



Materia: Microprocesadores y microcontroladores

Maestro: García López Jesús Adán

Alumno: Christian Alfredo Montero Martínez

Practica 11: Uso del Convertidor Analógico Digital del ATmega1280

Fecha: 19/05/2017

## Práctica No. 11

### Uso del Convertidor Analógico Digital del ATmega1280

Objetivo: Mediante esta práctica el alumno aprenderá la programación y uso básico del convertidor analógico digital del microcontrolador ATmega1280.

#### Equipo:

- Computadora Personal
- Módulo T-Juino

#### Teoría

Programación y uso del ADC (Diagrama, Funcionamiento, registros de configuración y operación).

#### Algunas de sus características

- 10 bits de resolución.
- 13 a 260uS de tiempo de conversión.
- Hasta 15000 muestras por segundo en su máxima resolución.
- 8 canales de entrada multiplexados a una sola terminación.
- 7 canales de entradas diferenciales.
- Rango de voltaje de entrada de 0 a Vcc.
- Voltaje de referencia seleccionable a 2.56V.

El Convertidor Analógico Digital del ATmega32 es por Aproximaciones Sucesivas con una resolución de 10 bits. El ADC se conecta a un multiplexor de 8 canales análogos el cual permite 8 voltajes de entrada en una sola terminación construido en los pines del puerto A. El voltaje de entrada de una sola terminación se refiere a 0 V (GND).

El dispositivo también soporta 16 combinaciones de voltajes de entrada diferenciales. Dos de las entradas diferenciales (ADC1, ADC0 y ADC3, ADC2) están equipadas con una etapa de ganancia programable, proveyendo pasos de amplificación de 0dB (1x), 20dB (10x), o 46dB (200x) en el voltaje de entrada diferencial antes de la conversión del A/D. Siete canales de entrada diferenciales análogos comparten una terminal común negativa (ADC1), mientras que cualquier otra entrada del ADC puede ser seleccionada como la terminal positiva de entrada. Si se usa una ganancia de 1x o 10x, una resolución de 8 bits se espera. Si se usa una ganancia de 200x, una resolución de 7 bits se espera.

El ADC contiene un circuito de Muestreo y Retención el cual asegura que el voltaje de entrada al ADC se mantenga constante durante la conversión. El diagrama a bloques se muestra en la siguiente figura.

## Diagrama de bloques

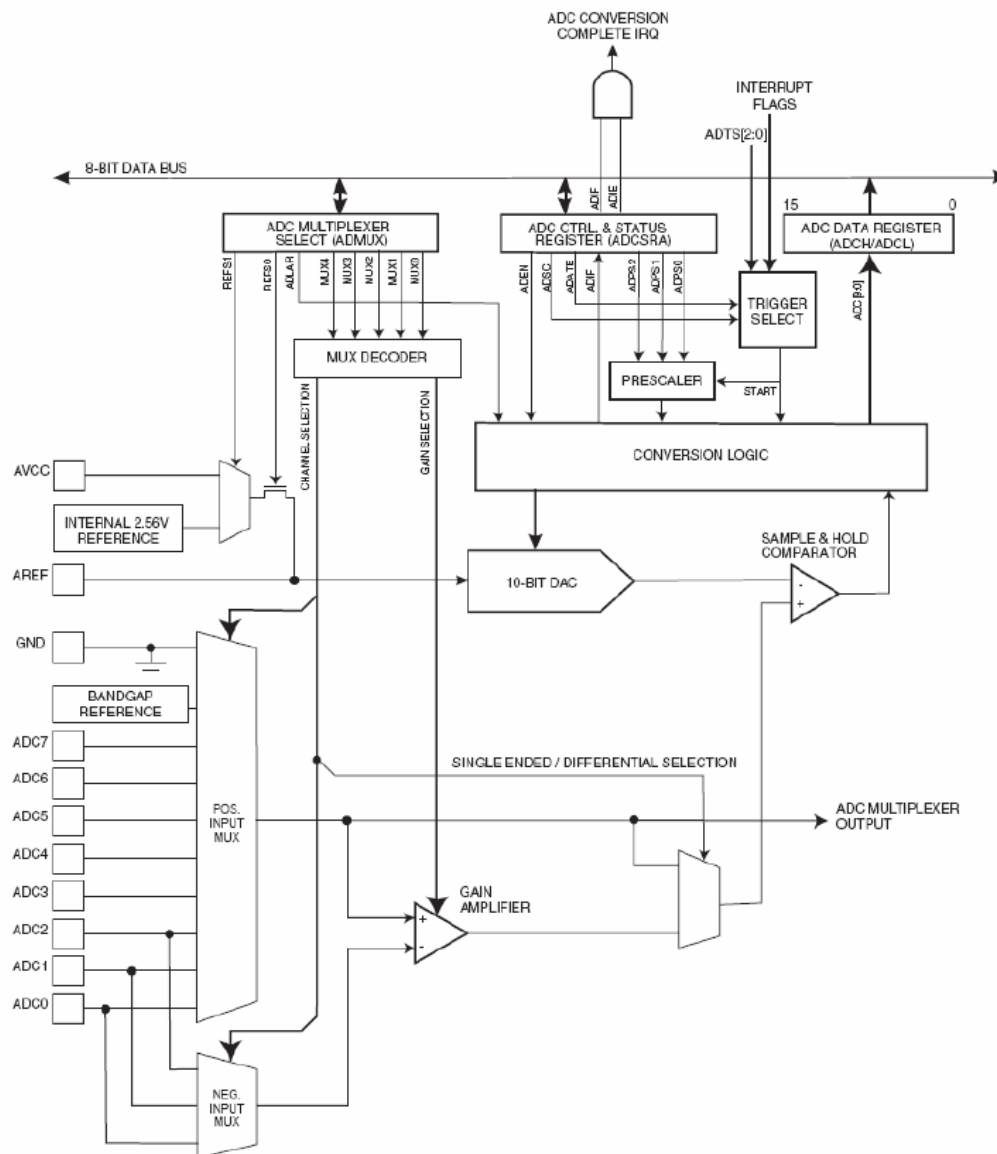


Figura No.1: Diagrama de bloques del ADC

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 más 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. El ADC convertidor analógico digital AVR necesita una tensión de referencia para poder trabajar adecuadamente, esta tensión de referencia Vref 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 Vref y el máximo número binario de 10 bits  $(2^{10} - 1) = 1023 = 11\ 1111\ 1111$  que representará la señal analógica se le conoce como resolución.

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

Si se toma como la Vref = 5V que es la tensión adecuada a la que trabaja el microcontrolador se tendrá que la resolución de su convertidor analógico digital AVR será:

$$Resolución = \frac{5V}{2^{10} - 1} \approx 4.9mV$$

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:

Voltaje	Representación binaria
0	00 0000 0000
4.9 mV	00 0000 0001
9.8 mV	00 0000 0010
14.7 mV	00 0000 0011
.	.
.	.
.	.
5V	11 1111 1111

Tabla No.1: Ejemplos voltajes en su representación binaria.

Como el módulo convertidor analógico digital AVR es de 10bits, 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:

	ADCH	ADCL
Justificación a la Derecha	0000 00XX	XXXX XXXX
Justificación a la Izquierda	XXXX XXXX	XX00 0000

Tabla No.2: Configuraciones de justificación del ADC.

El número de 16 bits almacenado entre ADCH y ADCL, donde las 10's X pueden ser 0 o 1 es el número de 10 bits que representa la señal analógica, para el caso de la justificación a la derecha, los 6 bits más significativos se ponen a 0 automáticamente, mientras que en 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 dato_analogico;

si la justificación es a la derecha
dato_analogico = ADC;

si la justificación es a la izquierda
dato_analogico = ADC>>6;
```

Figura No.2: Ejemplo de lectura del valor de ADC.

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

$$V_{volts} = \frac{dato_{analógico} * V_{ref}}{1023}$$

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

Para la utilización del convertidor analógico digital AVR del microcontrolador 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, junto con la función sei().

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 microcontrolador, si la frecuencia de trabajo del ADC es la mayor la conversión será más 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 más tiempo, se recomienda por tanto utilizar una frecuencia de trabajo del ADC intermedia por ejemplo de 100KHz.

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.

### Tiempo de conversión

Una conversión normal toma 13 ciclos de reloj del ADC. La primera conversión después de que el ADC se enciende (ADEN en ADCSRA se ponga a uno) toma 25 ciclos de reloj del ADC para inicializar la circuitería analógica. El muestreo y retención actual toma lugar 1.5 ciclos de reloj del ADC después del inicio de una conversión normal y 13.5 ciclos de reloj del ADC después del comienzo de la primera conversión. Cuando una conversión se completa, el resultado se escribe en el registro de datos del ADC, y ADIF se activa. En el modo de conversión única, ADSC se limpia simultáneamente. El software puede poner a uno ADSC de nuevo, y una nueva conversión será iniciada en la primera transición positiva del ciclo de reloj del ADC.

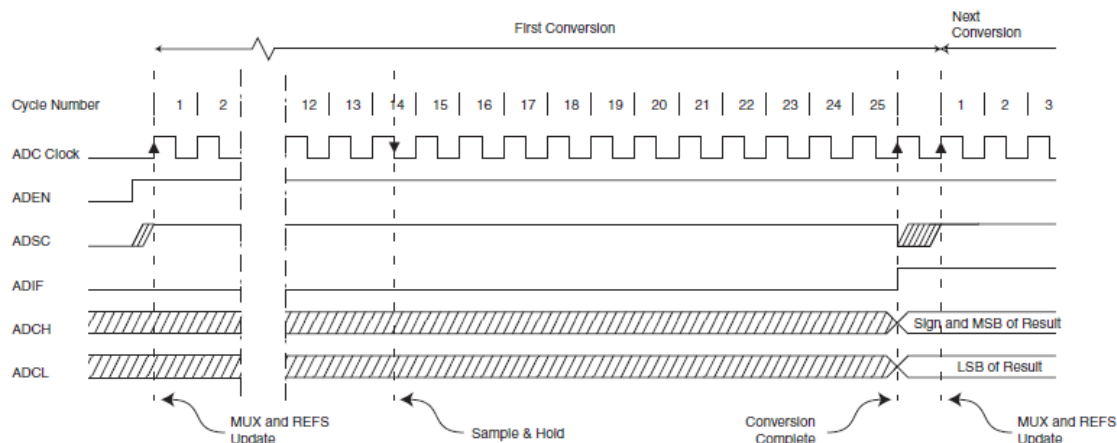


Figura No.3: Diagrama de tiempos de la primera conversi3n del ADC.

## Registro ADMUX

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura No.4: Registro ADMUX.

Los bits 7 y 6 de este registro son para elegir la tensi3n de referencia  $V_{ref}$  de la cual depender3 la resoluci3n para el m3dulo convertidor anal3gico digital AVR, la combinaci3n de bits para elegir la tensi3n de referencia ser3 hecha seg3n la siguiente tabla.

REFS1	REFS0	Voltage Reference Selection <sup>(1)</sup>
0	0	AREF, Internal $V_{REF}$ turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Internal 1.1V Voltage Reference with external capacitor at AREF pin
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Figura No.5: Configuraciones del voltaje de referencia.

El bit 5 es para elegir la justificaci3n del n3mero de 10 bits que representa la tensi3n anal3gica convertida a digital, dentro de los registros ADCH y ADCL que juntos hacen 16 bits, de esos 16 bits solo ser3n 3tiles 10 bits, los 6 bits restantes se ponen autom3ticamente a 0, se puede elegir entre una justificaci3n a la derecha o una

justificación a la izquierda, para una justificación a la derecha es necesario poner el bit ADLAR a 0, mientras que para una justificación a la izquierda hay que poner el bit ADLAR a 1.

Los bits 4, 3, 2, 1 y 0: El valor de estos bits seleccionan la combinación de entradas analógicas que están conectadas al ADC, si estos bits cambian durante la conversión, el cambio no tendrá efecto hasta que la conversión actual termine.

MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
000000	ADC0	N/A		
000001	ADC1			
000010	ADC2			
000011	ADC3			
000100	ADC4			
000101	ADC5			
000110	ADC6			
000111	ADC7			
001000 <sup>(1)</sup>	N/A	ADC0	ADC0	10×
001001 <sup>(1)</sup>		ADC1	ADC0	10×
001010 <sup>(1)</sup>		ADC0	ADC0	200×
001011 <sup>(1)</sup>		ADC1	ADC0	200×
001100 <sup>(1)</sup>		ADC2	ADC2	10×
001101 <sup>(1)</sup>		ADC3	ADC2	10×
001110 <sup>(1)</sup>		ADC2	ADC2	200×
001111 <sup>(1)</sup>		ADC3	ADC2	200×
010000	N/A	ADC0	ADC1	1×
010001		ADC1	ADC1	1×
010010		ADC2	ADC1	1×
010011		ADC3	ADC1	1×
010100		ADC4	ADC1	1×
010101		ADC5	ADC1	1×
010110		ADC6	ADC1	1×
010111		ADC7	ADC1	1×
011000	N/A	ADC0	ADC2	1×
011001		ADC1	ADC2	1×
011010		ADC2	ADC2	1×
011011		ADC3	ADC2	1×
011100		ADC4	ADC2	1×
011101		ADC5	ADC2	1×
011110	1.1V (V <sub>BG</sub> )	N/A		
011111	0V (GND)			
100000	ADC8	N/A		
100001	ADC9			
100010	ADC10			
100011	ADC11			
100100	ADC12			
100101	ADC13			
100110	ADC14			
100111	ADC15			

Figura No.6: Tabla de canales del ADC.



MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
101000 <sup>(1)</sup>	N/A	ADC8	ADC8	10×
101001 <sup>(1)</sup>		ADC9	ADC8	10×
101010 <sup>(1)</sup>		ADC8	ADC8	200×
101011 <sup>(1)</sup>		ADC9	ADC8	200×
101100 <sup>(1)</sup>		ADC10	ADC10	10×
101101 <sup>(1)</sup>		ADC11	ADC10	10×
101110 <sup>(1)</sup>		ADC10	ADC10	200×
101111 <sup>(1)</sup>		ADC11	ADC10	200×
110000		ADC8	ADC9	1×
110001		ADC9	ADC9	1×
110010		ADC10	ADC9	1×
110011		ADC11	ADC9	1×
110100		ADC12	ADC9	1×
110101		ADC13	ADC9	1×
110110		ADC14	ADC9	1×
110111		ADC15	ADC9	1×
111000		ADC8	ADC10	1×
111001		ADC9	ADC10	1×
111010		ADC10	ADC10	1×
111011		ADC11	ADC10	1×
111100		ADC12	ADC10	1×
111101	N/A	ADC13	ADC10	1×
111110	Reserved	N/A		
111111	Reserved	N/A		

Figura No.7: Continuación de la tabla de canales.

## Registro ADCSRA

Bit	7	6	5	4	3	2	1	0	
(0x7A)	<b>ADEN</b>	<b>ADSC</b>	<b>ADATE</b>	<b>ADIF</b>	<b>ADIE</b>	<b>ADPS2</b>	<b>ADPS1</b>	<b>ADPS0</b>	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura No.8: Registro ADCSRA.

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 más 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 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 preescalador 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 preescalador 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 especificaciones.

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figura No.9: Configuraciones del peescalador.

Por ejemplo si el microcontrolador AVR está trabajando a 1MHz, para obtener una frecuencia de trabajo del convertidor analógico digital de 100Khz o uno cercano que esté entre 50Khz y 200Khz, el preescalador a utilizar seria de 8, con el cual se obtiene una frecuencia de trabajo para el convertidor analógico digital AVR de 125Khz.

## Registro ADCSRB

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	MUX5	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura No.9: Registro ADCSRA.

Los bits 7, 5 y 4 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.

El bit 3 se usa en conjunto con MUX4:0 presentes en el registro ADMUX para seleccionar la combinación en la cual las entradas analógicas están conectadas al ADC, si esta combinación cambia durante la conversión el cambio no tendrá efecto hasta que la conversión actual termine.

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.

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Figura No.10: Configuración de la fuente de disparo.

## Registro DIDR0

Bit (0x7E)	7	6	5	4	3	2	1	0	
	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura No.11: Registro DIDR0.

—Bit 7 – 0: Cuando se le escribe a alguno de estos bits un 1 lógico, el buffer de la entrada digital del correspondiente ADC se deshabilita. El registro PIN al bit correspondiente siempre leerá un 0 mientras este bit tenga presente un 1 lógico. Cuando una señal analógica sea aplicada a alguno de los pines ADC0 – 7 y la entrada digital de este pin no se necesite, este bit debe ser alimentado con un 1 lógico para reducir el consumo de energía.

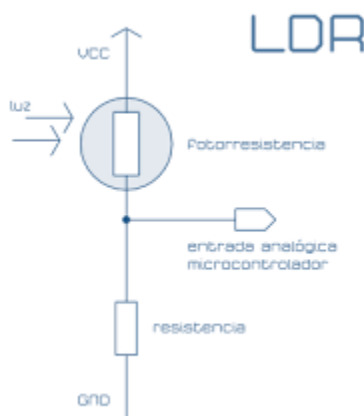
## Registro DIDR2

Bit (0x7D)	7	6	5	4	3	2	1	0	
	ADC15D	ADC14D	ADC13D	ADC12D	ADC11D	ADC10D	ADC9D	ADC8D	DIDR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura No.12: Registro DIDR0.

Bit 7 – 0: Cuando se le escribe a alguno de estos bits un 1 lógico, el buffer de la entrada digital del correspondiente ADC se deshabilita. El registro PIN al bit correspondiente siempre leerá un 0 mientras este bit tenga presente un 1 lógico. Cuando una señal analógica sea aplicada a alguno de los pines ADC15 – 8 y la entrada digital de este pin no se necesite, este bit debe ser alimentado con un 1 lógico para reducir el consumo de energía.

## Funcionalidad y características de la fotorresistencia



Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas, LDR, se originan de su nombre en inglés light-dependent resistor. Su cuerpo está formado por una célula o celda y dos patillas.

### Características

Un fotorresistor está hecho de un semiconductor de alta resistencia como el sulfuro de cadmio, CdS. Si la luz que incide en el dispositivo es de alta frecuencia, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar la banda de conducción. El electrón libre que resulta, y su hueco asociado, conducen la electricidad, de tal modo que disminuye la resistencia. Los valores del enlace típicos varían entre 1 M $\Omega$ , o más, en la oscuridad y 100  $\Omega$  con luz brillante. Las células de sulfuro del cadmio se basan en la capacidad del cadmio de variar su resistencia según la cantidad de luz que incide la célula. Cuanta más luz incide, más baja es la resistencia. Las células son también capaces de reaccionar a una amplia gama de frecuencias, incluyendo infrarrojo (IR), luz visible, y ultravioleta (UV). La variación del valor de la resistencia tiene cierto retardo, diferente si se pasa de oscuro a iluminado o de iluminado a oscuro. Esto limita a no usar los LDR en aplicaciones en las que la señal luminosa varía con rapidez. El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo. Esta lentitud da ventaja en algunas aplicaciones, ya que se filtran variaciones rápidas de iluminación que podrían hacer inestable un sensor (ej. tubo fluorescente alimentado por corriente alterna). En otras aplicaciones (saber si es de día o es de noche) la lentitud de la detección no es importante. Se fabrican en diversos tipos

y pueden encontrarse en muchos artículos de consumo, como por ejemplo en cámaras, medidores de luz, relojes con radio, alarmas de seguridad o sistemas de encendido y apagado del alumbrado de calles.

## Conclusiones y Comentarios

En el mundo real existen un sinnúmero de dispositivos interactuando con fenómenos físicos que nos proporcionan información de manera analógica, los microcontroladores por su parte manejan información digital, dado que en la mayoría de las aplicaciones del mundo real es necesaria la implementación de los dos tipos de dispositivos, el acoplamiento de dispositivos analógicos y digitales es un conocimiento necesario para el ingeniero en computación, en esta premisa radica la importancia del conocimiento del ADC, su correcta manipulación e implementación. Su implementación es relativamente sencilla apoyándonos de la hoja de especificaciones y nos permite elaborar en base a las capacidades del microcontrolador casi cualquier aplicación que requiera del acoplamiento de dispositivos analógicos y digitales.

## Bibliografía

[1] Atmel, “8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash” Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V datasheet, 2014 [Revisado: mayo 2017].

[2] EcuRed. (2011). Fotorresistores. [Revisado: mayo 2017], de EcuRed Sitio web: <https://www.ecured.cu/Fotorresistencia>

[3] (2012). Convertidor Analógico Digital AVR. [Revisado: mayo 2017], de . Sitio web: <http://microcontroladores-mrelberni.com/convertidor-analogico-digital-avr/>