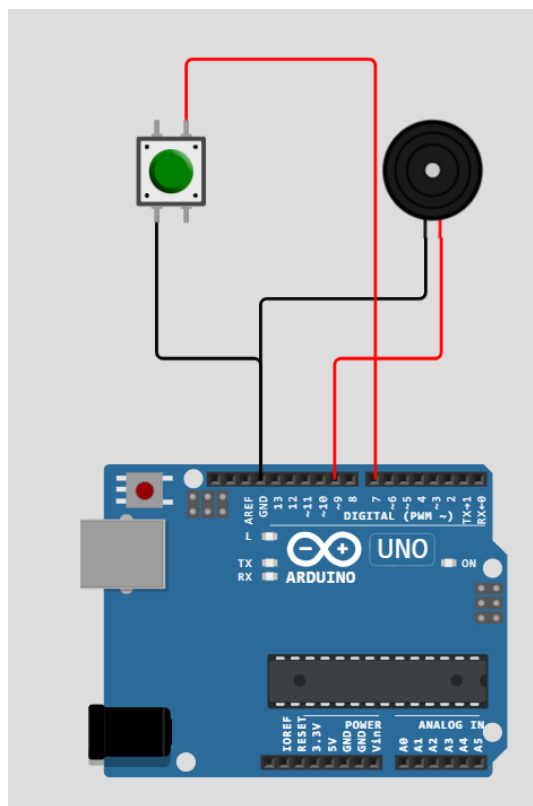


Relatório de Sistemas de Controle – Adicionando CONTROLE ao circuito.

Grupo: Ana Carolina Cardoso & Marco Bockoski & Mateus Henrique Cardoso

EXPLICAÇÃO

Em laboratório executamos um exemplo para ilustrar a adição de CONTROLE a um circuito através de um BOTÃO DE PRESSÃO. Utilizamos de exemplo um programa que utiliza um botão de pressão juntamente com um sensor buzzer. O botão e o buzzer são ligados a terminais digitais de um microcontrolador Arduino, assim como ligados ao terminal de referência nula (terra, GND) do microcontrolador.



MATERIAIS UTILIZADOS

- 4 fios condutores
- 1 cabo USB
- 1 Microcontrolador Arduino Uno
- 1 BUZZER
- 1 PUSHBUTTON
- 1 protoboard

MONTAGEM

As entradas digitais utilizadas nesse projeto são 7 e 9, juntamente do terminal de referência (GND). As entradas digitais corresponde a ligação do BUZZER e do BOTÃO com o Arduino, que são feitas por fios condutores.

CÓDIGO

```
1  int buzzer = 9;
2  int botao = 7;
3  int val = 0;
4
5  void setup() {
6      pinMode(buzzer, OUTPUT);
7      pinMode(botao, INPUT);
8  }
9
10 void loop() {
11     val = digitalRead(botao);
12     if(val == 1){
13         digitalWrite(buzzer, HIGH);
14         delayMicroseconds(10);
15         digitalWrite(buzzer, LOW);
16         delayMicroseconds(100);
17     }
18 }
19
```

O Script enviado para o Arduino é separável em 3 partes:

- Declaração de Variáveis
- Ligação com o Arduino e Definição de entrada/saída
- Acionamento do BUZZER

A primeira etapa se trata de criar variáveis correspondentes a cada terminal digital conectada ao BOTÃO e o BUZZER. Essas variáveis serão processadas no decorrer do código.

A segunda parte apresenta a função Setup, que corresponde aos processos anteriores à devida execução do código, geralmente nesse espaço declara-se especificações de

sensores, declarações de LCDs. Nesse caso, foi determinado como os terminais digitais 7 e 9 seriam utilizados, que no caso, funcionam como saída (BUZZER) e outro como entrada. Isso é definido pela função `pinMode()`.

Por fim, a função `loop()` corresponde aos comandos que serão executados por tempo indefinido no Arduino, no corpo desta função, há apenas 3 tipos de comandos:

- `DigitalWrite()`
- `delay()`
- `DigitalRead()`

Sendo a primeira função responsável por modificar o estado do BUZZER, assim, o primeiro argumento da função corresponde ao terminal em que o BUZZER está conectado e o segundo argumento da função corresponde aos estados digitais HIGH e LOW, que seriam respectivamente, ligado e desligado.

A segunda função corresponde no atraso (em milissegundos) do código. Pois o processamento do script é muito rápido, para as modificações no sistema serem perceptíveis ao olho humano, essas funções são responsáveis por permitir visualizar os comandos serem executados.

Ao acionar o botão, o buzzer é energizado por 200 microssegundos (por meio da função `DigitalRead()`), logo após isso o buzzer é desligado. Caso o botão não seja acionado o buzzer não fará som por tempo indeterminado.