Practica 1: Funcionamiento del ADC

Sistemas de Medicion y Control 18MPEDS0730
Ago-Dic 2025
Centro de Enseñanza Tecnica Industrial Plantel Colomos
Tgo. en Desarrollo de Software
Academia: Sistemas Electrónicos
Profesor: Diana Marisol Figueroa Flores

Emmanuel Buenrostro 22300891 7F1
Emiliano Arzate 22300929 7F1

27 de Agosto de 2025

§1 Objetivo

Objetivo General: Verificar el funcionamiento básico de un ADC de 8 bits y 10 bits.

Objetivos Específicos: Identificar la resolución utilizando un sistema de adquisición de datos, considerando una palabra de 8 bits y 10 bits, anexando un acoplamiento digital.

§2 Desarrollo Teórico

§2.1 Resumen

- **Aproximaciones Sucesivas (SAR):** El más común. Ofrece un buen **equilibrio** entre velocidad y resolución. Ideal para sistemas de adquisición de datos y control.
- **Delta-Sigma ($\Delta\Sigma$):** El de más **alta resolución**. Perfecto para audio de alta fidelidad y mediciones de precisión a baja velocidad.
- **Flash:** El más **rápido** que existe. Se usa en video, radares y osciloscopios, pero su resolución es limitada debido a su alto costo y consumo.
- **Doble Rampa:** El más **lento pero muy preciso** y con excelente rechazo al ruido. Comúnmente utilizado en multímetros digitales.

Referencias Bibliográficas

Horowitz, P., & Hill, W. (2015). *The Art of Electronics* (3rd ed.). Cambridge University Press.

Texas Instruments. (2022). *The ADC guide*. Recuperado de https://www.ti.com/lit/sl/slyy190/slyy190.pdf

§2.2 Material

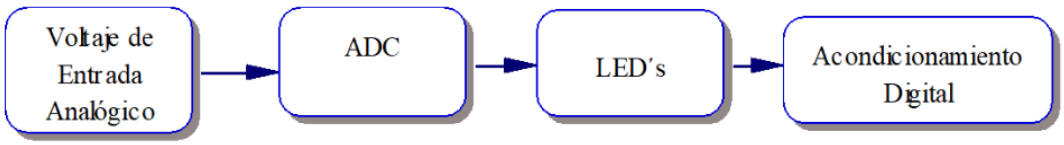
Cantidad	Material
1	Arduino
10	Leds
10	Resistecias 330 omhs
1	Potenciometro de toque fino
1	pantalla lcd
Muchos	Jumpers

§2.3 Características Eléctricas de los Componentes

Anexar características eléctricas de todos los componentes a utilizar, así como los voltajes y corrientes máximas de trabajo, distribución de terminales, etc.

Componente	**Características Eléctricas**
Arduino (UNO)	• Voltaje de operación: 5 V DC • Voltaje de entrada recomendado: 7–12 V (máx. absoluto: 6–20 V) • Corriente por pin I/O: 40 mA máx. • Corriente total en pin 3.3 V: 50 mA máx.
LEDs comunes	• Voltaje directo (Vf): 1.8–2.2 V (rojo/verde/amarillo), 2.8–3.3 V (azul/blanco) • Corriente directa típica: 10–20 mA (recomendado 10–15 mA) • Corriente inversa máxima: ≈100 µA
Resistencias 330 Ω (1/4 W)	• Valor nominal: 330 Ω ± 5% • Potencia máxima: 0.25 W • Voltaje máximo soportado: 250 V aprox.
Potenciómetro de toque fino	• Valor típico: 10 kΩ (multivuelta) • Tolerancia: ±10% • Potencia disipada: 0.25–0.5 W • Voltaje máximo: 50–100 V DC
Pantalla LCD 16x2 (HD44780)	• Voltaje de operación: 4.5–5.5 V (típico: 5 V) • Corriente típica: 1–2 mA por segmento (sin retroiluminación) • Retroiluminación LED: 5 V / 15–30 mA
Jumpers	• Corriente máxima: ∼1 A continuo • Voltaje soportado: hasta 50 V DC

§2.4 Diagrama a Bloques



§2.5 Calculos

La resolución de un Convertidor Analógico–Digital (ADC) se obtiene con la siguiente expresión:

$$\text{Resolución} = \frac{V_{\text{ref}}}{2^n}$$

donde V_{ref} es el voltaje de referencia y n es el número de bits.

- Para un ADC de 10 bits:

$$\Delta V_{10} = \frac{5\,V}{2^{10}} = \frac{5}{1024} \approx 4.88\,\text{mV}$$

- Para un ADC de 8 bits:

$$\Delta V_8 = \frac{5\,V}{2^8} = \frac{5}{256} \approx 19.53\,\text{mV}$$

El voltaje medido a partir del valor digital leído se expresa como:

$$V_{in} = \frac{\text{ADC}_{\text{valor}}}{2^n - 1} \cdot V_{\text{ref}}$$

Ejemplo: para un valor de $ADC = 512$ en un convertidor de 10 bits:

$$V_{in} = \frac{512}{1023} \cdot 5 \approx 2.50\,V$$

El error de cuantización corresponde a medio paso de resolución (medio LSB):

$$E_q = \pm\frac{1}{2} \cdot \Delta V$$

- En 10 bits: $E_q \approx \pm 2.44\,\text{mV}$
- En 8 bits: $E_q \approx \pm 9.77\,\text{mV}$

Por lo tanto, el ADC de 10 bits permite distinguir variaciones mucho más pequeñas de voltaje en comparación con el de 8 bits, aunque ambos son funcionales dependiendo de la aplicación.

§2.6 Tabla de Mediciones Teoricas

Agregar la tabla de mediciones Teóricas, con sus 10 voltajes a verificar.

§2.7 Tabla de Mediciones Teóricas

Agregar la tabla de mediciones Teóricas, con sus 10 voltajes a verificar.

§2.8 Tabla de Mediciones Teóricas (10 bits, ordenada)

ADC 10 bits (0–1023)	Binario (10 bits)	Voltaje (V)
174	0010101110	0.85
303	0100101111	1.48
372	0101110100	1.82
401	0110010001	1.96
413	0110011101	2.02
428	0110101100	2.09
528	1000010000	2.58
581	1001000101	2.84
645	1010000101	3.15
832	1101000000	4.06

§2.9 Tabla de Mediciones Teóricas (8 bits, ordenada)

ADC 8 bits (0–255)	Binario (8 bits)	Voltaje (V)
29	00011101	0.57
50	00110010	0.98
83	01010011	1.63
85	01010101	1.67
91	01011011	1.78
101	01100101	1.98
144	10010000	2.82
157	10011101	3.08
170	10101010	3.33
252	11111100	4.94

§3 Desarrollo Practico

§3.1 Proceso

Lo primero que tuvimos que hacer fue conectar los 10 leds con sus resistencias de 330 omhs para que no se quemen, despues de eso conectamos el potenciometro de ajuste fino la pata 1 la conectamos a vcc, la pata 3 a gnd y la pata de en medio la conectamos al pin A1 que es un pin analogico del arduino para poder interpretar esos voltajes y convertirlo en un adc , despues de eso conectamos la pantalla lcd y alli imprimimos el voltaje y los bits que representa ese mismo voltaje

§3.2 Codigos

ADC de 8 bits

```
#include <LiquidCrystal.h>

// --- Definici n de Pines ---
```

```
LiquidCrystal lcd(22, 23, 25, 24, 26, 27);
const int pinADC = A1;
// Usaremos solo 8 LEDs para el valor de 8 bits
const int leds[] = {42, 43, 44, 45, 46, 47, 48, 49}; // <-- CAMBIO (ahora 8 LEDs)

// --- Par metros de Calibraci n y Filtrado ---
const int SAMPLES = 16;
const float ALPHA = 0.25f;
const int DEADBAND = 1;
const float VREF = 4.78;

// --- Variables Globales ---
// 'shownADC' seguir  guardando el valor de 10 bits para mantener la
    sensibilidad del filtro
int shownADC_10bit = -1;
float ema = -1;

/**
 * Lee el pin anal gico varias veces y devuelve el promedio.
 */
int readADCavg(int samples) {
  long acc = 0;
  for (int i = 0; i < samples; ++i) {
    acc += analogRead(pinADC);
    delayMicroseconds(120);
  }
  return (int)(acc / samples);
}


void updateLCD(int adc_8bit) {

  float voltaje = (adc_8bit * VREF) / 255.0;

  // Muestra el valor ADC de 8 bits
  lcd.setCursor(5, 0);
  lcd.print("    ");
  lcd.setCursor(5, 0);
  lcd.print(adc_8bit);

  // Muestra el voltaje calculado
  lcd.setCursor(9, 1);
  lcd.print("     ");
  lcd.setCursor(9, 1);
  lcd.print(voltaje, 3);
}

/**
 * Muestra el valor ADC de 8 BITS en formato binario en los 8 LEDs.
 */
void updateLEDs(int adc_8bit) {
  for (int i = 0; i < 8; i++) {
    bool on = ((adc_8bit >> i) & 1);
    digitalWrite(leds[i], on ? HIGH : LOW);
  }
}

/**
 * Funci n de configuraci n inicial.
 */
void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Iniciando...");
  delay(600);

  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("ADC:");
  lcd.setCursor(0, 1); lcd.print("Voltaje:");


```

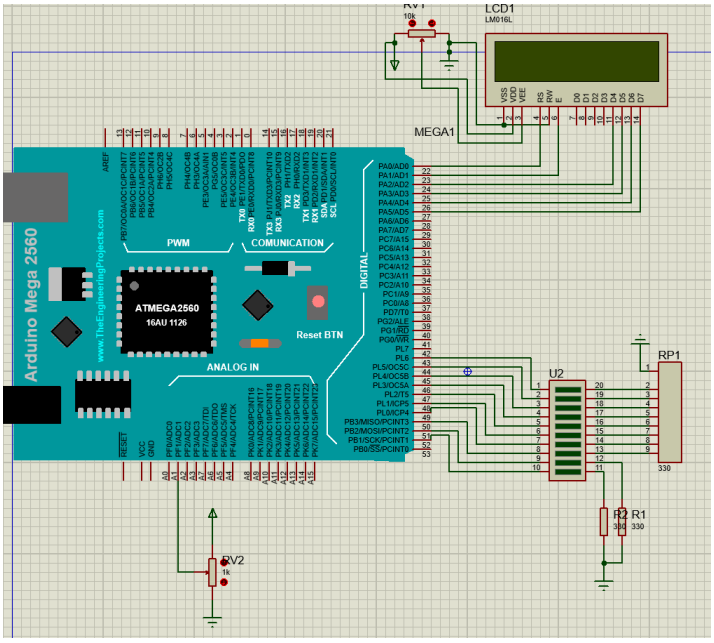
```
  for (int i = 0; i < 8; i++) {
    pinMode(leds[i], OUTPUT);
  }

  int init_10bit = readADCavg(SAMPLES * 2);
  ema = init_10bit;
  shownADC_10bit = init_10bit;


  int init_8bit = init_10bit >> 2;
  updateLCD(init_8bit);
  updateLEDs(init_8bit);
}

/**
 * Bucle principal del programa.
 */
void loop() {
  // El filtro sigue trabajando con la resoluci n completa de 10 bits
  int raw_10bit = readADCavg(SAMPLES);
  ema = ema + ALPHA * (raw_10bit - ema);
  int filtered_10bit = (int)lround(ema);

  if (abs(filtered_10bit - shownADC_10bit) >= DEADBAND) {
    shownADC_10bit = filtered_10bit;

    int adc_8bit = shownADC_10bit >> 2;


    updateLCD(adc_8bit);
    updateLEDs(adc_8bit);
  }

  delay(30);
}
```

ADC de 10 bits

```
#include <LiquidCrystal.h>

// --- Definici n de Pines ---
LiquidCrystal lcd(22, 23, 25, 24, 26, 27);
const int pinADC = A1;
const int leds[] = {42, 43, 44, 45, 46, 47, 48, 49, 50, 51}; // 10 LEDs


const int SAMPLES = 16;
const float ALPHA = 0.25f;
const int DEADBAND = 1;
const float VREF = 4.78;


int shownADC = -1;
float ema = -1;

/**
 * Lee el pin anal gico varias veces y devuelve el promedio.
 */
int readADCavg(int samples) {
  long acc = 0;
  for (int i = 0; i < samples; ++i) {
    acc += analogRead(pinADC);
    delayMicroseconds(120);
  }
  return (int)(acc / samples);
}

/**
 * Actualiza la pantalla LCD con el valor ADC y el voltaje calculado.
 */
void updateLCD(int adc) {

```

```
  float voltaje = (adc * VREF) / 1023.0;

  // Muestra el valor ADC
  lcd.setCursor(5, 0);
  lcd.print("    ");
  lcd.setCursor(5, 0);
  lcd.print(adc);

  // Muestra el voltaje calculado
  lcd.setCursor(9, 1);
  lcd.print("     ");
  lcd.setCursor(9, 1);
  lcd.print(voltaje, 3);
}

/**
 * Muestra el valor ADC en formato binario en los 10 LEDs.
 */
void updateLEDs(int adc) {
  for (int i = 0; i < 10; i++) {
    // Extrae cada bit del valor 'adc' y lo asigna a un LED
    bool on = ((adc >> i) & 1);
    digitalWrite(leds[i], on ? HIGH : LOW);
  }
}


void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Iniciando...");
  delay(600);


  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("ADC:");
  lcd.setCursor(0, 1); lcd.print("Voltaje:");


  for (int i = 0; i < 10; i++) {
    pinMode(leds[i], OUTPUT);
  }


  int init = readADCavg(SAMPLES * 2);
  ema = init;
  shownADC = init;


  updateLCD(shownADC);
  updateLEDs(shownADC);
}


void loop() {

  int raw = readADCavg(SAMPLES);
  ema = ema + ALPHA * (raw - ema);
  int filtered = (int)lround(ema);


  if (abs(filtered - shownADC) >= DEADBAND) {
    shownADC = filtered;
    updateLCD(shownADC);
    updateLEDs(shownADC);
  }

  delay(30); // Peque a pausa para no saturar
}
```

§3.3 Diagrama Eléctrico



§3.4 Tabla Comparativa

Agregar la tabla comparativa de valores teóricos y prácticos, en caso de tener gráficas anexarlas.

§3.5 Tabla de Mediciones Teóricas (10 bits)

Valores obtenidos considerando una resolución de 10 bits (0–1023).

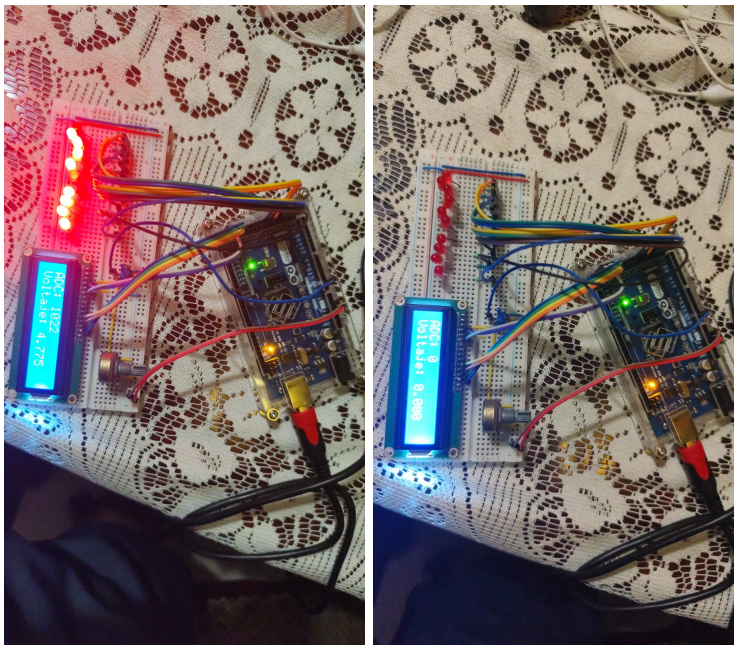
V analógico (V)	Binario (10 b)	V sistema (V)	ADC
0.793	0010101110	0.813	174
1.391	0100101110	1.411	303
1.718	0101110100	1.738	372
1.854	0110010001	1.874	401
1.910	0110011101	1.930	413
1.980	0110101100	2.000	428
2.447	1000010000	2.467	528
2.695	1001000101	2.715	581
2.994	1010000101	3.014	645
3.868	1101000000	3.888	832

§3.6 Tabla de Mediciones Teóricas (8 bits)

Valores obtenidos considerando una resolución de 8 bits (0–255).

V analógico (V)	Binario (8 b)	V sistema (V)	ADC
0.524	00011101	0.544	29
0.917	00110010	0.937	50
1.536	01010011	1.556	83
1.573	01010101	1.593	85
1.686	01011011	1.706	91
1.873	01100101	1.893	101
2.679	10010000	2.699	144
2.923	10011101	2.943	157
3.167	10101010	3.187	170
4.704	11111100	4.724	252

§4 Circuito real



§5 Observaciones y Conclusiones

§5.1 Observaciones

Esta practica presenta un problema principal, que es la exactitud, ademas de que muchas veces no se tienen los 5V completos y como es una variacion muy pequeña a veces hay mucho ruido.

§5.2 Conclusiones Personales

- Arzate: Esta practica estuvo dificil de hacer que el ruido desaparezca, pero de lo demàs fueron practicamente puras cosas que ya habiamos usado anteriormente.
- Emmanuel: Esta practica usa lo que hicimos en el semestre pasado en Embebidos I, donde usamos el AnalogRead del arduino. Sirve como un puente entre lo analogico y lo digital.