

Sistemas Embarcados

Introdução à programação multitarefa

Tiago Piovesan Vendruscolo



Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito aos autores originais. [4.0 international](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Exemplos de projetos multitarefas

- Os códigos desenvolvidos até aqui são totalmente sequenciais e não existe separação entre tarefas, com isso fica muito difícil ter previsibilidade de execução quando for necessário sincronizar partes do código, ou caso queira sincronizar com algum componente externo.
- Nessa aula, será visto como dividir o código em tarefas e definir tempos de execução.

Exemplos de projetos multitarefas

- Exemplo 1: Faça dois LEDs piscar em 1Hz.

```
#define LED1 7
#define LED2 8

void setup () {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}

void loop () {
    //Liga os LEDs
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    // Aguarda 500ms
    // "Trava" a execução
    delay(500);
    //Desliga os LEDs
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    delay(500);
}
```

Exemplos de projetos multitarefas

- Exemplo 2: Faça o LED1 alterar seu nível a cada 250ms e o LED2 alterar seu nível a cada 500ms.

```
#define LED1 7
#define LED2 8

void setup () {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}

void loop () {
    //Liga os LEDs
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    delay(250);
    digitalWrite(LED1, LOW);
    delay(250);
    digitalWrite(LED2, LOW);
    digitalWrite(LED1, HIGH);
    delay(250);
    digitalWrite(LED1, LOW);
    delay(250);
}
```

- E se fosse um a cada 350ms e o outro a 800ms?

Exemplos de projetos multitarefas

- Exemplo 3: Faça um LED1 alterar seu nível a cada 350ms e o LED2 alterar seu nível a cada 800ms

```
//Exemplo para a placa Arduino UNO
#define LED1 7
#define LED2 8
//Delay em ms para iniciar cada tarefa
#define tempo_tarefa1 350
#define tempo_tarefa2 800
//Valor do millis na ultima execução
unsigned long ultima_execucao1 = 0;
unsigned long ultima_execucao2 = 0;

void setup () {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}

void tarefa1() {
    //inverte o nível do LED
    digitalWrite(LED1, !digitalRead(LED1));
}

void tarefa2() {
    digitalWrite(LED2, !digitalRead(LED2));
}
```

```
void loop() {
    //Faz a comparação de tempo com a ultima execução
    if(millis() - ultima_execucao1 >= tempo_tarefa1){
        tarefa1();
        ultima_execucao1 = millis();
    }

    if(millis() - ultima_execucao2 >= tempo_tarefa2){
        tarefa2();
        ultima_execucao2 = millis();
    }
}
```

Função millis() retorna o tempo desde que o microcontrolador foi ligado. Ocorre overflow após 49 dias.

Função micros() o overflow ocorre após 70 minutos.

Exemplos de projetos multitarefas

- Exercício 1: Faça um código com as seguintes tarefas
 - *Tarefa 1: Um LED piscando em 1Hz*
 - *Tarefa 2: Faça a aquisição da entrada analógica a cada 1 segundo*
 - *Tarefa 3: Imprima na serial o valor da entrada analógica a cada 1 segundo*

Exemplos de projetos multitarefas

■ Exercício 1:

```
#define LED1 7
#define LED2 8
#define Entrada_analogica A0
//Delay em ms para iniciar cada tarefa
#define tempo_tarefa1 500
#define tempo_tarefa2 1000
#define tempo_tarefa3 1000
//Valor do millis na ultima execução
unsigned long ultima_execucao1 = 0;
unsigned long ultima_execucao2 = 0;
unsigned long ultima_execucao3 = 0;
unsigned long valor_analogico = 0;

void setup () {
    pinMode(LED1, OUTPUT);
    Serial.begin(9600);
}

void tarefa1(){
    //inverte o nível do LED
    digitalWrite(LED1, !digitalRead(LED1));
}

void tarefa2(){
    valor_analogico = analogRead(Entrada_analogica);
}

void tarefa3(){
    Serial.println(valor_analogico);
}
```

```
void loop(){
    //Faz a comparação de tempo com a ultima execução
    if(millis() - ultima_execucao1 >= tempo_tarefa1){
        tarefa1();
        ultima_execucao1 = millis();
    }

    if(millis() - ultima_execucao2 >= tempo_tarefa2){
        tarefa2();
        ultima_execucao2 = millis();
    }

    if(millis() - ultima_execucao3 >= tempo_tarefa3){
        tarefa3();
        ultima_execucao3 = millis();
    }
}
```

Monitor serial com
timestamp

```
15:10:56.863 -> 503
15:10:57.849 -> 506
15:10:58.841 -> 503
15:10:59.860 -> 507
15:11:00.870 -> 506
15:11:01.858 -> 503
15:11:02.842 -> 504
15:11:03.859 -> 507
```

Exemplos de projetos multitarefas

- Fonte: <https://github.com/ivanseidel/ArduinoThread>
- Baixe a pasta do projeto do moodle e descompacte com o nome “ArduinoThread” na pasta Library do software Arduino.
- Essa biblioteca ajuda a desenvolver projetos multitarefas simples, fazendo o gerenciamento das tarefas e mantendo o código mais limpo.

Exemplos de projetos multitarefas

- Exemplo 4: Faça o LED1 alterar seu nível a cada 500ms e o LED2 alterar seu nível a cada 100.

```
#include <Thread.h> //biblioteca para as tarefas

#define LED1 7
#define LED2 8

void piscaLed1() {
    digitalWrite(LED1, !digitalRead(LED1));
}

void piscaLed2() {
    digitalWrite(LED2, !digitalRead(LED2));
}

//Nomeia as threads
Thread threadLed1;
Thread threadLed2;
```

```
void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    //Seta o tempo para iniciar cada thread
    threadLed1.setInterval(500);
    //função chamada pela thread
    threadLed1.onRun(piscaLed1);
    threadLed2.setInterval(100);
    threadLed2.onRun(piscaLed2);
}

void loop() {
    //checa se esta na hora de rodar a thread
    //(de acordo com o tempo estipulado)
    //faz o teste com o millis
    if (threadLed1.shouldRun()) {
        threadLed1.run();
    }

    if (threadLed2.shouldRun()) {
        threadLed2.run();
    }
}
```

Exemplos de projetos multitarefas

- É possível tornar o código mais eficiente adicionando o controlador para as tarefas, ou seja, ele fará a “fila” das tarefas que serão executadas e também o controle de tempo.

ThreadController

Exemplos de projetos multitarefas

- Exemplo 5: Faça o LED1 alterar seu nível a cada 500ms e o LED2 alterar seu nível a cada 100.

```
#include <Thread.h>
#include <ThreadController.h>

#define LED1 7
#define LED2 8

//Controla todas as threads
ThreadController Controle_das_threads;
//Nomeia as threads
Thread threadLed1;
Thread threadLed2;

void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

    //Seta o tempo para iniciar cada thread
    threadLed1.setInterval(500);

    //função chamada pela thread
    threadLed1.onRun(piscaLed1);
    threadLed2.setInterval(100);
    threadLed2.onRun(piscaLed2);

    //Adiciona as threads criadas para o controlador
    Controle_das_threads.add(&threadLed1);
    Controle_das_threads.add(&threadLed2);
}
```

```
void piscaLed1() {
    digitalWrite(LED1, !digitalRead(LED1));
}

void piscaLed2() {
    digitalWrite(LED2, !digitalRead(LED2));
}

void loop() {
    //Executa o controlador de thread, com as thread carregadas
    //faz a checagem (shouldrun)
    Controle_das_threads.run();
}
```

Exemplos de projetos multitarefas

- Exercício 2: Faça um código com as seguintes tarefas
 - *Tarefa 1: Um LED piscando em 1Hz*
 - *Tarefa 2: Um LED piscando com frequência variável*
 - *Tarefa 3: Faça a aquisição da entrada analógica a cada 1 segundo, use o valor lido na entrada como intervalo da tarefa 2*
 - *Tarefa 4: Imprima na serial o valor da entrada analógica a cada 1 segundo*

Exemplos de projetos multitarefas

```
#include <Thread.h>
#include <ThreadController.h>

#define LED1 7
#define LED2 8
#define Entrada_analogica A0

int valor_analogico = 100;

//Controla todas as threads
ThreadController Gerenciador_das_threads;
//Nomeia as threads
Thread threadLed1;
Thread threadLed2;
Thread threadAD;
Thread threadSerial;

void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    Serial.begin(9600);

    //Seta o tempo para iniciar cada thread
    threadLed1.setInterval(500);
    //função chamada pela thread
    threadLed1.onRun(piscaLed1);
    threadLed2.setInterval(100);
    threadLed2.onRun(piscaLed2);
    threadAD.setInterval(1000);
    threadAD.onRun(leitura_AD);
    threadSerial.setInterval(1000);
    threadSerial.onRun(imprime_serial);
```

```
Gerenciador_das_threads.add(&threadLed1);
Gerenciador_das_threads.add(&threadLed2);
Gerenciador_das_threads.add(&threadAD);
Gerenciador_das_threads.add(&threadSerial);
}

void piscaLed1() {
    digitalWrite(LED1, !digitalRead(LED1));
}

void piscaLed2() {
    digitalWrite(LED2, !digitalRead(LED2));
}

void leitura_AD() {
    valor_analogico = analogRead(Entrada_analogica);
    threadLed2.setInterval(valor_analogico);
}

void imprime_serial() {
    Serial.println(valor_analogico);
}

void loop() {
    //Executa o controlador de thread, com as thread carregadas
    //faz a checagem (shouldrun)
    Gerenciador_das_threads.run();
}
```

Monitor serial com
timestamp

16:23:26.744	->	393
16:23:27.728	->	510
16:23:28.758	->	643
16:23:29.741	->	746
16:23:30.723	->	586

Próxima aula

- Desenvolvimento de códigos multitarefas de tempo real - FREERTOS - Tarefas