



Departamento de Electrónica
Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°5: Uso del ADC

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1º/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Autor			Seguimiento del proyecto									
Maximiliano	Porta	98800										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

21 de agosto de 2020

Índice

1. Objetivo	2
2. Descripción	2
2.1. Introducción	2
2.2. Funcionamiento del ADC	3
2.3. Diagrama de Conexión del Circuito Físico	4
2.4. Esquema en Bloques	5
2.5. Materiales Utilizados	5
2.6. Diagrama de Flujo	6
2.7. Lógica del Programa	6
2.8. Links de funcionamiento del circuito	6
3. Código	6
4. Resultados y Conclusiones	7
4.1. Resultados	7
4.2. Conclusiones	8

1. Objetivo

Consiste en aprender el funcionamiento del conversor analógico digital que tiene incluido el Atmega328p, como poder manipularlo para que al conectarle un potenciómetro a cualquiera de los terminales que tienen el ADC, que al variar su resistencia, que se traduce a una tensión para el terminal de dicho microcontrolador observar los números del 0 al 63 en binario.

2. Descripción

2.1. Introducción

Para el desarrollo de este trabajo de laboratorio se utilizo el diagrama de la Figura 1. Se observo en la hoja de datos del microcontrolador atmega328p la corriente máxima de entrada/salida ($I = 20mA$), usando las leyes de ohm se obtiene la ecuación 1, con una Resistencia de 220Ω y la tensión para un diodo led verde, se obtuvo que la corriente de salida total es $I_{Out} = 10,9mA$, por lo cual es menor a la máxima permitida por pin del arduino. Mediante la programación se busca poder controlar mediante un potenciómetro variar los números de 0 a 63 en binario, usando la entrada analógica del microcontrolador.

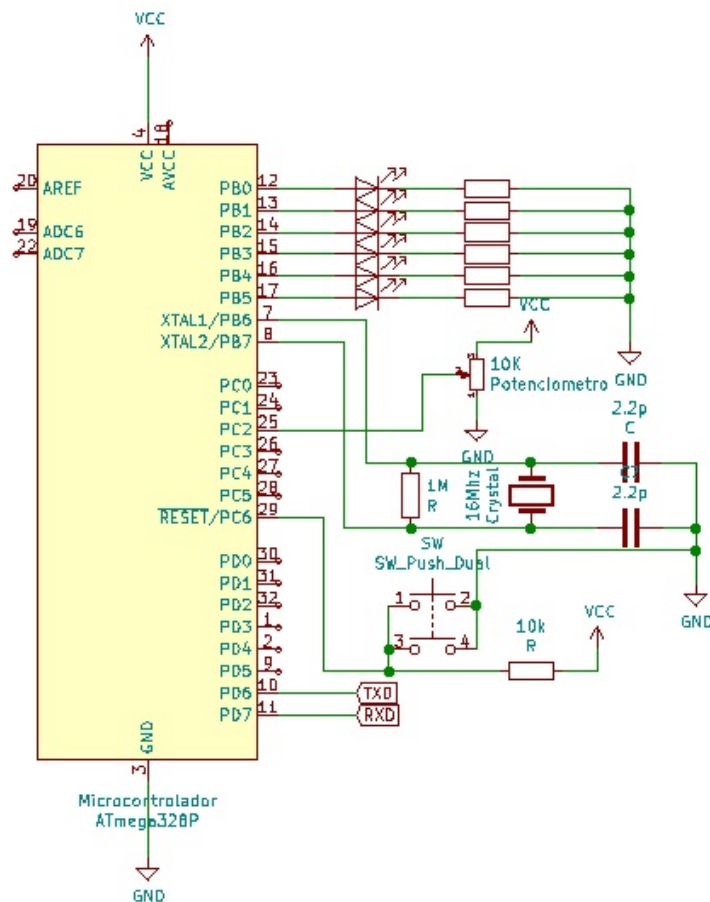


Figura 1: Diagrama de conexión

$$I_{Led} = \frac{V_{CC} - V_{Led}}{R_{Led}} \quad (1)$$

2.2. Funcionamiento del ADC

El ADC recibe una muestra de una señal analógica, su valor en un instante de tiempo, a partir de la cual genera un número (valor digital). Esto se puede hacer mediante interrupciones, cuando el microcontrolador ve que hay un cambio entra a un programa secundario y realiza alguna operación, luego retorna al programa principal. La función de un ADC es contraria a la que realiza un convertidor digital a analógico (DAC, digital-to-analog converter), un DAC genera un nivel de voltaje analógico a partir de un número o valor digital. En la figura 2 se ilustra la función de un ADC y un DAC (no viene incluido en un AVR).

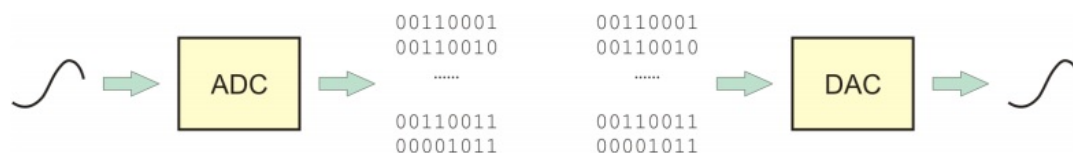


Figura 2: Funcionalidad de los conversores

Una señal analógica toma valores continuos a lo largo del tiempo, el proceso para convertirla a digital involucra 2 etapas: el muestreo y la cuantificación.

El muestreo consiste en tomar el valor de la señal analógica en un instante de tiempo. Cada muestra se va tomando cuando ocurre un intervalo de tiempo predefinido, conocido como periodo de muestreo, cuyo inverso es la frecuencia de muestreo. Para que la información contenida en la señal a digitalizar sea recuperada de manera correcta, se requiere que la frecuencia de muestreo sea por lo menos el doble de la frecuencia de la señal analógica. Por ejemplo, si se va a digitalizar una señal de audio acotada a 5 KHz, la frecuencia de muestreo por lo menos debe ser de 10 KHz.

La cuantificación consiste en la asociación de cada muestra con un número, que pertenece a un conjunto finito de valores, el número de bits utilizado por el convertidor determina el total de valores. Cada muestra analógica se asocia con el valor digital más cercano. Por ejemplo, si el convertidor es de 8 bits, el conjunto tiene 256 valores diferentes (0 a 255).

$$\text{Resolución} = \frac{V_{MAX}}{2^n - 1}$$

Según la resolución se puede observar en la siguiente figura como se realizara la toma de muestras de la señal analógica y como se cuantificara digitalmente.

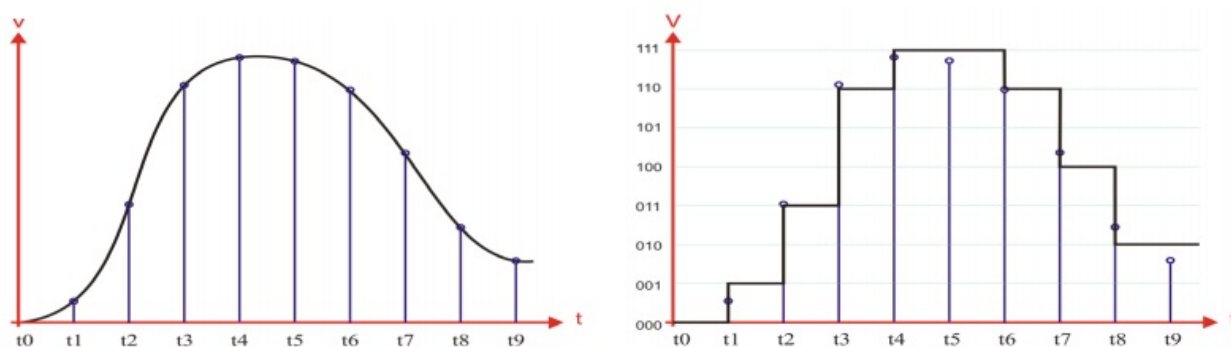


Figura 3: Resolución de ejemplo de una muestra analógica

El dato digital de 10 bits, resultante de la conversión, queda disponible en 2 Registros I/O: ADCH (para la parte alta) y ADCL (para la parte baja). La alineación puede cambiarse a la izquierda si se pone en alto al bit ADLAR (ADC Left Adjust Result) del registro ADMUX. Esta organización es útil en aplicaciones donde resulte suficiente una resolución de 8 bits o menos. En esos casos, sólo se emplearía al registro ADCH y se ignora el contenido del registro ADCL. En el registro ADMUX (multiplexor del ADC) también se pueden seleccionar otros parámetros, sus bits son:

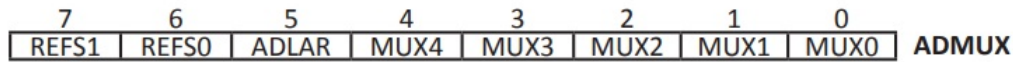


Figura 4: Registro ADMUX

Los Bits 7 y 6 Bits se utilizan para seleccionar el voltaje de referencia del ADC, en el atmega328p puedo seleccionar hasta 2.56 V(interno) o usar directamente Vcc. El bit 5 es el que se utiliza para acomodar a izquierda el resultado. Los bits del 0 al 4 se utilizan para elegir el canal deseado.

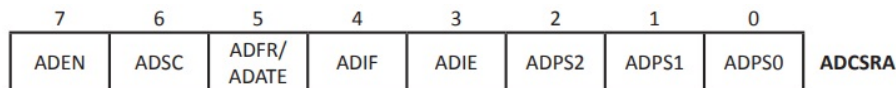


Figura 5: Registro ADCSRA

El bit 7 activa el ADC, el bit 6 se utiliza para iniciar la conversión al escribirle un 1, el bit 5 la función de este bit depende del dispositivo, establece un modo de carrera libre, es decir, al terminar una conversión inicia con la siguiente. El bit ADSC debe ponerse en alto para dar paso a las conversiones. El bit 4 es la bandera de fin de conversión para que funcione, hay que activar el bit 3 que activa el flag de fin de conversión, el flag se limpia automáticamente por hardware y los bits del 0 al 2 se utiliza para seleccionar el preescalador, en caso de trabajar con una plataforma de arduino Uno se debe configurar en 128 por que este utiliza por defecto el cristal externo de 16Mhz, con esto se evita que supere la frecuencia máxima del dispositivo.

2.3. Diagrama de Conexión del Circuito Físico

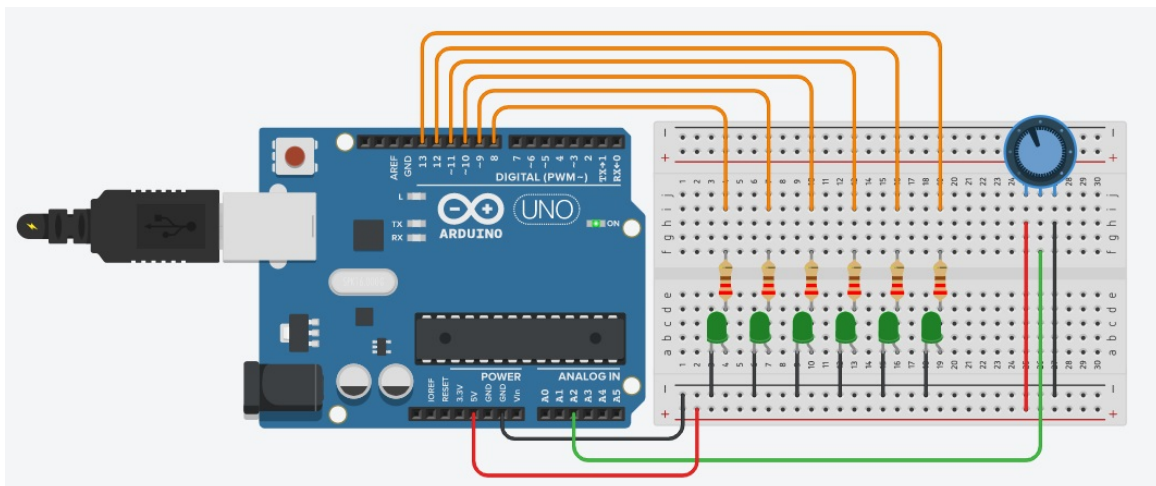


Figura 6: Diagrama del Circuito Físico

2.4. Esquema en Bloques

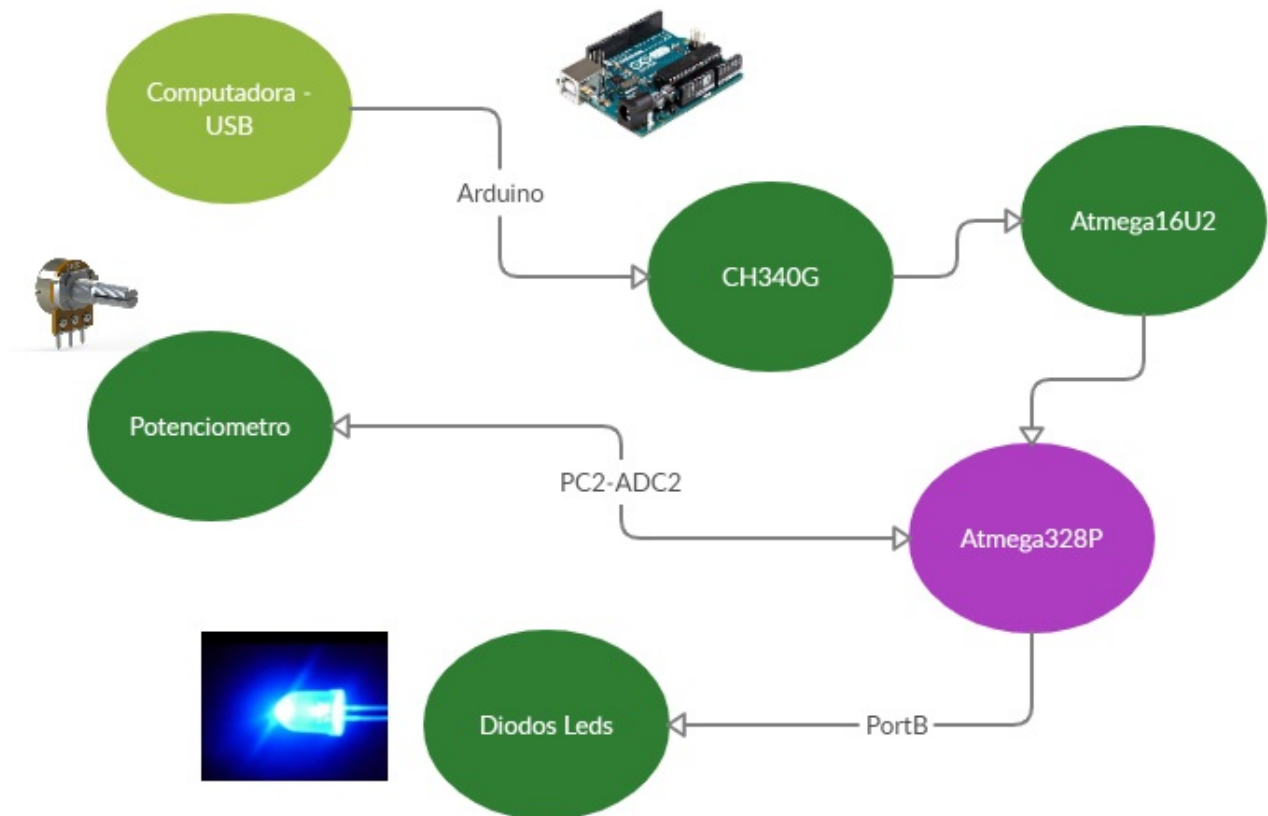


Figura 7: Diagrama en bloques

2.5. Materiales Utilizados

- x1(12.5 Ars)Potenciometro de $5K\Omega$
- x6(42 Ars)Diodo Led verde de 3mm (Alta Luminosidad).
- x6(10 Ars)Resistor de 220Ω de carbón 1/4 Watt.
- 1x(725 Ars)Placa Arduino Uno con USB.
- 1x(96.5 Ars)Placa Experimental.
- 1x(10 Ars)Cables de conexión.
- Costo del Proyecto : 895 Ars

2.6. Diagrama de Flujo

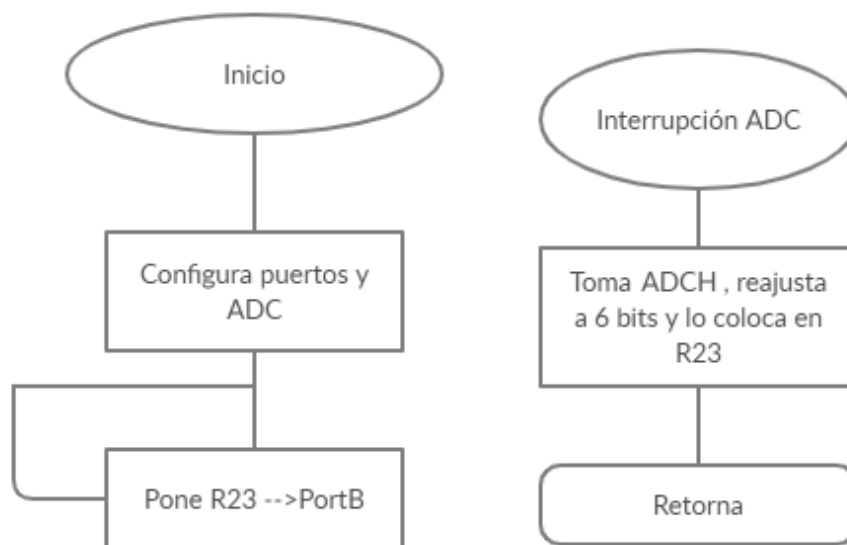


Figura 8: Diagrama de flujo

2.7. Lógica del Programa

Primero realizo todas las configuraciones de los puertos, configuro el conversor, activo las interrupciones y luego hago una ORI para que empiece a convertir. Para poder obtener los valores de 0 a 63, lo que hago es multiplicar ADCH por 63 y luego debería dividir por 255, pero si me quedo con la parte alta (R1) sería como dividir por 256, como las divisiones tienen que ser números naturales coinciden en todos los casos menos en el borde así que solo hago un reajuste en el caso de que ADCH tenga 255 y me ahorro de hacer la división sucesiva, por lo cual queda un programa mas rápido por que hace menos cuentas y mas sencillo.

2.8. Links de funcionamiento del circuito

Conversor Analógico-Digital: <https://youtu.be/LdAo1wNdlr0>

3. Código

```

.INCLUDE "m328pdef.inc"
.CSEG
    RJMP MAIN

.ORG ADCCaddr                ;0x002A
    RJMP INT_ADC

.ORG INT_VECTORS_SIZE

MAIN:
    LDI R22, 0xFF
    OUT DDRB, R22            ;todo el puerto B como salida
    LDI R22, 0x00
  
```

```

OUT DDRC, R22                                ; el puerto c como entrada (adc2)

LDI R17, 0X3F                                ; constante que multiplica

LDI R22, 0xAF
; ADCSRA-> ADC_Activado, Por_Interrupcion, Prescale 128(16Mhz)
STS ADCSRA, R22
LDI R22, 0x62
; ADMUX-> Vreferencia=Vcc, Corrimiento a izquierda, ADC2
STS ADMUX, R22
LDI R22, 0x00
STS ADCSRB, R22 ; free running

SEI                                           ; activo las interrupciones

LDS R22, ADCSRA
; escribo un 1 en adsc y arranca la conversion
ORI R22, 0x40
STS ADCSRA, R22

LDS R23, 0

LOOP:
; loop donde queda el programa esperando interrupciones
OUT PORTB, R23
RJMP LOOP

INT_ADC:

    LDS R23, ADCH
    CPI R23, 255
    BREQ ES_255
    MUL R23, R17
    MOV R23, R1
    RETI

ES_255:
    LDI R23, 0x3F
    RETI

```

4. Resultados y Conclusiones

4.1. Resultados

Se logro controlar mediante un potenciómetro la entrada analógica del microcontrolador y mediante un reescalado poder obtener a la salida los números binarios del 0 al 63.

4.2. Conclusiones

Aprendí a utilizar el conversor analógico digital que tiene el microcontrolador atmega328p, también a reescalar en caso de que quiera utilizar menos de 10bits en la salida, aprendí que hay que configurar todo y después activar el ADC.