ANALOG TO DIGITAL CONVERTER

~MIKROKONTROLLER~

Oleh :

Muhammad Alvian Akbar

2B D3 - TE

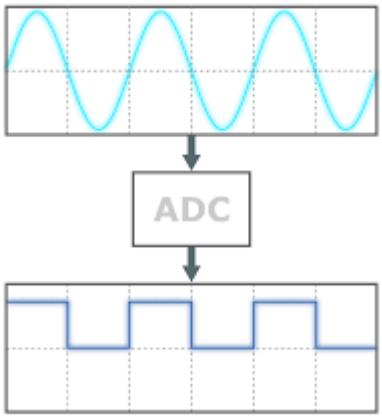
(16 / 2331110023)

Jurusan Teknik Elektro
Program Studi D-III Teknik Elektronika
Politeknik Negeri Malang
2024

I. Tujuan Praktikum

- ➞ Mahasiswa mampu mengetahui dasar dari ***Analog to Digital Converter.***
- ➞ Mahasiswa dapat memanfaatkan pengaplikasian ***Analog to Digital Converter*** pada sistem **mikrokontroler**, yaitu **Arduino.**
- ➞ Mahasiswa dapat mempraktikkan percobaan aplikasi ***Analog to Digital Converter*** dengan baik dan benar.

II. Dasar Teori



A. DEFINISI

Apakah itu ***Analog to Digital Converter*** atau yang biasa di singkat menjadi ***ADC?*** Jadi **ADC** itu adalah komponen elektronik yang sangat penting dalam sistem mikrokontroler. Fungsinya adalah untuk mengubah **sinyal analog**, seperti tegangan yang nilainya terus-menerus berubah (kontinyu), menjadi **sinyal digital** berupa angka diskrit yang dapat diproses oleh perangkat digital seperti Arduino. ADC menjadi jembatan antara dunia analog dan dunia digital. Contoh paling umum adalah membaca data dari sensor (seperti potensiometer atau sensor suhu) yang menghasilkan sinyal analog dan kemudian mengonversinya menjadi angka digital yang bisa digunakan untuk perhitungan atau pengontrolan.

B. PRINSIP KERJA

Prinsip kerja **ADC** ini dibagi menjadi 3 sektor. Yaitu :

1. **Input Analog**
   - Arduino menerima tegangan analog dari perangkat eksternal melalui pin analog (contoh: **A0**, **A1**, dst.).
   - Tegangan input biasanya dalam rentang **0 hingga 5V**, tergantung pada sumber daya dan konfigurasi.

2. **Proses Konversi**
   - Proses konversi menggunakan prinsip kuantisasi. ADC membagi rentang tegangan referensi menjadi sejumlah tingkat berdasarkan resolusi.
   - **Resolusi ADC pada Arduino adalah 10-bit**, artinya tegangan input dibagi menjadi 1024 tingkatan.

   Rumus Konversinya adalah :

$$‘\ \boldsymbol{Digital\,Value = fracVref \times (2^{res} - 1)}\ ‘$$

Misalnya, jika tegangan input adalah 2.5V dan tegangan referensi 5V, nilai digitalnya :

$$\boldsymbol{Digital\,Value = frac2.55 \times 1023 = 511}$$

3. **Output Digital**
   - Nilai digital berupa bilangan biner (0 hingga 1023 pada ADC 10-bit). Nilai ini digunakan dalam logika pemrograman atau diolah lebih lanjut oleh mikrokontroler.

C. FITUR PENTING DALAM ARDUINO

1. **Resolusi**
   - Resolusi menentukan seberapa banyak tingkat pembagian tegangan referensi. Arduino menggunakan **ADC 10-bit**, yang membagi tegangan input menjadi **1024 level**.
   - Dengan **resolusi tinggi**, hasil pembacaan **lebih presisi**, tetapi membutuhkan **waktu konversi yang lebih lama**.
2. **Tegangan Referensi (Vref)**

- Tegangan referensi adalah nilai maksimum yang dijadikan acuan ADC. Arduino biasanya memiliki **tegangan referensi default sebesar 5V**.
- Tegangan referensi bisa diubah untuk meningkatkan presisi pada rentang tegangan yang lebih kecil dengan menggunakan :
  ⇀ **DEFAULT:** 5V atau 3.3V (bergantung pada papan).
  ⇀ **EXTERNAL:** Menggunakan tegangan referensi eksternal yang dihubungkan ke pin AREF.
  ⇀ **INTERNAL:** Referensi internal (biasanya 1.1V pada sebagian besar Arduino).

3. **Pin Analog**
   - Pin analog pada Arduino digunakan untuk menerima sinyal tegangan analog. Jumlah pin bervariasi tergantung pada jenis Arduino.
     Contoh : Arduino Uno memiliki **6 pin analog (A0–A5).**
4. **Fungsi Bawaan**
   - Arduino menyediakan fungsi **"analogRead(pin)"** untuk membaca nilai ADC dari pin tertentu. Contoh :

```
int nilaiAnalog = analogRead(A0);
```

D. APLIKASI ADC DALAM ARDUINO

1. **Pengukuran Tegangan Analog**
   - Membaca nilai tegangan dari sensor seperti **potensiometer, LDR (Light Dependent Resistor)**, atau **termistor.** Nilai ini bisa dikonversi menjadi format yang dapat ditampilkan atau digunakan.
     Contoh kode sederhana :

```
int potPin = A0;
void loop() {
    int nilai = analogRead(potPin);
    float tegangan = nilai * (5.0 / 1023); // Konversi ke tegangan
    Serial.println(tegangan); // Tampilkan tegangan
    delay(500);
}
```

2. **Indikator Baterai**
    - Tegangan dari baterai diukur dan diubah menjadi persentase untuk menunjukkan kapasitasnya.
    - Dengan menggabungkan LED bar, setiap LED dapat dinyalakan sesuai level kapasitas baterai.
3. **Monitoring Sensor**
    - Membaca data analog dari sensor jarak, kelembapan, atau cahaya, lalu menampilkan hasil pengukuran pada LCD.

E. KEUNGGULAN VS KEKURANGAN

KEUNGGULAN

1. **Interaksi dengan Real Usage di Dunia Nyata**

   ADC memungkinkan Arduino membaca sinyal analog, menjadikannya alat yang fleksibel untuk berbagai aplikasi.

2. **High Resolution**

   ADC 10-bit cukup akurat untuk banyak proyek hobi dan aplikasi semi-profesional.

KEKURANGAN

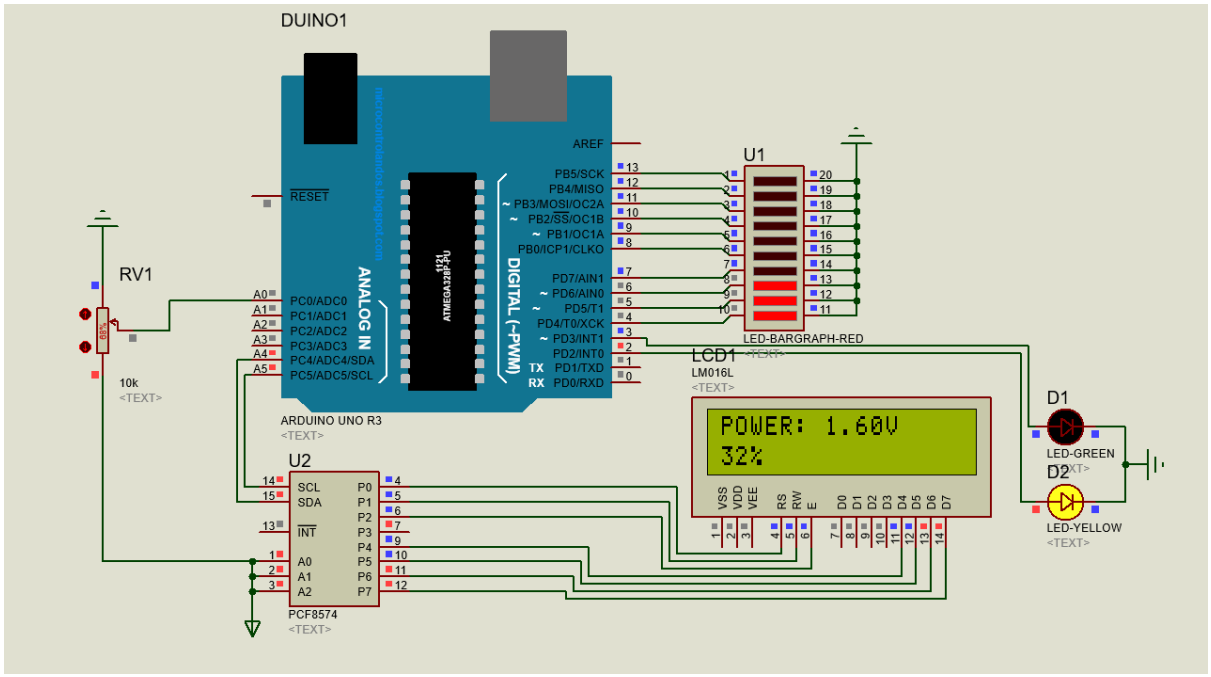
1. **Keterbatasan Kecepatan**

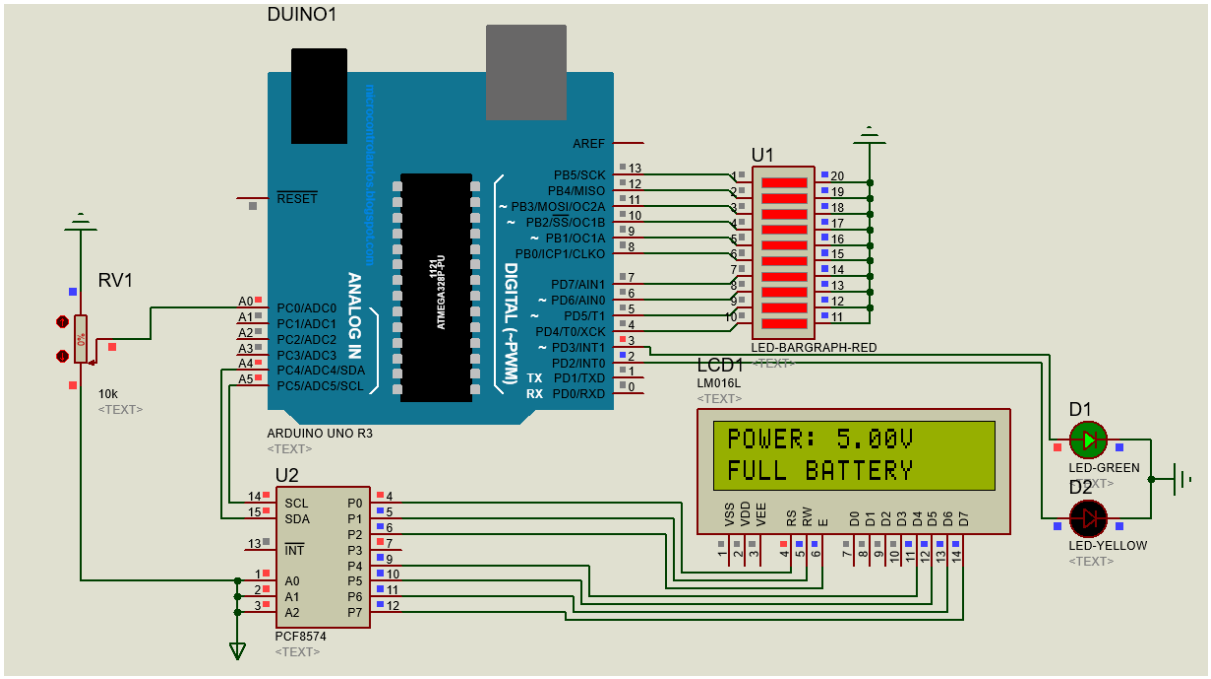
   ADC membutuhkan waktu untuk mengonversi, sehingga tidak ideal untuk sinyal analog yang berubah sangat cepat.

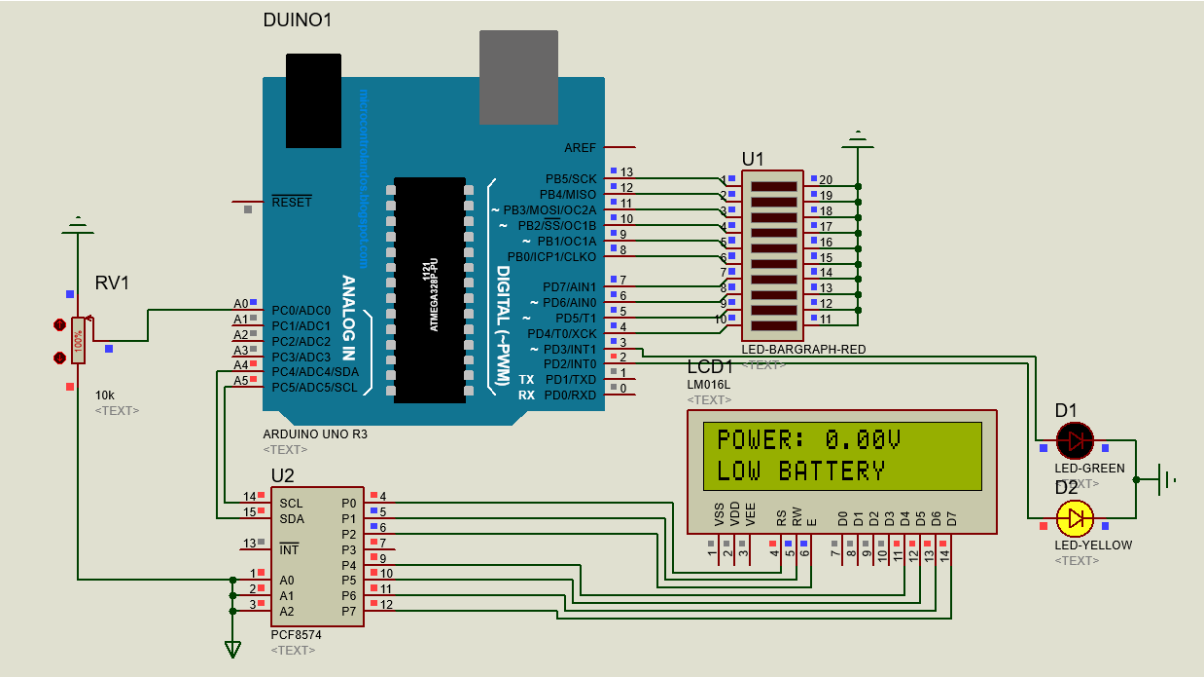
2. **Ketergantungan terhadap Vreferensi**

   Akurasi pembacaan sangat bergantung pada kestabilan tegangan referensi. Ada sedikit kekacauan dalam gelombang listriknya,hasil output akan menjadi tidak sepenuhnya akurat

III. Rangkaian Percobaan







Untuk detail program dan file project proteus nya, saya lampirkan juga pada link dibawah ini

G-DRIVE || **GITHUB**

IV. Kebutuhan Komponen

DAFTAR TAGIHAN ARDUINO ADC BATTERY LEVEL

Bill Of Materials For

Design Title
Author
Document Number
Revision
Design Created Sunday, December 15, 2024
Design Last Modified Sunday, December 15, 2024
Total Parts In Design 7

0 Modules

Quantity	References	Value	Stock Code	Unit Cost
Sub-totals:				$0.00

0 Capacitors

Quantity	References	Value	Stock Code	Unit Cost
Sub-totals:				$0.00

0 Resistors

Quantity	References	Value	Stock Code	Unit Cost
Sub-totals:				$0.00

2 Integrated Circuits

Quantity	References	Value	Stock Code	Unit Cost
1	U1	LED-BARGRAPH-RED		
1	U2	PCF8574		
Sub-totals:				$0.00

0 Transistors

Quantity	References	Value	Stock Code	Unit Cost
Sub-totals:				$0.00

2 Diodes

Quantity	References	Value	Stock Code	Unit Cost
1	D1	LED-GREEN		
1	D2	LED-YELLOW		
Sub-totals:				$0.00

3 Miscellaneous

Quantity	References	Value	Stock Code	Unit Cost
1	DUINO1	ARDUINO UNO R3		
1	LCD1	LM016L		
1	RV1	10k		
Sub-totals:				$0.00

Totals: $0.00

Sunday, December 15, 2024 11:05:49 PM

MADE BY KITTYAWN AKA. ALVIAN

V. PRATINJAU PROGRAM

```
// Code made with luv by Kittyawn 🤍 aka. Alvian
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define NUM_LEDS 10
const int ledPins[NUM_LEDS] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
LiquidCrystal_I2C lcd(0x27, 16, 2);

#define LEDLOW 2
#define LEDHIGH 3

float tegangan = 0.0;
int persen = 0;
unsigned long lastBlinkTime = 0;
bool ledLowState = false;

void setup() {
    lcd.begin(16, 2);
    lcd.backlight();

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

    pinMode(LEDLOW, OUTPUT);
    pinMode(LEDHIGH, OUTPUT);

    digitalWrite(LEDLOW, HIGH);
}

void loop() {
    unsigned long currentTime = millis();

    if (currentTime - lastBlinkTime >= 1000) {
        lastBlinkTime = currentTime;
        ledLowState = !ledLowState;
        digitalWrite(LEDLOW, ledLowState ? HIGH : LOW);
    }

    int nilaiAnalog = analogRead(A0);
    tegangan = nilaiAnalog * 5.0 / 1023.0;
    persen = (tegangan / 5.0) * 100;

    lcd.setCursor(0, 0);
```

```
    lcd.print("POWER: ");
    lcd.print(tegangan, 2);
    lcd.print("V ");

    lcd.setCursor(0, 1);
    if (persen == 0) {
        lcd.print("LOW BATTERY    ");
    } else if (persen >= 100) {
        lcd.print("FULL BATTERY   ");
    } else {
        lcd.print(persen);
        lcd.print("%           ");
    }

    int aktifLed = map(nilaiAnalog, 0, 1023, 0, NUM_LEDS);
    for (int i = 0; i < NUM_LEDS; i++) {
        if (i < aktifLed) {
            digitalWrite(ledPins[i], HIGH);
        } else {
            digitalWrite(ledPins[i], LOW);
        }
    }

    if (persen >= 100) {
        digitalWrite(LEDLOW, LOW);
        digitalWrite(LEDHIGH, HIGH);
    } else {
        digitalWrite(LEDHIGH, LOW);
    }
}
```