

EDASIM – KIT DE DESENVOLVIMENTO ARDUINO

Manual do Usuário e Guia de Projetos

Versão 1.0 – Revisão editorial em português
Baseada no manual fornecido para o Kit EDASIM.

Apresentação

O EDASIM – Kit de Desenvolvimento Arduino foi desenvolvido para facilitar o aprendizado de programação, eletrônica e sistemas embarcados por meio de atividades práticas. A placa reúne, em uma única plataforma, uma placa compatível com Arduino e diversos sensores, módulos de entrada e dispositivos de saída.

Os módulos são integrados à placa por meio de conexões na própria PCB. Dessa forma, muitos experimentos podem ser realizados sem a necessidade de cabos adicionais. Quando necessário, os módulos podem ser separados e utilizados individualmente por meio de cabos compatíveis.

Este manual apresenta uma sequência progressiva de atividades, começando pelos conceitos básicos de entradas e saídas digitais e avançando para sensores analógicos, comunicação I²C, display OLED e projetos integrados.

Objetivos de aprendizagem

- Compreender os fundamentos de sistemas de hardware de código aberto.
- Aprender conceitos básicos de programação com Arduino.
- Compreender princípios de leitura e comunicação com sensores.
- Desenvolver programas para controlar dispositivos de saída.
- Aplicar os conhecimentos adquiridos em projetos práticos.
- Explorar a integração entre sensores, atuadores e software.

Conteúdo do kit

Categoria	Módulo	Quantidade
Sensores	Sensor de temperatura e umidade	1
Sensores	Acelerômetro de 3 eixos	1
Sensores	Sensor de pressão atmosférica	1
Sensores	Sensor de luz	1
Sensores	Sensor de som	1
Entradas	Potenciômetro rotativo	1
Entradas	Botão	1
Saídas	LED	1
Saídas	Buzzer	1
Display	OLED 0,96"	1
Acessórios	Cabos	6
Acessórios	Cabo USB-C	Conforme kit fornecido

Visão geral do hardware

A placa EDASIM integra os principais recursos necessários para os experimentos apresentados neste manual. Os módulos são conectados internamente à placa e podem ser utilizados diretamente, sem a necessidade de cabos, quando permanecerem na configuração original da PCB.

Quando os módulos forem separados da placa, utilize os cabos apropriados e confirme sempre a identificação dos pinos antes de realizar a conexão.

Mapa de conexões padrão

Módulo	Interface	Pino / Endereço
LED	Digital	D4
Buzzer	Digital / PWM	D5
Display OLED 0,96"	I ² C	0x78 (padrão, conforme manual-base)
Botão	Digital	D6
Potenciômetro rotativo	Analógica	A0
Sensor de luz	Analógica	A6
Sensor de som	Analógica	A2
Temperatura e umidade	Digital	D3
Pressão atmosférica	I ² C	0x77 (padrão) / 0x76 (opcional)
Acelerômetro de 3 eixos	I ² C	0x63 (conforme manual-base)

Observação: Os endereços e pinagens foram preservados do documento-base e devem ser validados com o hardware EDASIM definitivo antes da publicação da versão final.

Separação dos módulos

Caso seja necessário utilizar um módulo separadamente, siga o procedimento de separação indicado para a PCB. Tome cuidado ao utilizar ferramentas cortantes e utilize proteção adequada.

1. Identifique os pontos de união entre o módulo e a placa principal.
2. Utilize uma ferramenta apropriada para romper cuidadosamente os pontos de união da PCB.
3. Após a separação, verifique se não houve danos às trilhas ou aos conectores.
4. Utilize um cabo compatível para realizar a conexão do módulo à placa controladora.

Demonstração inicial – Plug and Play

O kit possui uma demonstração inicial que permite explorar diversos recursos da placa sem a necessidade de iniciar imediatamente a programação. Ao alimentar a placa, o usuário pode navegar entre diferentes demonstrações utilizando o potenciômetro rotativo e o botão.

- Navegar: gire o potenciômetro rotativo.
- Selecionar: pressione rapidamente o botão.
- Sair da demonstração atual: mantenha o botão pressionado.
- LED e buzzer: utilizados como indicadores durante a interação.

Como começar com Arduino

1. Instalação da Arduino IDE

A Arduino IDE é o ambiente de desenvolvimento utilizado para criar, compilar e enviar programas para placas compatíveis com Arduino.

5. Baixe e instale a versão da Arduino IDE compatível com o seu sistema operacional.
6. Abra a Arduino IDE após a instalação.
7. Conecte a placa EDASIM ao computador utilizando o cabo USB.
8. Selecione a placa compatível com o hardware EDASIM.
9. Selecione a porta serial correspondente à placa.

2. Instalação do driver USB

A comunicação entre a placa e o computador depende do circuito conversor USB instalado no hardware. O manual-base faz referência ao conversor CP2102 e ao respectivo driver. Essa informação deve ser confirmada de acordo com a versão definitiva da placa EDASIM.

Observação: Caso o computador não reconheça a placa, verifique o driver USB, o cabo utilizado e a porta selecionada na Arduino IDE.

3. Primeiro programa – Hello World

O primeiro programa utiliza a comunicação serial para enviar uma mensagem ao computador. Esse exercício permite verificar se a placa está corretamente configurada e se o programa pode ser compilado e carregado.

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Ola, mundo!");  
    delay(1000);  
}
```

Após carregar o programa, abra o Monitor Serial e configure a velocidade para 9600 baud. A mensagem deverá aparecer aproximadamente uma vez por segundo.

Guia de atividades

Aula 1 – Controle de um LED

Nesta atividade, você aprenderá a controlar uma saída digital utilizando o LED integrado ao kit. O objetivo é fazer o LED ligar e desligar em intervalos regulares.

Objetivos

- Configurar um pino como saída.
- Utilizar digitalWrite().
- Compreender os estados HIGH e LOW.
- Utilizar delay() para controlar o tempo.

```
const int ledPin = 4;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

O LED permanecerá ligado durante aproximadamente um segundo e desligado durante aproximadamente um segundo, repetindo continuamente.

Aula 2 – Controle do LED com um botão

Nesta atividade, o botão será utilizado como entrada digital. Quando o botão for acionado, o programa controlará o LED de acordo com o estado lido na entrada.

```
const int buttonPin = 6;
const int ledPin = 4;
int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

A função digitalRead() realiza a leitura do estado lógico do botão. O programa utiliza uma estrutura condicional para decidir se o LED deve ser ligado ou desligado.

Aula 3 – Controle do LED com potenciômetro

O potenciômetro fornece uma leitura analógica que pode ser utilizada para controlar o comportamento do LED. Nesta atividade, a leitura do potenciômetro é convertida para uma faixa adequada ao controle por PWM.

```
int ledPin = 4;
int potentiometerPin = A0;
```

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
    int potentiometerValue = analogRead(potentiometerPin);  
    int outputValue = map(potentiometerValue, 0, 1023, 0, 255);  
    analogWrite(ledPin, outputValue);  
}
```

A função `analogRead()` lê o valor do potenciômetro. A função `map()` converte a faixa de leitura para uma faixa de 0 a 255, utilizada no controle PWM.

Aula 4 – Controle do buzzer

O buzzer é um dispositivo de saída que pode produzir sinais sonoros. Nesta atividade, o potenciômetro é utilizado para alterar o valor aplicado ao buzzer por PWM.

```
int buzzerPin = 5;  
int potentiometerPin = A0;  
  
void setup() {  
    pinMode(buzzerPin, OUTPUT);  
}  
  
void loop() {  
    int potentiometerValue = analogRead(potentiometerPin);  
    int outputValue = map(potentiometerValue, 0, 1023, 0, 255);  
    analogWrite(buzzerPin, outputValue);  
}
```

PWM significa Modulação por Largura de Pulso. A técnica controla a proporção de tempo em que um sinal permanece em nível alto e baixo. O resultado percebido depende do dispositivo e do circuito utilizado.

Aula 5 – Sensor de luz e controle do LED

O sensor de luz utiliza um elemento fotossensível para medir a intensidade luminosa do ambiente. A leitura analógica pode ser utilizada para controlar o brilho do LED.

```
int sensorPin = A6;
int ledPin = 4;

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

void loop() {
  int sensorValue = analogRead(sensorPin);
  int outputValue = map(sensorValue, 0, 1023, 0, 255);

  Serial.println(sensorValue);
  analogWrite(ledPin, outputValue);
  delay(30);
}
```

Abra o Monitor Serial para acompanhar os valores fornecidos pelo sensor. A função `map()` converte a faixa de leitura para o intervalo utilizado pelo PWM.

Observação: Dependendo da orientação elétrica do sensor e da forma como o circuito foi projetado, pode ser necessário inverter a relação entre luminosidade e brilho. Isso deve ser ajustado após validar o comportamento do hardware EDASIM.

Aula 6 – Controle do LED por som

Nesta atividade, o sensor de som será utilizado para detectar variações no nível sonoro do ambiente. Quando o valor medido ultrapassar um limite definido no programa, o LED será acionado.

```
int soundPin = A2;
int ledPin = 4;

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

void loop() {
  int soundState = analogRead(soundPin);
  Serial.println(soundState);

  if (soundState > 200) {
    digitalWrite(ledPin, HIGH);
    delay(100);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

O valor de 200 é apenas um limite inicial. Dependendo do sensor, do ambiente e do circuito, esse valor pode precisar de ajuste.

Aula 7 – Exibição de informações no display OLED

O display OLED permite apresentar informações diretamente no kit, sendo especialmente útil para visualizar valores de sensores e mensagens de status.

Bibliotecas

O manual-base utiliza a biblioteca U8g2/U8x8 para controlar o display. A biblioteca correspondente deve estar instalada na Arduino IDE antes da compilação do exemplo.

```
#include <Arduino.h>
#include <U8x8lib.h>

U8X8_SSD1306_128X64_ALT0_HW_I2C u8x8(U8X8_PIN_NONE);

void setup(void) {
    u8x8.begin();
    u8x8.setFlipMode(1);
}

void loop(void) {
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 0);
    u8x8.print("Ola, mundo!");
}
```

O programa inicializa o display, seleciona a fonte e posiciona o cursor para apresentar uma mensagem.

Aula 8 – Temperatura e umidade

O sensor DHT11 permite medir a temperatura e a umidade relativa do ambiente. Nesta atividade, os valores são apresentados no display OLED.

```
#include "DHT.h"
#include <Arduino.h>
#include <U8x8lib.h>

#define DHTPIN 3
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
U8X8_SSD1306_128X64_ALT0_HW_I2C u8x8(U8X8_PIN_NONE);

void setup(void) {
    Serial.begin(9600);
    dht.begin();
    u8x8.begin();
    u8x8.setPowerSave(0);
    u8x8.setFlipMode(1);
}

void loop(void) {
    float temp = dht.readTemperature();
    float humi = dht.readHumidity();

    u8x8.setFont(u8x8_font_chroma48medium8_r);

    u8x8.setCursor(0, 3);
    u8x8.print("Temp: ");
    u8x8.print(temp);
    u8x8.print(" C");

    u8x8.setCursor(0, 5);
    u8x8.print("Umid: ");
    u8x8.print(humi);
    u8x8.print(" %");
}
```



```
    delay(200);  
}
```

A função `readTemperature()` retorna a temperatura e `readHumidity()` retorna a umidade relativa. Os valores são armazenados em variáveis do tipo `float` e apresentados no display.

Observação: O endereço I²C e o controlador exato do display devem ser confirmados na placa EDASIM definitiva.

Aula 9 – Medição da pressão atmosférica

O sensor BMP280 é um sensor digital utilizado para medir a pressão atmosférica e a temperatura. A partir da pressão, também é possível estimar a altitude, desde que sejam utilizados parâmetros de referência adequados.

```
#include "Seeed_BMP280.h"
#include <Wire.h>

BMP280 bmp280;

void setup() {
  Serial.begin(9600);

  if (!bmp280.init()) {
    Serial.println("Sensor nao conectado ou com falha!");
  }
}

void loop() {
  float pressure;

  Serial.print("Temperatura: ");
  Serial.print(bmp280.getTemperature());
  Serial.println(" C");

  Serial.print("Pressao: ");
  pressure = bmp280.getPressure();
  Serial.print(pressure);
  Serial.println(" Pa");

  Serial.print("Altitude estimada: ");
  Serial.print(bmp280.calcAltitude(pressure));
  Serial.println(" m");

  Serial.println();
  delay(1000);
}
```

O programa inicializa o sensor e envia os valores de temperatura, pressão e altitude estimada para o Monitor Serial.

Aula 10 – Detecção de movimento

O acelerômetro de três eixos permite detectar movimentos e medir a aceleração nos eixos X, Y e Z. Essa informação pode ser utilizada em projetos de monitoramento de movimento, inclinação e vibração.

```
#include "LIS3DHTR.h"
#include <Wire.h>

LIS3DHTR<TwoWire> LIS(I2C_MODE);

void setup() {
  Serial.begin(9600);
  LIS.begin(Wire);
  delay(100);
  LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);
}

void loop() {
  if (!LIS) {
    Serial.println("Acelerometro nao conectado.");
    while (1);
  }

  Serial.print("X: ");
```

```
Serial.print(LIS.getAccelerationX());  
Serial.print(" Y: ");  
Serial.print(LIS.getAccelerationY());  
Serial.print(" Z: ");  
Serial.println(LIS.getAccelerationZ());  
  
delay(500);  
}
```

Os valores de aceleração dos três eixos são enviados ao Monitor Serial. A partir dessas informações, é possível desenvolver aplicações de detecção de movimento.

Projetos bônus

Projeto 1 – Lâmpada rítmica com música

Neste projeto, o buzzer reproduz uma sequência musical enquanto o LED acompanha o ritmo. A atividade combina controle de áudio, temporização e acionamento de uma saída digital.

O código-base utiliza uma tabela de frequências para representar as notas musicais e uma tabela de duração para controlar o ritmo. A função `tone()` gera o sinal sonoro e o LED é acionado durante a reprodução.

```
int tonePin = 5;
int ledPin = 4;

void setup() {
  pinMode(tonePin, OUTPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Inserir aqui a sequencia de notas e duracoes
  // conforme o projeto musical desejado.
}
```

Observação: O documento-base contém uma sequência extensa de notas. Na versão final, recomenda-se manter o código completo original somente após sua validação no hardware EDASIM.

Projeto 2 – Lâmpada inteligente com sensores de som e luz

O objetivo deste projeto é criar uma pequena lâmpada automática utilizando o sensor de som e o sensor de luz. O LED será acionado quando houver um nível sonoro acima do limite definido ou quando a luminosidade ambiente estiver abaixo de um determinado valor.

```
int soundPin = A2;
int lightPin = A6;
int ledPin = 4;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int soundState = analogRead(soundPin);
  int lightState = analogRead(lightPin);

  if (soundState > 50 || lightState < 10) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Os limites utilizados são exemplos e podem precisar de ajuste conforme as características do sensor e do ambiente.

Conceitos fundamentais

Entradas e saídas digitais

Uma entrada digital normalmente representa dois estados lógicos: HIGH e LOW. Botões e chaves podem ser utilizados como entradas, enquanto LEDs e outros atuadores podem ser utilizados como saídas.

Entradas analógicas

Entradas analógicas permitem medir sinais que variam dentro de uma faixa de valores. Sensores de luz, som e potenciômetros são exemplos de dispositivos que podem fornecer leituras analógicas.

PWM – Modulação por largura de pulso

PWM é uma técnica utilizada para controlar dispositivos por meio de um sinal digital pulsado. Alterando o ciclo de trabalho, é possível controlar, por exemplo, o brilho de um LED ou a resposta de determinados atuadores.

Comunicação I²C

I²C é um protocolo de comunicação serial que utiliza, normalmente, duas linhas principais: SDA, para dados, e SCL, para o relógio. Um controlador pode se comunicar com diferentes dispositivos conectados ao mesmo barramento, utilizando endereços para identificar cada dispositivo.

Monitor Serial

O Monitor Serial é uma ferramenta importante para acompanhar os dados enviados pela placa ao computador. Ele pode ser utilizado para visualizar leituras de sensores e auxiliar na identificação de problemas durante o desenvolvimento.

Serial Plotter

O Serial Plotter permite visualizar graficamente dados enviados pela porta serial. É especialmente útil para observar variações de sensores em tempo real.

Solução de problemas

- A placa não é reconhecida: verifique o cabo USB, o driver e a porta selecionada.
- O programa não é carregado: confirme a placa selecionada e a porta serial.
- O sensor não responde: verifique a alimentação, a conexão e a biblioteca utilizada.
- O display não apresenta dados: confirme o controlador e o endereço I²C.
- Os valores do sensor parecem incorretos: verifique a calibração e os limites utilizados no programa.
- O código apresenta erro de compilação: confirme se todas as bibliotecas necessárias estão instaladas.

Próximos passos

Depois de concluir as atividades deste manual, o usuário já possui uma base para desenvolver projetos próprios. A partir dos exemplos apresentados, é possível combinar sensores e atuadores, criar interfaces de usuário, desenvolver sistemas de automação e explorar novas bibliotecas e módulos compatíveis com Arduino.

A placa EDASIM também pode ser utilizada como plataforma de experimentação para novos circuitos e módulos, permitindo que o estudante avance gradualmente do aprendizado de programação para o desenvolvimento de projetos eletrônicos completos.

Observações para a próxima versão

Esta primeira versão foi preparada como uma revisão editorial e didática do conteúdo disponível no manual-base. As imagens existentes podem ser mantidas e substituídas posteriormente pelas novas imagens preparadas para o produto.

- Validar a pinagem definitiva da placa EDASIM.
- Confirmar o microcontrolador e a placa selecionada na Arduino IDE.
- Confirmar o conversor USB e o driver correspondente.
- Validar os endereços I²C de todos os módulos.
- Testar todos os códigos no hardware EDASIM.
- Substituir referências remanescentes ao hardware do fabricante original.
- Inserir novas fotografias e diagramas próprios.
- Revisar e padronizar as bibliotecas utilizadas.
- Adicionar QR Codes para códigos-fonte e materiais complementares.