

Bibliotecas

Prof. Dr. Roberto Kenji Hiramatsu
Prof. Dr. João Henrique Correia Pimentel

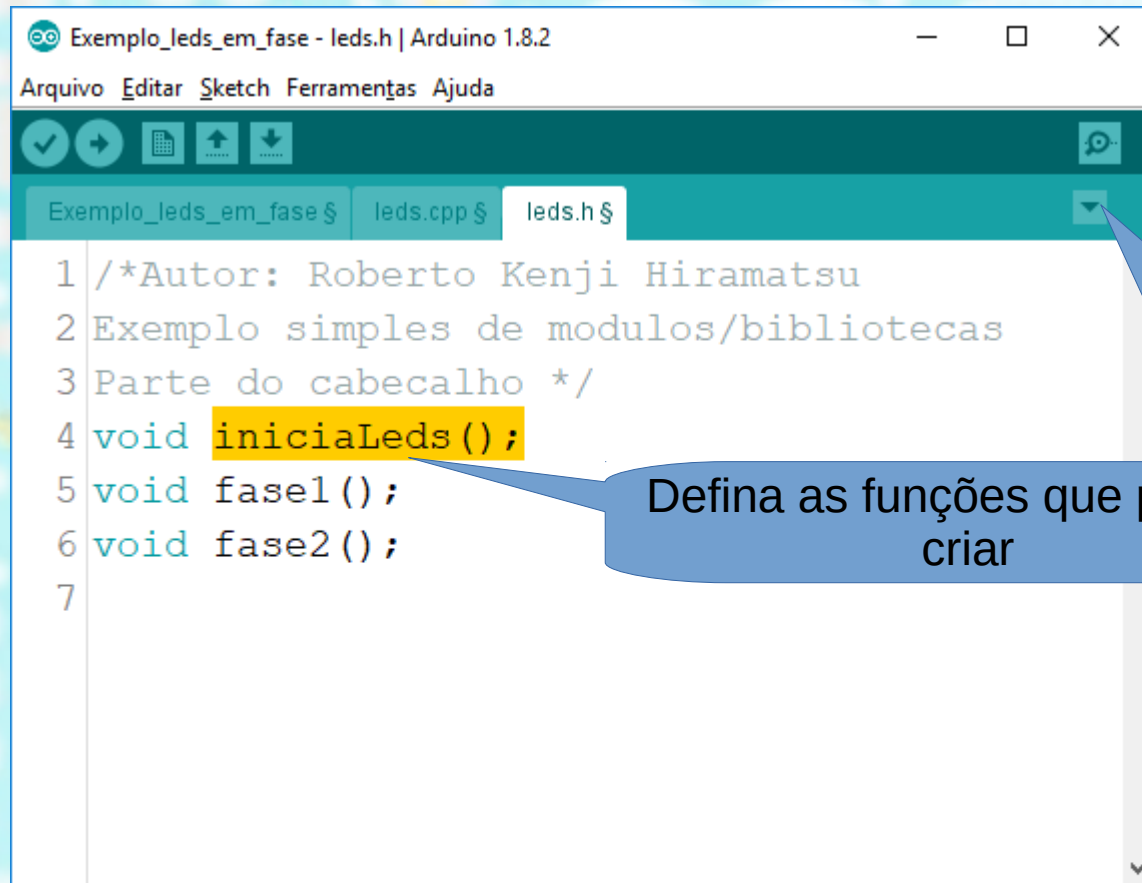
- 
- Bibliotecas
 - Crie o cabeçalho da biblioteca
 - Implemente as funções vazias com implementações mínimas
 - Usando a biblioteca leds no programa
 - Completando a implementação
 - Exercício separando as fase do semáforo
 - AVR libc no Arduino UNO R3 Biblioteca na plataforma
 - Um exemplo com operações matemáticas
 - Incluindo bibliotecas
 -
 - Instalando biblioteca de arquivo zip
 - Análise os exemplos da biblioteca instalada
 - Onde fica a instalação da biblioteca
 - O exemplo de blink da biblioteca Timer
 - Abstraindo conceito de classe/objeto

Bibliotecas

- São coleções de definições de tipos, constantes, macros, funções e classes definidas
- Existem as bibliotecas definidas na linguagem e que são incluídas junto a plataforma do microcontrolador.
- Bibliotecas fornecidas como contribuições da comunidade
- Uso de diretivas `#include` para incluir no programa/firmware.

–

Crie o cabeçalho da biblioteca



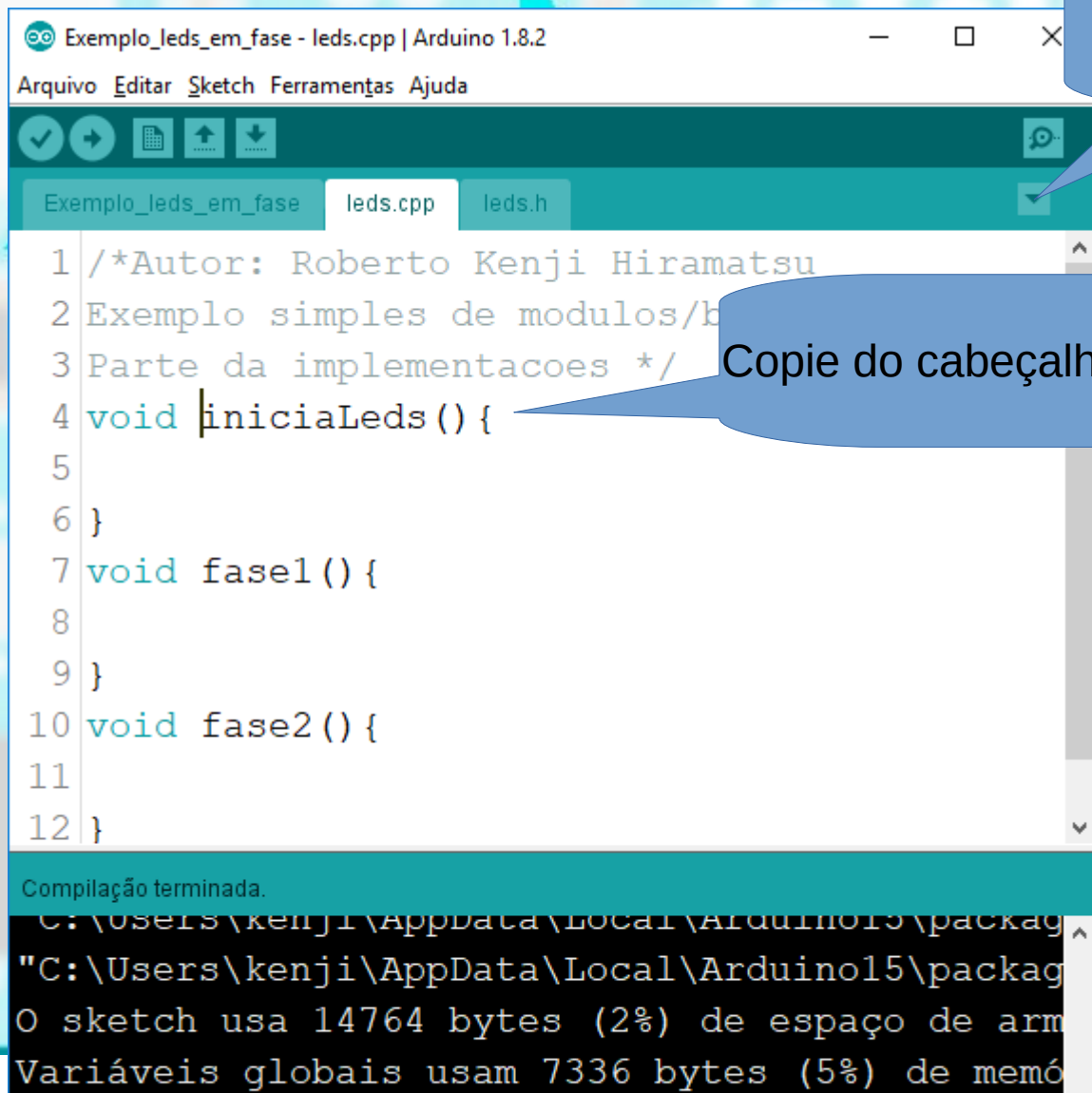
The screenshot shows the Arduino IDE interface with the title bar 'Exemplo_leds_em_fase - leds.h | Arduino 1.8.2'. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Ferramentas', and 'Ajuda'. The toolbar contains icons for opening, saving, and running. The tab bar shows three tabs: 'Exemplo_leds_em_fase \$', 'leds.cpp \$', and 'leds.h \$'. The 'leds.h' tab is active, displaying the following code:

```
1 /*Autor: Roberto Kenji Hiramatsu
2 Exemplo simples de modulos/bibliotecas
3 Parte do cabecalho */
4 void iniciaLeds();
5 void fase1();
6 void fase2();
7
```

Clique aqui para
criar o arquivo
leds.h

Defina as funções que planeja
criar

Implemente as funções vazias com implementações mínimas



```
Exemplo_leds_em_fase - leds.cpp | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda

Exemplo_leds_em_fase leds.cpp leds.h

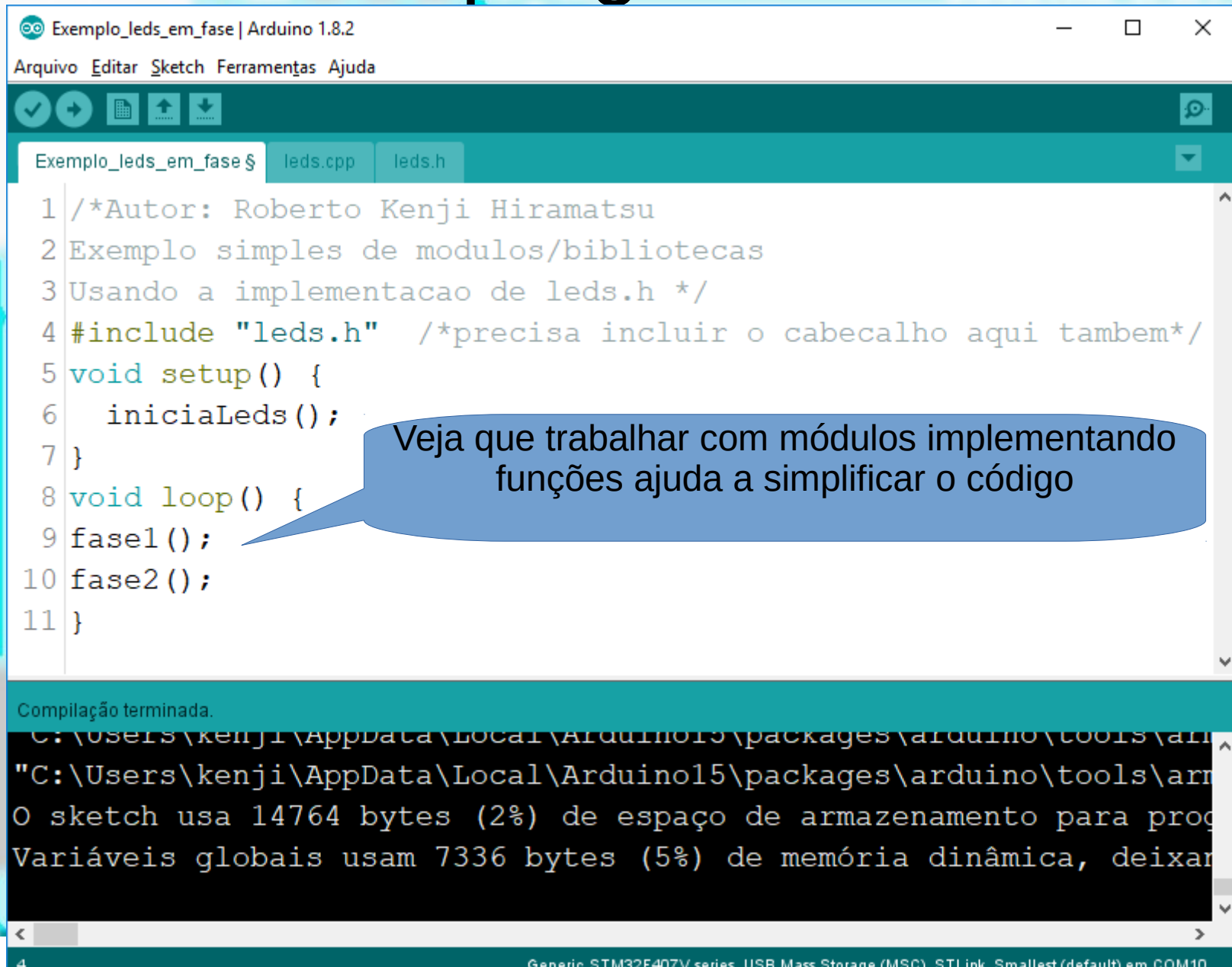
1 /*Autor: Roberto Kenji Hiramatsu
2 Exemplo simples de modulos/b
3 Parte da implementacoes */
4 void iniciaLeds() {
5
6 }
7 void fase1() {
8
9 }
10 void fase2() {
11
12 }

Compilação terminada.
C:\Users\kenji\AppData\Local\Arduino15\packag
"C:\Users\kenji\AppData\Local\Arduino15\packag
O sketch usa 14764 bytes (2%) de espaço de arm
Variáveis globais usam 7336 bytes (5%) de memó
```

Crie leds.cpp

Copie do cabeçalho e crie as funções mínimas

Usando a biblioteca leds no programa



```
Exemplo_leds_em_fase | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda

Exemplo_leds_em_fase$ leds.cpp leds.h

1 /*Autor: Roberto Kenji Hiramatsu
2 Exemplo simples de modulos/bibliotecas
3 Usando a implementacao de leds.h */
4 #include "leds.h" /*precisa incluir o cabecalho aqui tambem*/
5 void setup() {
6   iniciaLeds();
7 }
8 void loop() {
9   fase1();
10  fase2();
11 }

Compilação terminada.
C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\ar
"C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\ar
O sketch usa 14764 bytes (2%) de espaço de armazenamento para prog
Variáveis globais usam 7336 bytes (5%) de memória dinâmica, deixar
```

Completando a implementação

```
Exemplo_leds_em_fase - leds.cpp | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda

Exemplo_leds_em_fase leds.cpp § leds.h
4 #include <arduino.h>
5 int ledUm=2;
6 int ledDois=3;
7 void iniciaLeds() {
8     pinMode(ledUm, OUTPUT);
9     pinMode(ledDois, OUTPUT);
10 }
11 void fase1() {
12     digitalWrite(ledUm, HIGH);
13     digitalWrite(ledDois, LOW);
14     delay(1000);
15 }
16 void fase2() {
17     digitalWrite(ledUm, LOW);
18     digitalWrite(ledDois, HIGH);
19     delay(1000);
20 }
```

A biblioteca `arduino.h` é sempre necessária quando vai trabalhar com tipos e funções definidas na plataforma

O que acontece quando não incluimos?

Exercício separando as fase do semáforo

- Usando o exemplo do semáforo para pedestre tente subdividir os eventos de acionamento das luzes em funções específicas:
 - Inicial: luz verde na avenida + luz vermelha no pedestre
 - Fase 1: luz amarela na avenida + luz vermelha no pedestre
 - Fase 2: luz vermelha na avenida + luz verde no usuário
 - Fase 3: luz vermelha na avenida + luz vermelha piscante no pedestre
- Considere as fases 1, 2 e 3 podem ser configuradas com tempo

AVR libc no Arduino UNO R3

Biblioteca na plataforma

← → ↻ Seguro | <https://www.nongnu.org/avr-libc/user-manual/modules.html>

<alloca.h>: Allocate space in the stack

<assert.h>: Diagnostics

<ctype.h>: Character Operations

<errno.h>: System Errors

<inttypes.h>: Integer Type conversions

<math.h>: Mathematics

<setjmp.h>: Non-local goto

<stdint.h>: Standard Integer Types

<stdio.h>: Standard IO facilities

<stdlib.h>: General utilities

<string.h>: Strings

<time.h>: Time

<avr/boot.h>: Bootloader Support Utilities

<avr/cpufunc.h>: Special AVR CPU functions

<avr/eeprom.h>: EEPROM handling

<avr/fuse.h>: Fuse Support

<avr/interrupt.h>: Interrupts

<avr/io.h>: AVR device-specific IO definitions

<avr/lock.h>: Lockbit Support

<avr/pgmspace.h>: Program Space Utilities

<avr/power.h>: Power Reduction Management

▼ <avr/sfr_defs.h>: Special function registers

Additional notes from <avr/sfr_defs.h>

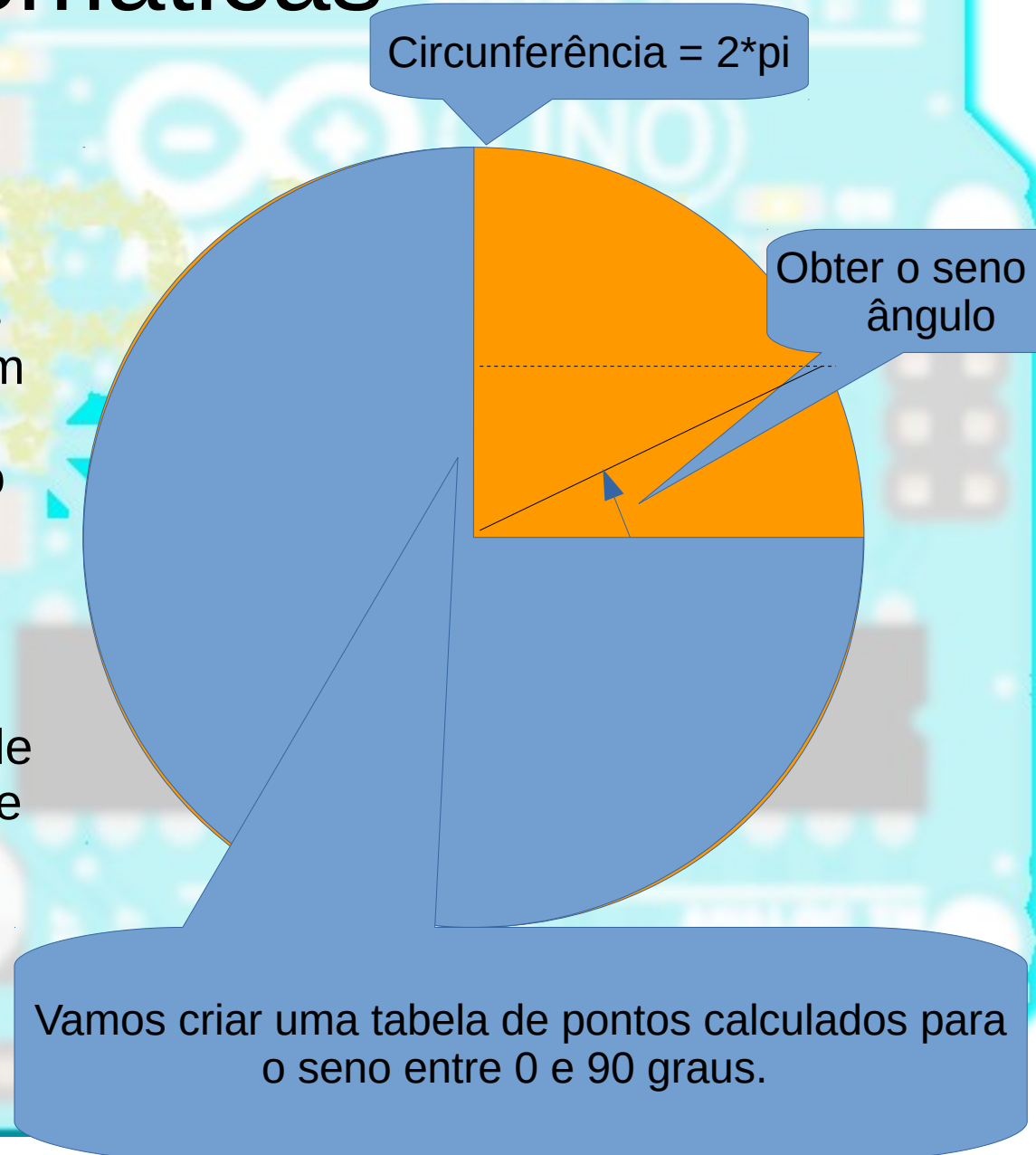
<avr/signature.h>: Signature Support

<avr/sleep.h>: Power Management and Sleep Modes

<avr/version.h>: avr-libc version macros

Um exemplo com operações matemáticas

- Operações matemáticas podem ser encontradas em `math.h`
 - As operações trigonométricas usam radianos então precisam tratamento em função da constante `pi` definida na como `M_PI` em `math.h`
 - No Uno R3 de 8bits
 - `float = double`
 - Se precisar precisar rapidez de processamento use tabelas de busca



Exemplo para operações matemáticas

Clique aqui para criar o arquivo matematico.h

Usamos o esquema `#ifndef #define` para evitar múltiplas inserções do cabeçalho

Na linguagem C se precisamos acessar uma variável em vários módulos, declaramos o modificador *extern*

Exemplo_sin - matematico.h | Arduino 1.8.2

Arquivo Editar Sketch Ferramentas Ajuda



Exemplo_sin

matematico.cpp

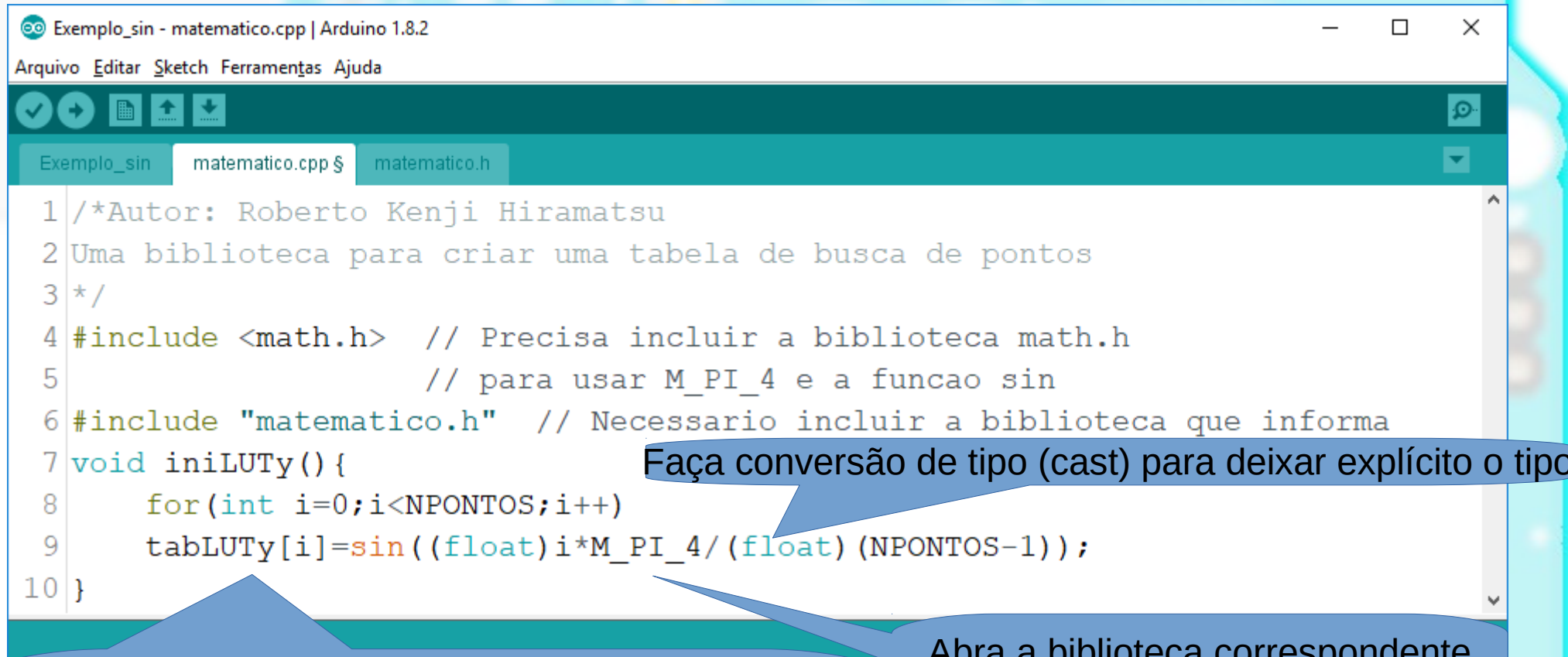
matematico.h

```
1 /*Autor: Roberto Kenji Hiram  
2 Uma biblioteca para criar uma tabela de busca de pontos  
3 */  
4 #ifndef __MATEMATICO__  
5 #define __MATEMATICO__  
6 #define NPONTOS 7 // Numero de pontos a preencher entre 0 90 graus  
7 extern float tabLUTy[NPONTOS]; // Para acesso da variavel do modulo da  
8 // tabela de busca - Look Up Table externo  
9 void iniLUTy(); // Funcao para iniciar a tabela  
10 #endif
```

Compilação terminada.

```
C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc  
"C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc  
O sketch usa 24208 bytes (4%) de espaço de armazenamento para programas. O máxi  
Variáveis globais usam 7368 bytes (5%) de memória dinâmica, deixando 123704 byt
```

Implementa inicialização

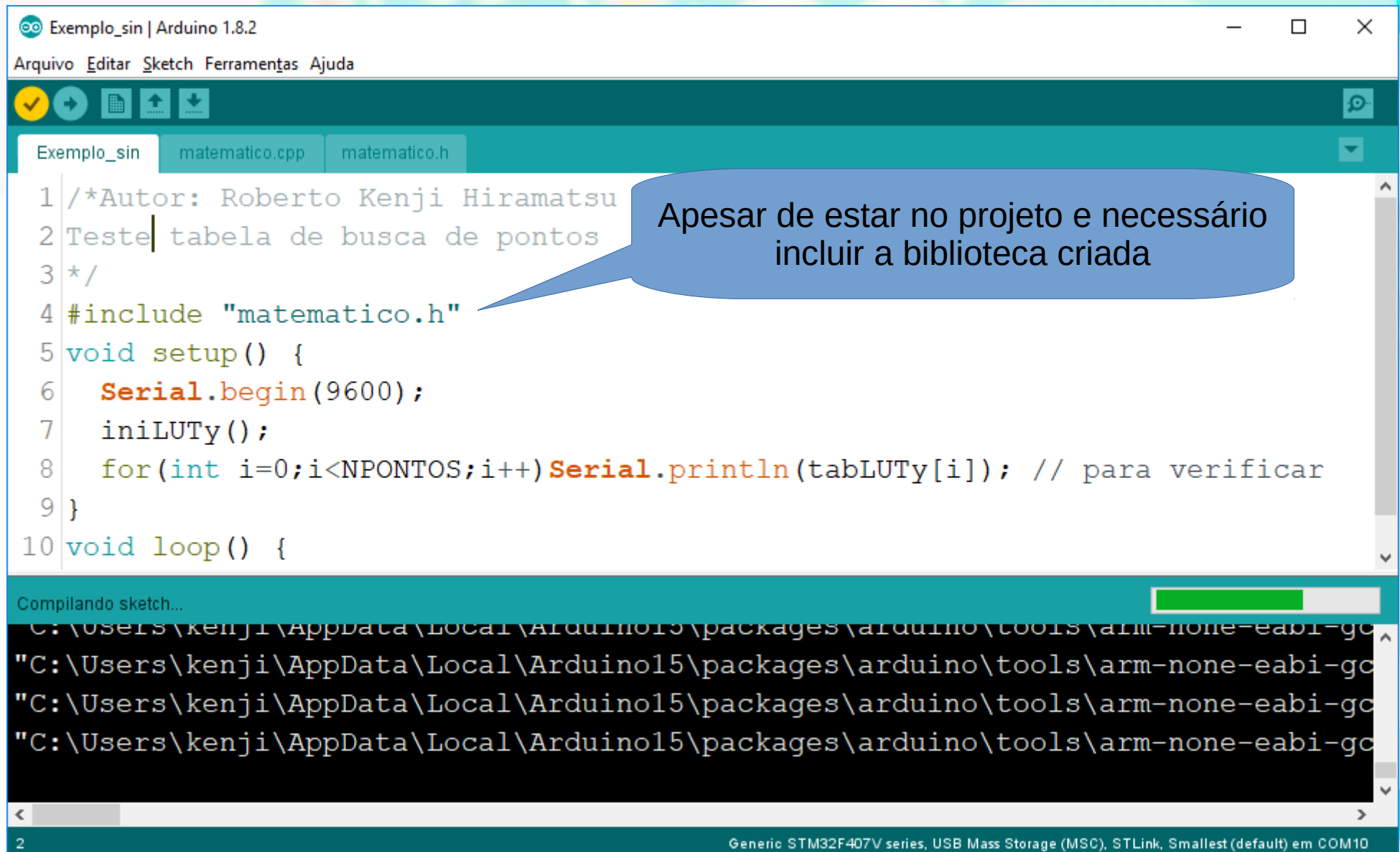


```
1 /*Autor: Roberto Kenji Hiramatsu
2 Uma biblioteca para criar uma tabela de busca de pontos
3 */
4 #include <math.h> // Precisa incluir a biblioteca math.h
5 // para usar M_PI_4 e a funcao sin
6 #include "matematico.h" // Necessario incluir a biblioteca que informa
7 void iniLUTy() {
8     for(int i=0;i<NPONTOS;i++)
9         tabLUTy[i]=sin((float)i*M_PI_4/(float)(NPONTOS-1));
10 }
```

A criação de tabela de buscas permite uma recuperação rápida da informação, pois processamento de ponto flutuante é “custoso” para uC. Criar tabelas de busca são interessantes para realizar conversões entre processamentos de entrada no uC e uma saída como o caso de controle de velocidade de motor em relação a carga do motor

Abra a biblioteca correspondente para ver as constantes e funções

Testando a biblioteca de usuário criada



The screenshot shows the Arduino IDE interface with a sketch named 'Exemplo_sin'. The sketch includes a custom library 'matematico.h' and tests its functionality by printing values from a table. A blue callout bubble highlights the `#include "matematico.h"` line with the text: 'Apesar de estar no projeto e necessário incluir a biblioteca criada'.

```
1 /*Autor: Roberto Kenji Hiramatsu
2 Teste| tabela de busca de pontos
3 */
4 #include "matematico.h"
5 void setup() {
6     Serial.begin(9600);
7     iniLUTy();
8     for(int i=0;i<NPONTOS;i++) Serial.println(tabLUTy[i]); // para verificar
9 }
10 void loop() {
```

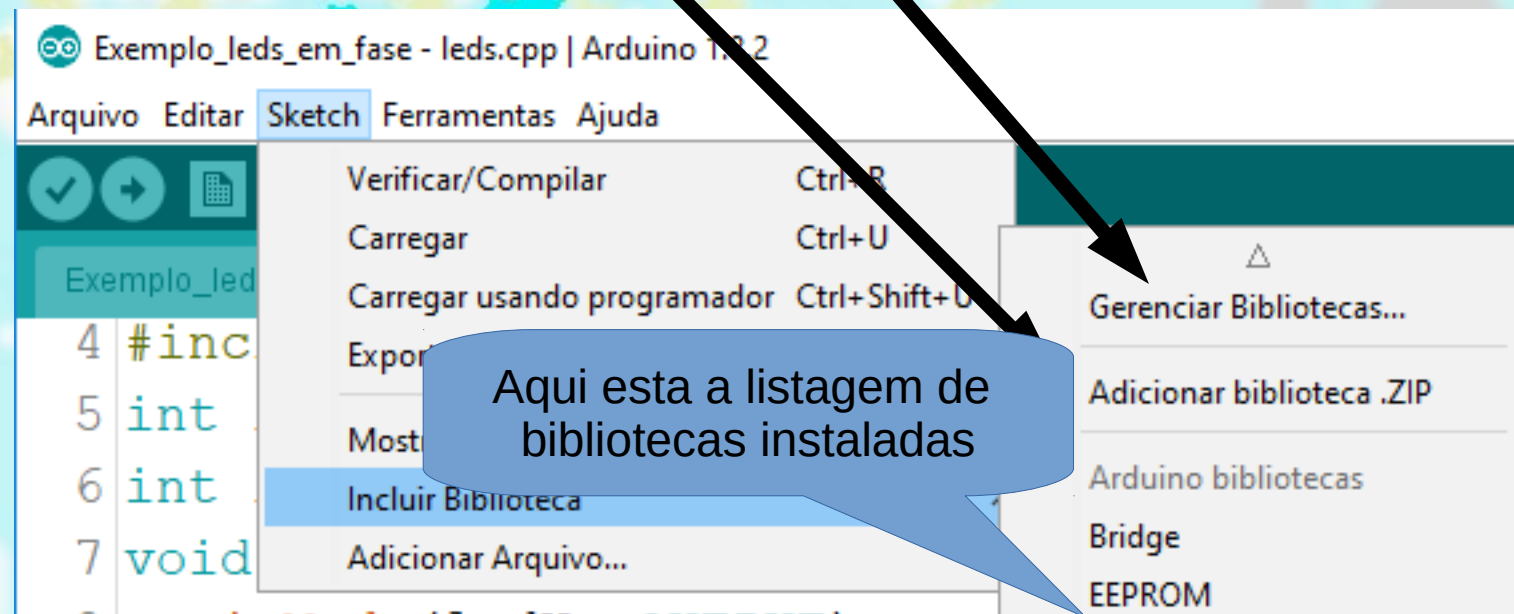
Compilando sketch...

C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc
"C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc
"C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc
"C:\Users\kenji\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc

2 Generic STM32F407V series, USB Mass Storage (MSC), STLink, Smallest (default) em COM10

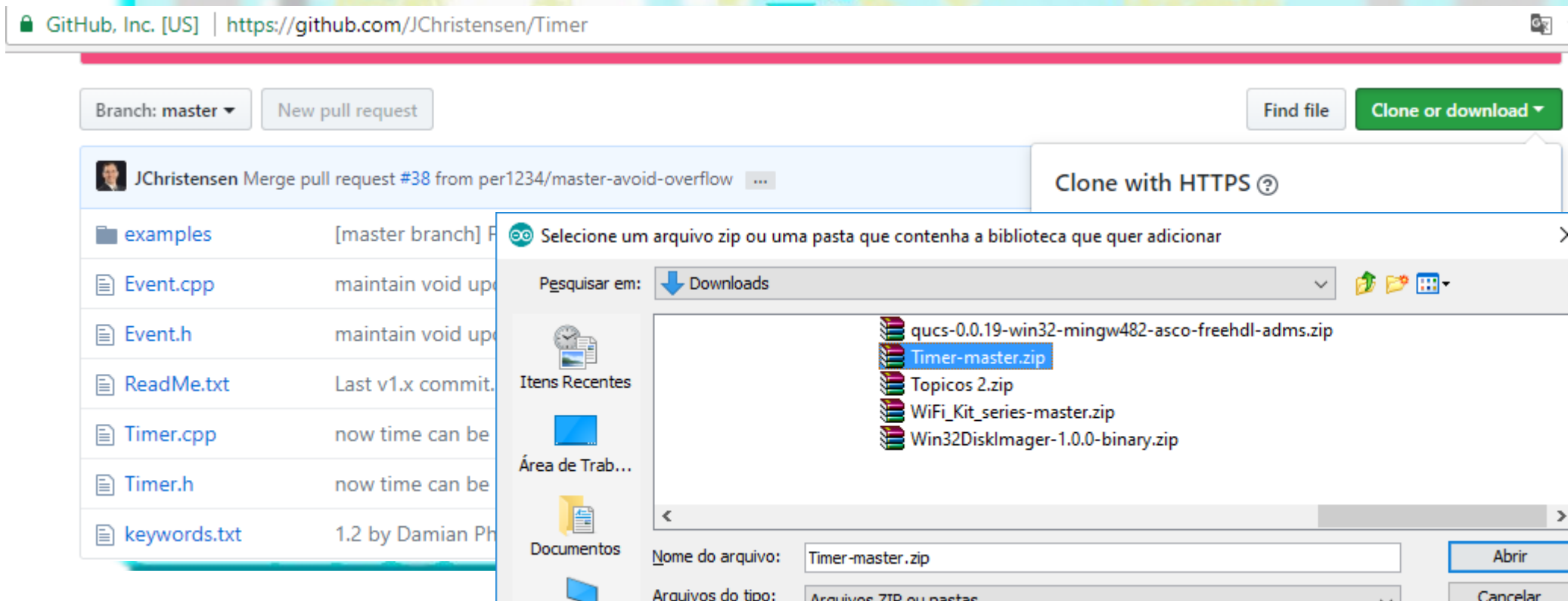
Incluindo bibliotecas

- Duas formas:
 - Disponíveis no gerenciador de bibliotecas
 - Inserido via arquivo zipado

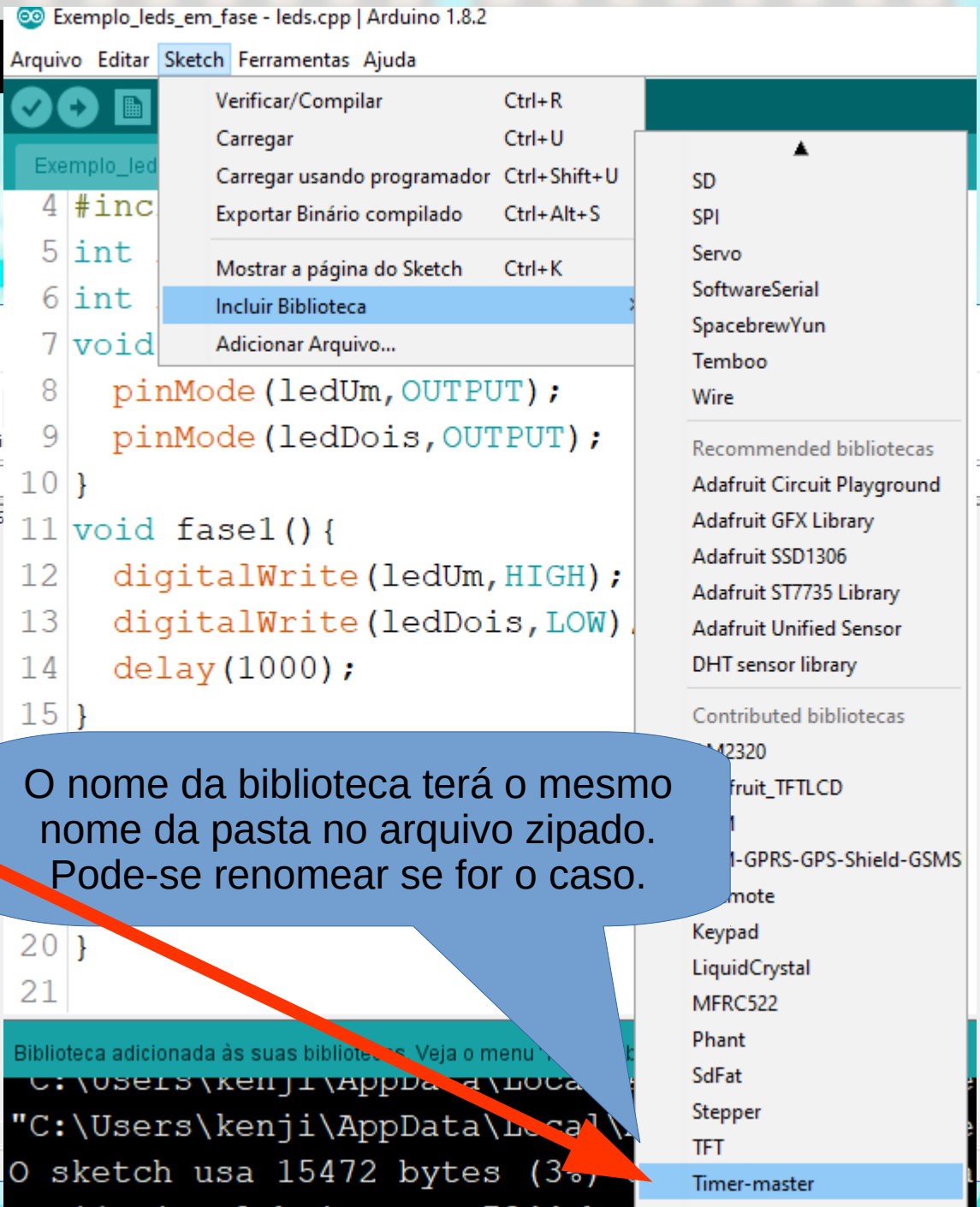
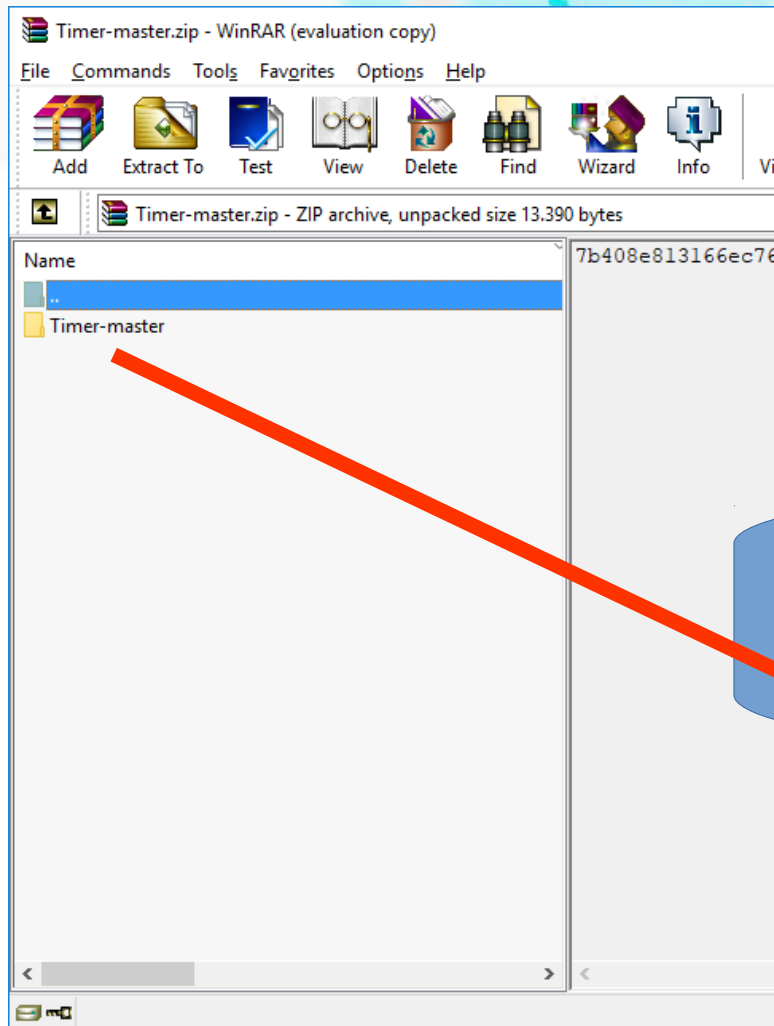


Incluindo um exemplo de biblioteca

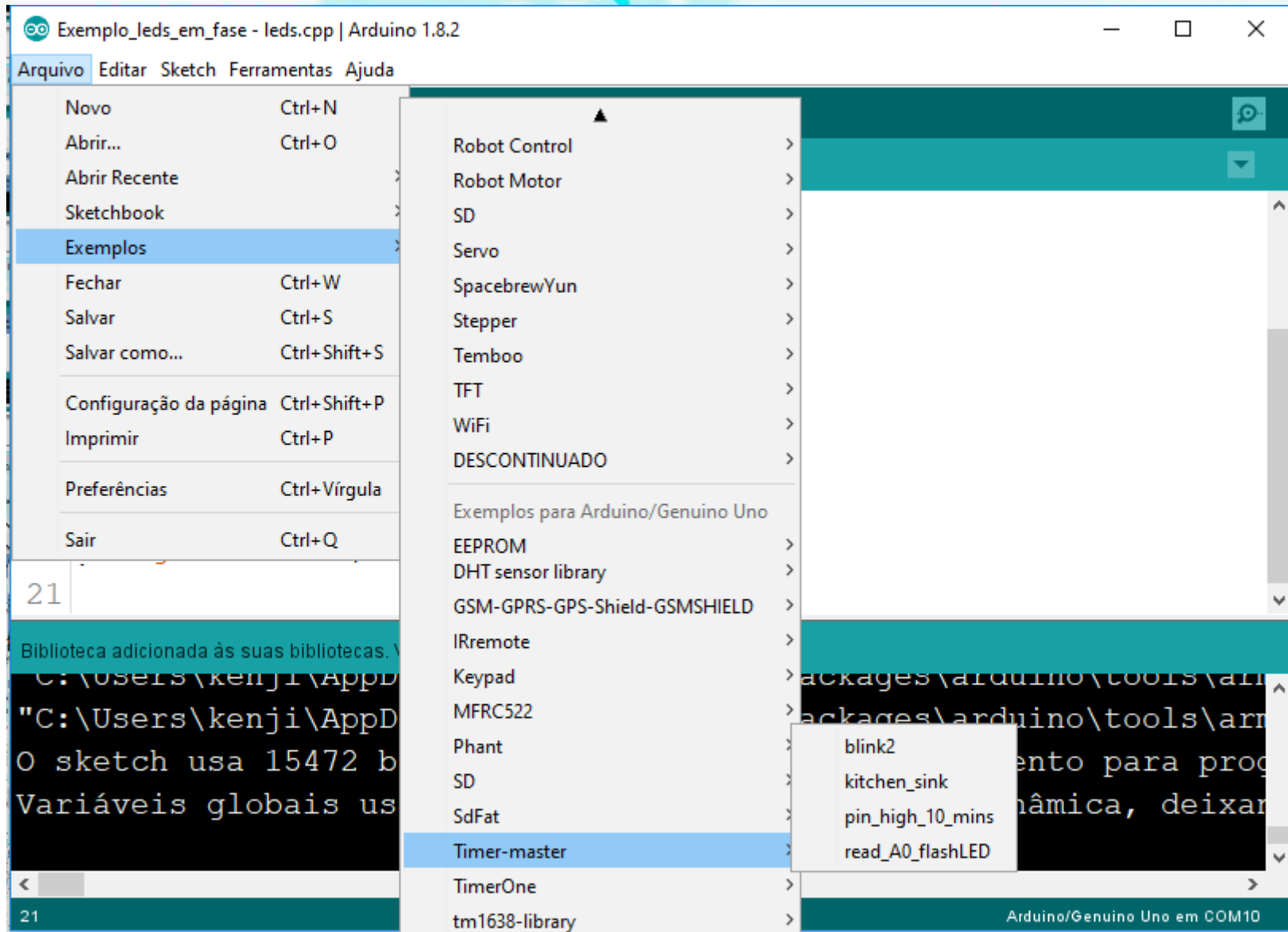
- Carregando biblioteca de timer
 - Baixe do site:
<https://github.com/JChristensen/Timer>
 - Adicione a biblioteca por meio de arquivo zip.



Instalando bibli



Análise os exemplos da biblioteca instalada



Onde fica a instalação da biblioteca

- Se necessário, inspecione a biblioteca.

Este Computador > Documentos > Arduino > libraries > Timer-master >

jEdit - Timer.h

File Edit Search Markers Folding View Utilities Macros Plugins Help

Timer.h (%USERPROFILE%\Documents\Arduino\libraries\Timer-master\)

```
20
21 #define TIMER_NOT_AN_EVENT (-2)
22 #define NO_TIMER_AVAILABLE (-1)
23
24 class Timer
25 {
26
27 public:
28     Timer(void);
29
30     int8_t every(unsigned long period, void (*callback)(void));
31     int8_t every(unsigned long period, void (*callback)(void), int repeatCount);
32     int8_t after(unsigned long duration, void (*callback)(void));
33     int8_t oscillate(uint8_t pin, unsigned long period, int8_t startingValue);
34     int8_t oscillate(uint8_t pin, unsigned long period, int8_t startingValue, int repeatCount);
35
36 /**
37  * This method will generate a pulse of
38  * call of this method and lasting for
39  */
40     int8_t pulse(uint8_t pin, unsigned long period, uint8_t startingValue);
41
42 }
```

Nome da classe – pode ser visto como definição de tipo de variável

Métodos na classe – visto como funções de uma classe

O parâmetro callback é o nome de uma função do tipo void <nome_função>() que é implementada pelo programador

O exemplo de blink da biblioteca Timer

Arduino IDE interface showing the blink2 sketch using the Timer library. The code is as follows:

```
8
9 #include "Timer.h"                                     //http://github.com/JChristensen/Timer
10 const int LED1 = 8;                                    //connect one LED to this pin (with appropriate current-limiting resist
11 const int LED2 = 9;                                    //connect another LED to this pin (don't forget the resistor)
12 const unsigned long PERIOD1 = 1000;                    //one second
13 const unsigned long PERIOD2 = 10000;                   //ten seconds
14 Timer t;
15 void setup(void){
16     pinMode(LED1, OUTPUT);
17     pinMode(LED2, OUTPUT);
18     t.oscillate(LED1, PERIOD1, HIGH);
19     t.oscillate(LED2, PERIOD2, HIGH);
20 }
21 void loop(void){
22     t.update();
23 }
24
```

Annotations explaining the code:

- Objeto instanciado de Timer. Normalmente para dispositivos**
Somente instanciamos uma vez
- Simplifica um pouco tratamento de acesso de variáveis em módulos**
- O método de um objeto é acessado usando . (ponto) entre o nome do objeto e seu método**
- Permite opera de forma independente vários leds com período definido e diferente.**

Abstraindo conceito de classe/objeto

Classe → Ser humano

Objeto → Pedro, Maria,
João, José, Ana, Julio
Paula



Dados → nome, cor, sexo, altura, idade
Variáveis na classe

Pedro.nome
Maria.altura

Ações → Saltar, correr, carregar
Métodos na classe



José salta
José corre
José carrega

Jose.salta()
Jose.corre()
Jose.carrega()



roberto.hiramatsu@ufrpe.br

Analizando o objeto do sensor DHT11

```
• #include "DHT.h"
• #define DHTPIN 2 // what digital pin we're connected to
• // Uncomment whatever type you're using!
• // #define DHTTYPE DHT11
• #define DHTTYPE DHT22
• // #define DHTTYPE DHT21
```

DHT dht(DHTPIN, DHTTYPE);

```
• DHT dht(DHTPIN, DHTTYPE);
• void setup() {
•   Serial.begin(9600);
•   Serial.println("DHTxx test!");
•   dht.begin();
• }
• void loop() {
•   // Wait a few seconds between measurements.
•   Delay(2000);
•   // Reading temperature and humidity
•   // Sensor readings may not be exact (to try again).
```

float h = dht.readHumidity();

```
float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
• // Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);
// Check if any reads failed and exit early (to try again).
```

```
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("Heat index: ");
Serial.print(hic);
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");
}
```