Escola de Engenharia da UFMG

Projeto de Sistemas Embutidos

Laboratório 2 - Pisca LED com Interrupção

Nome: Giovanni Martins de Sá Júnior

Matrícula: 2017001850

Semestre: 2023/2

Conteúdo

1 Descrição do GPIO

- Neste segundo laboratório, para configurar um pino de GPIO de entrada e sensível a interrupções no Arduino Uno, são feitas as seguintes configurações:

1. Conexões:

   - **Chave (Botão):** Conectou-se uma extremidade do botão a um dos pinos do botão ao terra no Arduino.

   - **Outra extremidade da chave:** Conecte a outra extremidade do botão a um resistor de pull-up interno do Arduino, que é ativado com o comando *"pinMode(chavePin, INPUT-PULLUP);"*.

   - **LED:** Conectou-se o LED de modo que o anodo fosse conectado ao *"ledPin"* no Arduino e o catodo seja conectado a um resistor e em seguida, ao terra do Arduino.

2. Configurações do Periférico de GPIO

   - Na implementação, o pino da Chave será definido pelo comando "*const int chavePin = 13;*"e o pino do LED é defino por "*const int ledPin = 2;*".

   - Além disso, o pino da chave como uma entrada com pull-up interno no setup com a linha "*pinMode(chavePin, INPUT PULL UP)*".

   - Com isso, o resistor de pull-up interno do Arduino será ativado no pino da chave, que quando o botão não estiver pressionado, o pino estará como HIGH. Quando o botão é pressionado, ele será conectado ao terra, fazendo com que o pino mude de estado para LOW.

3. Interrupções

   - Para o caso de se utilizar interrupções para detectar a pressão do botão, foi utilizado uma estratégia de debounce simples com o função "*chavePressionada()*".

2 Implementação do Código

- A seguir, é listado abaixo a implementação feita para este segundo laboratório:

```
#include <Arduino.h>

 const int chavePin = 13;  // Pino da chave no Arduino Uno
 const int ledPin = 2;   // Pino do LED no Arduino Uno

 enum Estados { ESPERA, CHAVE_PRESSIONADA, CHAVE_LIBERADA,
LED_ACESO, LED_APAGADO, LED_PISCANDO };
 Estados estado = ESPERA;

 unsigned long inicioEstado;
 const long duracaoPulso = 250; // 250 ms


 void setup() {
   Serial.begin(9600);
   pinMode(chavePin, INPUT_PULLUP);
   pinMode(ledPin, OUTPUT);
 }
 void loop() {
   switch (estado) {
     case ESPERA:
       Serial.println("Teste");
       if (chavePressionada()) {
         estado = CHAVE_PRESSIONADA;
         inicioEstado = millis();
       }
       break;

     case CHAVE_PRESSIONADA:
       Serial.println("Chave pressionada");
       if (!chavePressionada()) {
         estado = CHAVE_LIBERADA;
         inicioEstado = millis();
       }
       break;

```

```
36        case CHAVE_LIBERADA:
37          Serial.println("Chave solta");
38          if (millis() - inicioEstado >= duracaoPulso) {
39            estado = LED_ACESO;
40            digitalWrite(ledPin, HIGH);
41            inicioEstado = millis();
42          }
43          break;
44
45        case LED_ACESO:
46          Serial.println("1 Led");
47          if (millis() - inicioEstado >= 1000) {
48            estado = LED_APAGADO;
49            digitalWrite(ledPin, LOW);
50            inicioEstado = millis();
51          }
52          break;
53
54        case LED_APAGADO:
55          Serial.println("Led apagado");
56          if (millis() - inicioEstado >= 2000) {
57            estado = LED_PISCANDO;
58            inicioEstado = millis();
59          }
60          break;
61
62        case LED_PISCANDO:
63          Serial.println(millis() - inicioEstado);
64          if (millis() - inicioEstado >= 500) {
65            delay(400);
66            Serial.println("1");
67            digitalWrite(ledPin, !digitalRead(ledPin)); //
  Inverte o estado do LED
68          }
69          if (millis() - inicioEstado >= 2000) {
70            Serial.println("2");
71            estado = ESPERA;
72            digitalWrite(ledPin, LOW); // Garante que o LED est
   desligado no final
73          }
```

```
74            break;
75        }
76    }
77
78    bool chavePressionada() {
79        // Leitura da chave com debounce simples
80        static bool ultimaLeitura = HIGH; // Estado anterior da
    chave
81        bool leituraAtual = digitalRead(chavePin);
82        if (leituraAtual != ultimaLeitura) {
83            delay(5); // Debounce de 5 ms
84            leituraAtual = digitalRead(chavePin);
85            return true;
86        }
87        ultimaLeitura = leituraAtual;
88        return false; // Retorna true no ciclo de descida
89    }
90
```

Listing 1: Implementação do Código

3 Conclusão

A realização do segundo laboratório trouxe uma dificuldade maior dada a necessidade de implementação da máquina de estados finitos. Com isso, a maior parte dos testes funcionou conforme o esperado.