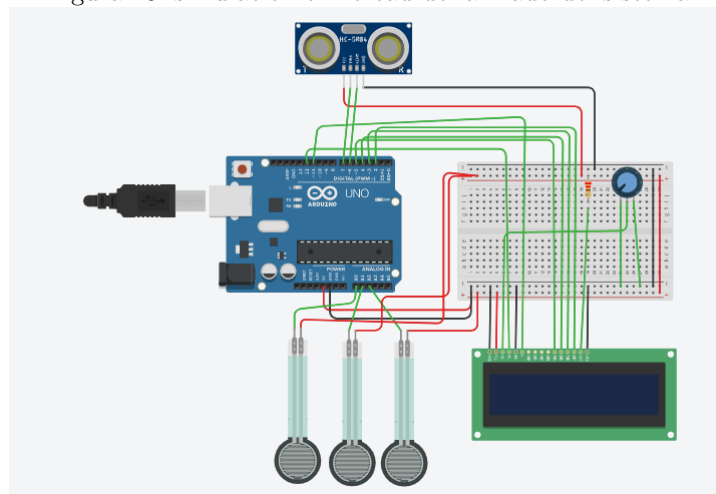


Figura 9: simulacion postura incorrecta postura 2 de 4



3.1.3. Simulación tinkercad

Figura 10: simulacion tinkercad del armado del sistema



4. Análisis de Resultados

4.1. programa completo

https://github.com/edwinsolano/SistemasEmbebidosProyecto1B/tree/main/SolanoEdwin_proyecto2B

4.2. codigo arduino

En el proyecto realizado se usan tres sensores de presión y un sensor ultrasónico todo simulado de tal manera que se pueda representar el sistema en funcionamiento, se analiza la presión de cada uno de los sensores y así se pronostica como esta sentada una persona en una silla de ruedas mediante una base de datos ya generada y usando los filtros para generar la matriz.

Se usa el filtro gaussiano para tener los datos de salida depurados y filtrados de tal manera que con esos datos se genera la matriz de entrenamiento, con esta matriz se aplica los algoritmos de knn y bayesiano y así poder predecir dependiendo de los datos de los sensores en que posición esta la persona si se encuentra en una

Además con el algoritmo de confusión se analiza los resultados de cada uno de los algoritmos KNN y bayesiano y así poder determinar la matriz con falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos con el fin de dar una respuesta correcta.

```

1  #include <SoftwareSerial.h> //libreria que permite la coneccion serial con otros p
2  #define rxCOM 5 //pin 5 para recibir datos
3  #define txCOM 3 //pin 6 para envio de datos
4  SoftwareSerial cxSerial = SoftwareSerial(rxCOM, txCOM); //conexion serial con otro
5
6  #define sensorpin A0// analog pin 0
7  #define sensorpin1 A1
8  #define sensorpin2 A2
9  #include <avr/wdt.h>
10 #include<MsTimer2.h>
11
12 #include <LiquidCrystal.h>
13 LiquidCrystal lcd(13,12,11,10,9,8);
14
15 //*****variables para ultrasonisco*****
16 long tiempo=0; //Donde se va a guardar el tiempo de duracion del pulso generado po
17 float distancia=0; //Donde se va a guardar la distancia calculada
18 int pinTrigger=7;
19 int pinEcho=6;
20 //*****variables para KNN*****
21 #include "datos.h"
22 //datos de prueba
23 // float datos_prueba [4]=[549.00,106.00,260.00,12.00];
24 int col;
25 int fil=0;
26 float potencia;
27 float raiz;
28 float etiqueta;
29 float dist_menor=2500; //agregar un valor fuera del rango
30 String tipo="";
31

```

Al momento de dar simulación en el proteus se tiene una interrupción o un botón de arranque del sistema que igual manera sirve para apagar el sistema y volver hacer el llamado de datos, en la terminal virtual cuando se inicia el sistema muestra los datos ingresados o leídos por Arduino de los sensores, hace el calculo por kNN y bayes y predicen la postura de la persona, después entra al perro guardián que no lee datos dentro de 20 segundos pasado los 20 segundos vuelve a leer los datos y manda la información anticipadamente reinicia las variables que almacenan los datos.

The screenshot shows a Jupyter Notebook with the following code and output:

```

# Seleccionamos los datos de entrenamiento
X_train = X[X['clase'] != 0]
y_train = y[X['clase'] != 0]

# Seleccionamos los datos de prueba
X_test = X[X['clase'] == 0]
y_test = y[X['clase'] == 0]

# Entrenamos el modelo
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Prediccion
prediccion = knn.predict(X_test)

# Mostramos la prediccion
print(prediccion)

```

The output of the notebook is:

```

array([1, 2, 3, 4])

```

The notebook also displays the following text:

Predicciones KNN y Bayes

Perro guardián

Datos que recoge de los sensores

Reinicio de variables

El resultado obtenido en este apartado es la interfaz grafica concetada simultaneamente con proteus y la virtual terminal el cual recibe los datos que esta contenido en la temrinal virtual de proteus y los pasa al codigo procesing para obtener un resultado.

Figura 13: interfaz grafica con virtual terminal



5. Conclusiones y Recomendaciones

5.1. Conclusiones

Al momento de generar la simulación se conecta por medio de un puerto COM virtual al Processing y dependiendo de los datos analizados con el algoritmo de bayes y KNN muestra una postura esta se define de acuerdo a la presión del sensor y la distancia del sensor ultrasónico.

Con el algoritmo de confusión es posible determinar cuantos verdadero positivos acertó el código o con respecto a los datos de los sensores identifica el resultado final y da un promedio de los aciertos y los aciertos falsos como también de los verdaderos falsos.

Es posible identificar en un algoritmo como trabaja el aprendizaje de maquina ya que en esta simulación al usar Arduino uno la base de datos o la matriz de datos se la tenía que limitar debido a la falta de memoria, caso contrario se tendría una mejor precisión al arrojar el resultado.

5.2. Recomendaciones

Si se tiene conectado la terminal virtual y un puerto COM con Arduino proteus y Processing, todo de manera virtual se debe buscar una manera la cual sea posible la comunicación con los diferentes softwares el cual se lo puede hacer con VSPE que es un creador de puertos virtuales.

Al momento de virtualizar todo hay que tener en cuenta que la respuesta no es la misma que teniendo un hardware físico, ya que la programación de estos esta basada a base de pruebas en algunos casos no funciona de igual manera un Arduino virtual que un Arduino físico.

Ya que se usa muchas variables se debe tener en cuenta el nombramiento de estas para no tener confusiones o conflictos con el programa posteriormente ya que algunas son de almacenamiento de datos.