

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA TRIÂNGULO MINEIRO
CAMPUS AVANÇADO UBERABA PARQUE TECNOLÓGICO
ENGENHARIA DE COMPUTAÇÃO**

**Carlos Alexandre Sousa Silva
Erick Silva de Andrade
Matheus Machado Pontes**

**RELATÓRIO REFERENTE À MONTAGEM DE PROTÓTIPO
DA DISCIPLINA DE PROJETOS INTERDISCIPLINARES DE
ENGENHARIA**

**UBERABA
JUNHO, 2022**

SUMÁRIO

1	INTRODUÇÃO	3
1.1	Objetivos	4
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	Descrição da Situação	4
2.2	Proposta de Solução	4
3	PROTÓTIPO	5
3.1	Componentes	5
3.2	Esquemas de Ligação	5
3.3	Códigos	5
4	CONCLUSÃO	6
5	BIBLIOGRAFIA	7

1 INTRODUÇÃO

Com o avanço da eletrônica e microeletrônica nos últimos anos, a tecnologia de ponta se estende para o desenvolvimento de aplicações de hardware e software. As tecnologias de prototipação e desenvolvimento de software, frutos de tal avanço, são utilizadas para desenvolver aplicações de hardware, em um ambiente controlado, por computador, ou em um ambiente controlado por um dispositivo móvel. Nesse universo de prototipação, uma das placas mais reconhecidas é a Arduino, que é uma plataforma de desenvolvimento de hardware e software livre, open-source, que permite a criação de pequenos projetos que podem ser utilizados em qualquer lugar, somado ao fato de ser uma placa modularizada, que efetivamente reduz o custo de sua produção e ajudou na popularização de tecnologias de prototipação modulares. (ARDUINO.CC)

Nesse contexto, no presente projeto a utilização da placa Arduino Uno R3 é realizada para criar um sistema que replica o jogo "Genius", um jogo de estratégia e memorização, que tem como objetivo o jogador memorizar uma sequência de cores, dentre as 4 cores (Verde, Amarelo, Vermelho e Azul), geradas aleatoriamente a partir de uma "seed" que é passada como parâmetro para o programa.

A utilização de leds (*light-emitting diode*), tem o objetivo de apresentar ao jogador a sequência de cores, que deve ser decorada, para dar continuidade ao jogo. Esses estão montados em uma placa de prototipagem, conectada com cabo jumper e resistores, com objetivo de evitar a utilização da solda de componentes eletrônicos, geralmente feitas em estanho sem chumbo. Cada led possui uma chave tátil, do tipo "push-button", com a cor respectiva.

1.1 Objetivos

Apresentar uma nova visão, aos públicos entusiastas de microeletrônica, representado por um projeto de um jogo já reconhecido, em uma escala menor e com baixo custo. Visando a interação do usuário com a eletroeletrônica, através de prototipagem, e linguagens de programação de baixo nível.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Descrição da Situação

Neste projeto a linguagem de programação "C ++", foi utilizada para escrever o código fonte, que é interpretado pela placa Arduino, sendo uma linguagem de baixo nível que permite uma interação mais efetiva. Sendo necessário o conhecimento básico dentre eles: laços de repetições e decisões; funções; variáveis; estruturas; tipos básicos da linguagem; funções inerentes do arduino (como: digitalRead(), digitalWrite(), pinMode()).

Sendo isso somado aos conhecimentos básicos de elétrica e eletrônica, como as leis de ohm e conhecimento básico da interação dos materiais utilizados na montagem do projeto.

2.2 Solução Proposta

Ao utilizar a placa arduino como uma plataforma de prototipagem, foi possível reduzir o custo do produto, se comparado com o jogo original, criando uma possibilidade de aprendizado ao mesmo tempo que promove acessibilidade.

// acessibilidade e aprendizado diminuindo o custo

//essa seção para explicar como será realizada a **solução dos problema**, métodos de comparação entre os valores calculados e uma proposta da situação problema.

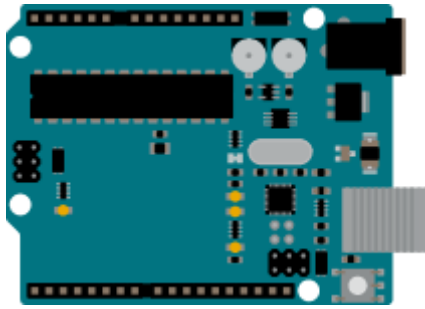
3 PROTÓTIPO

3.1 Componentes

A. Arduino

Arduino Uno R3 é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, desenvolvida com um microcontrolador ATmega328P com suporte de entrada/saída, com 14 portas digitais (com 6 saídas "PWM") e 6 portas analógicas, uma entrada USB e outra tipo power jack, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++. O objetivo do projeto é criar ferramentas que sejam acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e ferramentas mais complicadas. (ARDUINO.CC)

Figura 1 - Arduino Uno R3



Fonte: ARDUINO.CC (2022)

B. Placa de prototipagem

A placa de prototipagem permite montar circuitos eletrônicos de forma muito simples, já que não requer soldagem dos contatos.

C. Leds

O leds é um diodo emissor de luz, é usado para a emissão de luz em locais e instrumentos onde se torna mais conveniente a sua utilização no lugar de uma lâmpada.

D. Resistor

Um resistor é um dispositivo elétrico, com a finalidade de limitar a corrente elétrica em um circuito.

E. Cabo Jumper

Jumper é um pequeno condutor utilizado para conectar dois pontos de um circuito eletrônico. Geralmente utilizados para configurar placas de circuitos, como placas-mãe de computadores.

F. Chave tátil

Chave tátil é um dispositivo utilizado para controlar um processo. A chave tátil elétrica funciona fechando ou abrindo o circuito elétrico.

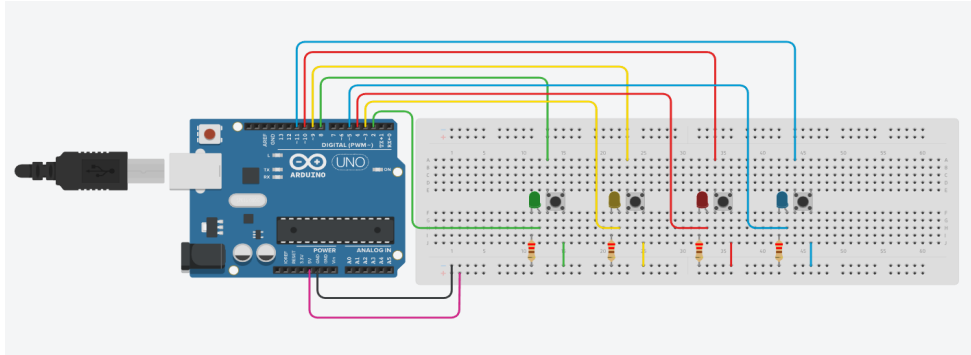
Placa de prototipagem

A placa de prototipagem permite montar circuitos eletrônicos de forma muito simples, já que não requer soldagem dos contatos.

3.2 Esquemas de Ligação

O esquema de ligação consiste da ligação de cabo jumper , do tipo macho-macho, na placa de prototipagem e na placa Arduino Uno R3, Figura 1. Com os leds ligados nas portas digitais de 2 a 5, e os botões nas portas de 8 a 11.

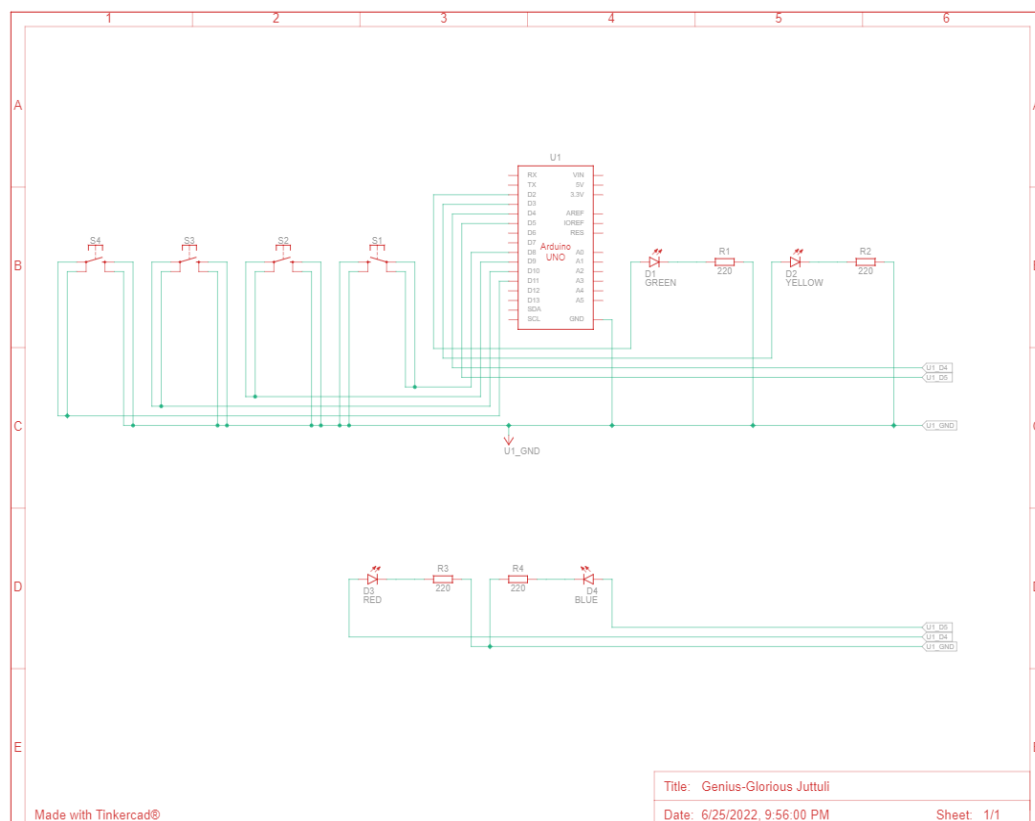
Figura 2 - Esquema geral do sistema



Fonte: AUTODESK TINKERCAD (2022)

No Esquema elétrico é representado de forma gráfica o circuito elétrico entre a placa e os demais componentes, onde os leds estão ligados de D2 a D5 e os botões de D8 e D11.

Figura 3 - Diagrama elétrico



Fonte: AUTODESK TINKERCAD (2022)

3.3 Códigos

Na figura 4 estão descritas as definições de variáveis do Arduino para leds, para botões, medidas de tempo para o jogo, e enumeração dos estados do jogo.

Figura 4 - Definição de variáveis do Arduino para leds, para botões, medidas de tempo para o jogo, e enumeração dos estados do jogo

```
// Define os pinos do arduino para os leds
#define LED_VERDE 2
#define LED_AMARELO 3
#define LED_VERMELHO 4
#define LED_AZUL 5

// Define os pinos do arduino para os botões
#define BOTAO_VERDE 8
#define BOTAO_AMARELO 9
#define BOTAO_VERMELHO 10
#define BOTAO_AZUL 11

// Define as medidas de tempo para o jogo
#define UM_SEGUNDO 1000
#define MEIO_SEGUNDO 500
#define QUARTO_SEGUNDO 250

// Tamanho do jogo
#define TAMANHO_SEQUENCIA 4

// Ind
#define INDEFINIDO -1

enum Estados { // Enumeração dos estados do jogo
    PRONTO_PARA_PROXIMA_RODADA,
    USUARIO_RESPONDENDO,
    JOGO_FINALIZADO_SUCESSO,
    JOGO_FINALIZADO_FALHA
};

int sequenciaLuzes[TAMANHO_SEQUENCIA];

int rodada = 0;

int ledsRespondidos = 0;
```

Fonte: silva (2022)

Na figura 5, temos duas funções que são necessárias para o início do jogo, e suas variáveis são visíveis no escopo do monitor serial, uma das funções do próprio Arduino.

Figura 5 - Inicializa o monitor serial do arduino e inicia duas funções necessárias para o jogo

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

    iniciaPortas();
    iniciaJogo();
}

void iniciaPortas(){
    pinMode(LED_VERDE, OUTPUT);
    pinMode(LED_AMARELO, OUTPUT);
    pinMode(LED_VERMELHO, OUTPUT);
    pinMode(LED_AZUL, OUTPUT);

    pinMode(BOTAO_VERDE, INPUT_PULLUP);
    pinMode(BOTAO_AMARELO, INPUT_PULLUP);
    pinMode(BOTAO_VERMELHO, INPUT_PULLUP);
    pinMode(BOTAO_AZUL, INPUT_PULLUP);
}

void iniciaJogo(){

    int jogo = analogRead(0); //Lê o valor do sensor de luz
    randomSeed(jogo); //Inicia o gerador de números aleatórios com o valor do sensor de luz

    for (int i = 0; i < TAMANHO_SEQUENCIA; i++) {
        sequenciaLuzes[i] = sorteiaCor(); //Sorteia uma cor e armazena na sequencia de luzes
    }
}
```

Fonte: silva (2022)

Na figura 6 é utilizado uma função nativa das linguagens de programação “C” e “C + +”, random(), que gera números pseudo aleatórios com base em uma “seed”.

Figura 6 - A função para sortear a cor, utilizando a função random(), gera números pseudo aleatórios.

```
int sorteiaCor(){
    return random(LED_VERDE, LED_AZUL + 1); // +1 para incluir o ultimo valor
}
```

Fonte: silva (2022)

Na figura 7 temos uma estrutura de switch case, visando facilitar a leitura, que é responsável por alterar os estágios do jogo, partindo da preparação da rodada, para a verificação de resposta do usuário e finaliza ao apresentar o resultado obtido pelo usuário.

Figura 7 - Troca os estados do jogo e determina o delay entre cada estado.

```
void loop(){
  switch (estadoAtual()){
    case PRONTO_PARA_PROXIMA_RODADA:
      Serial.println("Pronto para a proxima rodada");
      preparaNovaRodada();
      break;
    case USUARIO_RESPONDENDO:
      Serial.println("Usuario respondendo");
      processaRespostaUsuario();
      break;
    case JOGO_FINALIZADO_SUCESSO:
      Serial.println("Jogo finalizado com sucesso");
      jogoFinalizadoSucesso();
      break;
    case JOGO_FINALIZADO_FALHA:
      Serial.println("Jogo finalizado com falha");
      jogoFinalizadoFalha();
      break;
  }
  delay(QUARTO_SEGUNDO); // ou MEIO_SEGUNDO
}
```

Fonte: silva (2022)

Na figura 8 é preparado uma nova rodada, zerando todos os contadores relacionados a pontuação que foram utilizados em outros escopos;

Figura 8 - Prepara uma nova rodada, zerando o contador de leds

```
void preparaNovaRodada(){ // prepara a proxima rodada
  rodada++; // incrementa a rodada
  ledsRespondidos = 0; // zera o contador de leds respondidos
  if(rodada <= TAMANHO_SEQUENCIA){
    tocaLedsRodada();
  }
}
```

Fonte: silva (2022)

Na figura 9 existe uma função responsável por comparar a entrada do usuário com a resposta gerada pela função randômica, e imprimir esse resultado no monitor “serial” do próprio Arduino.

Figura 9 - Função para processar e verificar a resposta do usuário.

```
void processaRespostaUsuario(){ // verifica se o usuario respondeu corretamente
    int resposta = checaRespostaJogador();

    if(resposta == INDEFINIDO){
        return;
    }

    if(resposta == sequenciaLuzes[ledsRespondidos]){ // se a resposta for correta
        ledsRespondidos++;
    }
    else{
        Serial.println("Sequencia errada"); // se a resposta for errada
        rodada = TAMANHO_SEQUENCIA + 2;
    }
}
```

Fonte: silva (2022)

Na figura 10 é uma função que checa o estado do jogo, armazena qual rodada o jogador está e processa o resultado.

Figura 10 - Função para verificar o estado do jogo.

```
int estadoAtual() { // verifica o estado atual do jogo
    if(rodada <= TAMANHO_SEQUENCIA){
        if (ledsRespondidos == rodada){ // se o usuario respondeu corretamente
            return PRONTO_PARA_PROXIMA_RODADA; // pronto para a proxima rodada
        }
        else{
            return USUARIO_RESPONDENDO; // usuario esta respondendo
        }
    }
    else if(rodada == TAMANHO_SEQUENCIA + 1){ // se o usuario respondeu corretamente
        return JOGO_FINALIZADO_SUCESSO; // jogo finalizado com sucesso
    }
    else{
        return JOGO_FINALIZADO_FALHA;
    }
}
```

Fonte: silva (2022)

Na figura 11 a função manda a informação para o arduino de qual led deve ser aceso na sequência da rodada.

Figura 11 - Função para piscar os leds em sequência.

```
void tocaLedsRodada() { // toca as luzes da rodada
    for (int i = 0; i < rodada; i++){ // para cada led da rodada
        piscaLed(sequenciaLuzes[i]); // pisca o led da sequencia
    }
}
```

Fonte: silva (2022)

Na figura 12 é feita a apuração da entrada do usuário.

Figura 12 - Função para verificar se o usuário respondeu corretamente

```
int checaRespostaJogador(){ // verifica se o usuario respondeu corretamente
    if(digitalRead(BOTAO_VERDE) == LOW){ // se o usuario pressionou o botao verde
        return piscaLed(LED_VERDE); // pisca o led verde

    }
    if(digitalRead(BOTAO_AMARELO) == LOW){
        return piscaLed(LED_AMARELO);

    }
    if(digitalRead(BOTAO_VERMELHO) == LOW){
        return piscaLed(LED_VERMELHO);

    }
    if(digitalRead(BOTAO_AZUL) == LOW){
        return piscaLed(LED_AZUL);

    }

    return INDEFINIDO; // se o usuario nao pressionou nenhum botao
}
```

Fonte: silva (2022)

Na figura 13 é determinado o tempo(delay) que cada led vai ficar aceso durante as rodadas;

Figura 13 - Função para piscar o led durante a rodada.

```
int piscaLed(int portaLed){ // pisca o led da porta passada como parametro
    digitalWrite(portaLed,HIGH); // pisca o led
    delay(UM_SEGUNDO); // espera um segundo - delay do led aceso
    digitalWrite(portaLed,LOW); // desliga o led
    delay(MEIO_SEGUNDO); // espera um meio segundo - delay do led desligado

    return portaLed; // retorna a porta do led
}
```

Fonte: silva (2022)

Na figura 14 é a representação gráfica do resultado obtido pelo jogador, caso o jogador tenha ganhado, todos os leds vão piscar em sequência, do contrário todos piscarão simultaneamente.

Figura 14 - Função para finalizar o jogo e representar graficamente o resultado.

```
void jogoFinalizadoSucesso(){ //pisca todos os leds em sequencia
    piscaLed(LED_VERDE);
    piscaLed(LED_AMARELO);
    piscaLed(LED_VERMELHO);
    piscaLed(LED_AZUL);
    delay(MEIO_SEGUNDO);

}

void jogoFinalizadoFalha(){ // pisca todos os leds de uma vez
    digitalWrite(LED_VERDE,HIGH);
    digitalWrite(LED_AMARELO,HIGH);
    digitalWrite(LED_VERMELHO,HIGH);
    digitalWrite(LED_AZUL,HIGH);
    delay(UM_SEGUNDO);
    digitalWrite(LED_VERDE,LOW);
    digitalWrite(LED_AMARELO,LOW);
    digitalWrite(LED_VERMELHO,LOW);
    digitalWrite(LED_AZUL,LOW);
    delay(MEIO_SEGUNDO);

}
```

Fonte: silva (2022)

4 CONCLUSÃO

O desafio proposto foi alcançado, visto que conseguimos reduzir cerca de 20% do custo do aparelho original. Ainda vem a reforçar os conceitos da linguagem “C + +” e arduino, mediante outros conhecimentos da área de elétrica e eletrônica.

Por se tratar de um protótipo, em suas escalas e especificidades reduzidas, o projeto veio a ser um sucesso, como pode ser visto em:

<<https://www.youtube.com/watch?v=4Q9HWMtZUVI>>, e o código encontrado em:

<<https://github.com/Kiwiabacaxi/Genius>>.

5 BIBLIOGRAFIA

ARDUINO.CC. 2022. É uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++. 2022. Disponível em: <https://www.arduino.cc/en/hardware#classic-family>. Acesso em: 25 jun. 2022.

AUTODESK TINKERCAD. 2022. É um aplicativo Web gratuito para projetos 3D, eletrônica e codificação. Somos a introdução ideal à Autodesk, empresa líder global em tecnologia de projeto e criação... Disponível em : https://www.tinkercad.com/things/a9fVHelyhIo?sharecode=PX_5hVVjnOFCmPwR7_axMPqIa88Wh-ULtlBNylQ4fck> Acesso em: 25 jun. 2022

SILVA, Carlos A. S. **Repositório digital do jogo “Genius”**. Uberaba, 25 jun. 2022. Github: @Kiwiabacaxi. Disponível em: <https://github.com/Kiwiabacaxi/Genius>. Acesso em: 25 jun. 2022.