



Estos pines se pueden conectar internamente al módulo del convertor analógico, el microcontrolador tiene un solo módulo y estas entradas se seleccionan con una llave digital interna (multiplexador) para conectar una determinada entrada analógica al módulo analógico y así poder hacer una medición en ese canal analógico.

En el caso de un Arduino Uno, el valor de 0 voltios analógico es expresado en digital como 0x00 y el valor de 5V analógico es expresado en digital como 0x3F (1023).

Es decir que toda medición tendrá un número comprendido entre 0 y 1023, resolución de 10 bits.

Cuanta más resolución tiene un convertor más preciso será en su medición, por ejemplo Arduino DUE tiene una resolución de 12 bits, es decir que sus mediciones generan número entre 0 y 4095.

Una de las características claves del convertidor, es su número de bits, que define la resolución con la que podemos cuantificar esa conversión a digital. Cuanto mayor sea esta resolución mejor es la capacidad de aproximación al valor real que estamos midiendo.

Un ADC compara sucesivamente la señal que queremos medir en la entrada con una tensión de referencia interna altamente estabilizada.



Un ADC no proporciona valores absolutos, sino que proporciona una comparación cuantificada con relación a un valor de referencia.

Con una resolución de 10 bits tenemos lo siguiente,  $5V / 1024 = 0.0048 V$  (4.8 mV), entre cada medición hay 4.8 mV es decir que los voltajes superior 2.4 mV ya son tomados con el valor del siguiente escalón de medición.

Con 12 bits el resultado es  $5/4096 = 0.0012$  (1.2mV) separan los escalones de medición por lo tanto la incertidumbre en la medición está en el orden de 0,6 mV bastante más preciso que 2,4 mV.

Sin embargo, es justo decir que para todo lo que sea mediciones clásicas de temperatura, humedad, presión, etc., los 10 bits de un conversor son más que adecuados.

El datasheet de ATmega advierte de hacer lecturas rápidas entre pines analógicos (*analogRead*). Esto puede causar ruido eléctrico e introducir jitter en el sistema analógico.

Se aconseja que después de manipular pines analógicos en modo digital, añadir un pequeño retraso antes de pasar a leer otros pines analógicos.

También se puede cambiar la tensión máxima (siempre por debajo de  $V_{cc}$ ) que utiliza el ADC como referencia, es la llamada tensión de referencia y es la tensión contra la que todas las entradas analógicas hacen las conversiones.

Esta tensión de referencia se toma del pin *AREF*. Reducir el voltaje máximo del ADC tiene sentido para mejorar la resolución del ADC, sin embargo debe considerar que las mediciones se realizan tomando este voltaje como referencia por lo tanto debe ser un voltaje estable, si este voltaje se “mueve” también lo harán las mediciones.

Como se explicó anteriormente, con 5V la resolución es de  $5/1023 = 4,88$  mV para cada valor, pero para un sensor que no pasa de 3.3V la resolución es de  $3.3/1023 = 3.22$ mV.

El ADC interno también se puede utilizar en un modo de 8 bits, donde sólo se utilizan los 8 bits más significativos de la resolución de 10 bits completa, esto podría ser útil cuando se trabaja en ambientes ruidosos y sólo necesita 8 bits de resolución, el uso de este modo es un plus debido a que no es necesario dedicar más tiempo de CPU calculando los 10 bits completos.

El ADC también puede configurarse para que lleve a cabo una conversión y detenerse o puede ser configurado para funcionar en un modo de funcionamiento



libre de conversiones continuas siendo la primera opción la mejor cuando queremos leer diferentes pines, y el segundo es mejor cuando sólo tenemos que leer un pin y esto puede ahorrar algo de tiempo entre las conversiones.

También tenemos que tener cuidado de la frecuencia máxima de trabajo del ADC, este valor se especifica en la ficha hoja de datos del microcontrolador y es de **200 kHz**, este es el valor del reloj interno de la circuitería del ADC y se genera dividiendo el reloj principal ATmega, que en el caso del UNO es 16 MHz, este divisor del reloj se realiza mediante preescaladores y sólo hay un rango limitado de valores, por lo que la frecuencia máxima que podemos utilizar y estar dentro de la frecuencia máxima de trabajo es 125 kHz.

El siguiente preescalador supone usar el ADC a 250 kHz, en este caso no se puede garantizar la resolución de 10 bits, pero si una resolución de 8 bits. De todas formas en caso de necesitar un ADC más rápido se podría usar uno externo.

El ADC puede trabajar en dos modos, **single conversion mode** y **free running mode**.

En modo single conversion el ADC hace una sola conversión y para, pero en modo free running el ADC está continuamente convirtiendo, es decir, hace una conversión y luego comienza con la siguiente.

La comparación continúa dividiendo de nuevo la tensión y actualizando cada bit del registro ADC a 1 si el voltaje es alto en la comparación o 0 en el otro caso. Este proceso se realiza 10 veces (por cada bit de resolución del ADC) y genera como resultado la salida binaria que va desde 0 a 1023.

Los pines analógicos de Arduino también tienen todas las funcionalidades de los pines digitales, por lo tanto, si necesitamos más pines digitales podemos usar los pines analógicos.

Normalmente el uso del conversor analógico obliga a considerar distintos pasos y configuraciones en el hardware para obtener finalmente una medición fiable de una variable analógica. Afortunadamente el entorno de programación de Arduino hace transparente mucho de esta complicación, funciones y bibliotecas destinadas al manejo del conversor hacen que su implementación sea sencilla, sin embargo siempre se debe tener presente que detrás de un Arduino hay un microcontrolador y por tanto, a pesar que no lo veamos la complejidad electrónica existe y en puntuales situaciones deberemos lidiar con registros y configuraciones “a mano” por ejemplo cuando trabajemos con el ADC controlado por interrupciones.