BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRONICA

CONTROL III

REPORTE NO. 2

Arduino-Matlab

PROFESOR:

Dr. Jaime Cid Monjaraz

jcid@ece.buap.mx

Equipo:

Campos Pérez José Mario

Calderón Martínez Alfredo

Montes Ramírez Jared

Primavera 2013

1.-OBJETIVO

En esta primera práctica vamos a realizar un sistema controlado por la tarjeta Arduino Uno, que consiste en monitorear un sensor de temperatura, el cual el comportamiento de este lo vamos a visualizar en Matlab y posteriormente obtendremos una ecuación que represente a esa grafica obtenida.

2.-MARCO HISTORICO

- Arduino nació como un proyecto educativo allá por el año 2005 sin pensar que algunos años más tarde se convertiría en líder del mundo DIY (Do It Yourself).
- Su nombre viene del nombre del bar Bar di Re Arduino donde Massimo Banzi pasaba algunas horas, el cual a su vez viene del nombre de un antiguo rey europeo allá por el año 1002.
- Banzi dice que nunca surgió como una idea de negocio, es más nació por una necesidad de subsistir ante el eminente cierre del Instituto de diseño Interactivo IVREA en Italia. Es decir, al crear un producto open hardware (de uso público) no podría ser embargado. Es más hoy en día Arduino tiene la difícil tarea de subsistir comercialmente y continuar en continuo crecimiento.
- A la fecha se han vendido más de 250 mil placas en todo el mundo sin contar las versiones clones y compatibles.
- Para su creación participaron alumnos que desarrollaban sus tesis como Hernando Barragan (Colombia) quien desarrollo la plataforma de programación Wiring con la cual se programa el microcontrolador.
- Hoy en día con Arduino se pueden fabricar infinidad de prototipos y cada vez su uso se viene expandiendo más. Desde cubos de Leds, sistemas de automatización en casa (domótica), integración con el Internet, displays Twitter, kit analizadores de ADN.
- Google ha apostado por el proyecto y ha colaborado en el Android ADK (Accesory Development Kit), una placa Arduino capaz de comunicarse directamente con Smartphone Android para obtener las funcionalidades del teléfono (GPS, acelerómetros, GSM, abuses de datos) y viceversa para que el teléfono controle luces, motores y sensores conectados de Arduino.
- El primer prototipo fue desarrollado en el instituto IVRAE pero aún no se llamaba Arduino.
  Para la producción en serie de la primera versi[on se tomaron en cuenta algunas consideraciones: Economía (no > a 30 Euros), debía ser Plug and Play, utilizaron el color azul para marcar una diferencia con las placas convencionales, trabajar en todas las plataformas (Mac, Windows y Linux). Como se muestra en la **fig.1**

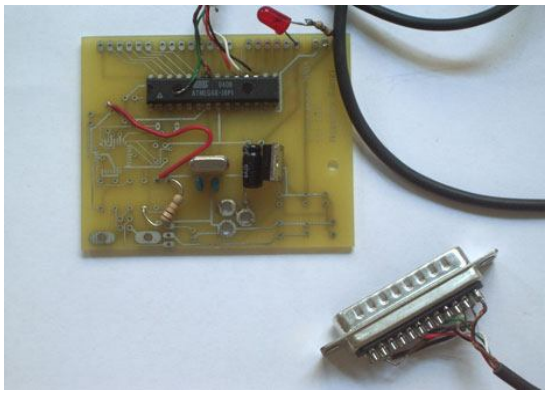


Fig. 1

- La primera producción fue de 300 unidades y se las dieron a los alumnos del Instituto IVRAE, (las ganancias fueron de sólo 1 dólar por placa), con el fin de que las probaran y empezaran a diseñar sus primeros prototipos.
- Uno de los primeros proyecto fue un reloj alarma, el cual no se apagaría hasta que no te pararas de la cama.
- Tom Igoe, profesor y padre de la computación física se unió al proyecto luego que se enterara del mismo a través de la web. El ofreció su apoyo para desarrollar el proyecto a grandes escalas.
- Hoy por hoy Arduino te permite crear cosas por ti mismo.
- Varias universidades como Standford y Carnegie Mellon y el MIT usan Arduino en sus campus.
- En la feria Maker Fair del 2011 se presentó la primera placa Arduino 32 Bit para trabajar tareas más pesadas. Entre ellas se presentó la impresora en 3D de MakerBot capaz de de imprimir en resina cualquier modelo en 3D.
- Esta plataforma abierta, puede ser utilizada para el aprendizaje así como para automatizar cualquier sistema mecatronico existen múltiples versiones de la tarjeta Arduino, para realizar esta práctica vamos a utilizar la versión Arduino UNO como se observa en la **fig. 2**

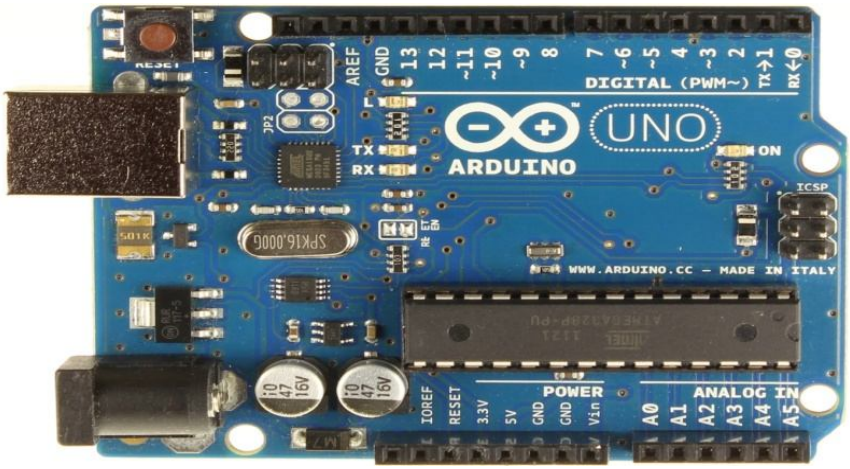


Fig. 2

3.- MARCO TEORICO

CARACTERISTICAS PRINCIPALES Arduino uno

Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 entradas / salidas digitales pines (de las cuales 6 se puede utilizar como salidas PWM), 6 entradas analógicas, un cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reset. Contiene todo lo necesario para apoyar el microcontrolador, basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA a CC o batería para empezar.

A continuación se describen las características de la tarjeta de adquisición de datos Arduino uno. Referencias en **Tabla 1.**

Características principales

Micro controladores	ATmega328
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Digital I/O Pines	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente por I DC / O Pin	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz
o, si el suministro de tensión a través de la toma de poder, acceder a él a través de esta clavija.	

Tabla 1

Alimentación Externa para Arduino uno

El Arduino UNO puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente. Externa (no USB) de potencia puede venir con un adaptador de corriente alterna a corriente continua (pared-verruga) o la batería. El adaptador se puede conectar al conectar un centro de 2.1mm-positivo clavija en jack de alimentación de la placa. Conduce de una batería se pueden insertar en los encabezados de pines Gnd y Vin del conector de alimentación.

La junta puede operar en un suministro externo de 6 a 20 voltios. Si se proporcionan menos de 7V, sin embargo, el pin de 5V puede proporcionar menos de cinco voltios y el tablero puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

VIN. La tensión de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (en lugar de 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de esta clavija, • 5V.

Este pin como salida una 5V regulada por el regulador en el tablero. La junta se puede suministrar corriente, ya sea a partir de la entrada de alimentación (7 - 12 V), el conector USB (5V), o el pasador de VIN de la junta (7-12V).

El suministro de tensión a través de los pines de 5V o 3.3V no pasa por el regulador, y puede dañar la placa. No se lo aconsejo.

3.3V Una tensión de alimentación 3,3 generado por el regulador de a bordo. Consumo de corriente máxima es de 50 mA.

GND. Pins de tierra.

Programación

Arduino Uno se puede programar con el software Arduino.

El ATmega328 en la Arduino Uno viene con un gestor de arranque que le permite cargar nuevo código a la misma sin el uso de un programador de hardware externo. Se comunica con el original STK500 protocolo.

También puede pasar por alto el gestor de arranque y programar el microcontrolador a través del ICSP (programación In-Circuit Serial) cabecera.

El ATmega16U2 (8U2 o en los tableros de REV1 y REV2) Código fuente del firmware disponible. El ATmega16U2 / 8U2 se carga con un cargador de arranque DFU, que puede ser activado por:

- En Rev1 juntas: conectar el puente de soldadura en la parte posterior de la tarjeta y luego reiniciar el 8U2.
- En Rev2 o placas posteriores: hay una resistencia que tirar de la línea 8U2/16U2 HWB a tierra, por lo que es más fácil de poner en modo DFU.

A continuación, puede utilizar el software FLIP de Atmel (Windows) o el programador DFU (Mac OS X y Linux) para cargar un nuevo firmware. O bien, puede utilizar el encabezado ISP con un programador externo (sobrescribir el gestor de arranque DFU).

Termistor

Es un sensor resistivo de temperatura. Su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con la temperatura. El término termistor proviene de **Therm**ally Sensitive Res**istor**. Existen dos tipos de termistor:

- NTC (Negative Temperature Coefficient) – coeficiente de temperatura negativo
- PTC (Positive Temperature Coefficient) – coeficiente de temperatura positivo

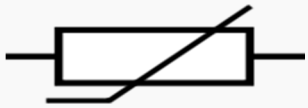
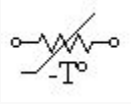
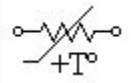


Fig.3.1

Símbolo genérico de un termistor.



- Símbolo NTC.



- Símbolo PTC.

Cuando la temperatura aumenta, los tipo PTC aumentan su resistencia y los NTC la disminuyen.

Su funcionamiento se basa en la variación de la resistencia de un semiconductor con la temperatura, debido a la variación de la concentración de portadores. Para los termistores NTC, al aumentar la temperatura, aumentará también la concentración de portadores, por lo que la resistencia será menor, de ahí que el coeficiente sea negativo. Para los termistores PTC, en el caso de un semiconductor con un dopado muy intenso, éste adquirirá propiedades metálicas, tomando un coeficiente positivo en un margen de temperatura limitado. Usualmente, los termistores se fabrican a partir de óxidos semiconductores, tales como el óxido férrico, el óxido de níquel, o el óxido de cobalto.

Sin embargo, a diferencia de los sensores RTD, la variación de la resistencia con la temperatura es no lineal. Para un termistor NTC, la característica es hiperbólica. Para pequeños incrementos de temperatura, se darán grandes incrementos de resistencia.

Por ejemplo, el siguiente modelo caracteriza la relación entre la temperatura y la resistencia mediante dos parámetros:

$$R_T = A \cdot e^{\frac{B}{T}}$$

Con

$$A = R_0 \cdot e^{\frac{-B}{T_0}}$$

Dónde:

- R_T es la resistencia del termistor NTC a la temperatura *T* (*K*)
- R_0 es la resistencia del termistor NTC a la temperatura de referencia T_0 (*K*)
- *B* es la temperatura característica del material, entre 2000 *K* y 5000 *K*.

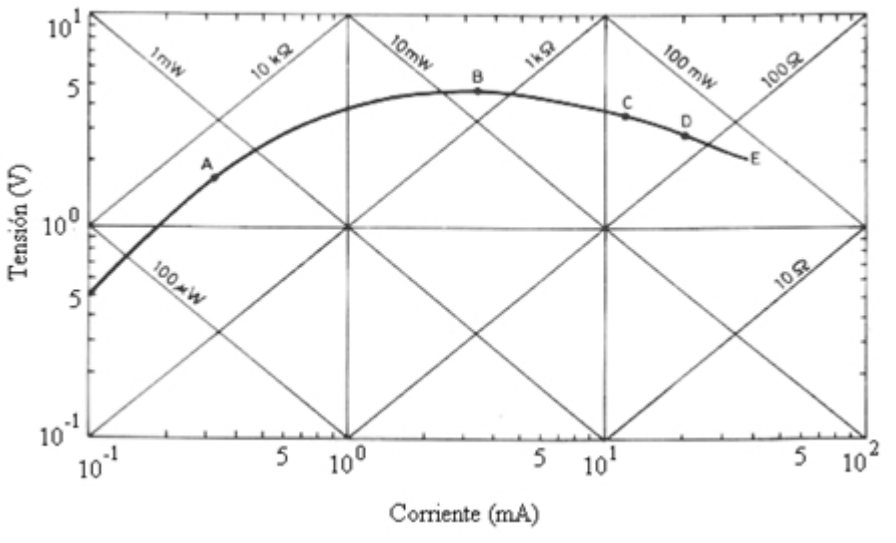
Por analogía a los sensores RTD, podría definirse un coeficiente de temperatura equivalente α, que para el modelo de dos parámetros quedaría:

$$\alpha = \frac{1}{R_T} \cdot \frac{dR_T}{dT} = -\frac{B}{T^2}$$

Puede observarse como el valor de este coeficiente varía con la temperatura. Por ejemplo, para un termistor NTC con *B* = 4000 *K* y *T* = 25 °C, se tendrá un coeficiente equivalente α = -0.045 K^{-1}, que será diez veces superior a la sensibilidad de un sensor Pt100 con α = 0.00385 K^{-1}.

El error de este modelo en el margen de 0 a 50 °C es del orden de ±0.5 °C. Existen modelos más sofisticados con más parámetros que dan un error de aproximación aún menor.

En la siguiente figura se muestra la relación tensión – corriente de un termistor NTC, en la que aparecen los efectos del auto calentamiento.



Grafica 3.1 Relación de tensión y corriente de un termistor NTC

Auto calentamiento.

A partir del punto A, los efectos del auto calentamiento se hacen más evidentes. Un aumento de la corriente implicará una mayor potencia disipada en el termistor, aumentando la temperatura de éste y disminuyendo su resistencia, dejando de aumentar la tensión que cae en el termistor. A partir del punto B, la pendiente pasa a ser negativa.

Características

Las termo resistencias más comunes se fabrican de alambres finos soportados por un material aislante y encapsulados. El elemento encapsulado se inserta dentro de una vaina o tubo metálico cerrado en un extremo que se llena con un polvo aislante y se sella con cemento para impedir que absorba humedad.

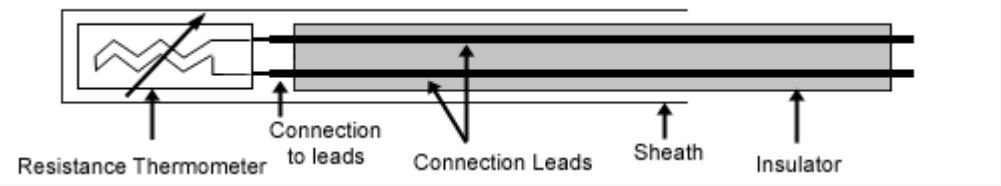


Diagrama esquemático de una resistencia termométrica, o resistencia detectora de temperatura (RTD).

Tipos

- 

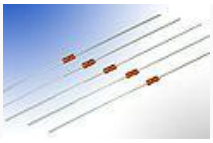
Termistor (tipo perla)

- 

Termistor (tipo SMD)

- 

Termistor (tipo disco)

- 

Termistor (axial)

- 

Sonda de medida

Material:

-1 Tarjeta de adquisición de datos Arduino uno

-1 termistor tipo PTC de acción positiva

-1 Protoboard

-1 OPAMP LM358

-3 resistencias (10kΩ -1MΩ -330Ω)

-Conexiones varias

Software a utilizar:
-Matlab
-Arduino

4.- DESARROLLO

Para poder realizar el monitoreo de una señal analógica con la tarjeta Arduino, que en este caso es el monitoreo de temperatura que es la variable física que se va a medir en nuestro sistema armamos el siguiente circuito que se muestra a continuación.

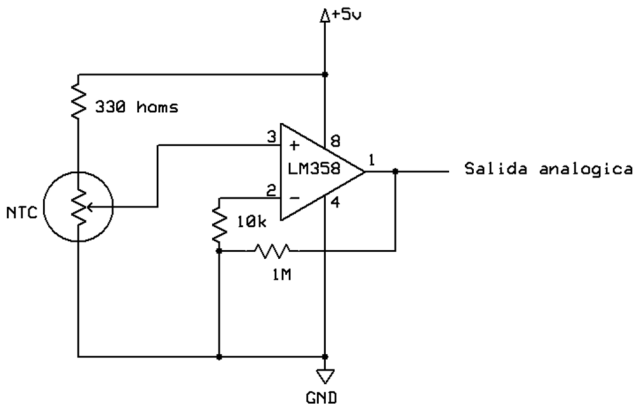


Fig. 4.1 Diagrama del termistor

El termistor esta en serie con una resistencia de 330Ω y la resistencia del termistor es de 100kΩ en este caso se utiliza el termistor de acción positiva PTC el cual al aumentar la temperatura también aumenta la resistencia, nuestra alimentación es de 5 volts y la salida que se obtiene esta dad en milivolts; al realizar la prueba a temperatura ambiente nos da una salida de 3.5 volts ya amplificada la cual se mantiene constante hasta que se eleva la temperatura.

Amplificando la señal de salida del termistor con el OPAMP Lm358 en configuración de ganancia 1000 y la alimentación del OPAMP es con la fuente de 5 volts, debido a que las características del CI sirven muy bien parta aplicaciones de instrumentación y no es necesaria una fuente simétrica.

Como se muestra en la gráfica obtenida en Matlab nuestra línea constante empieza a 3.5 volts y al aumentar la temperatura disminuye el voltaje el cual tiende a ser cero si se llegara a mantener la temperatura arriba de 80 grados, en este caso se muestra en un tiempo determinado el comportamiento de dicha medición. Tabla 4.1

NOTA:

Nuestro circuito sirve para aplicaciones generales, si se desea utilizar para control digital se tiene que implementar una compuerta NOT la cual nos servirá como un candado para que al llegar a 5 volts este pueda servir para otras aplicaciones con sistemas digitales.

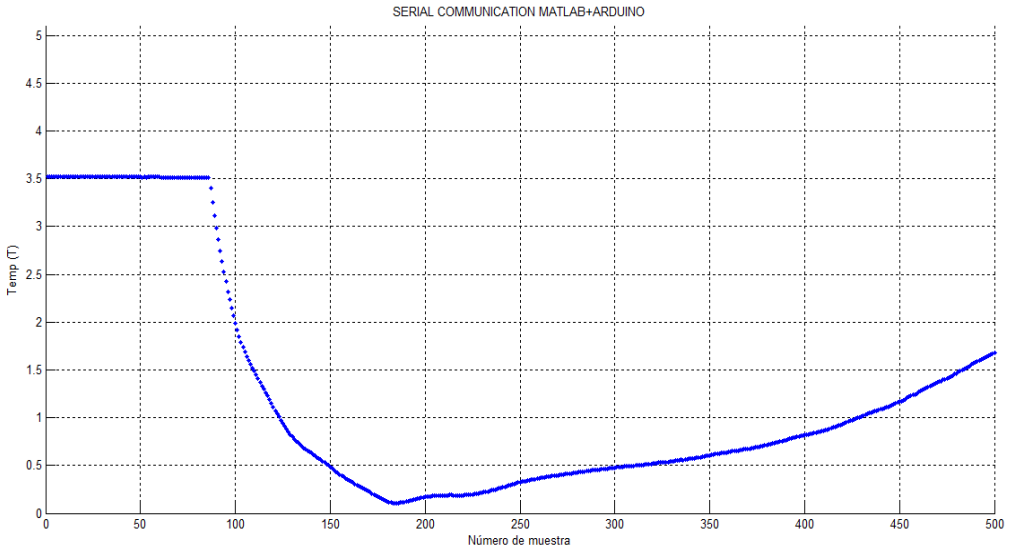


Tabla 4.1

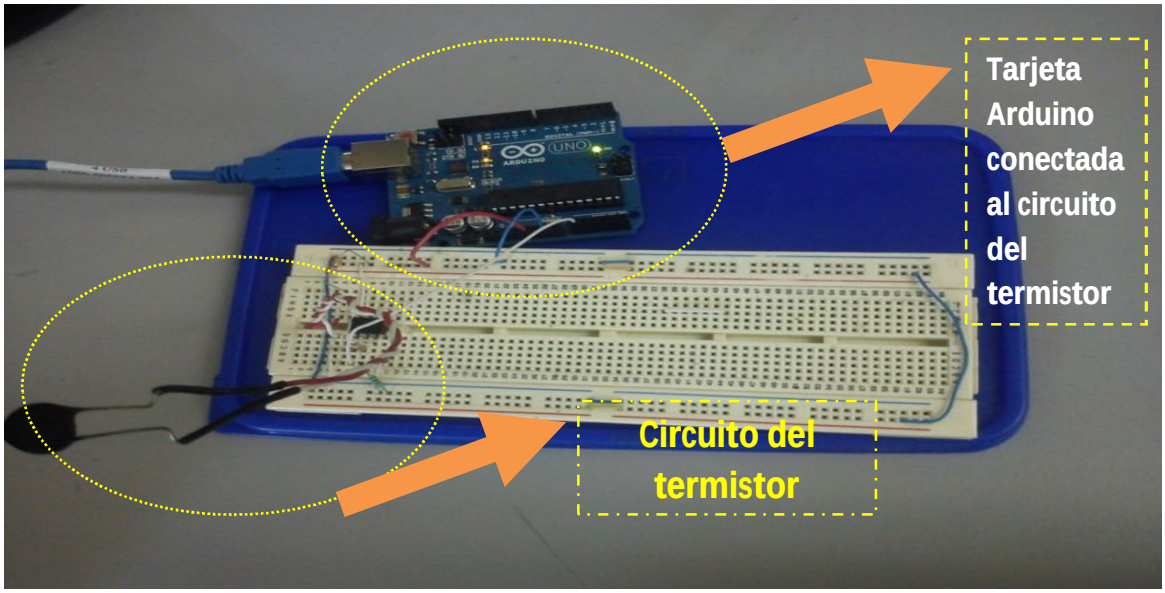
El código para Matlab:

```
function Matlab_Arduino(numero_muestras)

% Matlab + Arduino Serial Port communication
close all;
clc;
y=zeros(1,1000); %Vector donde se guardarán los datos
%Inicializo el puerto serial que utilizaré
delete(instrfind({'Port'},{'COM4'}));
puerto_serial=serial('COM4');
puerto_serial.BaudRate=9600;
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');
%Abro el puerto serial
fopen(puerto_serial);
%Declaro un contador del número de muestras ya tomadas
contador_muestras=1;
%Creo una ventana para la gráfica
figure('Name','Serial communication: Matlab + Arduino')
title('SERIAL COMMUNICATION MATLAB+ARDUINO');
xlabel('Número de muestra');
ylabel('Temp (T)');
grid on;
hold on;

%Bucle while para que tome y dibuje las muestras que queremos
while contador_muestras<=numero_muestras
        ylim([0 5.1]);
        xlim([contador_muestras-20 contador_muestras+5]);
        valor_potenciometro=fscanf(puerto_serial,'%d')';
        y(contador_muestras)=(valor_potenciometro(1))*5/1024;
        plot(contador_muestras,y(contador_muestras),'X-r');
        drawnow
        contador_muestras=contador_muestras+1;
end

%Cierro la conexión con el puerto serial y elimino las variables
```

El código que se realizo para Arduino:

```
int potenciometro_pin=0; / / Analog 0

int valor_potenciometro=0;


void setup(){

  Serial.begin(9600);

}


void loop(){

  valor_potenciometro=analogRead(potenciometro_pin);

  Serial.println(valor_potenciometro);

  delay(100);

}
```

Aquí se muestra acontinuacion como se hizo la conexión entre nuestro circuito y la tarjeta arduino



Conclusiones

Durante la realización de esta practica tuvimos que hacer un interfaz entre Matlab y Arduino el cual no fue difícil hacerlo, ya que hay tutoriales en los cuales nos indican paso a paso como hacer dicha interfaz.

El problema que se obtuvo al armar el circuito fue que el termistor no es lineal, esto implica que para poder realizar las mediciones teníamos que enfriarlo de manera forzada, la cual al hacer la amplificación con el opamp mejoro de manera que se pudo observar mejor el comportamiento del termistor.