

Materia:

Bases de datos.



Reporte#12

Uso del Convertidor Analógico Digital del ATmega1280.

Alumno:

Montoya Valdivia Omar
Antonio: 1252892

Profesor:

Jesús García

Programación y uso del ADC (Diagrama, Funcionamiento, regs. de conf. y operación).

El ADC convertidor analógico digital AVR, permite medir señales analógicas en forma digital, para ello el AVR cuenta con pines por donde le llegará la señal analógica, estos pines deben configurarse como entradas analógicas, el convertidor analógico digital AVR convierte la señal analógica que le llega al pin configurado como entrada analógica, en un número binario de 10 bits que representará la medida analógica, este número binario se guarda en sus registros ADCH y ADCL de 8 bits cada uno pero estos actúan como un solo registro de 16 bits, en el registro ADCH se guardan los bits mas significativos y en el registro ADCL se guardan los bits menos significativos, el número que representa la tensión analógica y guardado en forma binaria dentro de estos registros será de 10 bits para el ATMEGA2560.

El ADC convertidor analógico digital AVR necesita una tensión de referencia para poder trabajar adecuadamente, esta tensión de referencia V_{ref} normalmente será la tensión a la cual trabaja el AVR, aunque por programa se puede elegir otras tensiones de referencia; a la relación que hay entre la tensión de referencia V_{ref} y el máximo número binario de 10 bits $2^{10}-1=1023=111111111_2$ que representará la señal analógica se le conoce como resolución, por ejemplo para el caso del ATmega2560 se tendrá que la resolución del convertidor analógico digital AVR será:

$$Resolucion = \frac{V_{ref}}{2^{10} - 1} = \frac{V_{ref}}{1023}$$

Si se toma como la $V_{ref}=5V$ que es la tensión adecuada a la que trabaja el ATmega2560 se tendrá que la resolución de su convertidor analógico digital AVR será:

$$Resolucion = \frac{5V}{1023}$$

de donde

Resolución = $0,004887585533V \approx 0,0049V$ luego la resolución para el caso del convertidor analógico digital AVR de 10 bits con un voltaje de referencia de 5V será:

$$Resolución = 4.9mV$$

La resolución indica en este caso que a la tensión de referencia de 5V se le ha partido en 1023 partes iguales y cada una de esas partes equivalen a aproximadamente 4,9mV, al utilizar esto en forma digital indicará que para un 0 será 0V, si el voltaje aumenta desde 0 en 4,9mV se tendrá un 1, si aumenta 2 veces 4,9mV se tendrá un 2, si aumenta 3 veces 4,9mV se tendrá un 3, si aumenta 4 veces 4,9mV se tendrá un 4 etc.

Esto quiere decir, que al utilizar el convertidor analógico digital AVR, cada vez que la tensión analógica que llegue por el pin configurado como entrada analógica aumente en 4,9mV, el número que lo representa y almacenado en los registros **ADCH** y **ADCL** aumentará en 1, por ejemplo cuando la tensión analógica es 0, el número binario que lo representa será 0000000000, si la tensión analógica aumenta de 0 a 4,9mV el número binario que lo representa será 0000000001, si la tensión analógica aumenta de 4,9mV a $9,8=2*4,9\text{mV}$ el número binario que representa este valor será 0000000010, si la tensión analógica aumenta de 9,8V a $14,7=3*4,9\text{mV}$ el número binario que representa este valor será 0000000011, y así hasta que la tensión analógica se haga igual a la tensión de referencia, lo que ocurrirá cuando la tensión analógica aumente desde 0 de 4,9mV en 4,9mV unas 1023 veces lo cual es $1023*4,9\text{mV}$ que es un poquito mas de 5V porque la resolución se redondeo, el número binario que representa a los 5V será 1111111111.

Como el módulo convertidor analógico digital AVR es de 10bits, por lo que la resolución dependerá de la tensión de referencia.

El número binario de 10 bits que representará la tensión analógica, puede leerse desde los registros **ADCH** y **ADCL**, este número se puede guardar con una justificación a la derecha o una justificación a la izquierda, esto es así

000000xxxxxxxx es el número de 16 bits almacenado entre **ADCH** y **ADCL**, donde las 10 x que pueden ser 0 o 1 es el número de 10 bits que representa a la señal analógica, este es el caso la justificación a la derecha los 6 bits mas significativos se ponen a 0 automáticamente.

xxxxxxxx000000 es el número de 16 bits almacenado entre **ADCH** y **ADCL**, donde las 10 x que pueden ser 0 o 1 es el número de 10 bits que representa la señal analógica, este es el caso la justificación a la izquierda los 6 bits menos significativos se ponen a 0 automáticamente.

Para leer los registros **ADCH** y **ADCL** la hoja de datos indica que hay que leer primero el registro **ADCL** y luego el registro **ADCH**; utilizando el ATMEGA STUDIO se puede leer directamente desde el registro de 16bits al cual se le

llama ADC, para ello se crea una variable de 16 bits y dependiendo de si la justificación es a la derecha o a la izquierda se procederá así:

```
uint16_t medida_analógica;//se declara la variable de 16 bits
```

si la justificación es a la derecha se procede directamente a almacenar el número de 10 bits en la variable medida_analógica así:

```
medida_analógica=ADC;
```

si la justificación es a la izquierda se tendrá que hacer un corrimiento de 6 bits hacia la derecha para almacenar el número de 10 bits en la variable medida_analógica así:

```
medida_analógica=ADC>>6;
```

Luego el valor de la variable medida_analógica, mediante simples operaciones matemáticas se puede hacer que represente el valor en voltios, de la medida de la señal analógica que esta llegando a la entrada analógica del AVR como se ve a continuación:

El valor almacenado en la variable medida_analogica se multiplica por la resolución obteniéndose el valor en voltios de la señal analógica representada por el número binario de 10 bits.

```
medida_analógica=medida_analógica*Vref/1023;
```

Para la utilización del convertidor analógico digital AVR del ATmega2560 se tienen 4 registros para su control, los que son el **ADMUX**, el **ADCSRA**, el **ADCSRB** y el **DIDR0**; el ADC también puede producir **interrupciones** para lo cual se utilizan unos bits que forman parte del registro ADCSRA.

El módulo ADC convertidor analógico digital AVR necesita una frecuencia de trabajo adecuada para obtener conversiones confiables, esta frecuencia de trabajo para el convertidor analógico digital AVR tiene que estar comprendida entre los 50KHz y los 200Khz según la hoja de datos del ATmega88, lo que es común para otros microcontroladores AVR, si la frecuencia de trabajo del ADC es la mayor la conversión será mas rápida pero el valor obtenido no será muy confiable, por el contrario si la frecuencia de trabajo es la menor el valor obtenido será confiable pero la conversión tardará mas tiempo, se recomienda por tanto utilizar una frecuencia de trabajo del ADC intermedia por ejemplo de 100Khz.

La frecuencia de trabajo del ADC se obtendrá a partir de la frecuencia del trabajo del microcontrolador AVR, mediante el uso de prescaler como se comentará mas abajo donde se trata este tema.

El tiempo que tarda la conversión según la hoja de datos es para la primera conversión es de unas 25 veces el periodo de trabajo del ADC que viene a ser la inversa de la frecuencia de trabajo del ADC, las demás conversiones tardarán unas 13 veces el periodo de trabajo según la hoja de datos, lo que quiere decir que si se tiene por ejemplo una frecuencia de trabajo del convertidor analógico digital AVR de 100Khz, entonces su periodo de trabajo será de 10us, por lo que la conversión tardará unas 13 veces este valor, esto es unos 130us.

En lo que sigue es necesario tener la hoja de datos del microcontrolador que se esté utilizando, lo que se comenta mas abajo es para el caso del 2560, la forma en que se procede puede variar un poco en comparación con el convertidor analógico digital AVR de otros microcontroladores AVR, pero en general la manera de proceder es muy similar, pero eso si es muy importante siempre guiarse con la hoja de datos.

Registros:

En la siguiente imagen se tiene el registro ADMUX para el control del convertidor analógico digital AVR, se pueden ver los nombres que le corresponden a cada uno de sus bits.

25.8.1. ADC Multiplexer Selection Register

Name: ADMUX
Offset: 0x7C
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Los bits 7 y 6 de este registro son para elegir la tensión de referencia Vref de la cual dependerá la resolución para el módulo convertidor analógico digital AVR, la combinación de bits para elegir la tensión de referencia será hace según la siguiente tabla.

Table 25-3. ADC Voltage Reference Selection

REFS[1:0]	Voltage Reference Selection
00	AREF, Internal V_{ref} turned off
01	AV_{CC} with external capacitor at AREF pin
10	Internal 1.1V Voltage Reference with external capacitor at AREF pin
11	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Si se elige la combinación 00, se le indica al AVR que se utilizará una tensión de referencia externa diferente a la que utiliza el AVR para su funcionamiento, esta tensión será ingresada por el pin AREF de microcontrolador AVR.

Si se elige la combinación 01, se le indica al AVR que se utilizará la tensión a la cual funciona el microcontrolador AVR, esta tensión será ingresada por el pin AV_{CC} del microcontrolador AVR, además recomienda colocar un condensador entre este pin y tierra para evitar ruidos.

La combinación 10 se le indica al AVR que se utilizará como tensión de referencia una tensión interna de 1.1V del microcontrolador AVR, en este caso recomienda colocar un condensador entre el pin AREF y tierra para evitar ruidos.

Si se elige la combinación 11, se le indica al AVR que se utilizará como tensión de referencia una tensión interna de 2.56V del microcontrolador AVR, en este caso recomienda colocar un condensador entre el pin AREF y tierra para evitar ruidos.

El bit 5 es para elegir la justificación del número de 10 bits que representa la tensión analógica convertida a digital, dentro de los registros ADCH y ADCL que juntos hacen 16 bits, de esos 16 bits solo serán útiles 10 bits, los 6 bits restantes se ponen automáticamente a 0, se puede elegir entre una justificación a la derecha o una justificación a la izquierda

000000xxxxxxxx es el número de 16 bits almacenado entre ADCH y ADCL, donde las 10 x que pueden ser 0 o 1 es el número de 10 bits que representa a la señal analógica, este es el caso la justificación a la derecha y para ello hay que poner el bit 5 ADLAR a 0, los 6 bits mas significativos se ponen a 0 automáticamente.

xxxxxxxxxx000000 es el número de 16 bits almacenado entre ADCH y ADCL, donde las 10 x que pueden ser 0 o 1 es el número de 10 bits que representa la señal analógica, este es el caso la justificación a la izquierda y para ello hay que poner el bit 5 ADLAR a 1 los 6 bits menos significativos se ponen a 0 automáticamente.

Los bits 4, 3, 2, 1 y 0 son para elegir la entrada analógica a utilizar, el ATMEGA88 tiene 6 pines que se pueden utilizar como entradas analógicas, los cuales se encuentran en el puerto C y son llamados ADC0, ADC1, ADC2, ADC3, ADC4, ADC5, la elección de la entrada analógica a se hace de acuerdo a los valores de estos bits como se muestra en la siguiente tabla, por ejemplo, si se quiere elegir la entrada analógica 3 se utilizará la combinación para estos bits de 00011.

Table 25-4. Input Channel and Gain Selections

MUX[4:0]	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			

01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000	N/A	ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11110	1.1V (V_{BG})	N/A		
11111	0V (GND)			

En la siguiente imagen se tiene el registro ADCSRA para el control del convertidor analógico digital AVR, se pueden ver los nombres que le corresponden a cada uno de sus bits.

25.8.2. ADC Control and Status Register A

Name: ADCSRA
Offset: 0x7A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

El bit 7 ADEN es para activar o desactivar el convertidor analógico digital AVR, cuando este bit es puesto a 1 el convertidor está activo y listo para usarse, si este bit es puesto a 0 el convertidor estará apagado no pudiendo utilizarse.

El bit 6 ADSC se pondrá a 1 para iniciar la conversión analógica digital AVR, cuando la conversión de analógico a digital termina este bit se pone a 0 en forma automática, lo que indica que la conversión a terminado.

El bit 5 ADATE al poner este bit a 1 se habilita la opción de disparo o activación automática del convertidor analógico digital AVR, por alguna señal que será elegida de acuerdo al registro **ADCSRB** que se verá mas adelante.

El bit 4 ADIF este bit es el que detecta cuando se ha producido una interrupción por terminación de una conversión analógica digital AVR, para detectar la interrupción hay que ponerlo a 0, al terminar una conversión se pondrá automáticamente a 1, para seguir detectando más interrupciones luego de que haya ocurrido alguna hay que ponerlo nuevamente a 0, al utilizar la rutina de interrupciones en C/C++ con el atmel studio este bit se pone automáticamente a 0 dentro de la rutina de interrupciones.

El bit 3 ADIE este bit es utilizado para habilitar la interrupción por la terminación de una conversión analógica digital AVR, para habilitar esta interrupción hay que poner a 1 este bit.

Los bits 2, 1 y 0 son para elegir el prescaler a utilizar para obtener la frecuencia de trabajo del convertidor analógico digital AVR, mediante la combinación de valores de estos bits, a partir de la frecuencia de trabajo del microcontrolador AVR utilizado, los prescaler que se pueden utilizar son los que se indican es la siguiente tabla para el ATmega88, los valores de la frecuencia de trabajo para

el convertidor analógico digital AVR tiene que estar comprendido entre los 50KHz y los 200Khz según la hoja de datos

Table 25-5. Input Channel Selection

ADPS[2:0]	Division Factor
000	2
001	2
010	4
011	8
100	16
101	32
110	64
111	128

En la siguiente imagen se tiene el registro ADCSRB para el control del convertidor analógico digital AVR, se pueden ver los nombres que le corresponden a cada uno de sus bits.

25.8.5. ADC Control and Status Register B

Name: ADCSRB
Offset: 0x7B
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		ACME				ADTS2	ADTS1	ADTS0
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Los bits 7, 5, 4 y 3 no son utilizados por lo que se les suele poner a 0.

El bit 6 trabaja con el comparador analógico AVR, en el caso de utilizar solo el convertidor analógico digital AVR no influye por lo que se le puede poner a 0.

Los bits 2, 1 y 0 dependen del bit 5 del registro ADCSRA al poner este bit a 1 se habilita la opción de disparo o activación automática del convertidor analógico digital AVR si se pone a 0 no pasará nada, mediante las combinaciones de los bits 2, 1 y 0 del registro ADCRB, se elige cual será el evento que haga que el convertidor analógico digital AVR se habilite automáticamente, en la siguiente tabla se tiene el evento a elegir según la combinación valores de estos bits, se tienen 8 posibilidades para el disparo automático.

Si la combinación es 000 y se ha puesto a 1 el bit 5 del registro ADCSRA el convertidor analógico digital AVR se habilitará automáticamente tras cada conversión, si la combinación es por ejemplo 110 el convertidor analógico digital AVR se disparará cada vez que el timer1 se desborde.

Table 25-6. ADC Auto Trigger Source Selection

ADTS[2:0]	Trigger Source
000	Free Running mode
001	Analog Comparator
010	External Interrupt Request 0
011	Timer/Counter0 Compare Match A
100	Timer/Counter0 Overflow
101	Timer/Counter1 Compare Match B
110	Timer/Counter1 Overflow
111	Timer/Counter1 Capture Event

25.8.6. Digital Input Disable Register 0

When the respective bits are written to logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC7...0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Name: DIDR0

Offset: 0x7E

Reset: 0x00

Property: -

Bit	7	6	5	4	3	2	1	0
	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ADC0D, ADC1D, ADC2D, ADC3D, ADC4D, ADC5D, ADC6D, ADC7D: ADC Digital Input Disable

Del bit 7 al bit 0 mediante estos bits al ponerlos a 1 se permite que el pin utilizado para la entrada analógica solo funcione con el módulo convertidor analógico digital AVR, y la parte interna del pin que tiene que ver con su comportamiento digital no se vea afectado, se puede decir que es una forma de deshabilitar el comportamiento digital del pin utilizado como entrada analógica.

Código Morse:

El código morse, también conocido como alfabeto morse o clave morse, es un sistema de representación de letras y números mediante señales emitidas de forma intermitente.

La duración del punto es la mínima posible. Una raya tiene una duración de aproximadamente tres veces la del punto. Entre cada par de símbolos de una misma letra existe una ausencia de señal con duración aproximada a la de un punto. Entre las letras de una misma palabra, la ausencia es de aproximadamente tres puntos. Para la separación de palabras transmitidas el tiempo es de aproximadamente tres veces el de la raya.

Toda correspondencia entre dos estaciones deberá comenzar con la señal de llamada. Para llamar, la estación que llama transmitirá el distintivo de llamada (no más de dos veces) de la estación requerida, la palabra DE seguida por su propia señal de llamada y la señal -. - a menos que haya reglas especiales peculiares al tipo de aparato utilizado.

Signo	Código	Signo	Código	Signo	Código
A	· —	N	— ·	0	— — — — —
B	— · · ·	Ñ	— — · — —	1	· — — — —
C	— · — ·	O	— — —	2	· · — — —
CH	— — — —	P	· — — ·	3	· · · — —
D	— · ·	Q	— — · —	4	· · · · —
E	·	R	· — ·	5	· · · · ·
F	· — — ·	S	· · ·	6	— · · · ·
G	— — ·	T	—	7	— — — · ·
H	· · · ·	U	· · —	8	— — — · ·
I	· ·	V	· · · —	9	— — — — ·
J	· — — —	W	· — —	,	· — · — —
K	— · —	X	— · · —	,	— · — · — —
L	· — · ·	Y	— — — —	?	· · — — —
M	— —	Z	— — · ·	"	· — · · ·
				!	— — · · — —

Conclusión:

Al realizar la practica aprendí correctamente el funcionamiento del ADC del microcontrolador ATMEGA 2560, gracias a ello pude conocer el funcionamiento de los registros de control del ADC y como configurarlos según la aplicación que se necesite además de también sincronizarlo con el timer0 para realizar otras tareas. En este caso lo aplique para leer cambios de voltaje con la foto resistor para que se vieran reflejados y conforme a los cambios del ADC medir el tiempo en alto o en bajo según el mensaje que se estaba transmitiendo.