



Laboratorio de Microprocesadores - 86.07

## Trabajo Práctico Obligatorio N°1

Profesor:	Ing. Guillermo Campiglio
Cuatrimestre/Año:	1°/2020
Turno de las clases prácticas	Miércoles
Jefe de trabajos prácticos:	Ing. Pedro Ignacio Martos
Docente guía:	Ing. Fabricio Baglivo, Ing. Fernando Pucci
Autores	
Mariano Federico	Guglieri 99573
Seguimiento del proyecto	

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación		Firma J.T.P

Coloquio	
Nota final	
Firma profesor	

# Índice

<b>1. Objetivos</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>2</b>
2.1. configuración del adc . . . . .	2
<b>3. Diagrama en bloques</b>	<b>2</b>
<b>4. Esquemático y listado de componentes</b>	<b>3</b>
<b>5. Diagrama de flujos</b>	<b>3</b>
<b>6. Códigos</b>	<b>4</b>
<b>7. Resultados</b>	<b>4</b>
<b>8. Conclusiones</b>	<b>4</b>

## 1. Objetivos

El objetivo de este trabajo consiste en implementar un programa que permita visualizar, a través de los LEDs, el valor en binario tomado del ADC conectado a un potenciómetro. Para ello se realiza una rutina de conversión analógico/digital. El ADC se trabajará a manera de conversión simple y se tomarán solo 8 de los 10 bits, de los cuales se ajustarán para que la salida vaya de 0 a 63, que será representada por los LEDs conectados al microcontrolador.

## 2. Desarrollo

Para la realización del trabajo se utilizó la plataforma de desarrollo de Atmel, Atmel Studio versión 7.0, en donde se implementó el Software. Para la parte física se requirió de un Arduino UNO, el cual sirvió como programador para el integrado ATMEGA328P, además de suplirlo con la energía necesaria para su funcionamiento. Se utilizaron seis resistores junto a dos seis leds de color verde conectados al puerto b, un potenciómetro conectado al pin de adc, un resistor de  $1k\Omega$  y varios cables para unir el circuito.

El código del programa consta de una primera parte en donde se configura lo esencial; los puertos de entrada y salida, el stack y los registros de adc. Dentro del main se inicializa la conversión ADC. La subrutina de ADC se encarga de cargar un valor analógico convertido a digital en un registro y sacarlo por el puerto B, encendiendo los leds correspondientes que, puestos en fila, representan el número en binario que se encuentra dentro del registro. Cabe aclarar que como sólo se utilizarán 6 leds en vez de 8, previamente se realiza una conversión lineal del valor cargado en el registro antes de sacarlo por el puerto B. Este se divide por cuatro.

La configuración de los registros del conversor analógico digital, al ser un poco más complejos de lo que se venía viendo, se detallarán con mejor detenimiento a continuación.

### 2.1. configuración del adc

El registro de control y de estado del ADC (ADSCRA) es el primero en setearse. En dicho registro se setean los flags que habilitan la interrupción de conversión, la conversión del ADC y la velocidad de conversión. Para la velocidad de conversión se tuvo en cuenta que la plataforma arduino opera con un cristal de 16MHz, por ende la velocidad se seteo en un factor de división con respecto al clock de 128.

En el registro de selección del multiplexer del ADC (ADMUX) se seteo como referencia de tensión a  $V_{cc} = 5V$ , se eligió el canal ADC0 para la entrada del conversor y se ajustó el registro de datos del ADC (ADLAR) para que el resultado de la conversión esté a la izquierda del mismo. Este ajuste se hace ya que no se utilizarán más de 8 bits y no se requiere de la parte baja del registro (ADCL).

## 3. Diagrama en bloques

El diagrama de conexiones en bloque del trabajo se presenta a continuación:

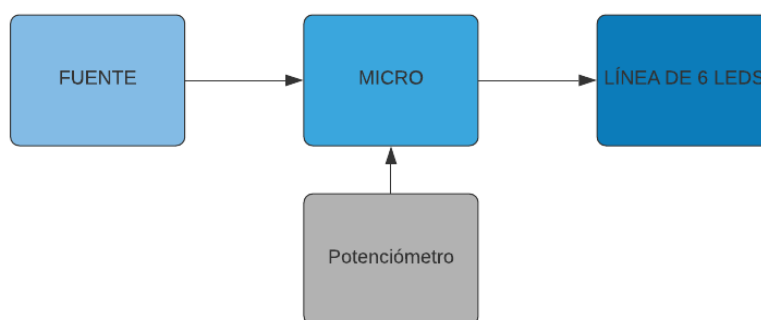


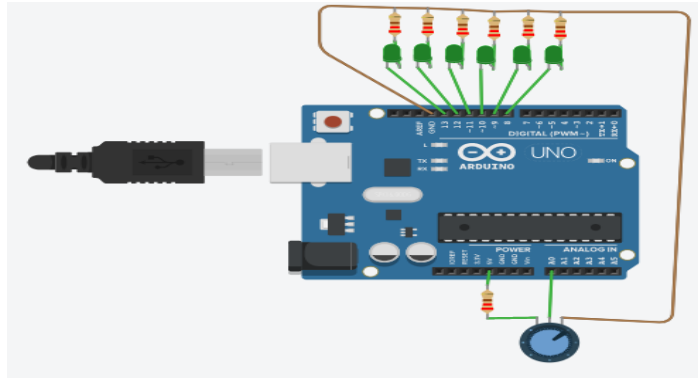
Figura 1: Diagrama de conexiones en bloque

Se observa que el funcionamiento surge primero de la pc donde se encuentra el código que será descargado en el micro y a su vez sirve como una fuente. El micro recibe información del estado del potenciómetro. Según a qué nivel de tensión se encuentre la entrada nivelada por la perilla del potenciómetro, los leds se comportarán de tal manera de mostrar un número en binario relacionado a la tensión que entra. Esto es lo que se ve en el diagrama.

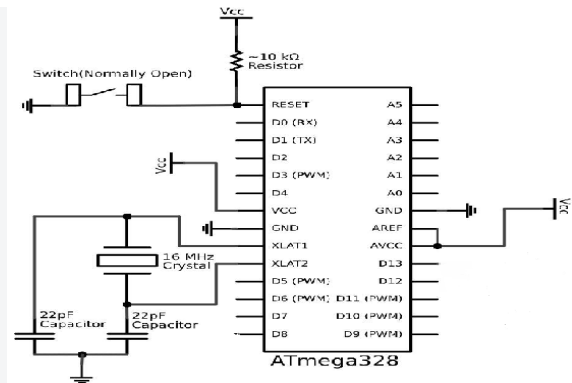
## 4. Esquemático y listado de componentes

Para el circuito se utilizaron los siguientes componentes:

- 7 resistores de  $220\Omega$
- 1 potenciómetro lineal de  $10k\Omega$
- 6 leds de color verde
- 12 cables para protoboard
- 1 placa arduino con atmega328p
- 1 protoboard



(a) Esquemático del circuito implementado



(b) Circuito con cristal de 16MHz

Figura 2: Esquemático del circuito

## 5. Diagrama de flujos

A continuación se presentan los diagramas de flujos. El primero consiste en el programa principal más la configuración previa. La configuración de puertos se puso todo en un mismo bloque ya que esto ya se vió en los trabajos anteriores. La rutina de interrupción analógico digital se encuentra al lado.

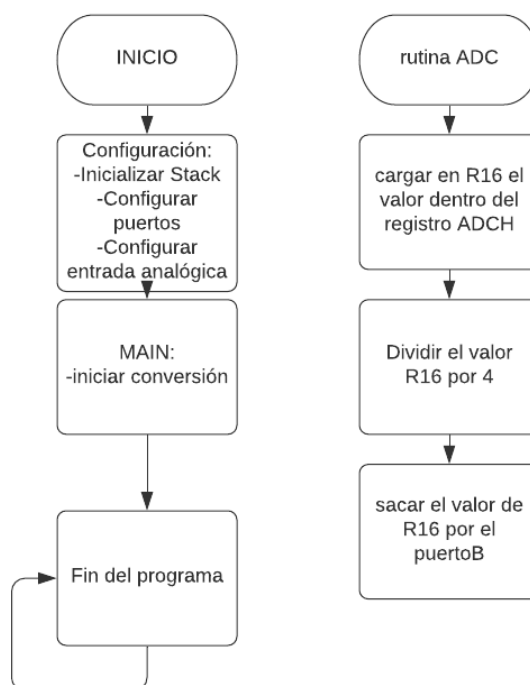


Figura 3: Diagrama de flujos

## 6. Códigos

A continuación se presenta el código:

```
1 .include "m328pdef.inc"
2
3 .cseg
4 ;direcciones de codigo
5 .org 0x0000
6 jmp configuracion
7 .org ADCCaddr
8 rjmp adc_isr
9 .org INT_VECTORS_SIZE
10
11 configuracion:
12 ;inicializo el stack
13 ldi R16, HIGH(RAMEND)
14 out SPH, R16
15 ldi R16, LOW(RAMEND)
16 out SPL, R16
17
18 ;declaro el PORTB como salida
19 ldi R16, 0xff
20 out DDRB, R16
21
22 ;declaro el PORTC como entrada
23 ldi R16, 0x00
24 out DDRC, R16
25
26 ;configuro la entrada analgica:
27 ;ADCEnable, Interrupt enable, Prescales 128
28 ldi R16, 0xaf
29 sts ADCSRA, R16
30 ;Vref=Vcc, LeftAdjust, ADC0
31 ldi R16, 0x60
32 sts ADMUX, R16
33
34 sei
35
36 main:
37
38 lds R16, ADCSRA ; start conversion
39 ori R16, 0x40
40 sts ADCSRA, R16
41
42
43
44 fin: jmp fin
45
46
47 adc_isr: ;Lo que hace el adc, carga un valor en el registro y chau
48 lds R16, ADCH
49 lsr R16
50 lsr R16
51 out PORTB, R16
52 reti
```

## 7. Resultados

En los resultados se pudo observar lo que se esperaba sin mayor inconveniente. Es decir, se vió como la tira de leds mostraba el número contenido en el registro en relación a la tensión que dejaba pasar el potenciómetro.

## 8. Conclusiones

Si bien en este trabajo no se hizo uso extenso de todas las opciones que permiten los registros de ADC, dio un buen acercamiento a todas las posibilidades que este micro tiene para la conversión de datos analógicos a digitales. A su vez, se tuvo que tener en consideración ciertas cuestiones a la hora del diseño de la parte física del proyecto, como el hecho de elegir la tensión de referencia y prestarle más atención al hecho de que el arduino trabaja con un cristal de 16MHz. Si no se hubiese tenido esto en cuenta, el desempeño del circuito se vería afectado.