

ARDUINO Y LAS PUERTAS ANALÓGICAS


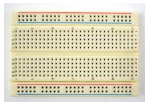




Convertidores Analógico a digital ADCs

Home • Arduino Y Las Puertas Analógicas

OBJETIVOS

- ★ Conocer los potenciómetros.
- ★ Comprender la conversión analógica a digital.
- ★ Aprender a usar las puertas analógicas de Arduino.

MATERIAL REQUERIDO.

| | |
|---|--|
|  | Arduino Uno o similar. Esta sesión acepta cualquier otro modelo de Arduino |
|  | Una Protoboard . |
|  | Un diodo LED . |
|  | Un potenciómetro de 10K Ω |
|  | na resistencia de 330 Ohmios. |
|  | Algunos cables de Protoboard.. |

LOS POTENCIÓMETROS

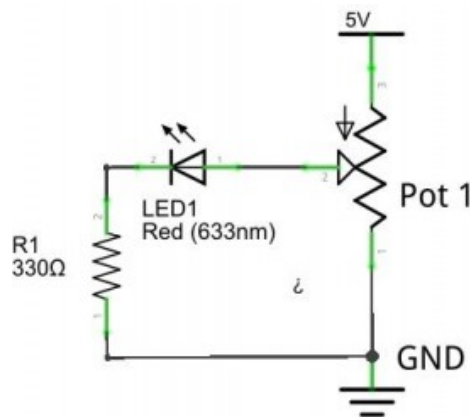
Hasta ahora hemos usado siempre resistencias fijas, de un valor dado. Pero a veces es conveniente disponer de una señal variable para controlar el circuito que nos interesa. Imaginad el volumen de un equipo de música, o el dial que sintoniza una emisora en una radio FM.

Un potenciómetro es, simplemente, un mecanismo para proporcionar una resistencia variable.

Hay potenciómetros de tantos tamaños, formas y colores como podáis imaginar, pero al final son una resistencia fija de un valor dado (10 kΩ en nuestro caso actual) y un mecanismo que permita deslizar un dial conductor sobre esa resistencia, que nos permita tomar una parte de ese valor.

Por eso un potenciómetro siempre tiene 3 pines en fila. Los del extremo se comportan como una resistencia del valor de fondo de escala del potenciómetro, y un pin central que va tomando valores de resistencia en función del movimiento que hagamos con el ajuste.

Vamos a montar un circuito como este (en el que el potenciómetro está rotulado Pot1):



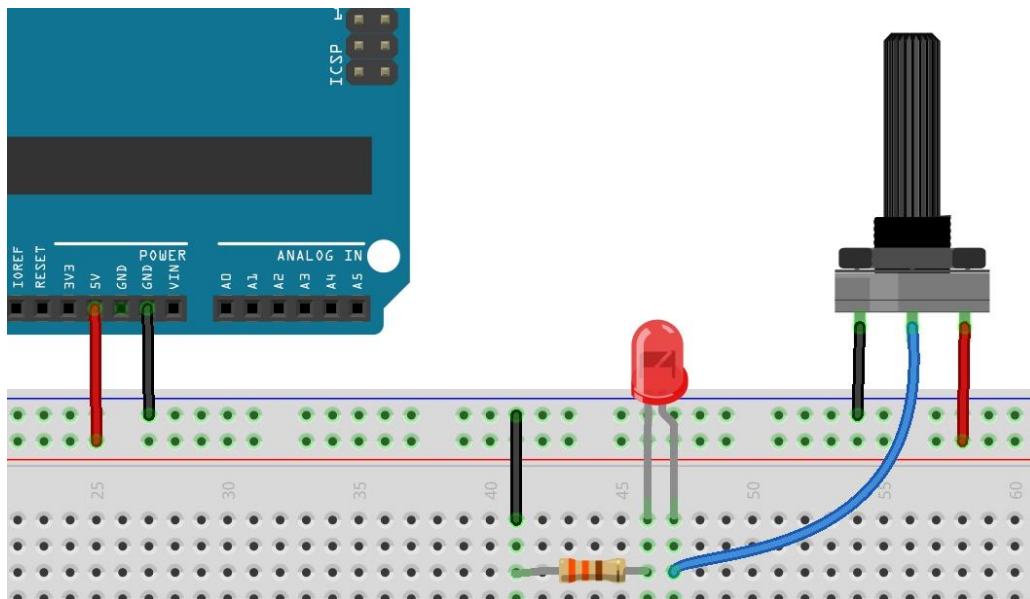
La idea es conectar 5V y GND a los extremos del Potenciómetro (no importa cual es uno y otro) y luego conectar el pin central al positivo de un LED y el negativo a GND directo, pasando por una resistencia de limitación.

De este modo cuando giremos el potenciómetro estaremos modificando la tensión que aplicamos a la entrada del LED, que variará entre 0 y 5V (Aunque ahora parezca extraño es muy sencillo) y habremos conseguido un regulador de intensidad del LED.

- ✔ Con una resistencia de 10k la intensidad en el circuito será de: $5V / 10.000\Omega = 0,5\text{ mA}$ Muy poco para conseguir iluminar el LED que requiere unos 20 mA. Así que durante la mayor parte del giro del potenciómetro el LED estará apagado.
- ✔ **Importante:** No olvides la resistencia R1. Aunque el potenciómetro limite la intensidad, hay un momento en que llegará a cero y ahí y tu LED fallecerá en acto de servicio.

CIRCUITO PARA PROTOBOARD

El montaje en la protoboard sería similar a esto ya que vamos a utilizar el Arduino simplemente para dar tensión al circuito y nada más. Veréis que la intensidad de la luz varía de forma continua al girar el potenciómetro.



- ✓ Recuerda que debido al exceso de resistencia del potenciómetro de prueba, durante la mayor parte del giro del ajuste el LED estará apagado.
- ✓ Nótese que en este caso utilizamos nuestro Arduino simplemente como fuente de alimentación para dar tensión al circuito.

ARDUINO Y LAS ENTRADAS ANALÓGICAS

Con Arduino hemos visto que podemos influir en el mundo exterior aplicando salidas todo / nada en los pines digitales y también que usando PWM podemos simular bastante satisfactoriamente señales analógicas en algunos de esos pines.

También hemos visto cómo detectar pulsaciones de botones, definiendo como entradas los pines digitales. Pero en muchas ocasiones los sensores que usamos para supervisar el mundo exterior, nos entregan una señal analógica. Es el caso de los sensores de temperatura o distancia, de presión o PH, de intensidad de corriente en un circuito o de caudal de agua en una tubería.

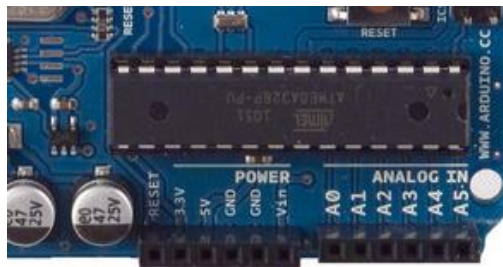
Para leer este tipo de señales continuas necesitamos un convertidor analógico a digital (o ADC por sus siglas en inglés) y que nos permite leer el valor de una señal analógica en un momento dado.

Estos convertidores toman una muestra del valor actual de la señal y nos entregan su *valor instantáneo*, medido en Voltios.

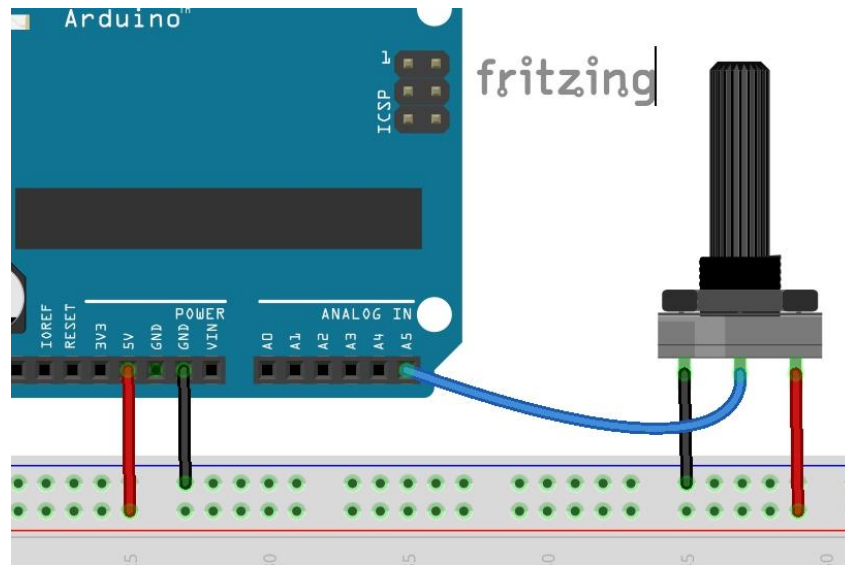
Mediante la lectura repetida de muestras a lo largo del tiempo podemos reconstruir la señal original con mayor o menor precisión, dependiendo de la exactitud de nuestra medida y de la velocidad a la que pueda tomar esas muestras.



Arduino UNO dispone de seis convertidores analógico a digital, nominados de A0 hasta A5, rotuladas como ANALOG IN:



Veamos cómo usar las entradas analógicas con un circuito como este, en el que damos tensión a los extremos de un potenciómetro y conectamos el pin central (el variable) a la entrada de la puerta A5 de Arduino:



- ✓ Parece buen momento para destacar que los convertidores ADC leen valores de tensión y no resistencia, por lo tanto, lo que vamos a leer es la **calda de tensión** en el potenciómetro a medida que giramos el ajuste.

La primera curiosidad es que no necesitamos declarar en el setup() que vamos a usar una puerta analógica. Y la segunda es que para tomar una muestra (leer) del pin A5, usaremos la instrucción:

```
int Val = analogRead(A5);
```

- ✓ Los convertidores de Arduino UNO y Mega son de 10 bits de resolución por lo que nos devolverá valores entre 0 y $2^{10} = 1.024$ para tensiones entre 0 y 5V. En cambio el Arduino DUE dispone de convertidores de 12 bits por lo que el valor de sus lecturas estará entre 0 y 10^{12} o sea 4.096, es decir tiene mejor resolución(pero sólo puede leer hasta 3,3V).
- ✓ Asegúrate de no usar sensores que puedan dar más de 5V máximo (con Arduino UNO y Mega), ya que dañarías el chip principal de Arduino.

Vamos a escribir un programa que lea el valor del pin A5 y lo envíe a la consola para que podamos visualizarlo.

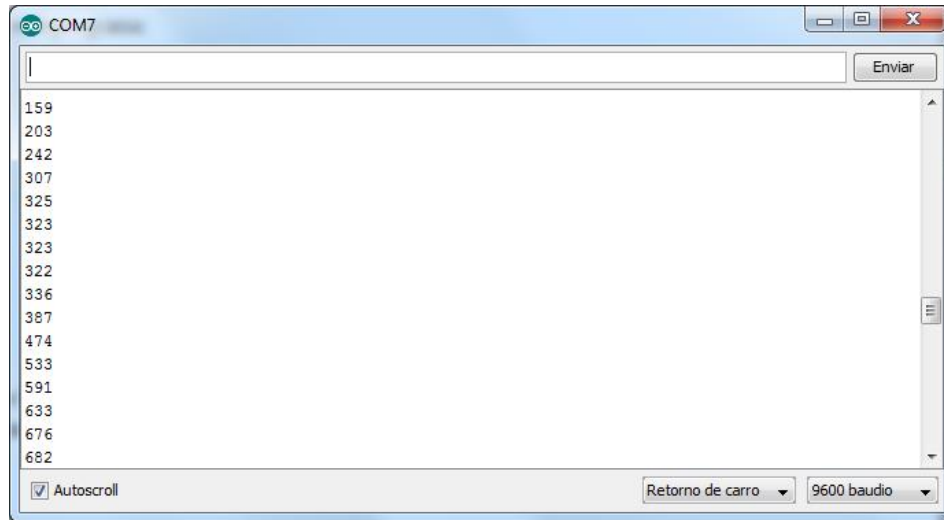
USANDO LAS PUERTAS ANALÓGICAS

Prueba este programa:

```
void setup()
{
  Serial.begin(9600); // Iniciamos la puerta serie
}
```

```
void loop()
{
    int Lectura = analogRead(A5) ;
    Serial.println( Lectura);
    delay(200) ;
}
```

Cuando lo vuelques, arranca la consola y veras que a medida que giras el ajuste las lecturas varían de forma continua reflejando la posición del potenciómetro, las lecturas reflejan la caída en voltios en el.



No puedo resistirme a proponeros esta prueba: Desconecta el potenciómetro de la puerta A5 y observa los resultados que arduino envía a la consola. ¿Porque salen esos valores?

- ✔ Al no estar el A5 conectado a ninguna referencia válida, está flotando y los valores que captura son muestra de esa incoherencia. En realidad lo que está haciendo tu Duino es captar ruido aleatorio de radiofrecuencia e intentar darle sentido, pero lo tiene mal, como podeis ver.
- ✔ No obstante en condiciones normales los valores que leerá serán relativamente bajos. ¿Quieres que las oscilaciones crezcan en valor?. Fácil. Ponle una antena. Vale un simple cable de protoboard conectado desde el A5 a nada (O si coges el otro extremo entre los dedos, tu mismo haras de antena). Acabas de construir el receptor de Radio frecuencia mas inutil del mundo

UN ÚLTIMO COMENTARIO

Decíamos en una sección anterior, que la fidelidad con que podemos muestrear una señal analógica dependía, básicamente, de la resolución de la muestra y de la velocidad a la que podíamos muestrear la señal (Sample Rate en inglés).

Ya dijimos que la familia Arduino, dispone de convertidores de 10 bits por lo que nuestra resolución es de $2^{10} = 1.024$ y en el caso del DUE de $2^{12} = 4.096$. Pero hasta ahora no hemos visto a qué velocidad podemos tomar muestras con nuestro Arduino. Vamos a comprobarlo, con este mismo circuito.

Tenemos una función llamada `millis()` que nos indica en milisegundos el tiempo transcurrido desde que iniciamos Arduino y la podemos usar para ver cuantas muestras podemos tomar por segundo



TIENDA SCRATCH ARDUINO FORO PROYECTOS CONTACTO

```
void loop()
{
    unsigned long T ;
    int n = 0 ;
    T = millis();

    while (millis() <= T + 1000)          // Mientras no pase un Segundo = 1000 mS
    {
        analogRead( A5) ;
    }
}
```

```

        n++ ;           // Contamos cada vez que leemos
    }
    Serial.println(n);
}

```

- ✔ Hemos usado un `unsigned long` para guardar `millis` porque es el tipo que Arduino usa internamente para su reloj. Sería un error manejar `millis` con un `int` porque su valor máximo es 32.767 y midiendo milisegundos el contador desbordaría en poca más de 32 segundos.

Si corréis este programa en un Arduino UNO os dará, poco más o menos, un resultado de 8.940 muestras o lecturas por segundo. No está mal.

Es adecuado para muestrear señales que no varíen demasiado rápido con el tiempo, como son casi todos los sensores habituales en la industria, pero que se quedará corto si queréis muestrear señales de audio.

- ✔ Para jugar con audio es mejor usar un Arduino DUE. Tiene una velocidad de reloj 4 veces más rápida (os hará falta), capacidad de muestreo a velocidad de audio (40Khz) y auténticos convertidores DAC (digital to analog converters).
- ✔ De hecho no es complicado aumentar la velocidad de muestreo hasta unas 20.000 muestras por segundo con un Arduino UNO, pero para eso tenemos que puentear Arduino y saltar a programar el chip interior Atmega 328. No es momento para ello, pero hay formas.

RESUMEN DE LA SESIÓN

- ★ Ya conocemos el uso del potenciómetro.
- ★ Hemos presentado los conceptos básicos en la conversión analógica to digital.
- ★ Aprendimos a leer las puertas analógicas de Arduino.
- ★ Sabemos que podemos leer las puertas analógicas unas 8.900 veces por segundo con una resolución de 10 bits, o sea entre 0 y 1.024.
- ★ Conocimos la función `millis()`.

[ANTERIOR](#)
[SIGUIENTE](#)

(9) COMMENTS



STARKMANN

01 Sep 20

Reply

He de confesar que del todo no lo estoy asimilando, sin embargo este tutorial de hoy ha servido para varias cosas. Para saber que $2^{10} = 1.024$ y que no se escribirlo al ordenador.

No se por que, pero algo me ha empujado a conectar al arduino al pc para suministrar corriente aun sin cargar el programa en el primer esquema y sabia o tenía la impresión (llámalo intuición) de que iba a funcionar, me refiero a donde el arduino se limita a dar corriente. Luego al leerlo me he afianzado en esa idea y me ha ayudado a comprender los pasos de hoy.

Por otro lado he comprendido un poco la lectura de mi arduino sobre el mundo exterior y para lo pequeñito que es, me sorprende lo que llega a ser capaz de procesar este aparatito, aunque supongo que en electrónica será habitual, me dan ganas de rodearme de pequeños aparatos de estos y llegar así a comprender el mundo exterior.

Luego llegamos a la programación donde voy sin rumbo, y he de copiar fielmente las indicaciones, ya que ando mas perdido que el limbo en ese parámetro.

Sin embargo sus indicaciones (las de este tutorial) y los comentarios recibidos en estas líneas me empujan a seguir adelante con unas ganas e interés del cual quiero hacer saber que estoy AGRADECIDISIMO a las personas que contribuyen a que esto sea así.



admin

01 Sep 2015

Reply

Starkman, Te recomiendo que te vayas tomando tu tiempo para digerir las cosas. EL mndo de la programación requiere una serie de conceptos que deben ir asentándose y especialmente cuando te acercas por primera vez. Nos ha pasado a todos. Recuerda la sensación cuando intentaste aprender a andar en bici, lo que cuesta y la frustración de caerte una y otra vez, pero si insistes, poco a poco las piezas van encajando y acabarás disfrutando. Un saludo y recuerda que estamos para lo que necesites



Jesús Sánchez

01 Sep 2015

Reply

Hola STARKMANN. Pasate por el foro y allí consulta lo que quieras. Siempre hay alguien que te hecha un cable (nunca mejor dicho...) Para eso está.

Saludos.



STARKMANN

03 Sep 2015

Reply

Gracias admin y Jesús.

La verdad es que no dispongo de la mitad del tiempo que tuve en mi juventud, es por ello que intento seguir adelante confiando en asimilar los conceptos, aunque tengo que tropezar una y otra y otra vez para aprender.

Así que sigo vuestros consejos, y ya me he registrado en el foro, estoy a falta de recibir el mail para poder completar el registro.



Jesús Sánchez

04 Sep 2015

Reply

Cuando entres al foro, pregunta todo lo que necesites, ya sea electrónica digital (programación) como electrónica analógica. Verás que avanzas enseguida.



Fernando Sanchez

13 Dic 2015

Reply

Antes de nada quiero agradeceros vuestro trabajo. Es muy útil para gente que como yo está empezando en esto de la electrónica y la programación. Me gustaría que si es posible me resolvieseis una duda; al ver los datos de lectura del potenciómetro en el monitor el mínimo no me lo da en 0, sino en 100 y pico y el máximo no me llega a 1023 sino a 600 y pico.

Muchas gracias de nuevo por vuestro trabajo y por compartirlo con todos.



admin

13 Dic 2015

Reply

Hola Fernando, podría ser posible que tu potenciómetro no estuviera bien, pero es mucho más probable que haya algún error en la forma de leerlo o que la tensión que le estás poniendo no sea 5V sino 3.3, o cualquier otro despiste que no será muy grave



Fernando Sanchez

15 Dic 2015

Reply

Muchas gracias por contestar admin. La verdad es que si, los potenciómetros que tengo son bastante maluchos, pero no soy capaz de encontrar unos que encajen bien en la proto. De todas formas, presionándolo si que hace contacto bien y debería leer correctamente. Bueno, cambiaré de pot y seguiré probando. Muchas gracias de nuevo



Mazuelas

10 Feb 2016

Reply

Disculpad, pero si llevas el valor de la resistencia variable a 0, no cortocircuitas la placa?, por cierto acabo de empezar con un starter Kit y vuestra página es la que me ha resultado más útil, gracias por compartir vuestro conocimiento.

GIVE A REPLY

Conectado como Javier. ¿Quieres salir?

Message

Post comment

Comments Protected by WP-SpamShield for WordPress



Sí, agrégame a tu lista de correos.

CATEGORIAS DE LOS PRODUCTOS

Selecciona una categoría ▼

Copyright © 2014 Redline Asesores All Rights Reserved. | Redline Asesores

