

Sistemas Microcontrolados

Experimento 04 - Saída analógica

Aluno: Deivid da Silva Galvão RA: 2408740

Aluno: Eduardo Yuji Yoshida Yamada RA: 2320606

1 Introdução

Neste experimento, exploramos a interface entre sinais temporizados e dispositivos de saída analógicos utilizando um microcontrolador Arduino. O objetivo principal foi desenvolver um sistema capaz de modular a intensidade luminosa de três LEDs de forma independente e com fases distintas. Para alcançar esse objetivo, utilizamos a função `millis()` para gerar temporizações não bloqueantes e distribuir atrasos específicos a cada LED, permitindo que eles operem com diferentes fases de variação de brilho, criando um efeito visual sequencial.

2 Materiais e Métodos

2.1 *Materiais Utilizados*

- 3 LEDs brancos;
- 3 resistências de 220 Ω ;
- Jumpers (fios para conexão);
- Placa Arduino Mega 2560;
- Protoboard para montagem do circuito.

2.2 *Esquemático do Circuito*

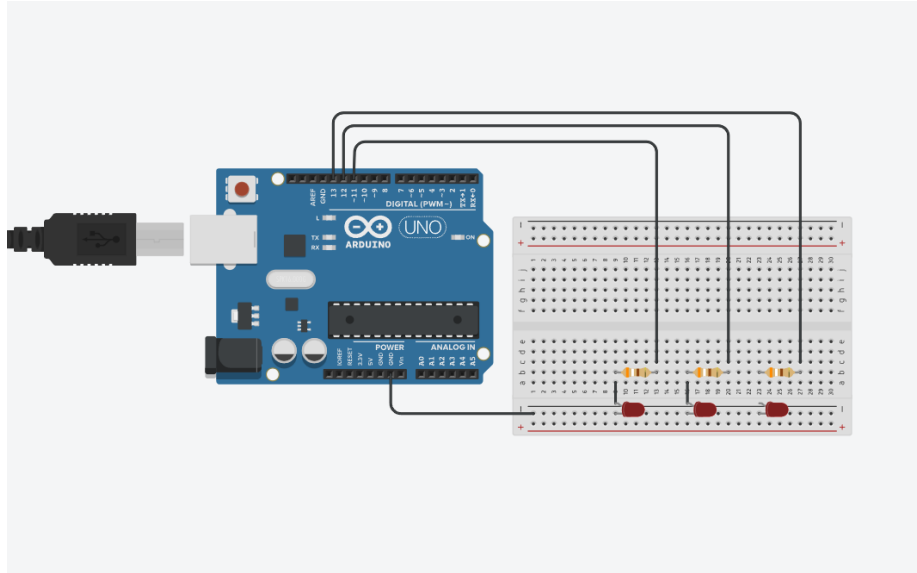


Figure 1: Circuito montado no Tinkercad

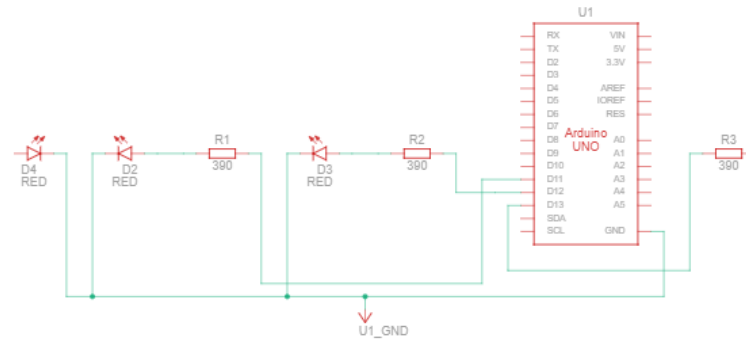


Figure 2: Diagrama esquemático do circuito no Tinkercad

2.3 Código Fonte Comentado

// Definicao dos pinos para cada LED

```

int LED1 = 11;
int LED2 = 12;
int LED3 = 13;

// Variaveis para controle temporal
unsigned long tempo1, tempo2, tempo3;

void setup() {
    // Configura os pinos como saida
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}

void loop() {
    // Calcula os tempos com diferentes fases
    tempo1 = (millis() % 600);           // LED1 sem defasagem
    tempo2 = ((millis() - 200) % 600);   // LED2 com defasagem
                                         // de 200ms
    tempo3 = ((millis() - 400) % 600);   // LED3 com defasagem
                                         // de 400ms

    // Chama as funcoes de controle para cada LED
    controleLED1();
    controleLED2();
    controleLED3();
}

// Funcao para controle do LED1
void controleLED1() {
    if (tempo1 < 300) {
        // Aumenta o brilho linearmente nos primeiros 300ms
        analogWrite(LED1, map(tempo1, 0, 300, 0, 255));
    } else {
        // Diminui o brilho linearmente nos proximos 300ms
        analogWrite(LED1, map(tempo1, 300, 600, 255, 0));
    }
}

// Funcao para controle do LED2
void controleLED2() {
    if (tempo2 < 300) {
        analogWrite(LED2, map(tempo2, 0, 300, 0, 255));
    } else {
        analogWrite(LED2, map(tempo2, 300, 600, 255, 0));
    }
}

```

```

// Funcao para controle do LED3
void controleLED3() {
    if (tempo3 < 300) {
        analogWrite(LED3, map(tempo3, 0, 300, 0, 255));
    } else {
        analogWrite(LED3, map(tempo3, 300, 600, 255, 0));
    }
}
}

```

Listing 1: Código para controle PWM dos LEDs

3 Resultados

O sistema implementado demonstrou um controle eficiente da intensidade luminosa dos três LEDs de forma independente. Através da análise do código e da observação prática, podemos descrever os seguintes aspectos:

- **Inicialização:** As variáveis são declaradas associando cada LED a um pino digital específico (11, 12 e 13) e definindo variáveis temporais do tipo `unsigned long` para armazenar os valores de temporização.
- **Configuração:** Na função `setup()`, os pinos correspondentes aos três LEDs são configurados como saídas utilizando a função `pinMode()`.
- **Ciclo principal:** Na função `loop()`, utiliza-se a função `millis()` para criar ciclos de 600 milissegundos com diferentes defasagens:
 - *LED1*: sem deslocamento temporal (referência)
 - *LED2*: com defasagem de 200ms
 - *LED3*: com defasagem de 400ms
- **Controle PWM:** Cada função de controle verifica em qual metade do ciclo (0-300 ms ou 300-600 ms) o LED se encontra e ajusta o brilho proporcionalmente usando `analogWrite()` em conjunto com `map()`.

O efeito visual observado foi de três LEDs variando seu brilho de forma sequencial, criando uma onda de luz que percorria os LEDs em intervalos regulares. A abordagem não bloqueante permitiu que o microcontrolador pudesse executar outras tarefas simultaneamente, se necessário.

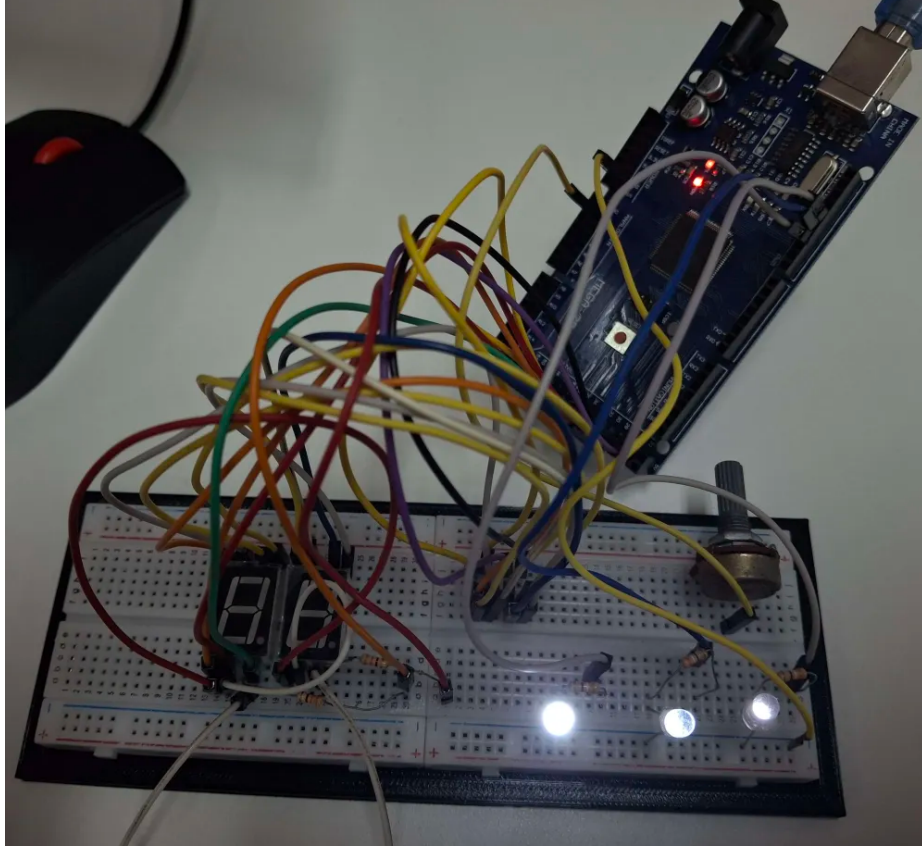


Figure 3: Montagem física do circuito no laboratório

4 Conclusão

Por fim, pode-se concluir que o experimento demonstrou de forma prática a aplicação de técnicas de modulação por largura de pulso (*PWM*) combinadas com temporizações dinâmicas para o controle de intensidade luminosa de *LEDs*. Ao utilizar a função `millis()` com diferentes deslocamentos temporais, foi possível implementar de maneira eficaz o controle progressivo da intensidade dos *LEDs*, proporcionando um efeito de aumento gradual e redução gradual do brilho, evidenciando o potencial do Arduino para gerar saídas visuais intercaladas de forma não bloqueante. Essa abordagem reforça a importância da integração entre hardware e software em sistemas microcontrolados, servindo como base para o desenvolvimento de aplicações mais complexas em controle de iluminação.