



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

## Conversor Analógico-Digital

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1º/2020									
Turno de las clases prácticas			Miercoles 19 hs									
Jefe de trabajos prácticos:			Pedro Ignacio Martos									
Docente guía:			Pedro Martos, Fabricio Baglivo, Fernando Pucci									
Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Leonel	Mendoza	101153										

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

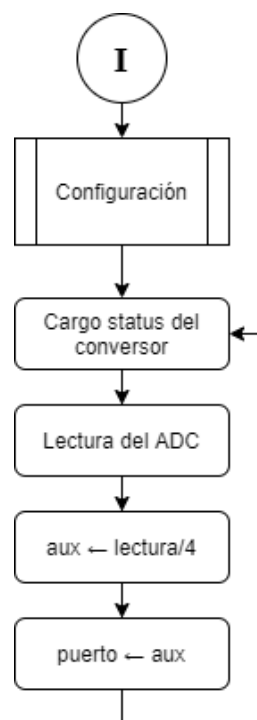
Coloquio	
Nota final	
Firma profesor	



## 5. Listado de componentes

- Microcontrolador *ATmega328p* y programador USBasp (Arduino UNO) [AR\$ 950]
- 6x LED [AR\$ 60]
- 3x Resistencia ( $220\ \Omega$ ) [AR\$ 12]
- Potenciometro ( $10\ k\Omega$ ) [AR\$ 90]

## 6. Diagrama de Flujo



## 7. Código de programa

```
.include "m328pdef.inc"

; * * * * *
;   START MACROS   ;
; * * * * *

.MACRO SET_SP ;[auxGPR]
    LDI @0, low(RAMEND)
    OUT SPL, @0
    LDI @0, high(RAMEND)
    OUT SPH, @0
.ENDM

.MACRO SET_X ;[LABEL to data memory]
    LDI XL, low(@0)
    LDI XH, high(@0)
.ENDM

.MACRO SET_Y ;[LABEL to data memory]
    LDI YL, low(@0)
    LDI YH, high(@0)
.ENDM

.MACRO SET_Z ;[LABEL to prog memory]
    LDI ZL, low(@0 << 1)
    LDI ZH, high(@0 << 1)
.ENDM

; * * * * *
;   END MACROS   ;
; * * * * *

.DEF aux = R16

.CSEG

.ORG 0X0000                ; En esta direccion escribo la instruccion JMP conf
JMP conf

.ORG INT_VECTORS_SIZE     ; Direccion donde escribir el codigo

conf:
    SET_SP    aux

    LDI      aux, (0 << PC0)    ; inicializo puerto de ADC
    OUT      DDRC, aux

    LDI      aux, 0xFF          ; inicializo puerto D como salida
    OUT      DDRD, aux
```

```
LDI    aux, 0b10000111    ; inicializo ADC Enable, Prescaler 128
STS    ADCSRA, aux
```

```
LDI    aux, 0b01100000    ; Vref=Vcc, Ajusto izq, ADC0
STS    ADMUX, aux
```

```
SEI
```

main:

```
LDS    aux, ADCSRA
ORI    aux, 0x40          ; or 1 al bit de ADSC (Start Conversion)
STS    ADCSRA, aux
LDS    aux, ADCH          ; tomo los 8 msb del resultado del adc al tener left
LSR    aux                ; adjust, ya que necesito solo los ultimos 6 bits
LSR    aux                ; los divido por 4 para quedarme con los 6 msb
OUT    PORTD, aux        ; pongo el resultado en PORTD (LEDS)
RJMP   main
```

## 8. Resultados

Se logró representar la tensión en el nodo del divisor resistivo del potenciómetro, con 6 LEDs (como 6 bits), mediante una lectura del nodo por el pin del ADC (PC0); de forma casi instantánea y en tiempo real, a medida que se mueve el potenciómetro, se puede leer en binario con los LEDs la tensión que se tiene en el divisor resistivo (si se lee  $0,078 \cdot n$ ), o bien se puede saber el valor de los resistores en el divisor resistivo (como  $156,25 \cdot n$ )

## 9. Conclusiones

Mediante la configuración del conversor analógico-digital, se pudo leer y almacenar el valor de tensión del divisor resistivo de una forma conveniente (en registro `ADMUX`, `ADLAR = 1`) para poder tomar la parte de interés (6 MSB) directamente, haciendo menos operaciones. Con esta información, se representa el valor en binario usando 6 LEDs como bits, mediante manejo de puertos.