

Saída Analógica

Experimento 2

1st Iago Macarini Brito

Engenharia de Computação

Universidade Tecnológica Federal do Paraná

Apucarana, Brasil

iagomacarini@alunos.utfpr.edu.br

2nd Luis Henrique Ferracciu Pagotto Mendes

Engenharia de Computação

Universidade Tecnológica Federal do Paraná

Apucarana, Brasil

luis.mendes.2020@alunos.utfpr.edu.br

Resumo—Este relatório apresenta o experimento para verificar saídas analógicas utilizando um Arduino Mega 2560 a fim de simular um sistema de controle de brilho de três LEDs usando *Pulse-Width Modulation* (PWM) e temporização não bloqueante. Cada LED executa rampas de aumento e redução no brilho, tendo inícios em intervalos de tempos diferentes, havendo sobreposição de transições.

Palavras-Chave—Arduino, Diodo, PWM, Microcontrolador, Saída analógica

I. INTRODUÇÃO

O relatório em questão se trata em desenvolver um experimento para compreender o funcionamento do *Pulse-Width Modulation* (PWM), para controlar, de forma suave, o brilho de três LEDs com lógica de tempo não bloqueante. Este método aplica rampas de subida e descida em intensidades luminosas sobrepostas, fazendo com que cada LED tenha seu ciclo de brilho, em que, enquanto um está no processo de diminuição, o outro está aumentando o brilho.

A aplicação desta função, no desenvolvimento para Arduino, utiliza a função `millis()`, que funciona executando diferentes tarefas simultaneamente sem interromper o fluxo principal do código. Desta forma, cada LED pode atuar independentemente e ter seu próprio ciclo sem a interferência de outro LED.

II. MATERIAIS E MÉTODOS

Para a realização deste experimento e verificação dos resultados, foram utilizados os materiais a seguir:

A. Materiais Utilizados

- 1 Arduino Mega 2560;
- 1 Protoboard;
- 1 Fonte de alimentação USB 5V.
- 3 Diodos Emissores de Luz (LEDs);
- 3 Resistores de 330Ω ;
- Jumpers macho-macho;

B. Montagem do Sistema

A implementação deste sistema foi realizada com o uso de 3 LEDs para representar a diferença de tempo entre as subidas e descidas independentes, controlados diretamente pelo Arduino e controlados com PWM. A montagem do circuito foi realizada ligando o ânodo de cada diodo, com um

resistor de 330Ω (instalados na protoboard) entre as portas PWM da placa: 13, 12 e 11, respectivamente e o cátodo dos mesmo ao GND da placa.

Com a montagem finalizada e os componentes já posicionados, o código a seguir foi utilizado para o controle do brilho dos LEDs, fazendo com que eles funcionasse da maneira solicitada na atividade.

C. Código Implementado

A partir do código a seguir, é possível verificar que as declarações "led1", "led2" e "led3" representam as portas que foram utilizadas para controlar os LEDs. Logo abaixo, estão as declarações que indicam o tempo de início do ciclo de cada diodo (**intervalo**), o tempo de subida da rampa (**tempoAumento**) e o tempo de descida da rampa (**tempoReducao**), todos funcionando em milissegundos. O cálculo abaixo indica o tempo total entre os ciclos completos, funcionando com a soma entre o tempo de subida e descida.

Já, na função `loop()`, a primeira declaração (**tempo**), implementa o `millis()`, que faz com que cada LED tenha seu ciclo único sem a interferência entre cada um, seguido pela função **controlarFadeLED** que define a qual diodo a função se refere, determina que deve ser uma função utilizando PWM (a partir da variável `tempo`), e determina o início do ciclo de cada um, já com o atraso solicitado.

Por fim, está implementada a função **controlarFadeLED()** que realiza o cálculo a partir dos valores definidos na chamada de função, e faz com que os LEDs iniciem o ciclo em 0s, 200ms e 400ms, explicada no trecho acima.

```

const int led1 = 13;
const int led2 = 12;
const int led3 = 11;

const int intervalo = 200;
const int tempoAumento = 300;
const int tempoReducao = 300;
const int tempoCiclo = tempoAumento +
    tempoReducao;

void setup(){
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
}

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

    controlarFadeLED(led1, tempo, 0);
    controlarFadeLED(led2, tempo, intervalo);
    controlarFadeLED(led3, tempo, 2 * intervalo)
        ;
}

void controlarFadeLED(int pin, unsigned long
    tempoAtual, int atrasoInicio){
    unsigned long tempoRelativo=(tempoAtual-
        atrasoInicio)%(3*intervalo);

    int valorPWM=0;

    if (tempoRelativo<tempoAumento){
        // Fase de subida
        valorPWM=map(tempoRelativo, 0,
            tempoAumento, 0, 255);
    }
    else if (tempoRelativo<tempoCiclo){
        // Fase de descida
        valorPWM=map(tempoRelativo-tempoAumento,
            0, tempoReducao, 255, 0);
    }
    else{
        valorPWM = 0;
    }
    analogWrite(pin, valorPWM);
}

```

Código 1. Código desenvolvido para implementação do sistema

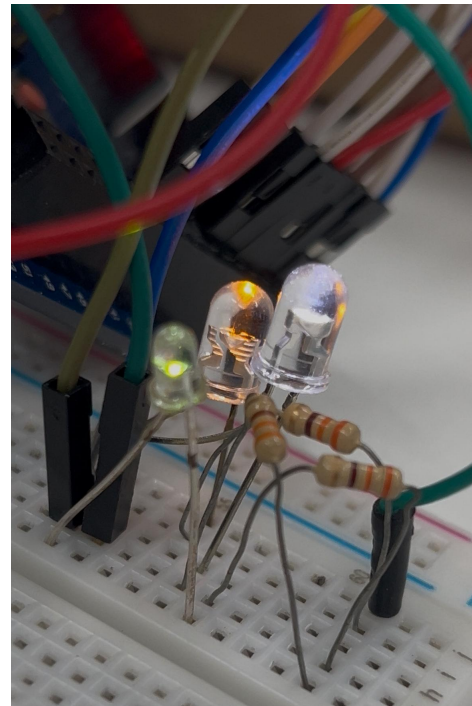


Figura 1. Foto do Circuito em Funcionamento

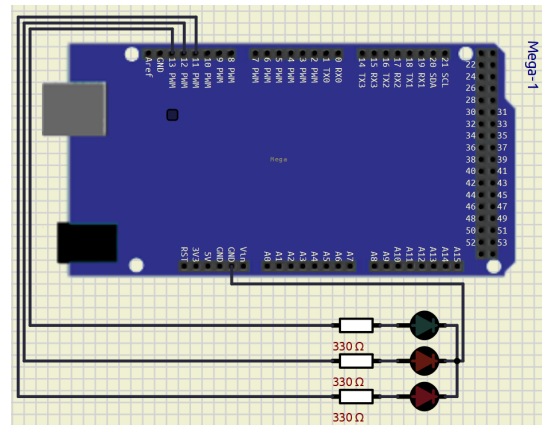


Figura 2. Esquema Elétrico Representativo do Circuito

III. CONCLUSÃO

Com a implementação deste experimento, foi possível determinar, de forma prática, a capacidade do Arduino Mega 2560 de controlar a luminosidade de três LEDs com a utilização de *Pulse-Width Modulation* (PWM) e da temporização não bloqueante, sem recorrer à função **delay()**. Ao aplicar rampas lineares de subida e descida de intensidade, com intervalos de 200ms entre cada, verificou-se que cada diodo completa seu ciclo dentro de 300ms, havendo sobreposição suave para as transições, garantindo fluxo contínuo. Com a aplicação do **millis()** foi demonstrada a eficácia ao sincronizar múltiplas saídas sem interrupções, mantendo o loop principal.