

CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
SISTEMAS EMBARCADOS

Vinícius de Oliveira Perpétuo

TRABALHO 01

LAB.03 - EX.: 1, 2.

Brasília - DF

2021

Vinícius de Oliveira Perpétuo

TRABALHO 01

LAB.03 - EX.: 1, 2.

Trabalho inicial apresentado à disciplina de Sistemas Embarcados da Universidade de Brasília, como parte dos requisitos para conclusão da matéria.

Orientador: Prof. Aderbal Botelho

Brasília - DF

2021

SUMÁRIO

1 INTRODUÇÃO	4
2 OBJETIVO	4
3 DESENVOLVIMENTO	5
4 CONCLUSÃO	9

INTRODUÇÃO

Os sistemas de tempo real são processos que estão diretamente ligados a situações onde a exatidão temporal é imprescindível para realizar determinada tarefa. Através do laboratório será possível estudar determinados algoritmos para solução dos problemas relacionados à concorrência para o desenvolvimento de sistemas de tempo real.

OBJETIVO

Construir algoritmos para sincronização e comunicação.

DESENVOLVIMENTO

Exercício 01 - BLINK

O exemplo Blink consiste em fazer o LED piscar quando solicitado, utilizando um LED que será embutido na porta 13. É um exercício simples que facilita em um primeiro contato com o sistema.

Materiais:

- Placa Arduino;
- LED;
- Resistor de 220 ohms;

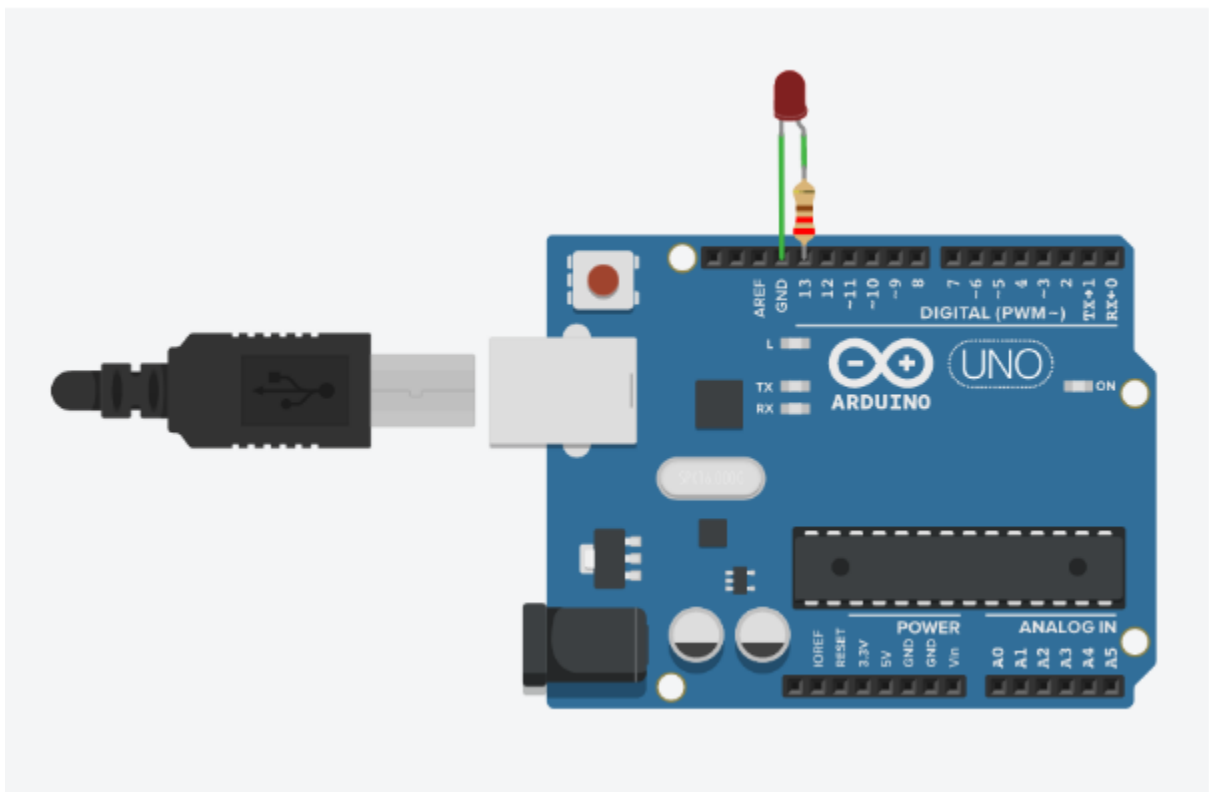
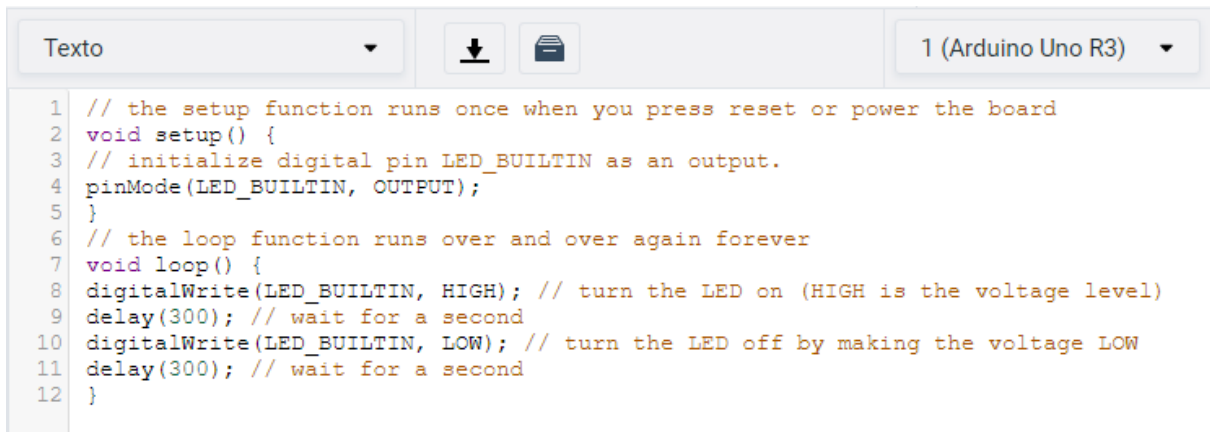


Figura 01: Esquemático do tutorial Blink.

Código utilizado para colocar o sistema em funcionamento:



```
1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6 // the loop function runs over and over again forever
7 void loop() {
8   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
9   delay(300); // wait for a second
10  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
11  delay(300); // wait for a second
12 }
```

Figura 02: Código utilizado no sistema.

Breve explicação do código:

```
void setup() {
```

```
  pinMode(LED_BUILTIN, OUTPUT);
```

```
  /*Inicia o LED*/
```

```
}
```

```
void loop() {
```

```
  digitalWrite(LED_BUILTIN, HIGH);
```

```
  /*Liga o LED*/
```

```
  delay(1000);
```

```
  /*Aguarda 1 segundo*/
```

```
  digitalWrite(LED_BUILTIN, LOW);
```

```
  delay(1000);
```

```
  /*Aguarda 1 segundo*/
```

Exercício 02 - DIGITALREADSERIAL

Este exemplo ajuda a obter conhecimentos sobre como controlar e ler as entradas digitais, através de uma comunicação serial entre a placa Arduino e o seu computador pela entrada USB.

Materiais:

- Placa Arduino;
- Um botão ou chave (switch);
- Resistor de 10k ohm;
- Jumpers;
- Protoboard.

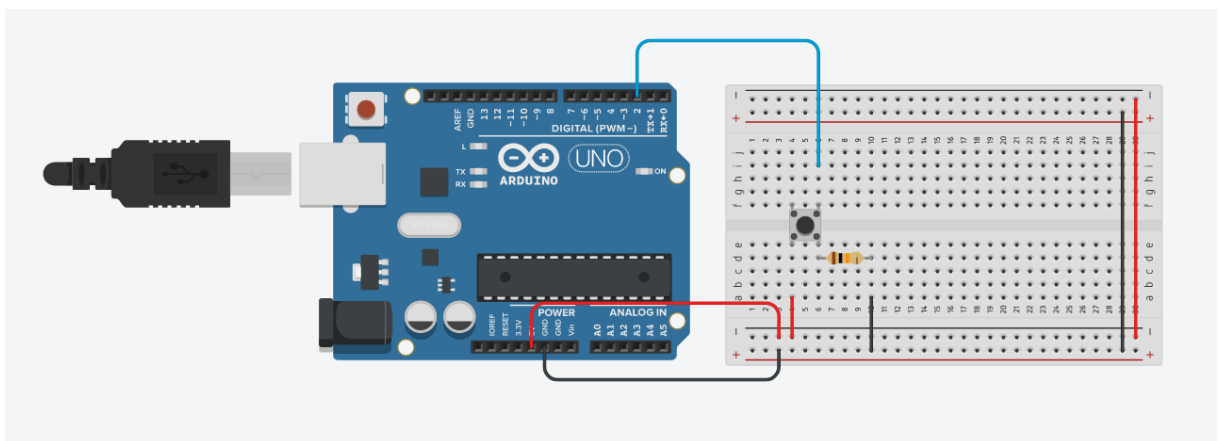


Figura 03: Esquemático para leitura do serial.

Código utilizado para colocar o sistema em funcionamento:

```

1 // digital pin 2 has a pushbutton attached to it. Give it a name:
2 int pushButton = 2;
3 // the setup routine runs once when you press reset:
4 void setup() {
5 // initialize serial communication at 9600 bits per second:
6 Serial.begin(9600);
7 // make the pushbutton's pin an input:
8 pinMode(pushButton, INPUT);
9 }
10 // the loop routine runs over and over again forever:
11 void loop() {
12 // read the input pin:
13 int buttonState = digitalRead(pushButton);
14 // print out the state of the button:
15 Serial.println(buttonState);
16 delay(1); // delay in between reads for stability
17 }

```

Figura 04: Código utilizado no sistema.

Breve explicação do código:

```

int pushButton = 2;
/*Salva o local de conexão do botão (porta número2)*/

void setup() {

Serial.begin(9600);
/*Inicia em 9600 bits*/

pinMode(pushButton, INPUT);
/*Lança INPUT ao apertar o botão*/
}

void loop() {

int buttonState = digitalRead(pushButton);
/*Identifica a situação do botão*/

Serial.println(buttonState);
/*Imprime a situação do botão*/

```



```
delay(1);  
/*Adiciona um atraso entre as leituras*/  
  
}
```

CONCLUSÃO

Tendo em vista os aspectos observados conclui-se que os RTOS (*Real Time Operating Systems*) são sistemas operacionais indispensáveis, portanto é de suma importância o desenvolvimento dos algoritmos para solução dos problemas relacionados à concorrência para o desenvolvimento de sistemas de tempo real.