



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico 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									
Docentes guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Fecha de presentación:			22/07/2020									
Alumno			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Maximiliano Daniel	Reigada	100565										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P

Índice

1. Objetivo	1
2. Descripción	1
3. Diagrama de conexiones en bloques	1
4. Circuito esquemático	2
5. Listado de componentes	2
6. Diagrama de flujo del Software	3
7. Código de programa	4
8. Resultados	5
9. Conclusiones	5

1. Objetivo

El presente trabajo práctico tiene como objetivo progresar con el manejo del conversor analógico digital del que dispone el microcontrolador ATmega328P, controlando el encendido de LEDs dispuestos en un puerto configurado como de salida, a partir de la posición en que se encuentra un potenciómetro conectado en un pin analógico de entrada.

2. Descripción

Mediante la implementación de una placa de microcontrolador de código abierto basado en el microchip ATmega328P, denominada Arduino Uno, se busca desarrollar un programa con el que visualizar a través de 6 LEDs conectados al puerto B, el valor en binario tomado del ADC contactado a un potenciómetro. El ADC se trabaja a manera de conversión simple y se toman solo 8 de los 10 bits, de los cuales se ajusta para que la salida obtenga valores que van de 0 a 63, y sea representada por medio de LEDs conectados al microcontrolador.

Para tomar los 8 de los 10 bits disponibles, se procede a setear en 1 el bit ADLAR del registro ADMUX para así obtener en el registro ADCH los 8 bits más significativos del resultado de la conversión (esto equivale a dividir el resultado total de la conversión por 4). Por otra parte, para expresar este resultado por medio de los 6 LEDs se divide al valor leído de ADCH por 4, desplazándolo dos bits a derecha.

3. Diagrama de conexiones en bloques

A continuación, puede verse el diagrama de conexiones en bloques correspondiente a este trabajo práctico:

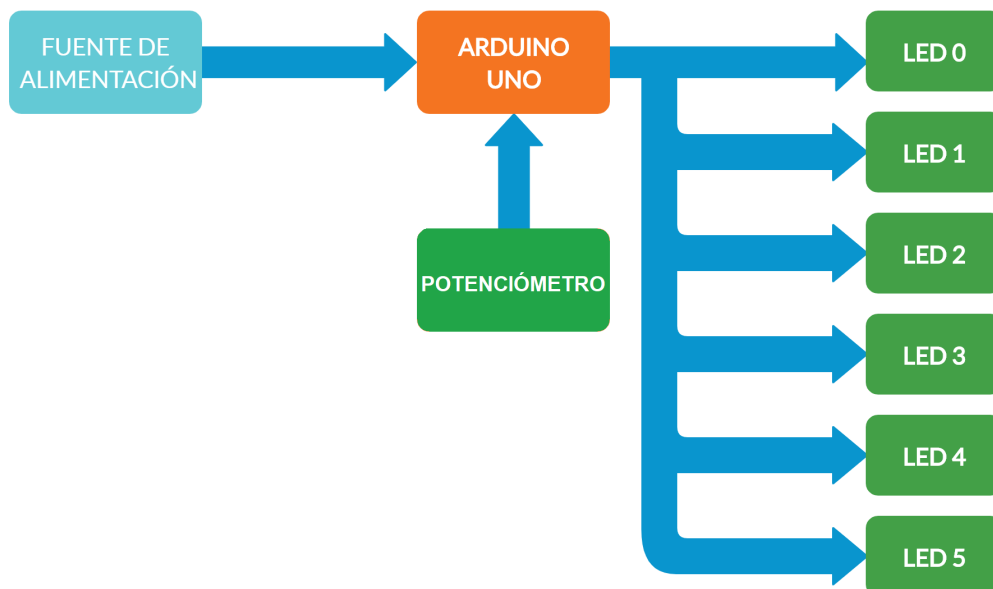


Figura 1: Diagrama de conexiones en bloques.

4. Circuito esquemático

Se procede a realizar el esquemático del circuito a implementar. Cabe destacar que, si bien en la realidad los componentes se encuentran conectado al Arduino Uno y este a su vez a una fuente de tensión externa, resulta más útil representar en el esquema solo los componentes electrónicos esenciales del sistema, conectados a referencias de tensión de un determinado valor:

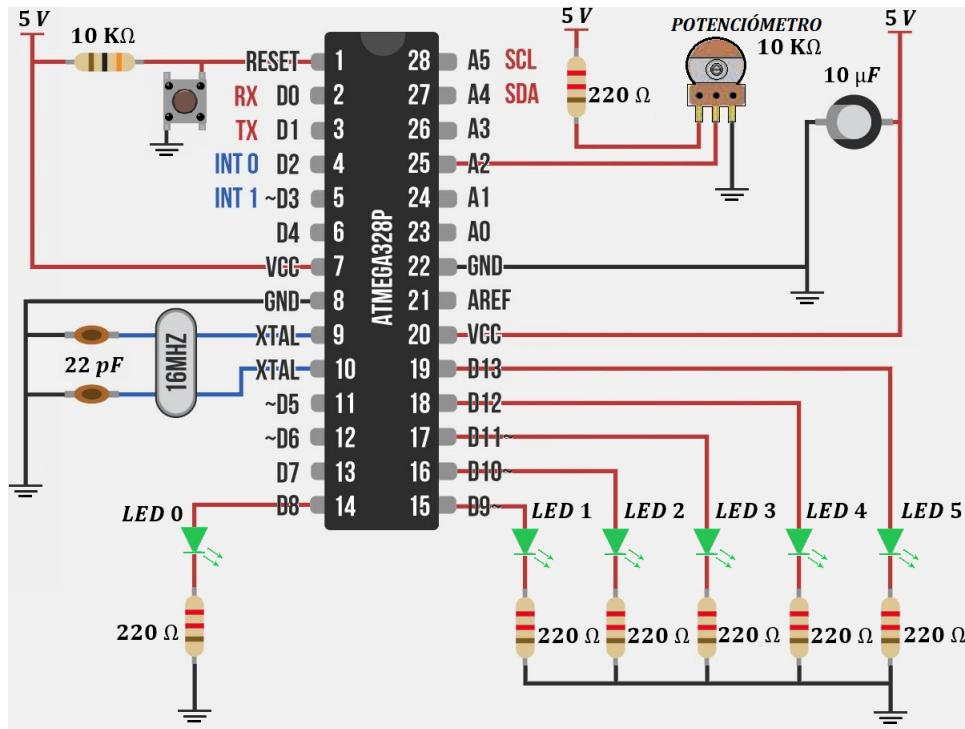


Figura 2: Circuito esquemático.

5. Listado de componentes

En el desarrollo de este trabajo práctico se utilizan los siguientes componentes:

- 6 LED verde de 5mm
- 7 Resistores de 220 Ω
- 1 Potenciómetro de 10 kΩ
- 10 Cables de conexión
- 1 Protoboard
- 1 Arduino Uno (con microcontrolador ATmega328P incluido)

6. Diagrama de flujo del Software

A continuación, se explicitan los diagramas de flujo correspondientes al código desarrollado:

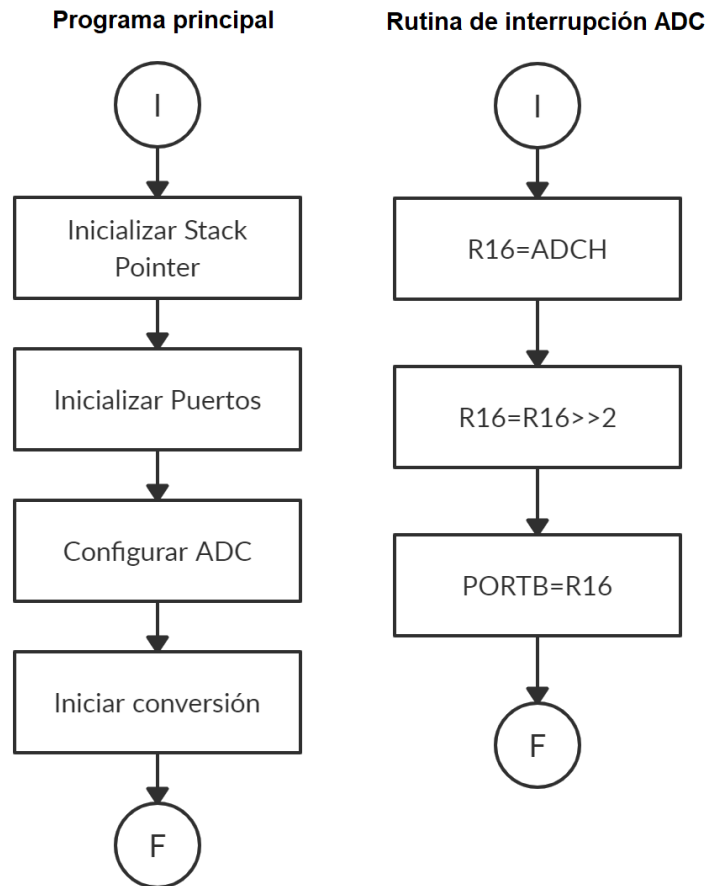


Figura 3: Diagramas de flujo.

7. Código de programa

A continuación, puede verse el código desarrollado para este proyecto:

```

1  ;*****
2  ; Código correspondiente al ejercicio planteado en el Trabajo Práctico 5.
3  ;
4  ; Alumno: Reigada Maximiliano Daniel
5  ; Padrón: 100565
6  ;*****
7
8  .INCLUDE "m328pdef.inc"
9
10 .DEF AUX=R16
11
12 .CSEG
13 .ORG 0X0000
14     RJMP    config
15
16 .ORG ADCSRA
17     RJMP    isr_adc
18
19 .ORG INT_VECTORS_SIZE
20
21 config:
22     LDI      AUX, HIGH(RAMEND)    ;Inicializo el SP al final de la RAM.
23     OUT      SPH, AUX
24     LDI      AUX, LOW(RAMEND)
25     OUT      SPL, AUX
26
27     LDI      AUX, 0X00            ;Declaro al puerto C como entrada.
28     OUT      DDRC, AUX
29
30     LDI      AUX, 0XFF            ;Declaro al puerto B como salida.
31     OUT      DDRB, AUX
32
33     LDI      AUX, 0XAF            ;Habilito el ADC, el disparo automático, la
34     STS      ADCSRA, AUX          ;interrupción de conversión y seteo el prescaler en
35                                   ;128 para no superar la máxima velocidad de conversión.
36
37     LDI      AUX, 0X62            ;Configuro como tensión de referencia externa la
38     STS      ADMUX, AUX           ;del pin AVCC, ajusto el resultado a izquierda y
39                                   ;selecciono PA2 como canal de entrada analógico.
40
41     SEI                                   ;Habilito interrupciones globales.
42
43 main:
44     LDS      AUX, ADCSRA
45     ORI      AUX, (1<<ADSC)
46     STS      ADCSRA, AUX          ;Inicio conversión poniendo al bit ADSC en 1.
47
48 end:
49     RJMP     end
50
51 isr_adc:
52     LDS      AUX, ADCH            ;Leo el resultado de la conversión dividido por 4.
53
54     LSR      AUX                  ;Divido el anterior valor por 4 nuevamente
55     LSR      AUX                  ;para poder representarlo mediante 6 bits.
56
57     OUT      PORTB, AUX           ;Al resultado anterior, lo muestro por el puerto B.
58
59     RETI

```

8. Resultados

Luego de desarrollar el código descripto en las secciones anteriores y armar el circuito especificado, se logra representar mediante los 6 LEDs dispuestos en el puerto B del microcontrolador, el valor adaptado de la conversión obtenida a partir del ADC, de la tensión regulada por medio del potenciómetro conectado al pin PA2. Así mismo, se verifica que los posibles valores que se muestran van de 0 a 63, siendo estos los valores en los extremos de posición del potenciómetro.

Finalmente, se procede a realizar un listado de los componentes utilizados y sus costos:

Componentes	Costos
Leds verde de 5mm	\$42,00
Resistores de 220 Ω	\$38,50
Potenciómetro de 10k Ω	\$52,00
Cables de conexión	\$20,00
Protoboard	\$232,00
Arduino Uno	\$720,00
TOTAL	\$1104,50

9. Conclusiones

Tras haber realizado todos los pasos pedidos en el enunciado de este trabajo práctico, resta destacar las conclusiones que la experiencia ha aportado.

A partir de este trabajo se logra afianzar el manejo del conversor analógico digital del que puede disponer un microcontrolador, poniendo en práctica la representación en un formato discreto de distintos valores de tensión que, en un principio, no cumplen con esta característica.

Es necesario hacer énfasis en todos los parámetros configurables del ADC de los que se tuvo que tomar nota y especificar para esta práctica en particular. Así mismo, aquellas opciones de uso que no fueron implementadas, como por ejemplo las posibles configuraciones del registro ADCSRB, quedan como buenas herramientas a tener en cuenta en futuras aplicaciones.