LICENCIATURA EN INGENIERÍA EN ELECTRÓNICA

MODALIDAD: PROYECTO TECNOLÓGICO

TITULO DEL PROYECTO:

SISTEMA DE ADQUISICIÓN Y MONITOREO EN TIEMPO REAL DE LAS VARIABLES CINEMÁTICAS DE DOS CAJAS DE VELOCIDADES

PRESENTA:

RAMÍREZ AGUNDIZ JULIO CÉSAR

MATRÍCULA: 209301188

ASESOR:

ING. ROMY PÉREZ MORENO

TRIMESTRE LECTIVO:

15-P

MÉXICO D.F. ENERO 2016

Yo, ROMY PÉREZ MORENO, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma

Yo, RAMÍREZ AGUNDIZ JULIO CÉSAR, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma

Resumen

La digitalización de una señal analógica ha sido de mucha ayuda para un correcto y más completo análisis de la señal, hoy en día existen muchas técnicas de muestreo de una señal analógica para convertirla en una señal digital, en este proyecto utilizaremos el muestreo por medio de una tarjeta de adquisición de datos y procesando dichos datos obtenidos en el software de Matlab.

En la actualidad se cuenta con dispositivos que permiten medir la posición de algún objeto como son los potenciómetros de precisión, con ellos se puede calcular la cinemática de un objeto en movimiento.

El proyecto consiste en la digitalización de las señales analógicas recibidas por el potenciómetro de precisión y con ellas calcular las variables cinemáticas de posición, velocidad y aceleración dependiendo de lo que solicite el usuario con un propósito didáctico.

Palabras clave: Digitalización de señales, Arduino, Matlab, Muestreo en tiempo real, Tarjeta de adquisición de datos.

Abstract

Analog signal digitization has been very helpful for a complete signal analysis, today there are many sampling techniques to convert an analog signal to a digital signal, this project will use a data acquisition board and the data processing will be done in Matlab.

Nowadays we have devices that measure the position of things for example precision potentiometers, with them you can calculate the kinematics of a moving object.

The project will digitize analog signals received by the precision potentiometer and calculate the kinematic variables including position, velocity and acceleration according to the user requests with a didactic purpose.

Keywords: Digitization of analog signal, Arduino, Matlab, Real-time sampling, DAQ

CONTENIDO

FIGURAS

TABLAS

I. INTRODUCCIÓN

Los engranes son utilizados cuando se quiere transmitir movimiento angular entre dos ejes de forma sincronizada [1].

Los engranes se pueden combinar de formas muy diversas, a los arreglos de engranes se les conoce como tren de engranes, las formas más básicas de combinar los engranes son el tren simple, el tren compuesto y tren planetario.

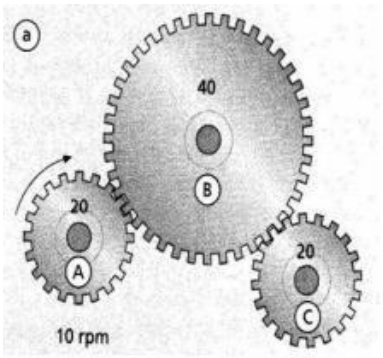


Fig. 1 Tren de engranes simple

En la figura 1 se muestra un tren simple, cada engrane está montado sobre un eje independiente y fijo al marco de referencia.

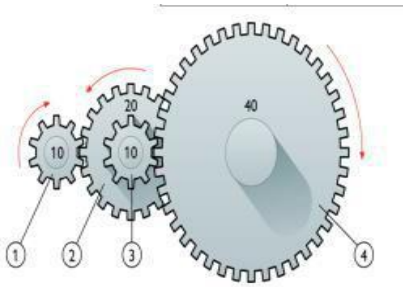


Fig. 2 Tren compuesto de engranes

Un tren compuesto se muestra en la figura 2, donde sus ejes están sin movimiento y cada uno contiene uno o dos engranes y necesariamente las velocidades angulares de los engranes que comparten cada eje es la misma.

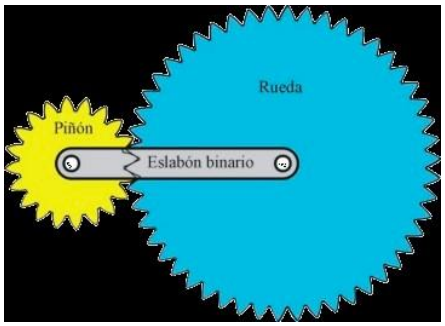


Fig. 3 Tren planetario

El movimiento de los engranes en un tren planetario es más complejo que en los dos anteriores, el nombre del tren planetario lo toma por la semejanza que tiene el movimiento del engrane piñon con el de un satélite (rueda) natural alrededor de un planeta, por lo tanto al engrane piñon se le conoce como satélite y al engrane con eje fijo como engrane sol.

En un tren de engranes planetario, la pieza que mantiene a un radio constante al engrane satélite del sol se llama brazo y puede ser una simple barra o un engrane con eje fijo que soporte al engrane satélite.

Dentro de los usos y aplicaciones que se le dan a los trenes de engranes, se encuentran las cajas de velocidades, donde dentro de estas se pueden encontrar combinaciones de un tren simple con un tren planetario y más.

Se puede analizar la cinemática de los sistemas que contienen algún tipo de tren de engranes o combinaciones de los mismos, para ello se cuentan con modelos matemáticos que dan solución a dichos sistemas.

Fig. 4 Potenciómetro de precisión

En la actualidad se cuenta con dispositivos que permiten medir la posición de algún objeto. En la figura 4 observamos los potenciómetros de precisión, con ellos se puede calcular la cinemática de un objeto en movimiento.

Para adquirir la señal de un sensor y digitalizarla se requiere de una tarjeta de adquisición de datos. La adquisición de datos o adquisición de señales, consiste en la toma de muestras del sistema analógico para generar datos que puedan ser manipulados por un sistema digital.

Existen muchas tarjetas de adquisición de datos, desde uso en aplicaciones muy sencillas hasta la adquisición de señales más complejas.

Fig. 5 Tarjeta de adquisición de datos Arduino

La tarjeta captadora de datos Arduino (figura 5), es una plataforma que esencialmente recibe los datos del sensor conectado a la entrada análoga, los digitaliza y los envía a la PC, para ser almacenados y luego procesados. El corazón de la tarjeta es un microcontrolador que toma los datos a sus entradas y los manipula o procesa según las instrucciones en el algoritmo que se ha diseñado para realizar la tarea que se desea.

Para procesar las señales adquiridas por una tarjeta de adquisición se utilizan distintos tipos de software entre los cuales destacan Matlab y LabVIEW.

Matlab (Matrix Laboratory) es un lenguaje de programación técnico-científico que básicamente trabaja con variables vectoriales y matriciales. Es muy útil debido a que contiene procesamiento de señales que se puede acoplar con Arduino para el procesamiento de señales analógicas.

Dispositivos como los sensores generan un ruido eléctrico que al digitalizar la señal recuperada por el sensor y procesada por la tarjeta de adquisición se ve afectada por el ruido generado y aparece una componente de alta frecuencia, a pesar de que el sistema se encuentre sin movimiento.

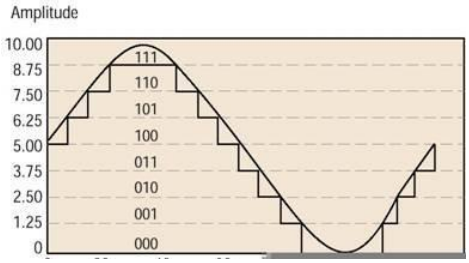


Fig. 6 Digitalización de una señal

Matlab permite crear toda una interfaz de análisis para señales adquiridas la cual puede contener desde filtrado de la señal hasta combinaciones de señales muestreadas y hacer el muestreo de las señales en tiempo real.

II. JUSTIFICACIÓN

El interés de diseñar e implementar los algoritmos y el hardware necesario para el correcto análisis de los mecanismos con trenes de engranes de tipo planetario surge como la propuesta de solución a la necesidad en las UEA´s de Mecanismos (1133060) y Laboratorio de mecanismos (1133009), de comprobar experimentalmente los valores de los modelos matemáticos que se revisan en clase.

La posibilidad de contar con dispositivos electrónicos para la digitalización de las posiciones, velocidades y aceleraciones de los engranes brindará una mejor exactitud en comparación con una medición analógica.

III. ANTECEDENTES

Los prototipos didácticos de cajas de velocidades planetarias, fueron diseñados en la UAM Azcapotzalco como parte de un proyecto de integración en el área de Ingeniería Mecánica, con el propósito de tener algunos de los mecanismos físicos de los modelos matemáticos que se analizan en la UEA de Mecanismos y el Laboratorio de mecanismos.

Juan Carlos Olayo Garay en su Proyecto de Integración Diseño, manufactura y construcción de un prototipo didáctico de una caja de velocidades planetaria [2] diseñó y construyó una caja de velocidades planetaria. Esta caja de velocidades está formada por un par de trenes de engranes: uno simple y otro planetario. En total existen 4 engranes en movimiento a los cuales se les instaló un potenciómetro de precisión de 10 kΩ.

Gerardo Montesinos Cruz en su Proyecto de Integración Diseño y construcción de un tren planetario con brazo dentado [3] diseñó y construyó un torno sumador mecánico empleando un tren planetario y un tren simple. Este mecanismo cuenta en total con 4 engranes en movimiento a los cuales también se les instaló un potenciómetro de precisión de 10 kΩ.

A diferencia del proyecto de Olayo Garay, el de Gerardo Montesinos tiene dos grados de libertad, es decir, tiene dos entradas de movimiento.

En la UEA de Procesamiento Digital de Señales, estuvimos trabajando con una interfaz gráfica que se realizó en Matlab donde analizamos los espectros y realizamos el filtrado de señales analógicas adquiridas por una tarjeta de adquisición de datos.

IV. OBJETIVOS

I. Objetivo General

• Diseñar e implementar un sistema de adquisición monitoreo y registro de la cinemática de dos cajas de velocidades didácticas.

II. Objetivos específicos

- Implementar el código en Arduino para la adquisición de las señales.
- Diseñar e implementar el filtro pasa bajas en Matlab.
- Desarrollar la interfaz gráfica en Matlab para la visualización en tiempo real.

V. DESARROLLO DEL PROYECTO

a) Principios básicos de la digitalización de señales

Muchas variables físicas son de naturaleza analógica y pueden tomar cualquier valor dentro de un rango continuo de éstos. Como ejemplos de variables de este tipo se incluyen la temperatura, presión, intensidad luminosa, señales de audio, velocidad rotacional y velocidad de flujo entre otras.

Por ejemplo, la salida de voltaje de un amplificador de audio hacia los altavoces. Este voltaje es una cantidad analógica porque cada uno de sus posibles valores produce una respuesta diferente en el altavoz, y por lo tanto su valor exacto si es significativo.

Una cantidad digital tiene un valor que se especifica por una de las posibilidades como un 0 o 1, ALTO o BAJO, falso o verdadero, y así sucesivamente. En la práctica una cantidad digital, como un voltaje, podría tener un valor dentro de cualquiera de los rangos especificados.

Por ejemplo, para la lógica TTL, se sabe que:

De 0.0 V	a	0.8 V equivale a un 0 Lógico
De 2.0 V	a	5.0 V equivale a un1 Lógico

Ahora los valores exactos de los voltajes no son significativos, ya que los circuitos digitales responden de la misma manera para todos los voltajes que se encuentran dentro de un rango dado.

Por otra parte, los circuitos analógicos procesan las variables físicas no muy rápidamente, ya que manejan 10 posibles valores y los circuitos digitales procesan variables físicas o señales más rápidamente ya que tienen únicamente dos posibles valores, 0 y 1.

Es por ello que se crearon los convertidores analógico-digitales, para poder aumentar la velocidad del procesamiento de las señales. Este proceso de conversión se lleva a cabo mediante un muestreo de la señal.

Para este caso en particular convertimos el valor resistivo del potenciómetro que se encuentra unido a cada engrane significativo de las cajas de velocidades, en una variación de voltaje, esta conversión se llevara a cabo tomando el valor resistivo de 0 a 1024 que registra la tarjeta Arduino ilustrado en la siguiente tabla.

Resistencia Ω	Valor de Arduino
0	0
100	10
500	50
1k	100
5k	500
9k	900
10k	1024

Tabla 1 Caracterización del potenciómetro con Arduino

La siguiente tabla (Tabla 2) ilustra la caracterización de grados de giro vs resistencia.

Grados °	Resistencia Ω
0	0
10	200
20	504
30	786
38	1k
40	1.08k
50	1.34k

Tabla 2 Caracterización del potenciómetro grados vs resistividad

En la tabla 3 observamos el comportamiento del potenciómetro caracterizado por los grados de giro vs la variación de voltaje.

Grados	Variación de voltaje
0°	0
1°	0.01464
5°	0.0732
10°	0.1611
90°	1.3623

Tabla 3 Caracterización del potenciómetro grados vs voltaje

La variación de voltaje en comparación con los grados de giros fue un parámetro que calculamos con la siguiente formula en el software Matlab

v1(muestras)=(valor(n)*5/1024);

Fórmula 1. Conversión a voltaje del valor entregado por Arduino a Matlab

Donde v1(muestras) representa el valor del potenciómetro 1 convertido a voltaje dependiendo de la muestra que se toma dentro de un ciclo

Valor (n) representa el valor de 0 a 1024 que entrega Arduino y es dividido entre 1024 para tener una escala de 0 a 1 y multiplicado por 5 para obtener un valor de voltaje de 0V a 5V.

b) Adquisición de la señal mediante Arduino

Nos vamos al software de Arduino y desarrollamos un algoritmo capaz de adquirir y registrar las señales mandadas por el potenciómetro a la entrada analógica A0 hasta A3 de

la tarjeta Arduino UNO, mediante la comunicación serial y asignando un retardo definido por el usuario al final del codigo.

c) Algoritmo para leer puerto serie en Arduino

La instrucción serial.begin en Arduino sirve para enviar los datos registrados por el puerto serie y puedan ser trabajarlos por algún otro software de procesamiento en este caso en particular ocuparemos Matlab.

d) Registro de la señal con Matlab

En Matlab se desarrollo un algoritmo capaz de leer el puerto serie donde Arduino manda la señal registrada y Matlab la procesa de tal forma que obtenemos aquí la relación de voltaje vs grados.

De esta manera ya tenemos los datos que necesitamos en tiempo real con un comunicación directa desde el potenciómetro hasta Matlab con la tarjeta de adquisición de datos Arduino.

Con los datos obtenidos realizamos el cálculo de la posición de cada engrane, es decir ocuparemos la tabla 3 para hacer la conversión de voltaje a magnitud de grados mediante la siguiente formula

$$pn(muestras)=vn(muestras)/0.01464;$$

Fórmula 2. Conversión de voltaje a grados

pn representa la posición del engrane n almacenándose en un vector.

El siguiente cálculo es la velocidad; cada muestra del vector de posición se resta con su anterior valor y está diferencia se divide entre el tiempo de muestreo y así obtenemos el vector velocidad.

$$veln(muestras)=(abs\ (deltapn(muestras)))/(im);$$

Fórmula 3. Formula de la velocidad en °/s

Este parámetro puede ser calculado con 2 unidades ya sea grados/segundo ó rpm para esta segunda la fórmula es la siguiente.

velnrpm(muestras)=veln(muestras)*0.166667;

Fórmula 4. Formula de la velocidad en rpm

Lo mismo para el vector aceleración, cada muestra del vector velocidad se resta con su valor anterior y dividimos esta diferencia entre el tiempo de muestreo y así obtenemos el vector aceleración.

acln(muestras)=(abs(deltavn))/(im);

Fórmula 5. Formula de la aceleración

Cada una de estas variables es almacenada dentro de un vector el cual a petición del usuario imprime el comportamiento en una gráfica además de haber mostrado el comportamiento en tiempo real de cada engrane.

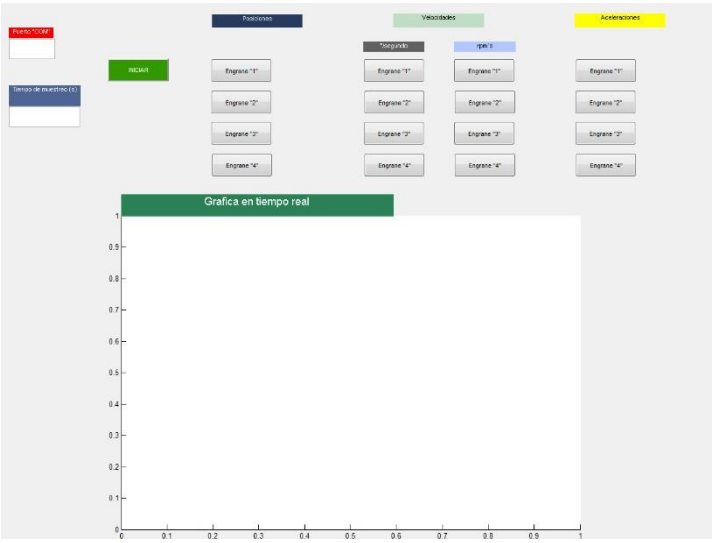


Fig. 7 Interfaz Gráfica para el usuario

Para probar que los algoritmos funcionan de manera correcta utilizamos un par de potenciómetros conectados a la tarjeta de Arduino y se obtuvieron los siguientes resultados.

e) Digitalización del potenciómetro fijo en 250°

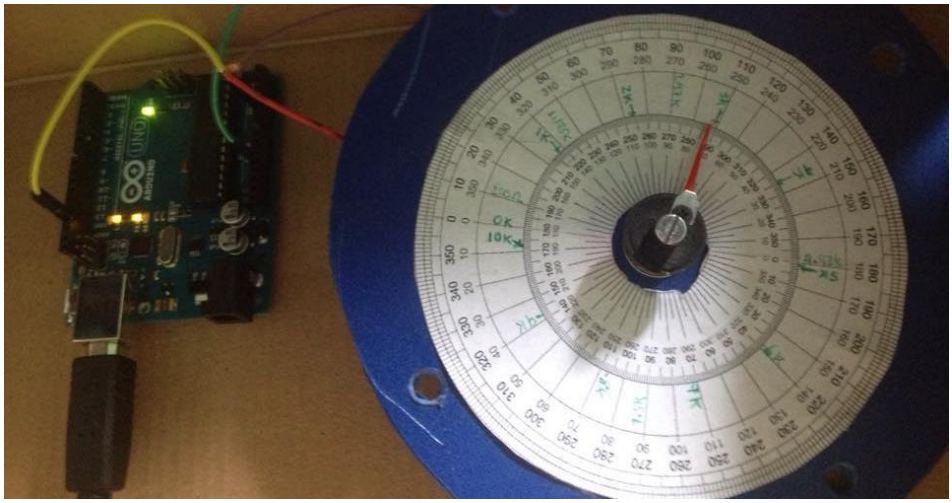


Fig. 8 Potenciómetro fijo en 250°

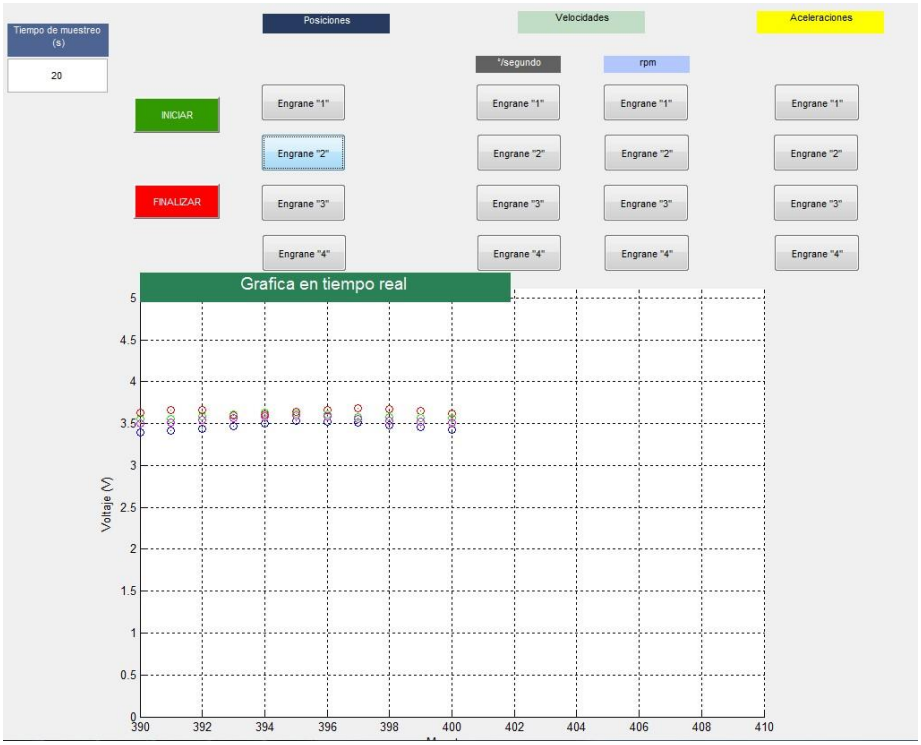


Fig. 9 Interfaz Gráfica con muestreo en tiempo real

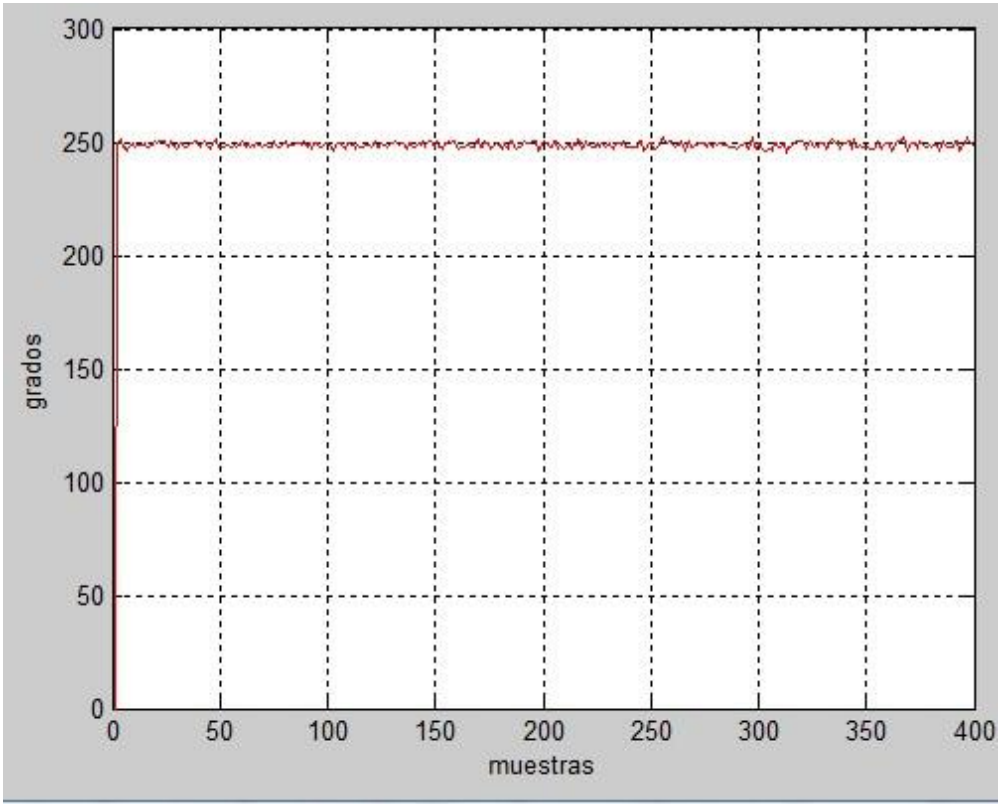


Fig. 10 Grafica de posición del engrane

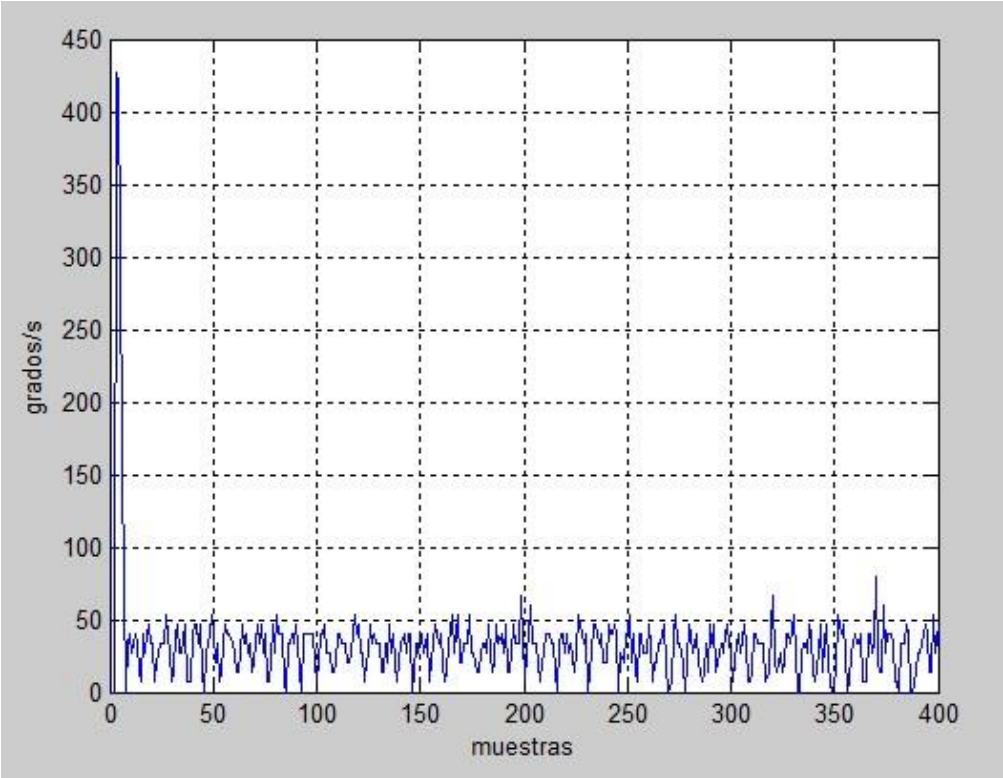


Fig. 11 Grafica de la velocidad en grados/s

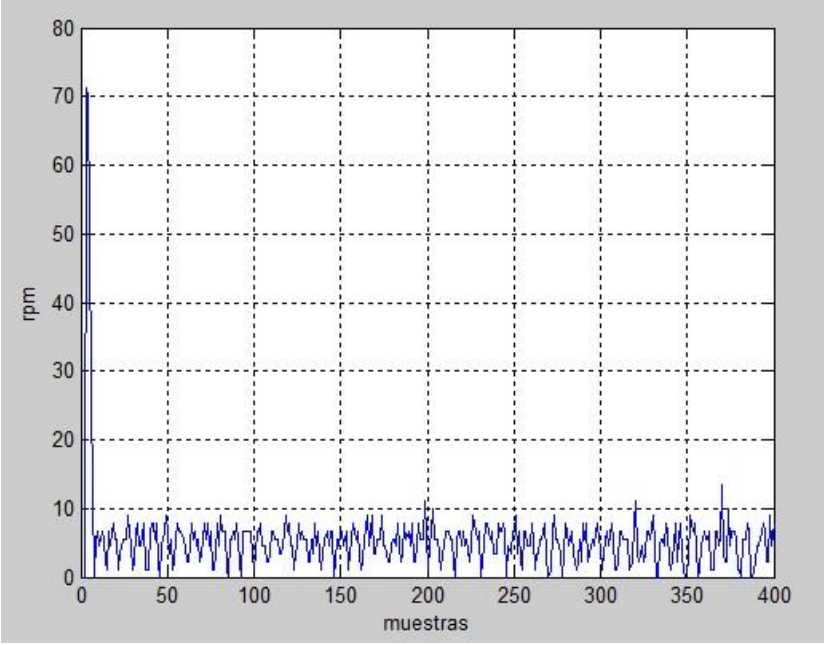


Fig. 12 Grafica de la velocidad en rpm

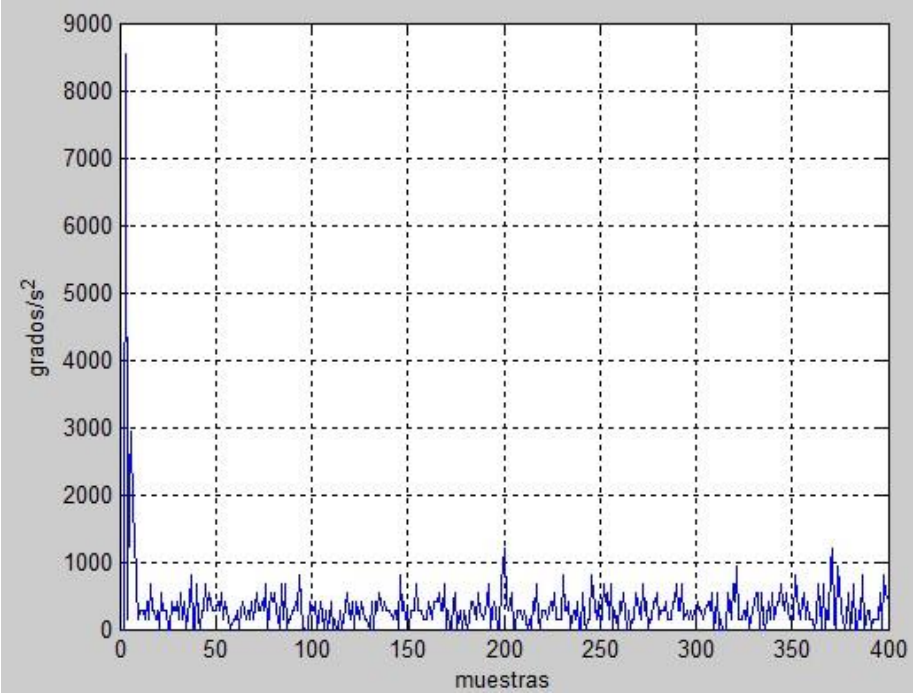


Fig. 13 Grafica de la aceleración

f) Análisis de resultados

En la figura 8 observamos que el potenciómetro está colocado fijo en 250° para lo que esperamos que no haya variación en las gráficas de posición, velocidad y aceleración.

En la figura 9 se observa el muestreo en tiempo real con la opción a elegir la variable de la que queremos ver su comportamiento.

En seguida observamos en la figura 10 que la posición esta fija en 250° con una oscilación muy pequeña que se puede considerar despreciable.

La figura 11 y 12 nos muestra el comportamiento de la velocidad en grados/s y rpm respectivamente en donde podemos observar que de igual forma se muestra de manera constante con una pequeña oscilación muy cercana a cero

Y finalmente observamos que la figura 13 que corresponde a la aceleración es de igual forma cercana a cero con una oscilación que puede ser despreciable.

Esto mismo se hace para cada entrada analógica de la tarjeta Arduino para comprobar su funcionamiento.

g) Instalación en los potenciómetros de las cajas didácticas

Una vez hecho lo anterior, el siguiente paso fue instalar este sistema en las cajas didácticas, para esto se soldaron cables en las 3 terminales de cada potenciómetro con un largo considerable para que no obstruya el funcionamiento de las cajas y colocar el nombre de cada potenciómetro de acuerdo al color que proporciona la interfaz gráfica y el lugar donde se conectan en la tarjeta Arduino relacionados de la siguiente manera.

Numero de engrane	Entrada de Arduino	Color asignado
1	A0	AZUL
2	A1	ROJO
3	A2	VERDE
4	A3	MAGENTA

Tabla 4 Correspondencia de cada engran

h) Instalación a la caja didáctica #1

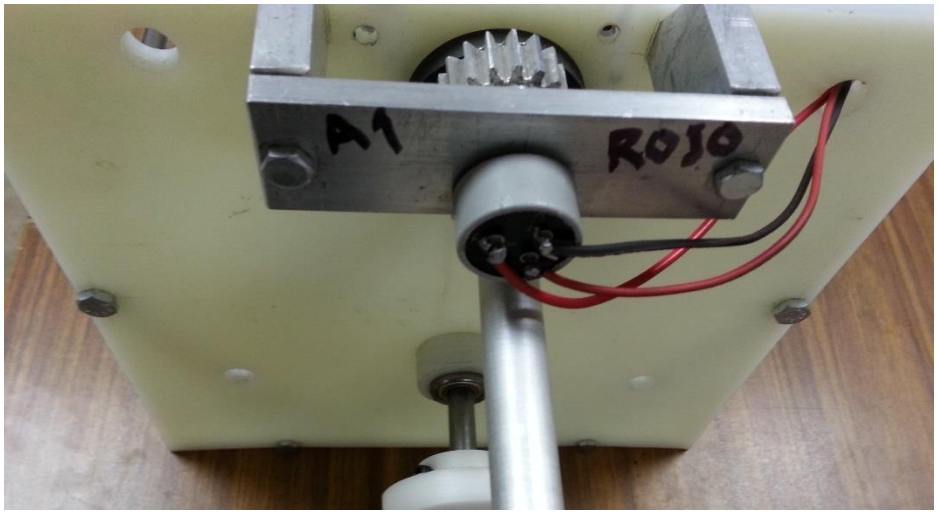


Fig. 14 Potenciómetro asignado a la entrada A1 color rojo

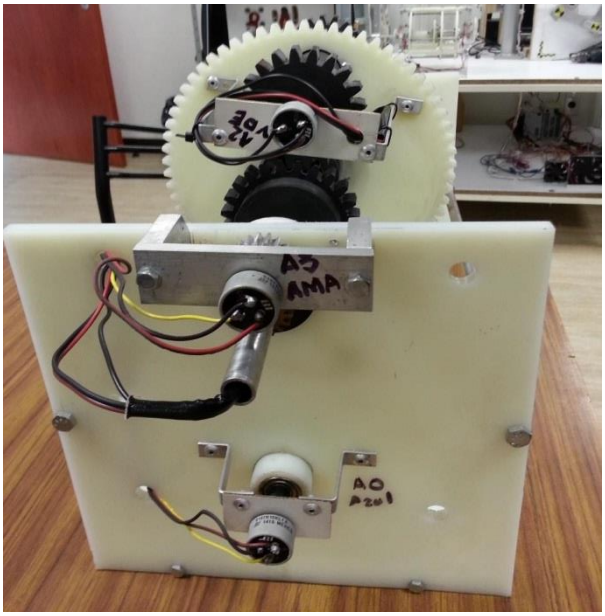


Fig. 15 Potenciómetros restantes asignados de acuerdo a la tabla 4

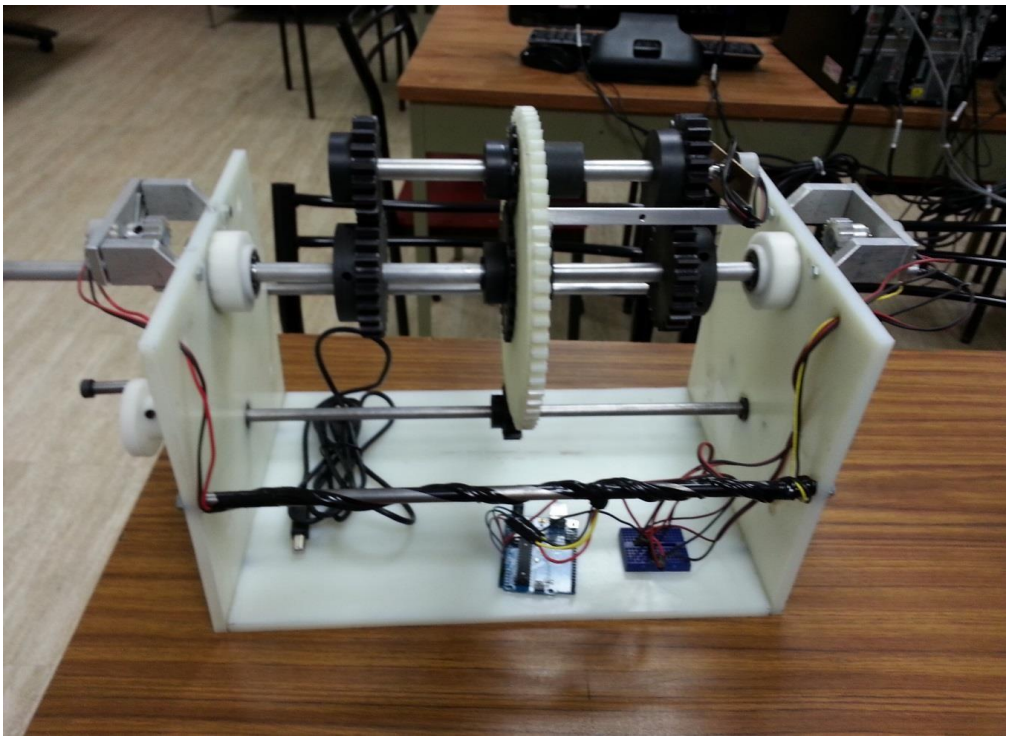


Fig. 16 Caja didáctica #1 terminada

i) Instalación a la caja didáctica #2

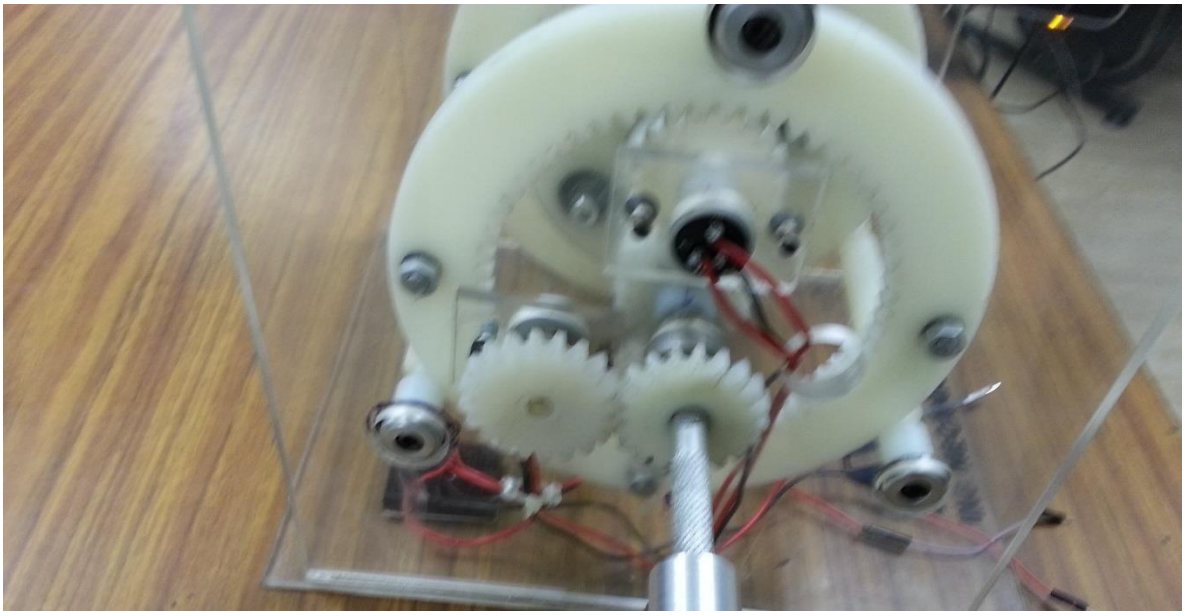


Fig. 17 Instalación a la caja #2

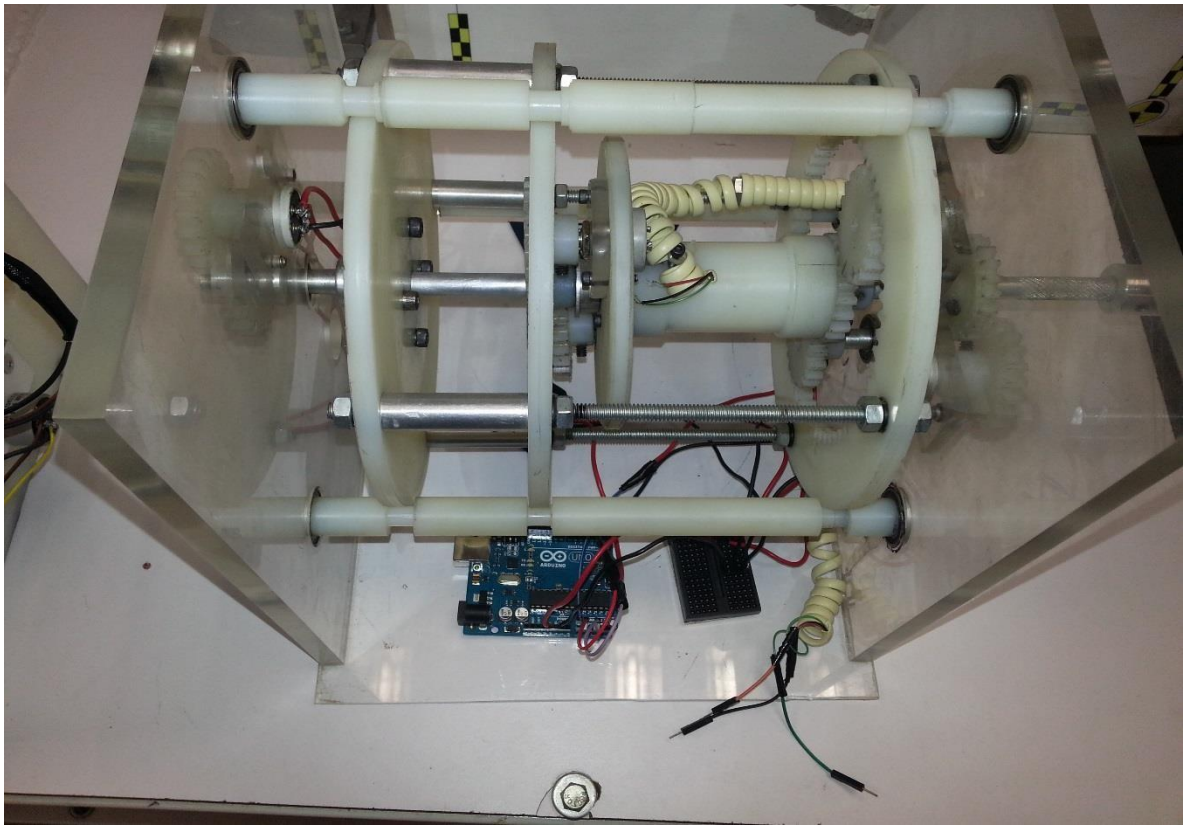


Fig. 18 Colocación de cable telefónico en caja #2

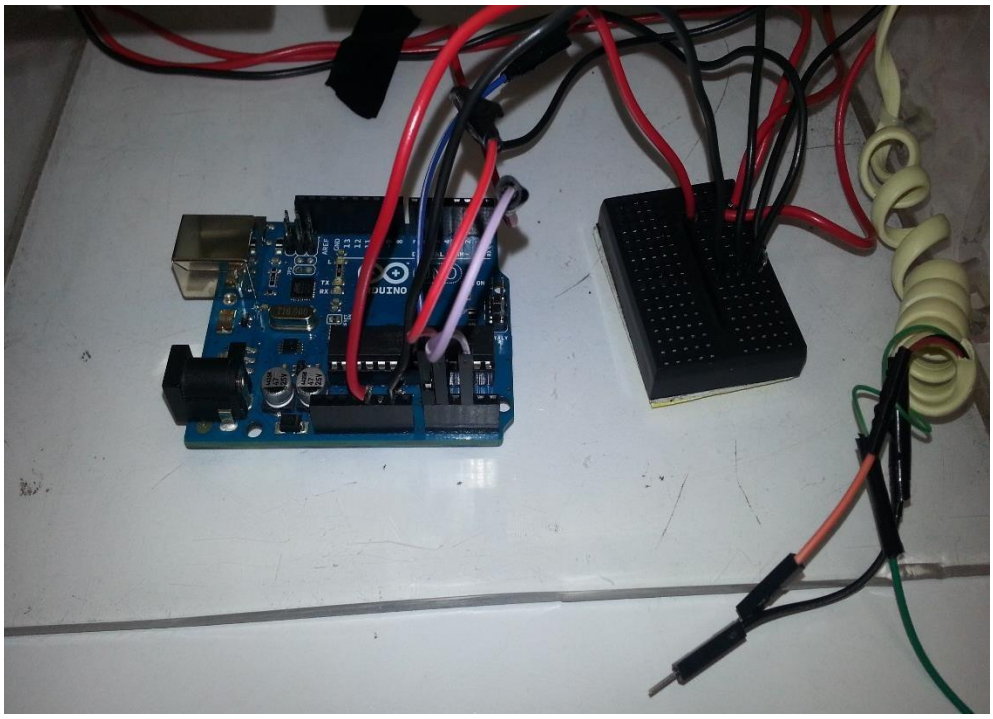


Fig. 19 Mini protoboard instalada en la caja 2

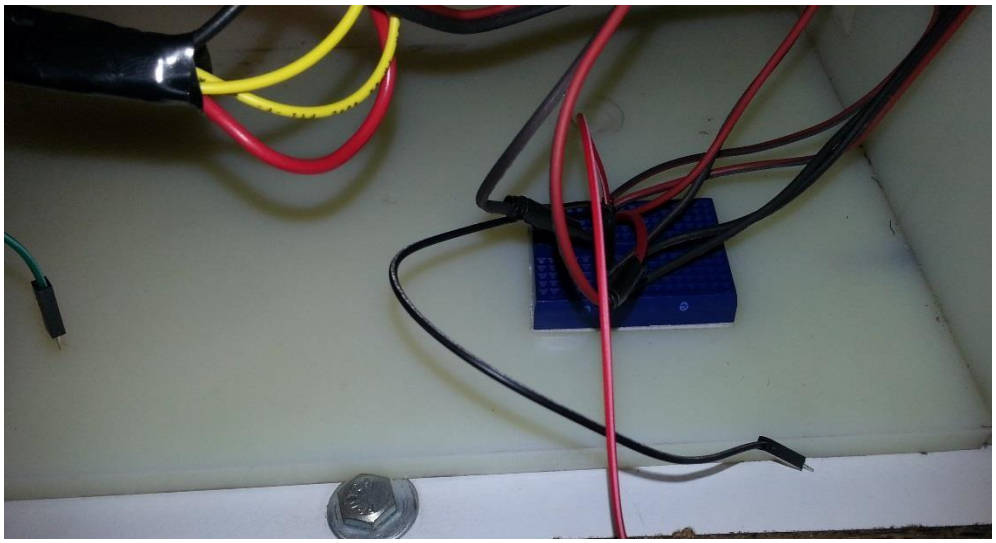


Fig. 20 Mini protoboard instalada en la caja 1

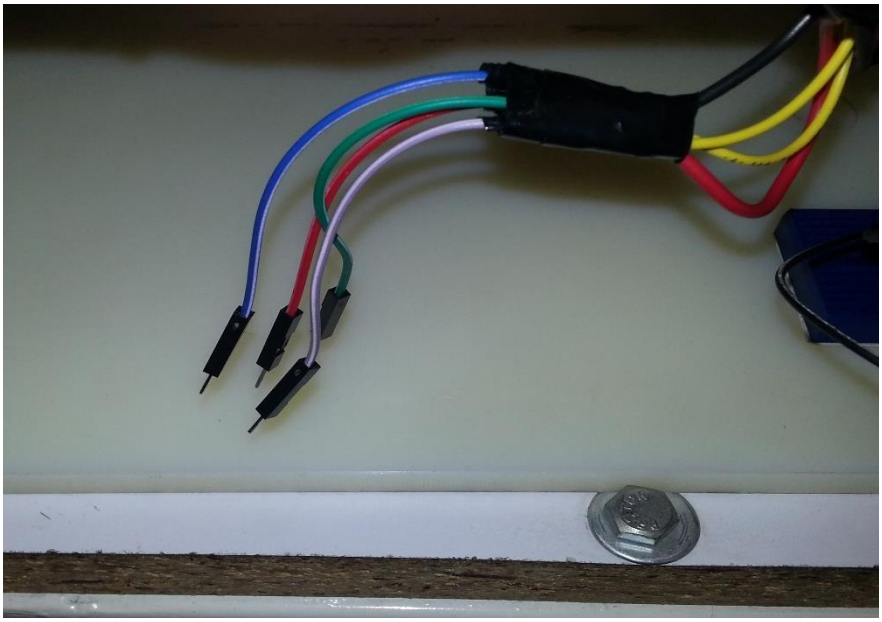


Fig. 21 Asignación de color a cada cable proveniente de los potenciómetros

Como se observa en la figura 18 se colocó un cable telefónico en las terminales del potenciómetro de la caja 2 ya que permite mayor flexibilidad al movimiento de los engranes.

En las figuras 19 y 20 se observa que se colocó una protoboard mini en cada caja para asignar un canal de voltaje y otro de tierra para la alimentación de los potenciómetros tomando esta alimentación directo de la tarjeta Arduino.

VI. DIAGRAMAS DE FLUJO

Diagrama de flujo Arduino

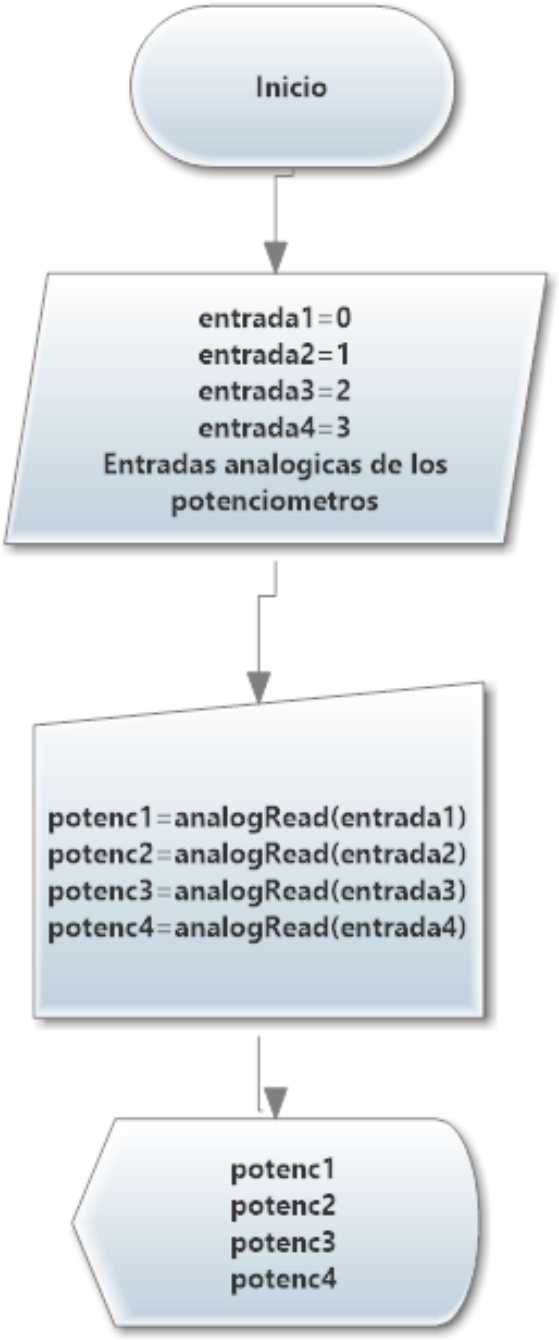
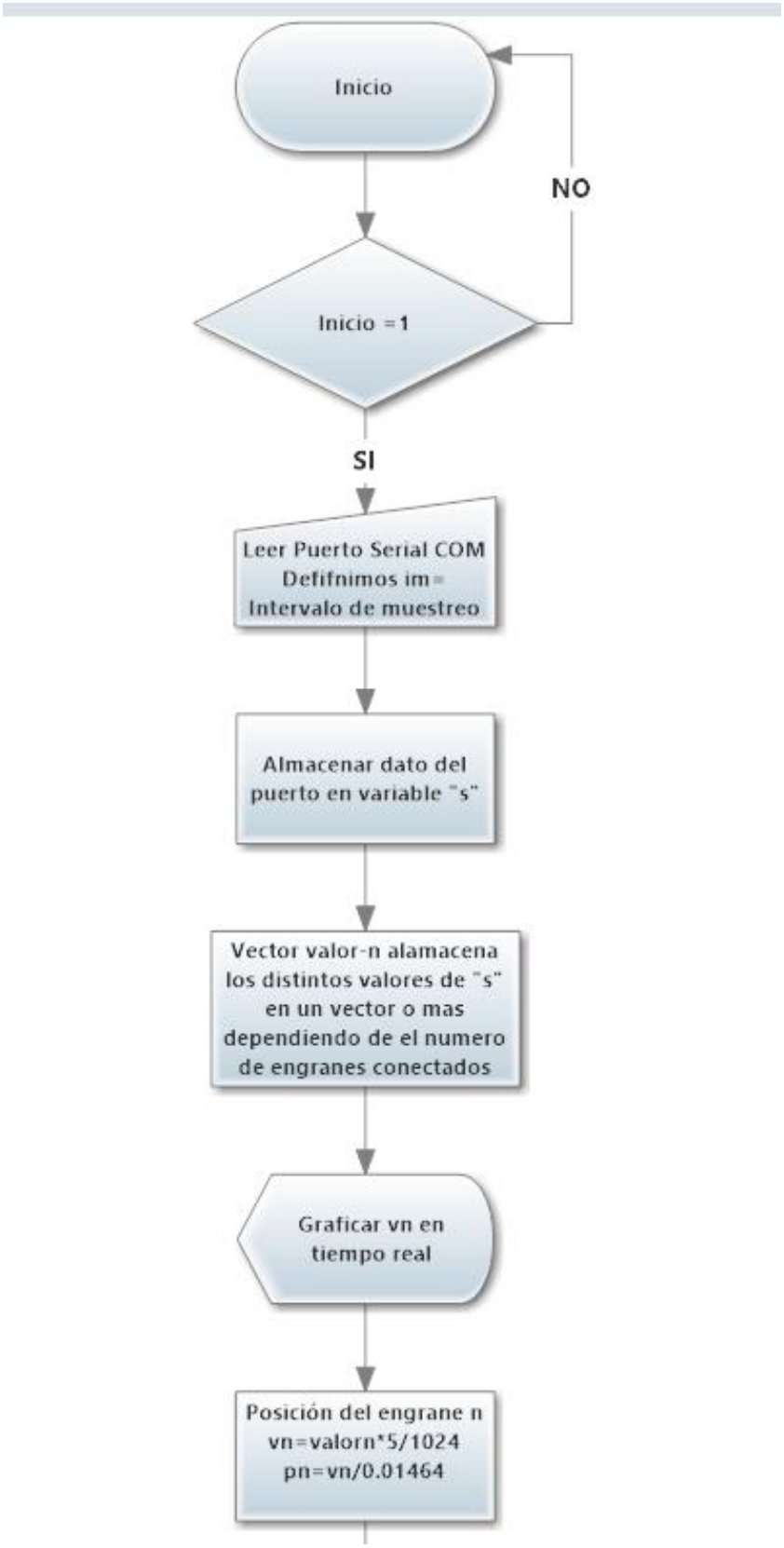
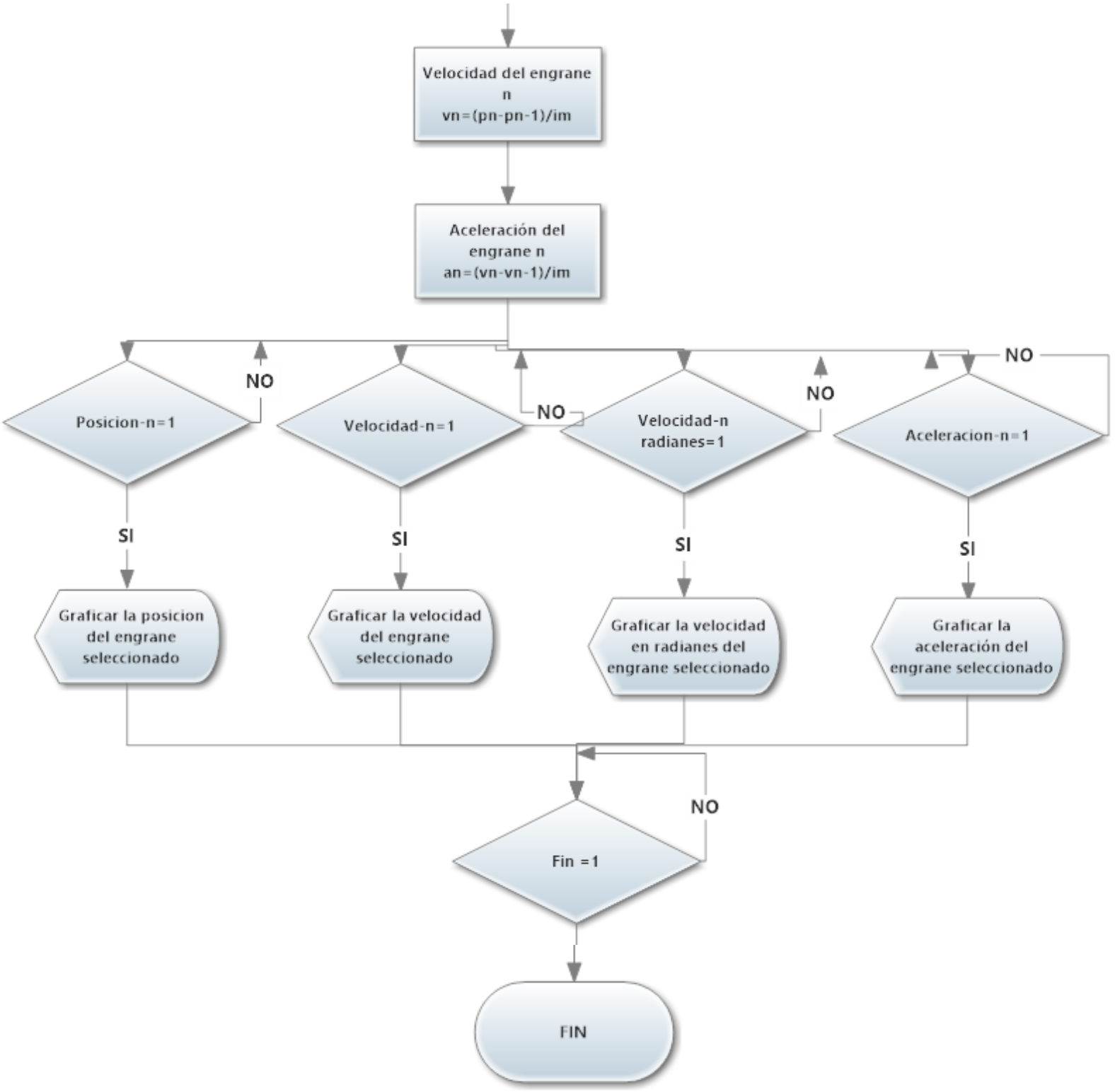


Diagrama de flujo Matlab




Velocidad del engrane n
vn=(pn-pn-1)/im
Aceleración del engrane n
an=(vn-vn-1)/im
Posicion-n=1
NO
SI
Graficar la posicion del engrane seleccionado
Velocidad-n=1
NO
SI
Graficar la velocidad del engrane seleccionado
Velocidad-n radianes=1
NO
SI
Graficar la velocidad en radianes del engrane seleccionado
Aceleracion-n=1
NO
SI
Graficar la aceleración del engrane seleccionado
Fin =1
NO
FIN

VII. MANUAL DE USUARIO

A. CONECTAR HARDWARE

1. Tomamos cualquiera de las cajas de didácticas de engranes y observamos que hay 6 terminales que están sin conectar, conectamos los cables que se ven en la figura 23 a las entradas analógicas de Arduino que se ven en la figura 22 en el orden que se muestra en la figura 24.

   Los cables que salen de la mini protoboard (rojo y negro) los conectamos a las salidas de Arduino que dicen 5V (ROJO) y GND (NEGRO), de tal forma que se vea como la figura 24.

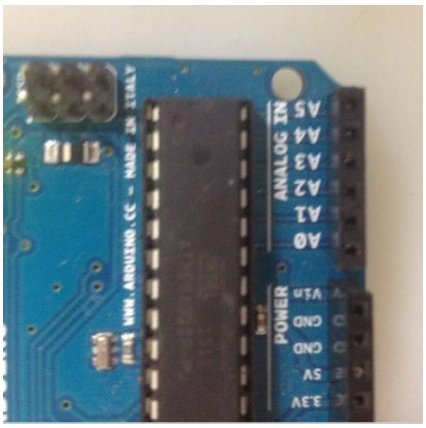


Fig. 22

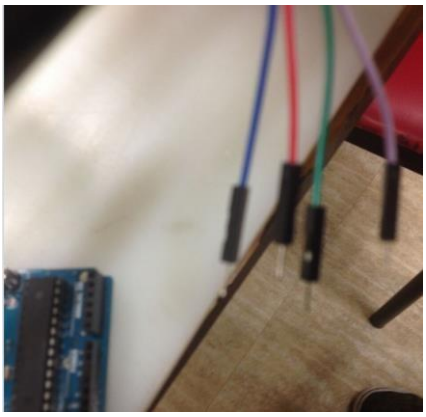


Fig. 23

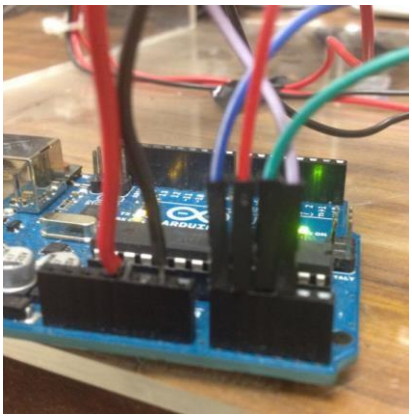


Fig. 24

2. Conectamos la placa Arduino a la pc con el cable USB.

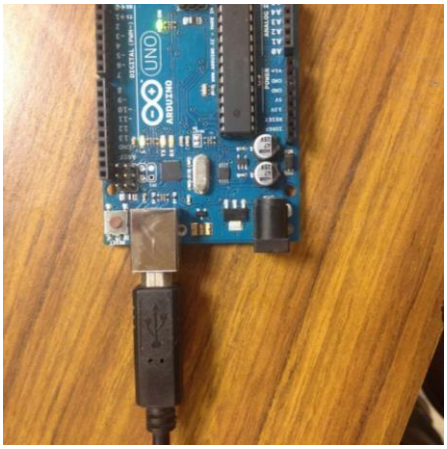


Fig.25

Fig. 26

B. CARGAR EL SOFTWARE EN LA TARJETA ARDUINO UNO

3. Vamos a inicio y buscamos el icono de Arduino y esperamos a que abra.

Fig. 27

Fig. 28

4. Una vez en el sketch de Arduino hacemos clic en "ARCHIVO" en la parte superior izquierda de la ventana y le damos clic en abrir, luego nos dirigimos a la carpeta contenedora del archivo Com_serial.ino (Nota: Es importante que el archivo este dentro de una carpeta de igual nombre que el archivo, de lo contrario no se compilara correctamente).

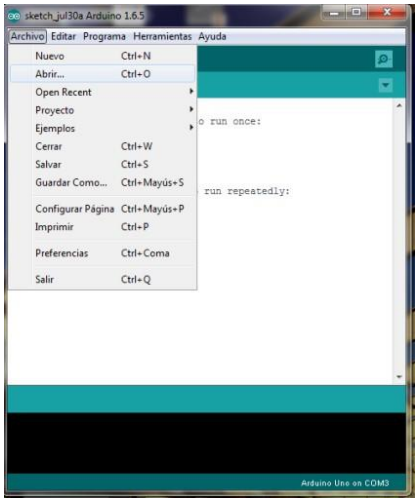


Fig. 29

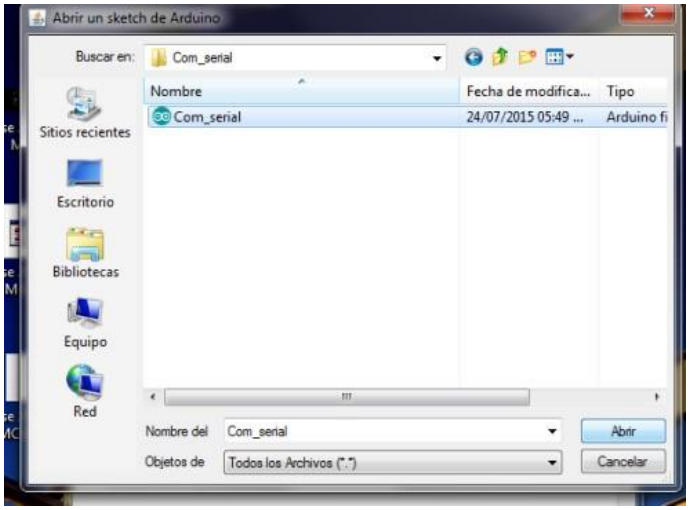


Fig. 30

5. Nos aparecerá un código como el que se muestra en la imagen siguiente, para continuar damos clic en la opción de "HERRAMIENTAS", damos clic en la opción de "PLACA" y escogemos la placa de Arduino que estamos utilizando, en este caso es Arduino UNO. De igual forma en la opción de "HERRAMIENTAS" damos clic en "PUERTO" y escogemos el puerto donde conectamos el Arduino (Nota: Si nos aparece más de un puerto COM disponible verificamos en administrador de dispositivos el puerto al que está conectado el Arduino para asignarlo de manera correcta como se muestra en la Figura 33 y 34).

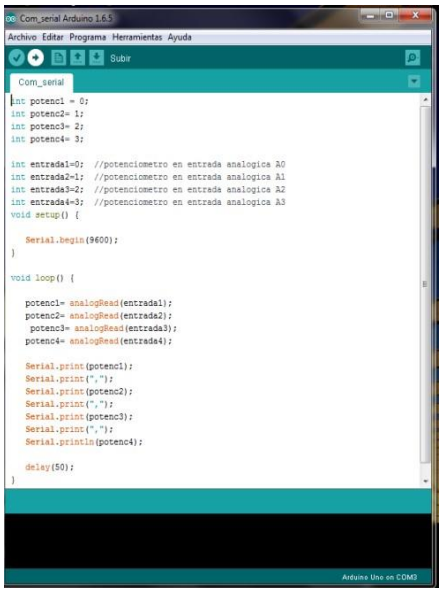


Fig. 31

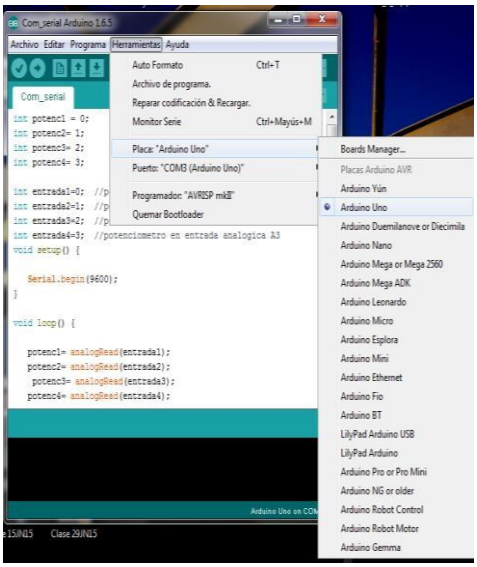


Fig. 32

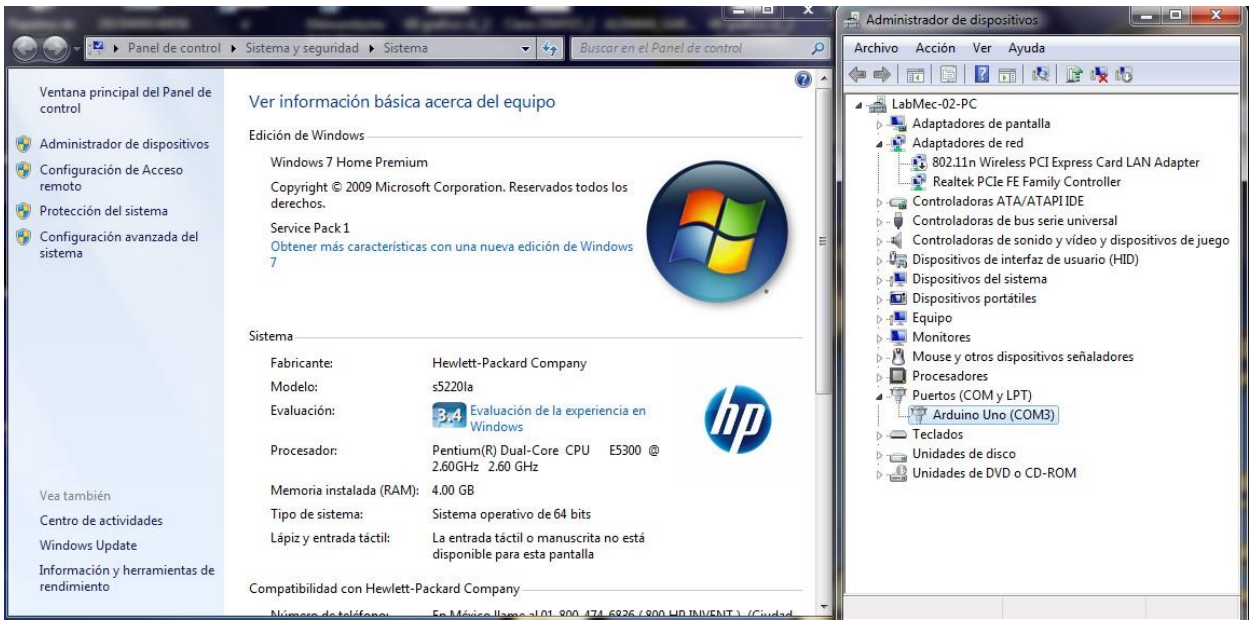


Fig. 33

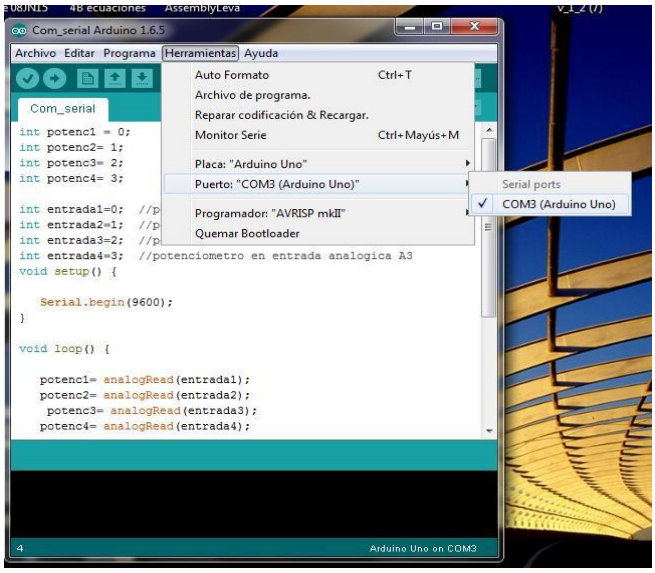


Fig. 34

6. Una vez asignado el puerto y la placa vamos a subir el programa a Arduino, para lo cual damos clic en la flecha que se encuentra debajo de la pestaña de archivo y nos tendrá que aparecer el texto “SUBIDO” en la parte inferior de la ventana como se muestra en la figura 36.

Hasta este momento tenemos cargado el programa en la placa Arduino (Nota: Si no apareció la leyenda “SUBIDO” y/o apareció cualquier otro texto necesitas cerrar todas las ventanas y volver a repetir todos los pasos anteriores).

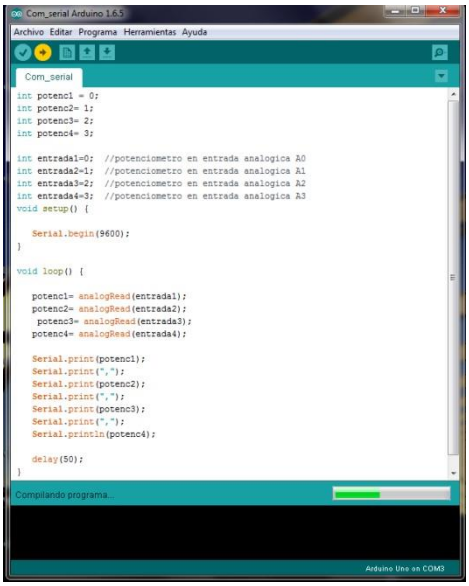


Fig. 35

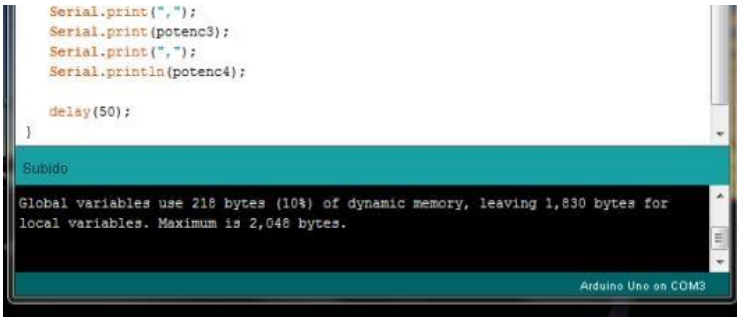


Fig. 36

C. ABRIR LA INTERFAZ GRÁFICA EN MATLAB

7. El siguiente paso es abrir el código en Matlab con el nombre de Digitalizacion_matlab.m, por lo que primero necesitamos abrimos el programa Matlab que se encuentra en INICIO/TODOS LOS PROGRAMAS y ahí buscamos el icono de Matlab.

   En la ventana de Matlab vamos a la opción de tres puntos "…" y nos pedirá indicar una carpeta, ahí buscamos la capeta "Matlab_com_serial" y la seleccionamos.

   Se deberá aparecer el contenido de la carpeta en la parte izquierda de la ventana de Matlab y ahí damos doble clic sobre "Digitalizacion_matlab.m" y esperamos a que abra.

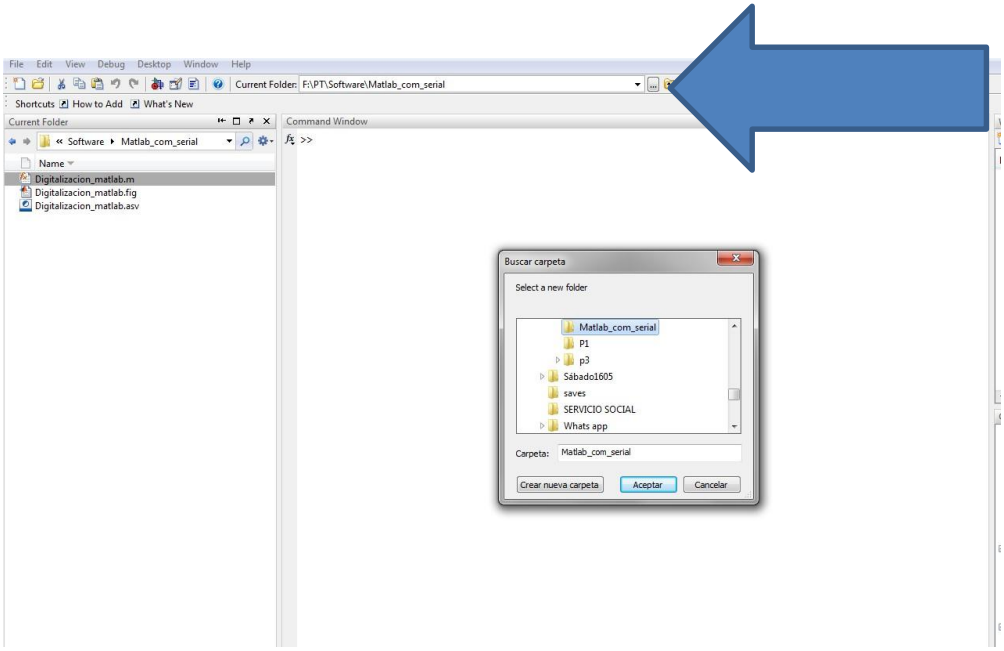


Fig. 37

8. Se abrirá la siguiente venta, ahí buscamos la parte de asignación de puerto serial que está en un recuadro de símbolos de porcentajes (%) alrededor de la línea 155 a la 170 y ahí nuevamente asignamos el COM que seleccionamos en Arduino en las dos líneas que lo piden.

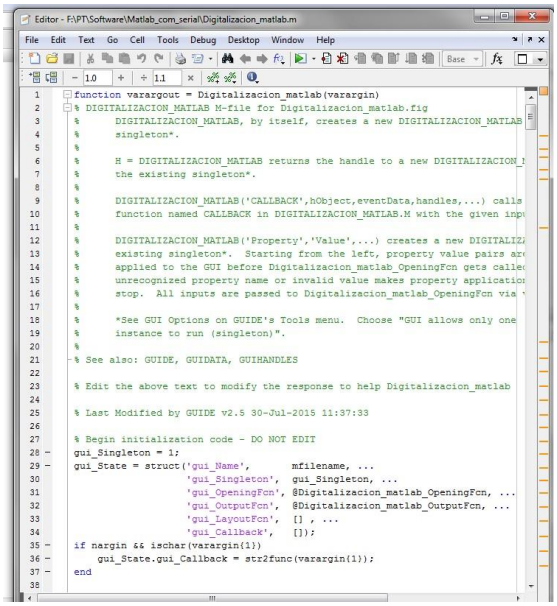


Fig. 38

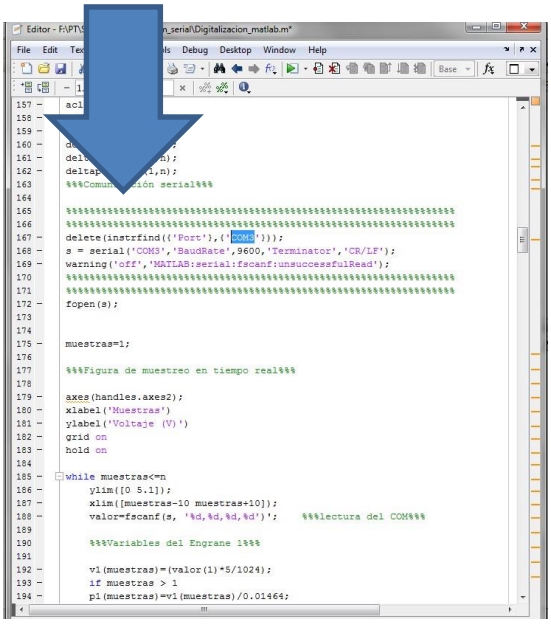


Fig. 39

9. Ahí mismo buscamos la asignación de tiempo de muestreo alrededor de la línea 90 y ponemos el mismo valor que en Arduino en este caso son 0.05s.

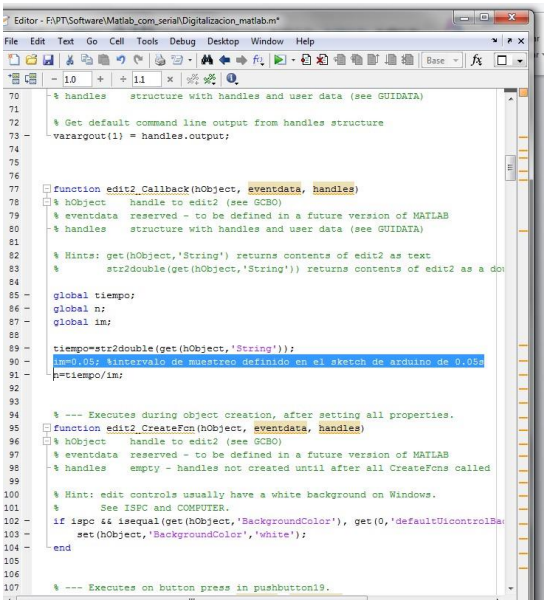


Fig. 40

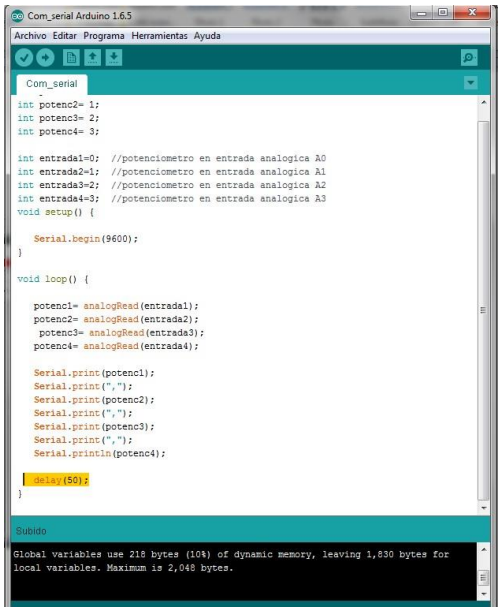


Fig. 41

10. Después de hacer correctamente los pasos anteriores podemos correr el programa dando clic en la parte superior de la ventana donde hay un triángulo verde y nos deberá salir la Interfaz Gráfica como la figura 43.

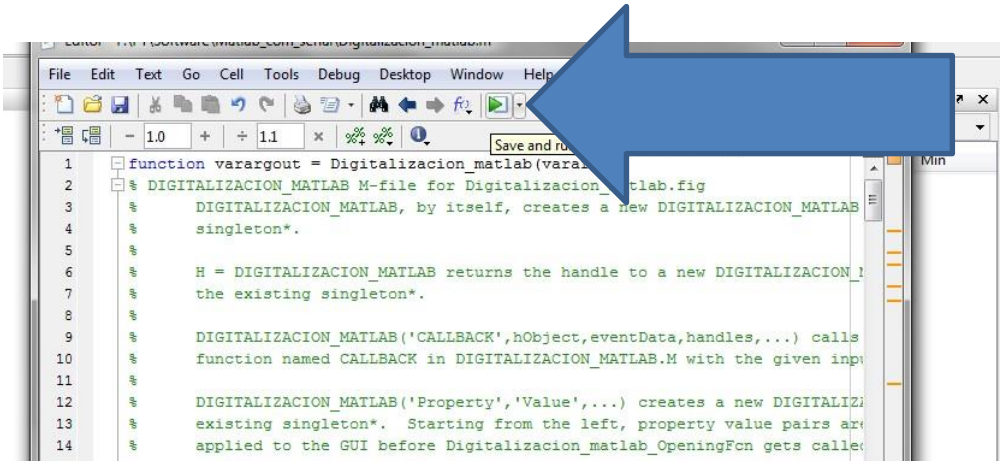


Fig. 42

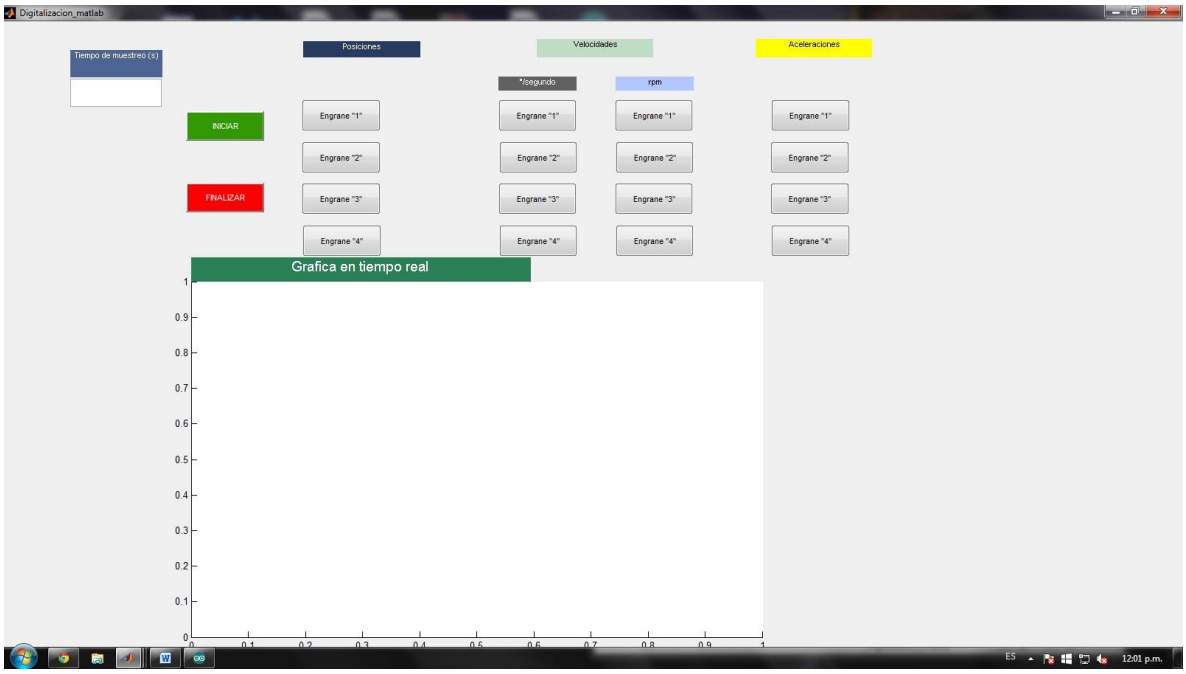


Fig. 43

D. MUESTREO DE LA SEÑAL

11. En la interfaz gráfica indicamos el tiempo que queremos muestrear la señal y damos clic en iniciar, si se han hecho correctamente los pasos del 1 al 8 no debería haber ningún problema al iniciar y se debería ver como en la figura 45 el muestreo de los 4 engranes en tiempo real.

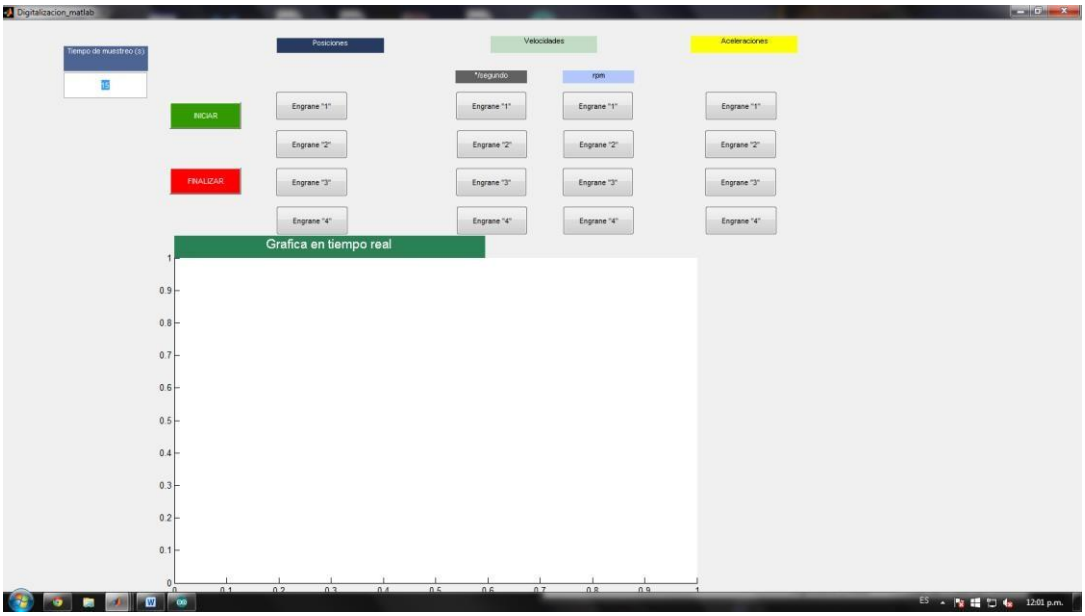


Fig. 44

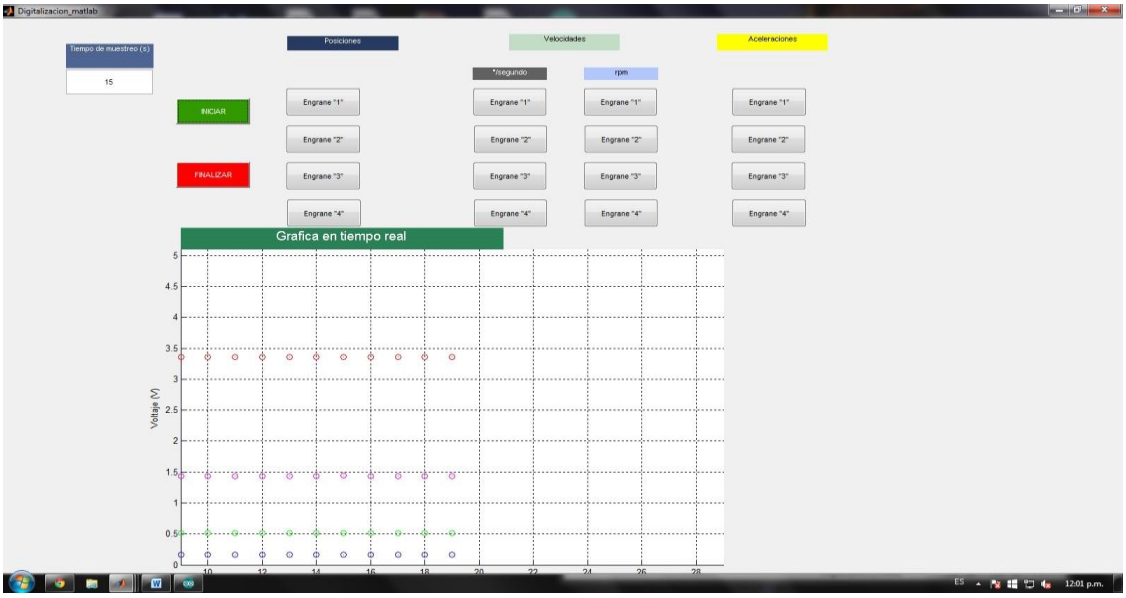


Fig. 45

12. Para hacer el análisis de la posición, velocidad y aceleración podemos desplegar las gráficas seleccionando el botón correspondiente a la variable que queremos estudiar.

    En la figura 46 se muestran las 4 graficas del comportamiento del engrane 1

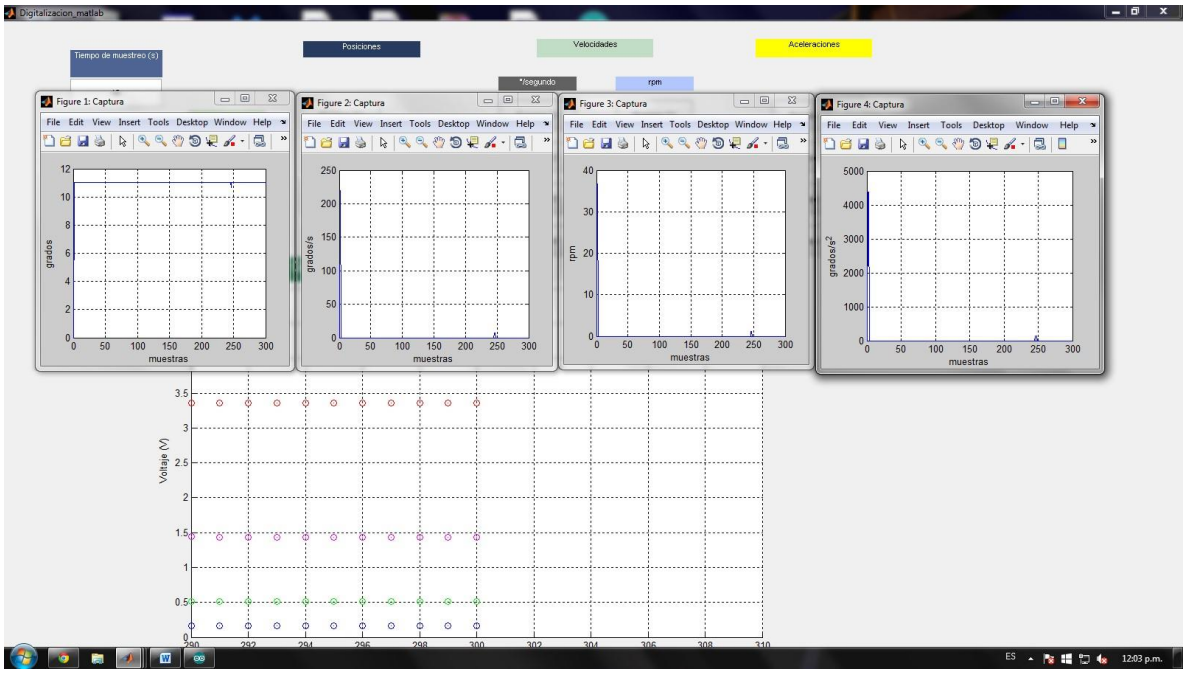


Fig. 46

E. CERRAR COMUNICACIÓN

13. Finalmente para cerrar la comunicación serial de Matlab con Arduino damos clic en el botón FINALIZAR que se encuentra en la interfaz gráfica en color rojo, esto cerrará las ventanas abiertas y la comunicación serial, por ultimo cerramos Matlab y Arduino y procedemos a desconectar la placa.

VIII. CONCLUSIONES

Con la digitalización de las señales de cada engrane podemos hacer un análisis más completo más detallado y más preciso del comportamiento cinemático de cada uno y hacer comparaciones con los cálculos teóricos.

Con la realización de este proyecto se tiene una gran herramienta para implementar en los cursos que requieran del análisis de cinemática de engranes ya que la señal que obtenemos con este dispositivo es fiable mediante la comunicación de Arduino y Matlab.

Existen dispositivos como son actuadores, sensores entre otros que nos permiten digitalizar un proceso mecánico en tiempo real con el fin de tener un mejor manejo de las propiedades cinemáticas de dicho proceso.

IX. REFERENCIAS

[1] Pérez Moreno Romy, 2006, “Análisis de Mecanismos y Problemas Resueltos”, 2a. Ed. Alfaomega.

[2] Olayo Garay Juan Carlos, 2015. "Diseño, manufactura y construcción de un prototipo didáctico de una caja de velocidades planetaria". Proyecto de Integración en Ingeniería Mecánica. Universidad Autónoma Metropolitana, Unidad Azcapotzalco.

[3] Montesinos Cruz Gerardo, 2015. "Diseño y construcción de un tren planetario con brazo dentado". Proyecto de Integración en Ingeniería Mecánica. Universidad Autónoma Metropolitana, Unidad Azcapotzalco.