

RA1000

Termómetro Digital



7 de abril de 2024

Universidad Panamericana

Erwin Alberto Lopez Hurtado

# Objetivo de la práctica

El objetivo de esta práctica es diseñar y construir un termómetro digital utilizando un microcontrolador ATmega16A y un sensor de temperatura LM35.

# **Materiales necesarios**

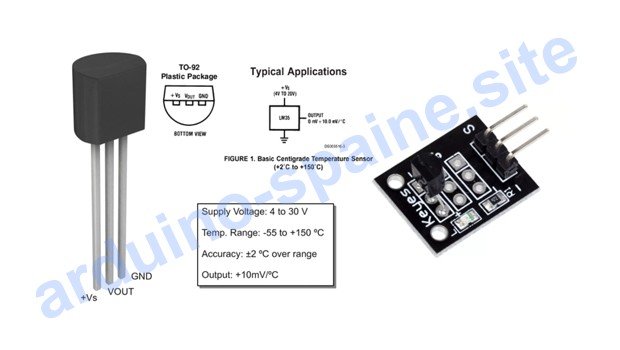
* Microcontrolador ATmega16A
* Sensor de temperatura LM35
* Placa de desarrollo para el ATmega16A
* Cables de conexión
* Fuente de alimentación
* Computadora con software de programación
* Multímetro (para medir la precisión del sensor)
* Diodos 1N914
* Resistencia de 18k

# **Planificación:**

Para la realización de este proyecto se utilizará un sensor LM35, este sensor de temperatura nos servirá para las mediciones correspondientes y luego comunicárselo directamente al microcontrolador.

Rango de temperatura de funcionamiento:

El rango de temperatura de funcionamiento del sensor LM35 es de -55°C a 150°C. Este rango lo hace adecuado para una amplia gama de aplicaciones que abarcan desde la temperatura ambiente hasta temperaturas extremas.



Precisión:

La precisión típica del sensor LM35 es de ±0.5°C a 25°C. La precisión puede variar ligeramente según las condiciones de operación, pero generalmente se mantiene dentro de un rango aceptable para muchas aplicaciones de medición de temperatura

La precisión garantizada por el fabricante es de +/-1°C a 25°C y +/-1,5°C a -55°C o +150°C para la versión menos precisa, lo que es más que suficiente para la mayoría de las aplicaciones. Otro gran punto fuerte del sensor LM35 es su ejemplar linealidad: menos de 1°C de error en todo el rango de -55°C a +150°C. Como cada grado Celsius corresponde a 10mV (es decir, 0,01 voltios) y la salida del sensor es (casi) perfectamente lineal.

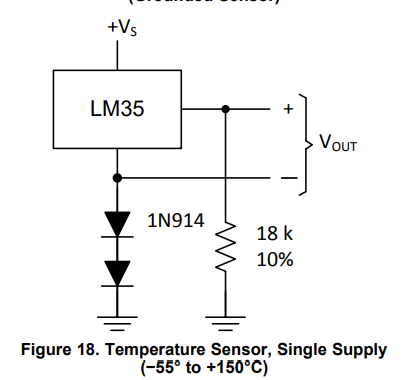
# Conexiones eléctricas:

Para la conexión de este se cuenta existen 2 formas de hacerlo.

Si queremos solamente contar con temperaturas arriba de 0 entonces seria con 3 pines:

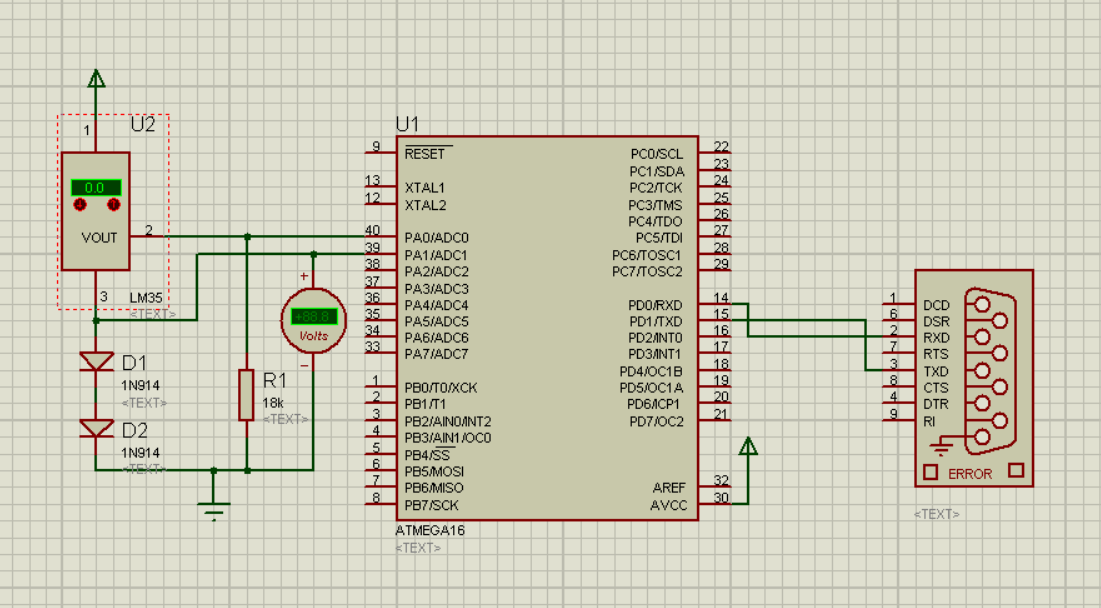
* Vs+ es la alimentación del circuito integrado y debemos conectarla a una tensión de entre 3v y 5v
* GND
* Vout es la tensión de salida del sensor el cual ira al pin del ADC que leerá el valor.

Por otra parte si requerimos todo el rango desde -55 a 150, necesitaremos realizar una conexión un tanto diferente. El siguiente diagrama muestra como seria la conexión.



Para este proyecto en Proteus conectaremos de la forma que tengamos todo el rango, esto no afectara de ninguna forma el físico ya que de no requerirse la parte negativa, solamente no se conectara y funcionara perfectamente igualmente.

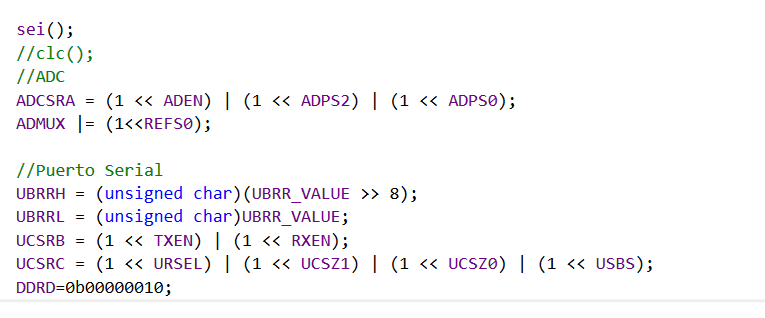
(Nota: la conexión no requiere una resistencia entre el positivo y el micro, sin embargo se le suele colocar por seguridad y que el circuito siempre sea preciso y no tengo fluctuaciones.)



A su vez será conectado el puerto serial, esto debido a que lo estaremos utilizando para el envió de datos y poder manejarlos de manera mas profesional en una interfaz de Python.

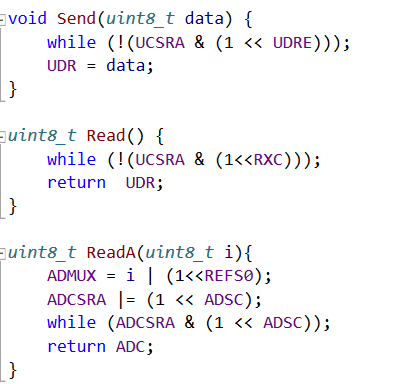
# Programación:

Lo primero a realizar es configurar el adc y el puerto serial que estaremos utilizando.



Aquí configuramos el ADC con un preescaler de 32, como nota, estaremos trabajando con el microcontrolador a 2mhz por conveniencia y facilidad. De igual forma configuramos el puerto serial a 9600 Bauds, para hacer los envíos lo más rápido posible, a 8 bits, sin paridad y con 2 bits de parada.

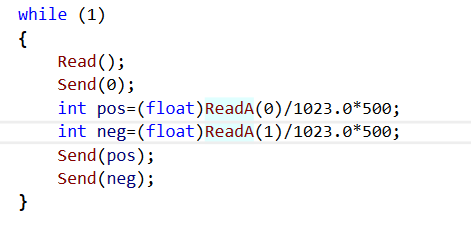
Para este trabajo contamos con 3 funciones principales que debemos entender.



Tenemos primero la función Send, que se encarga de mandar los datos por serial, se espera hasta poder y luego los envía. A su vez contamos con la función Read, esta espera hasta que haya algo que leer en el puerto serial, una vez lo haya lee el valor y lo regresa.

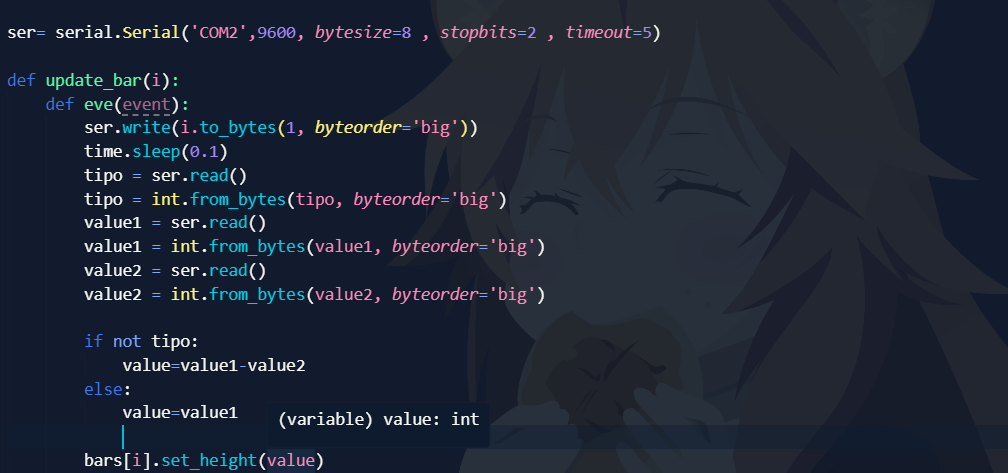
Por último nuestra función para leer de ADC (ReadA), esta coloca el valor del pin que queremos leer, se espera hasta que el ADC termine su conversión y luego nos regresa ese valor.

Este programa estará conectado por medio de serial con un programa de Python, entonces realmente lo único necesario que el micro debe hacer el leer y mandar los datos.



El código anterior se encarga de primero esperar una señal que le indique que lea, después manda el modo en que se encuentra el micro, ya sea en modo de negativos y positivos (0), o en modo de positivos solamente (1). Después lee ambos valores del ADC, tanto los positivos como los negativos, y los manda por serial.

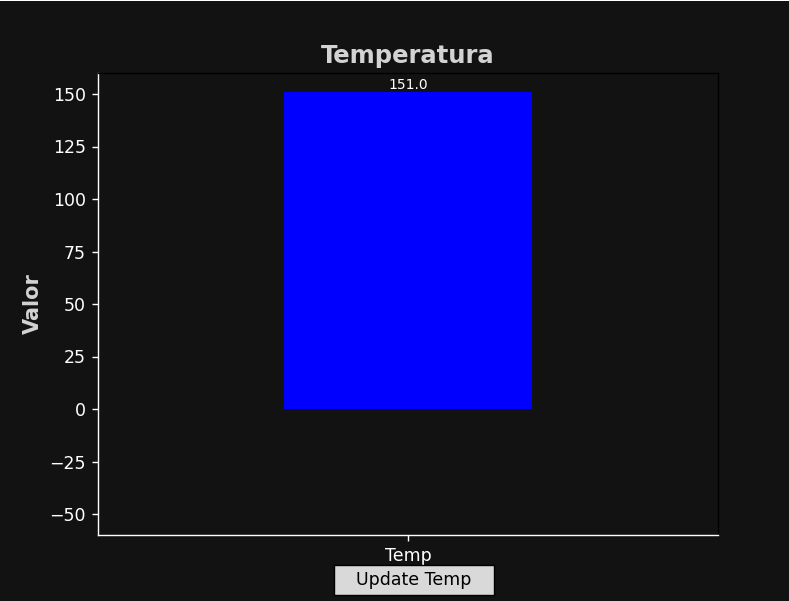
Para el código de Python, realmente mucho del mismo son cosas visuales y de Matplotlib, entonces esto es lo único relevante.



Esta parte se encarga primero de mandar la señal de lectura al micro, luego lee el tipo en el que esta, lee ambos valores y dependiendo el tipo asignara a (value), un valor distinto.

Luego este valor es mostrado como la altura de una barra en matplotlib.

(El código completo, estará adjuntado en la entrega final)



# Medición y calibración:

Como ya comentamos anteriormente, el lm35 tiene una precisión de +- 1 grado. Sin embargo un problema con el ADC de solo 10 bits es que no toma todos los decimales la mayoría de las veces. Por esta razon bastantes de nuestras lecturas se verán sesgadas por +-1 grado. Esto no es tanto problema, pero la única manera de resolverlo seria utilizando un ADC de mayor magnitud que pueda convertir los decimales de una manera más precisa.

# Prueba y análisis:

Nuestro código finalmente funciona en ambos aspectos, ya sea solo temperaturas positivas o ambas. Entonces realizamos la prueba de ambas formas y obtuvimos muy buenos resultados.

La mayoría de nuestras mediciones, como ya antes mencionamos tuvieron un error de +-1 grado. Esto debido al ADC que no puede convertir tan eficientemente los valores. Sin embargo nos pareció un resultado bastante aceptable y que no debería obstaculizar mucho el funcionamiento de este. Por desgracia tanto el ADC, como el puerto serial resultaron un inconveniente a la hora de manejar los datos. Ya que interfieren en el envió de la información precisa por el sensor. El ADC al no poder leer mas de 10 bits, no nos entrega una precisión absoluta en los valores que arroja el sensor.

Por otra parte el serial al solo contar con 8 bits. Resulta imposible mandar un valor mas preciso que no supere los 255 que nos da el puerto.

Estos resultados aunque desalentadores al principio debido a las limitantes. Abren muchas posibilidades para futuros usos. Tal vez utilizando un microcontrolador distinto seamos capaces de aprovechar al máximo la precisión que nos otorga este sensor. Mas sin embargo por esta ocasión concluimos que fue un trabajo excelente y que resulta de muchísima utilidad para la realización de mediciones básicas de temperatura utilizando el lm35.

# Bibliografía

*Hoja de datos de LM35, información de producto y soporte | TI.com*. (n.d.). <https://www.ti.com/product/es-mx/LM35>

Admin. (2024, January 29). *[GUÍA] Arduino Sensor de temperatura lm35 + código, conexión*. Arduino Spain. https://sp.arduino-france.site/lm35-arduino/