



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

## Trabajo Práctico N°5:

### Uso del ADC

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1°/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			-									
Autor			Seguimiento del proyecto									
Francisco	Rossi	99540										

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivo . . . . .	2
1.2. Descripción . . . . .	2
<b>2. Materiales</b>	<b>2</b>
<b>3. Diagrama en Bloques</b>	<b>2</b>
<b>4. Esquemático</b>	<b>3</b>
<b>5. Diagrama de flujo</b>	<b>4</b>
<b>6. Código</b>	<b>5</b>
<b>7. Resultados</b>	<b>6</b>
<b>8. Conclusiones</b>	<b>6</b>

## 1. Introducción

En el siguiente informe se explica el diseo de un programa escrito en lenguaje Assembler con el cual se lee una entrada analogica y se la convierte a digital mediante el ADC 2 del ATMEGA328p. La salida digital ser mostrada a travs de 6 LEDs conectados al puerto 6, que mostrar los valores digitales convertidos a valores entre 0 y 63.

### 1.1. Objetivo

El objetivo es comenzar a utilizar el conversor analgico digital, en este caso como conversor de 8 bits.

### 1.2. Descripción

Se conectan 6 LEDs con un resistor en serie de  $220\ \Omega$  a cada pin  $\in [PB0, PB5]$ , los mismo se interpretaran como valores de 0 al 63 en binario, los cuales correspondern a los valores tensin de la entrada del conversor ADC conectado al puerto  $PC2$  ( $ADC2$ ) se controla la tensin de entrada entre valores de 0 a 5 Volts determinados por el divisor resistivo entre  $R_V$  y  $R_{var}$  donde  $R_V = 220\ \Omega$  y  $R_{var}$  un resistor variable de  $50\ k\Omega$  como se muestra en el esquemático de la figura **Fig. 3**.

## 2. Materiales

Se utilizaron los siguiente materiales para el proyecto:

- 6 LEDs (120\$ (Pesos Argentinos))
- 7 Resistor de  $220\Omega$  (28\$ (Pesos Argentinos))
- 1 Resistor variable de  $50\ k\Omega$  (115\$ (Pesos Argentinos))
- 1 Microcontrolador ATmega328p (Utilizando el integrado con el Arduino Uno) (700\$ (Pesos Argentinos))

## 3. Diagrama en Bloques

En la **Fig. 1** se muestra un diagrama en bloques del circuito.

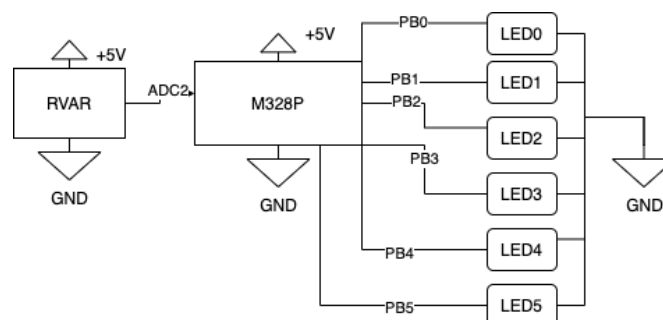


Figura 1: Diagrama en bloques.

## 4. Esquemático

En las **Fig. 3** y **Fig. 2** se muestra como se conect el arduino con los LEDs, el potenciómetro y los resistores de  $220\ \Omega$ .

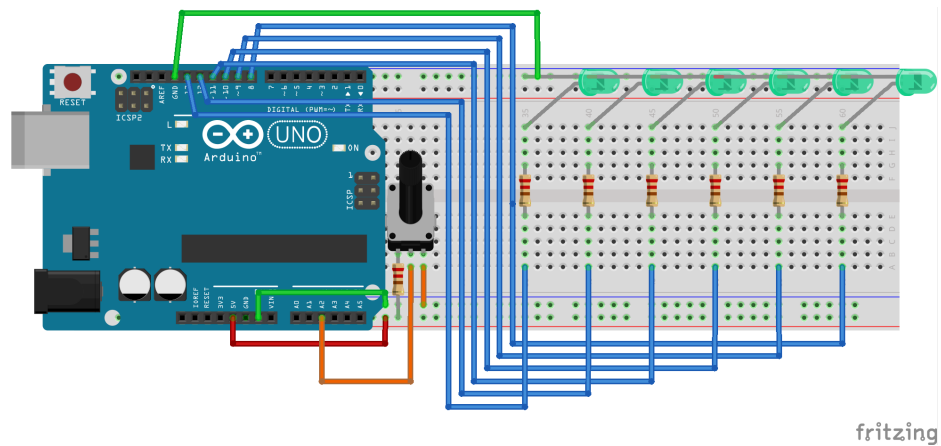


Figura 2: Coneccin entre el arduino, los LEDS y el pulsador

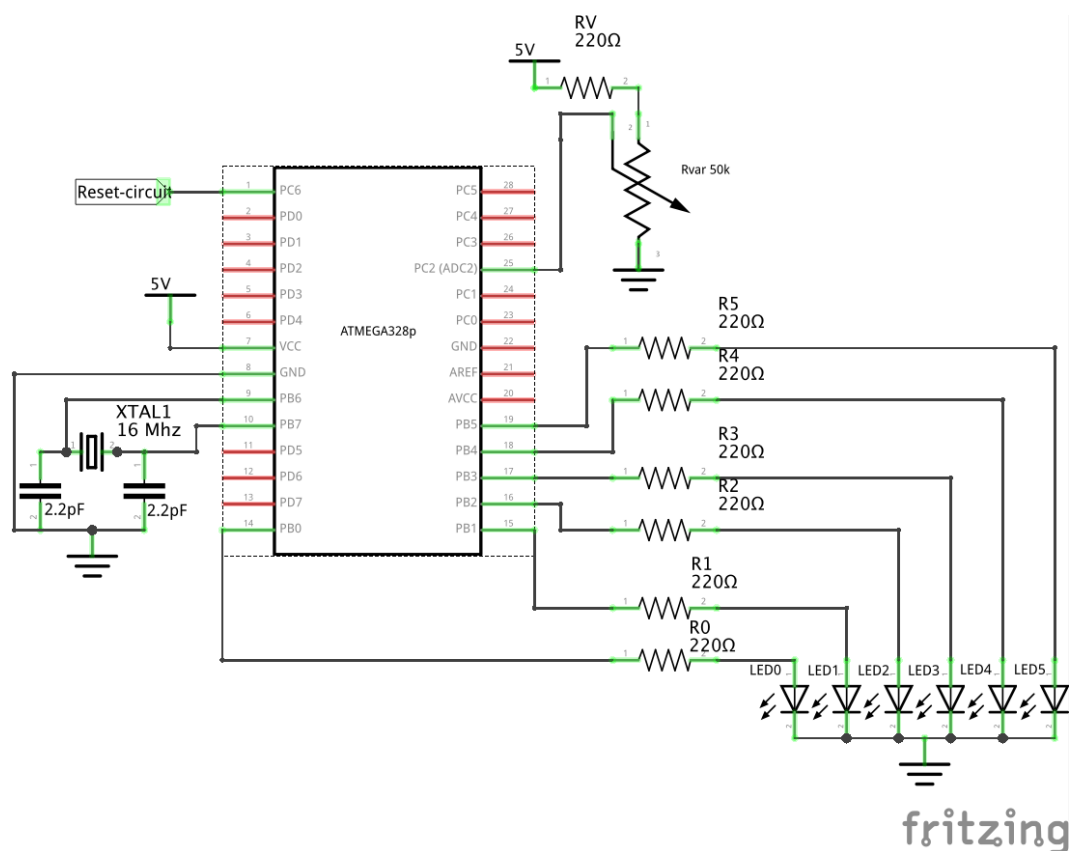


Figura 3: Esquemático del circuito implementado.

## 5. Diagrama de flujo

En la **Fig. 4** se muestra el diagrama de flujo del programa.

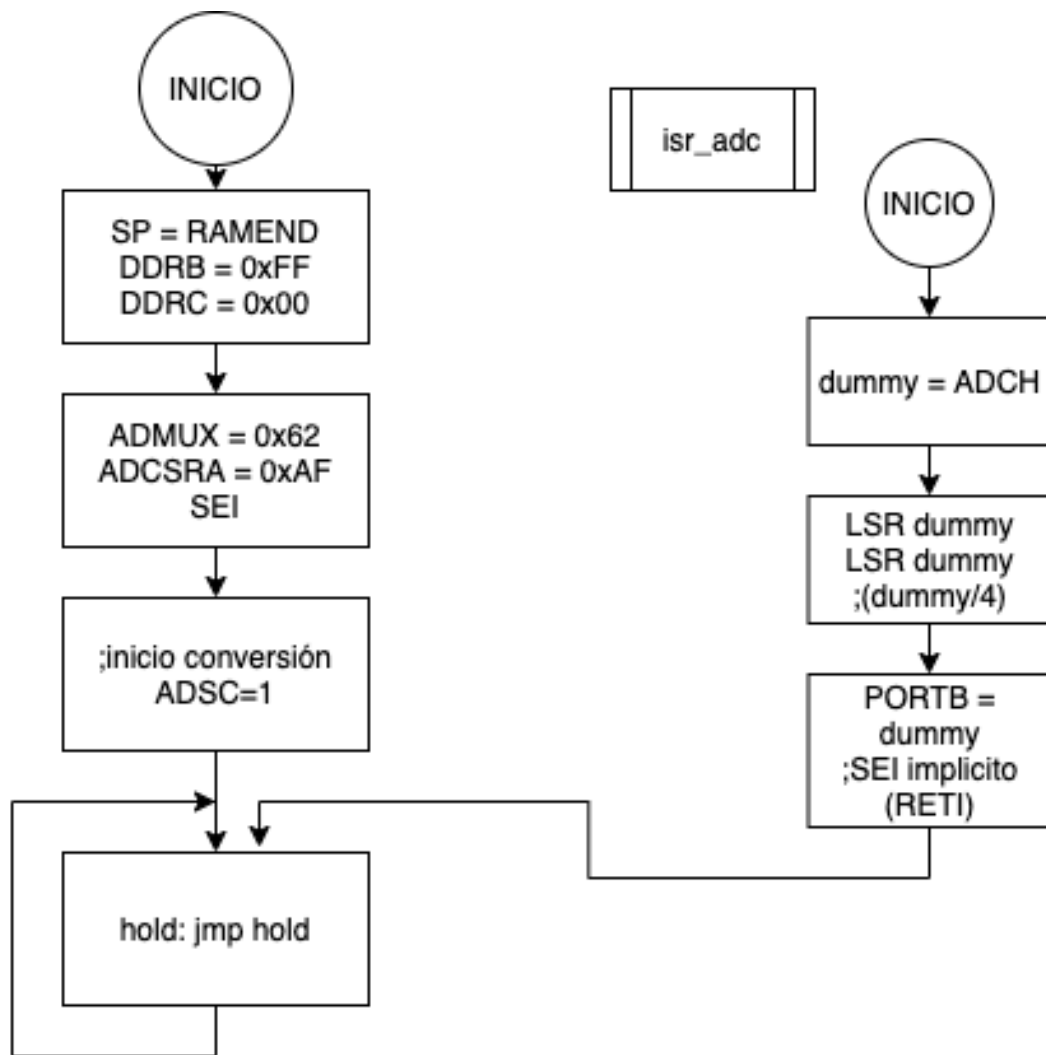


Figura 4: Diagrama de flujo.

## 6. Código

El siguiente código consta de tres partes, en primer lugar la configuración donde se inicializa el **stack pointer**, se configuran los puertos como entradas y salidas y los registros que afectan el comportamiento del ADC. Se modificaron dos de estos, el ADMUX, el ADCSRA, los cuales se configuraron como indican las **Tablas 1 y 2**. También es importante agregar que ADCSRB se dejó en cero con lo cual el **Trigger Source** está en modo **Free running**.

Tabla 1: Configuración del registro ADMUX.

REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
0	1	1	-	0	0	1	0

Tabla 2: Configuración del registro ADCSRA.

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
1	0	1	0	1	1	1	1

Esta configuración equivale a configurar el ADC de la siguiente manera:

- Vcc como Vref del ADC
- Ajuste a izquierda para utilizar solo 8 bits del conversor.
- el multiplexor selecciona a ADC2
- ADC habilitado
- Interrupción del ADC habilitada
- Autotriggger habilitado
- Prescaler = 128

```

1 ; Autor: Francisco Rossi
2 ; Padron: 99540
3 ; 86.07 Laboratorio de Microprocesadores – FIUBA
4 ; Catedra: Miercoles
5 ; Fecha: 19 de julio de 2020
6
7 .include "m328pdef.inc"
8
9 .def dummy = r25
10
11 .macro set_sp
12     ldi dummy, low(RAMEND)
13     out spl, dummy
14     ldi dummy, high(RAMEND)
15     out sph, dummy
16 .endm
17
18 .macro set_port_as_out
19     ldi dummy, 0xFF
20     out @0, dummy
21 .endm
22
23 .macro set_port_as_in
24     ldi dummy, 0x00
25     out @0, dummy
26 .endm
27
28
```

```
29 .cseg
30 .org 0x0000
31     jmp config
32
33 .org 0x002A
34     jmp     isr_adc
35 .org INT.VECTORS_SIZE
36
37 config:
38     set_sp
39
40     set_port_as_out    DDRB
41     set_port_as_in     DDRC
42
43     ; adc
44
45     ldi    dummy, 0x62; 0b01100010(VREF = VCC ,ajustar a la izq, para que sea de 8 bits ADLAR = 1,
46         MUX 0010 ADC2)
47     sts    ADMUX, dummy
48     ldi    dummy, 0xAF; 0b10101111 (ADCENABLE,no arranco conv,autotrigger,prescaler=128, ADIE=1)
49     sts    ADCSRA, dummy ; ADEN=1 ahbilito adc, ADSC= 1 inicio conversion prescaler en 128 para que
50         no supere la velocidad de conversion del ADC
51     ; adcsrb en 0. free running
52
53     sei ; habilito interrupciones
54 main:
55
56     lds    dummy,ADCSRA
57     ori    dummy,(1<<6) ; inicio conversion
58     sts    ADCSRA,dummy
59 hold:
60     jmp    hold
61
62 isr_adc:
63     ; leo entrada
64     lds    dummy,ADCH
65     ; division por 4 para que vaya entre 0 y 63
66     lsr    dummy
67     lsr    dummy
68     ; salida por PORTB
69     out    PORTB,dummy
70
71     reti
```

## 7. Resultados

Se logr diseñar un programa para el microcontrolador ATMEGA328p con el cual se utiliza el conversor ADC para leer una entrada analgica entre 0 y 5 Volts y generar una salida que va entra 0 y 63 en código binario y esta representada por seis LEDs conectados al puerto B del microcontrolador.

## 8. Conclusiones

Se logr generar un programa que realic la tarea de configuracin del ADC en este caso de 8 bits y lea una entrada analgica generando con manipulando la misma digitalmente y luego poder utilizar esos valores como salida.