

**UNIVERSIDADE DO VALE DO RIO DOS SINOS (UNISINOS)
GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

**MATHEUS DE FREITAS WEBER
GABRIEL BERWANGER SILVEIRA**

**EXERCÍCIO - AULA 04
Conversor A/D**

**São Leopoldo
2025**

MATHEUS DE FREITAS WEBER
GABRIEL BERWANGER SILVEIRA

EXERCÍCIO - AULA 04
Conversor A/D

Trabalho apresentado para a matéria
Circuitos microprocessados pelo Curso
de Engenharia de Controle e Automação
e Engenharia da Computação da Uni-
versidade do Vale do Sinos (UNISINOS),
ministrada pelo Prof. Jean Schmith.

São Leopoldo
2025

Sumário

1	Introdução	3
2	Metodologia	4
2.1	Materiais Utilizados	4
2.2	Diagrama de Conexões	4
2.3	Fluxograma	6
2.4	Código Fonte	7
3	Resultados	7
4	Conclusão	9
4.0.1	Link do TinkerCAD	9

1 Introdução

A comunicação serial entre microcontroladores é uma técnica fundamental em sistemas embarcados, permitindo a troca eficiente de dados entre dispositivos. A arquitetura Cliente-Servidor é amplamente utilizada para organizar essa comunicação, onde um dispositivo atua como cliente, solicitando serviços ou dados, enquanto o outro atua como servidor, respondendo às solicitações do cliente. Este trabalho tem como objetivo implementar um sistema de comunicação serial entre dois microcontroladores utilizando a arquitetura Cliente-Servidor, com o intuito de controlar e monitorar o estado de LEDs através de botões físicos.

2 Metodologia

2.1 Materiais Utilizados

- 1x Placas Arduino Uno
- 1x Protoboard
- 1x Resistores de 220 ohms
- 1x Fotorresistor (LDR)
- Fios de conexão (jumpers)
- Computador com software Arduino IDE instalado

2.2 Diagrama de Conexões

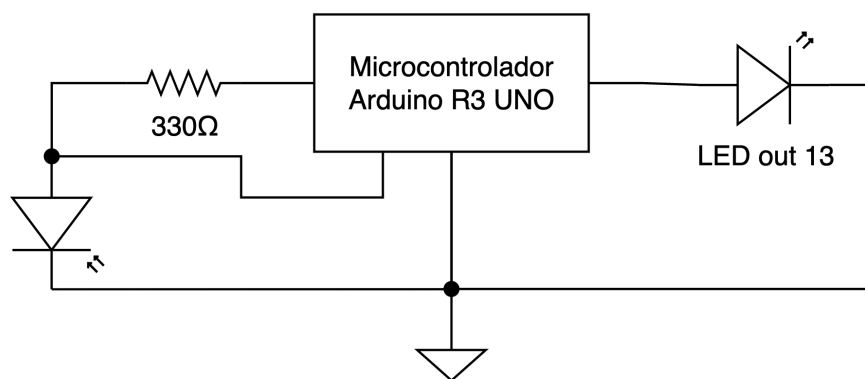
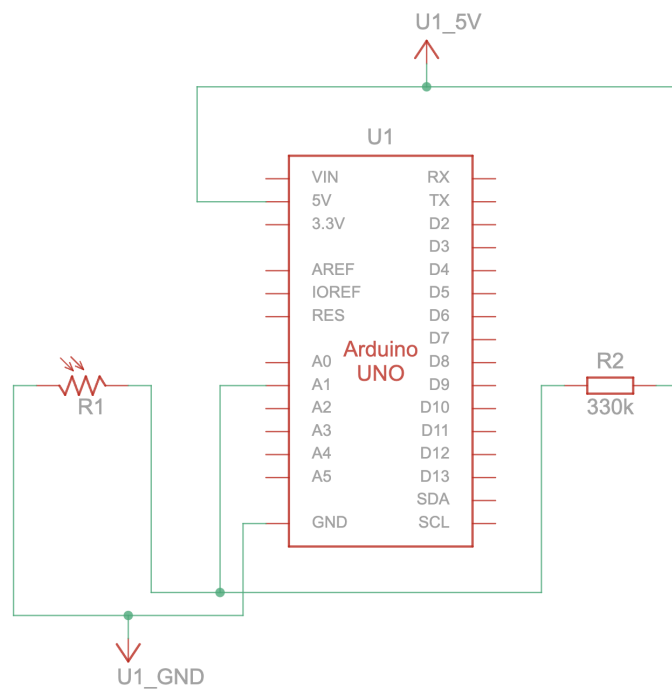
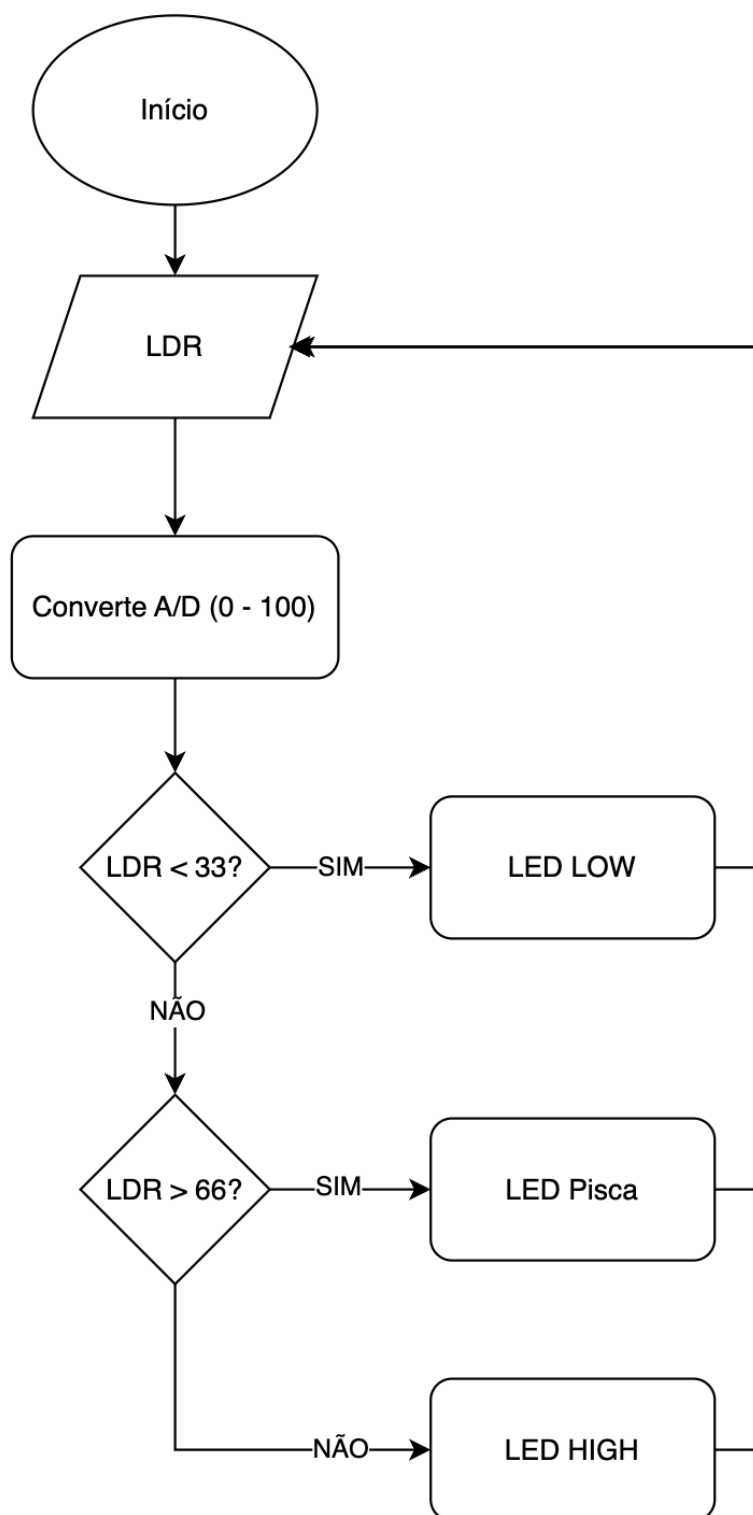


Diagrama de conexões entre os componentes e o Arduino Uno.



Montagem no TinkerCAD.

2.3 Fluxograma



Fluxograma do funcionamento do sistema.

2.4 Código Fonte

Código.

```
// C++ code
//

#define maxAnalogInput 1024
#define ledPin 13

bool ledStatus = false;

void desligaled() { digitalWrite(ledPin, LOW); }

void ligaled() { digitalWrite(ledPin, HIGH); }

void piscaled() {
    if (!ledStatus) {
        desligaled();
    } else {
        ligaled();
    }
    ledStatus = !ledStatus;
}

void setup() {
    pinMode(A1, INPUT);
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    int leitura = analogRead(A1);
    int mapLeitura = map(leitura, 0, maxAnalogInput, 0, 100);
    Serial.println(mapLeitura);
    if (mapLeitura < 33)
        desligaled();
    else if (mapLeitura > 66)
        piscaled();
    else
        ligaled();
    delay(500);
}
```

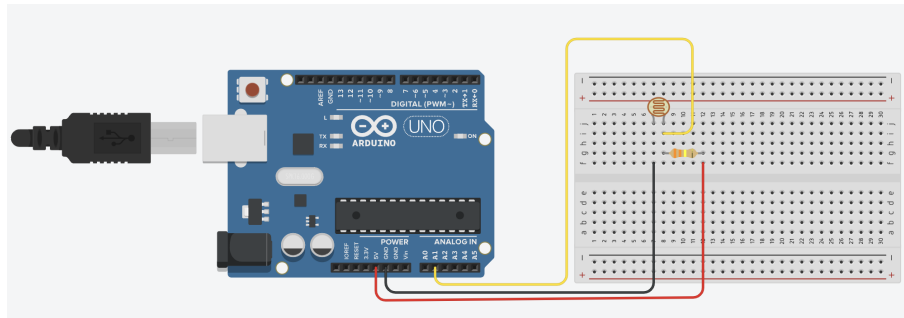
3 Resultados

O conversor A/D foi implementado com sucesso utilizando o microcontrolador Arduino Uno. A leitura analógica do fotoresistor (LDR) foi realizada na porta A1, e os valores foram mapeados para uma escala de 0 a 100. Com base nesses valores, o estado do LED conectado ao pino 13 foi controlado conforme as seguintes condições:

- Se o valor lido for menor que 33, o LED será desligado.
- Se o valor lido estiver entre 33 e 66, o LED será ligado.

- Se o valor lido for maior que 66, o LED piscará.

O sistema respondeu adequadamente às variações de luz detectadas pelo LDR, demonstrando a eficácia do conversor A/D na leitura de sinais analógicos e no controle do estado do LED.



Resultado da leitura analógica e controle do LED.

4 Conclusão

O projeto de conversor A/D utilizando o microcontrolador Arduino Uno foi concluído com sucesso, demonstrando a capacidade do sistema em ler sinais analógicos e controlar dispositivos digitais com base nessas leituras. A implementação mostrou-se eficaz para aplicações que requerem monitoramento de variáveis ambientais, como luminosidade, e controle de atuadores, como LEDs. Futuramente, este sistema pode ser expandido para incluir múltiplos sensores e atuadores, além de integrar comunicação com outros dispositivos para aplicações mais complexas em sistemas embarcados.

4.0.1 Link do TinkerCAD

<https://www.tinkercad.com/things/3dWNFKCmIui-epic-habbi-bigery/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard&sharecode=99L9c9x7W0UA92vSro-QPvIvRXCA00ue0CbV7t59LMk>