

Sistemas Embarcados e plataforma Arduino

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

O que é um sistema embarcado?

- É um sistema computacional de propósito específico

Ar-condicionado digital

Controle de de
injeção eletrônica

ABS





- O que é um microcontrolador?

- É um circuito integrado que contém as partes típicas de um computador e opcionalmente periféricos adicionais.

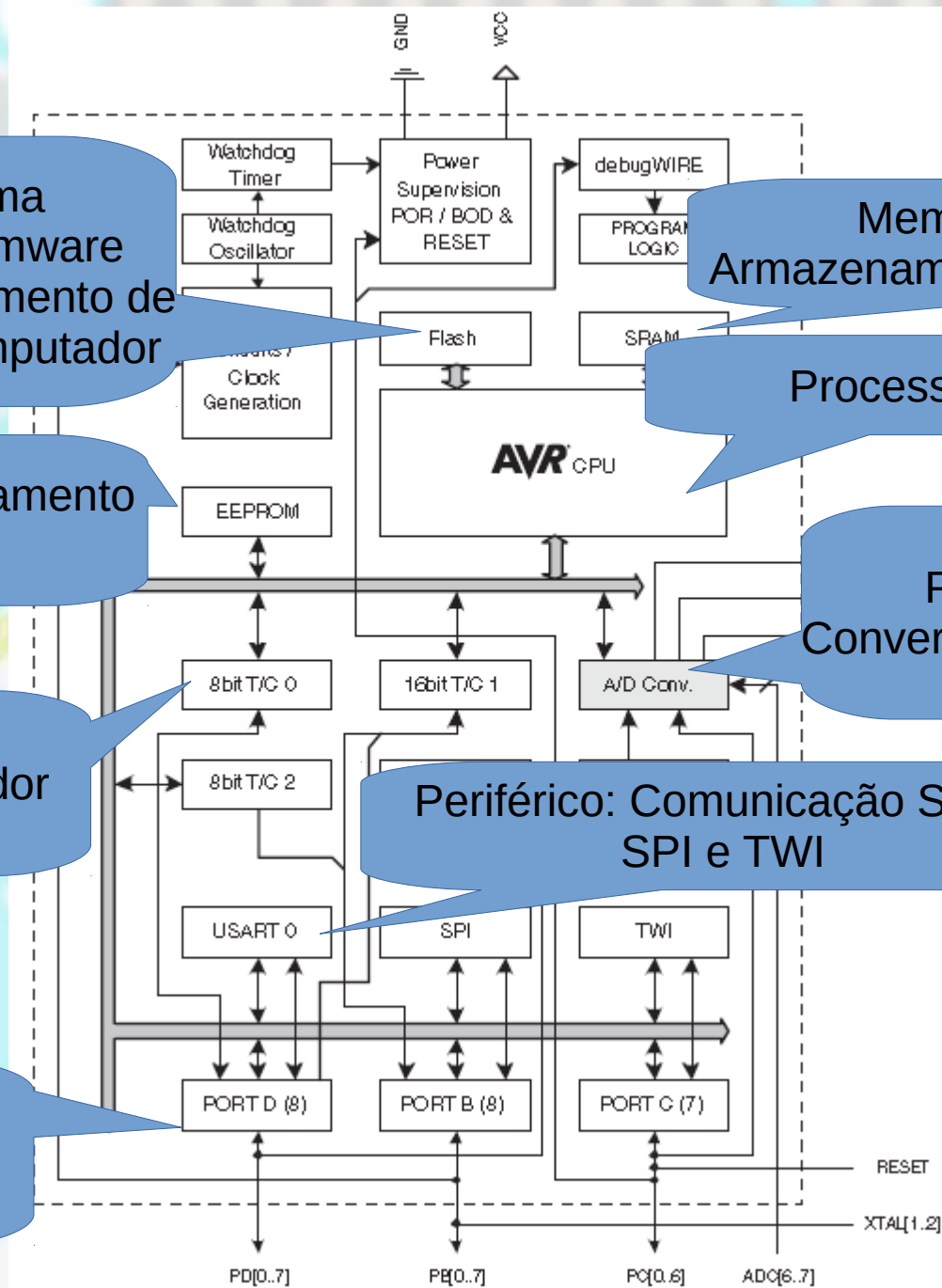
-

Memória de Programa
Armazenamento de Firmware
Equivalente ao armazenamento de
programas no HD do computador

EEPROM – Armazenamento
de usuário

Periférico: Temporizador

Entrada/Saída
Digital/Analógico
Multipropósito



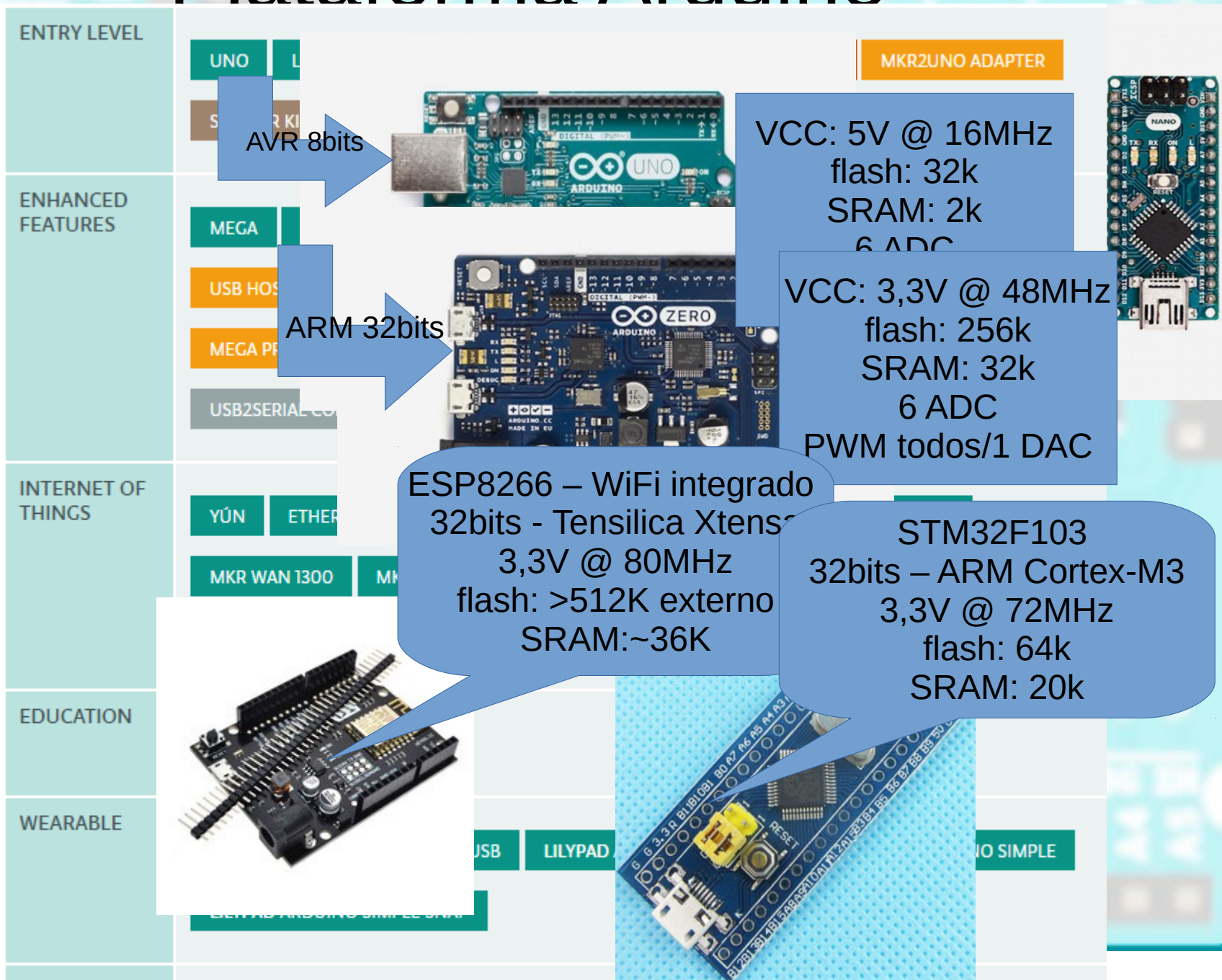
Memória RAM
Armazenamento de variáveis

Processador

Periférico: ADC
Conversor analógico digital

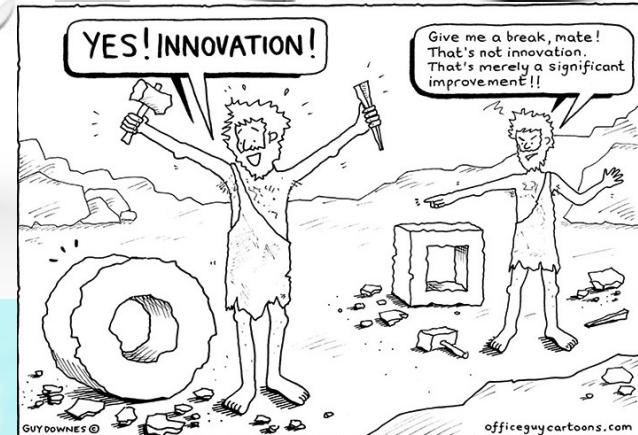
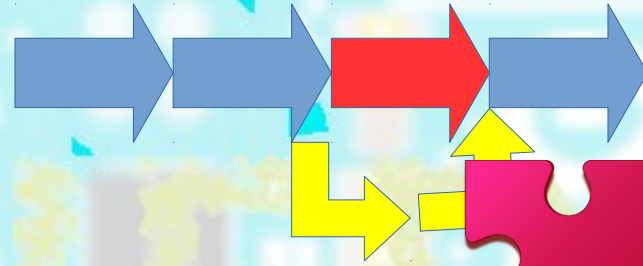
Periférico: Comunicação Serial
SPI e TWI

Plataforma Arduino

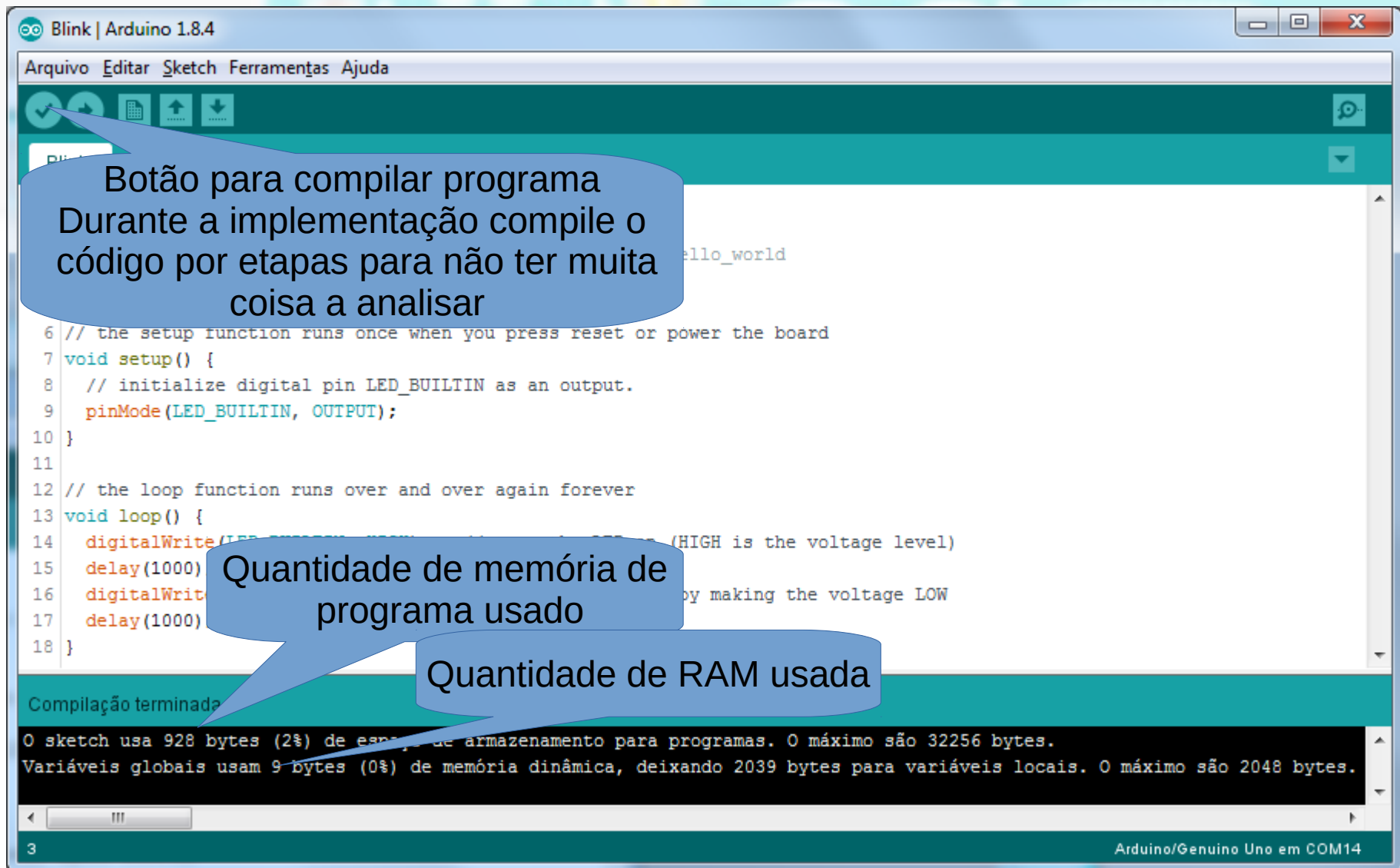


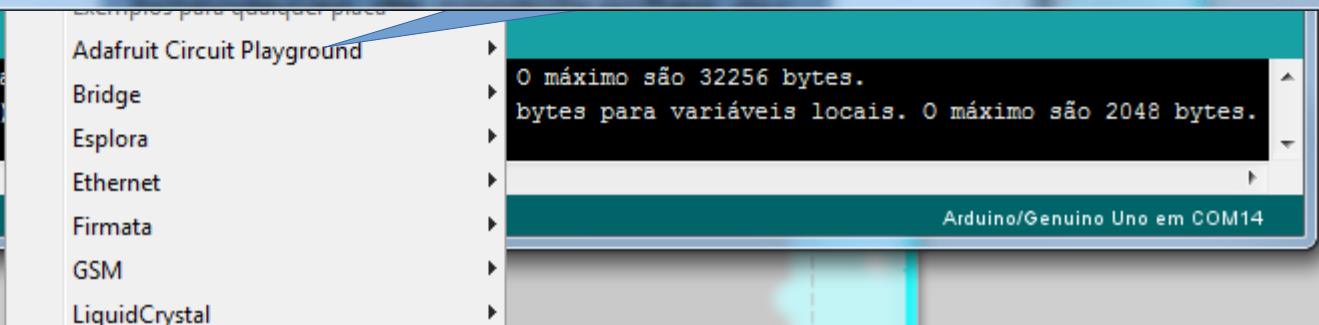
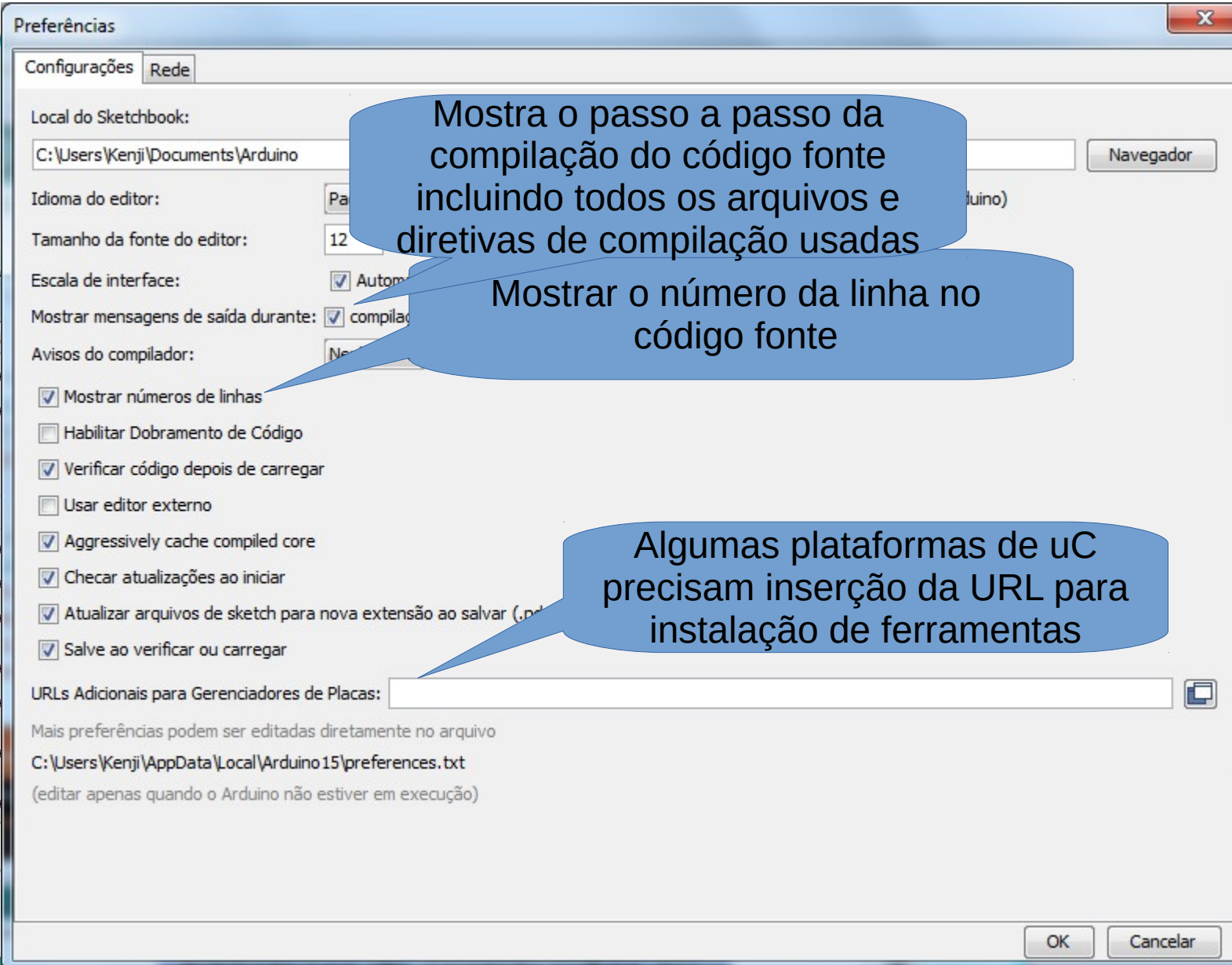
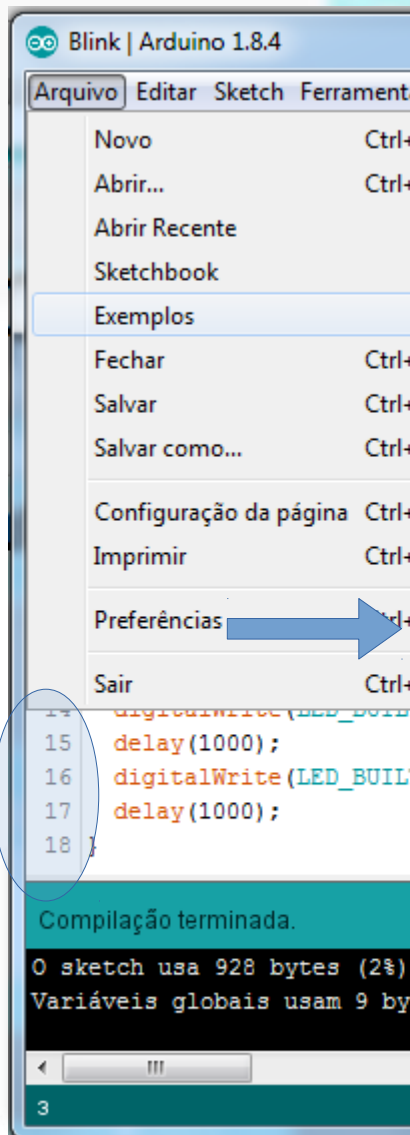
Motivações para criar um bom programa

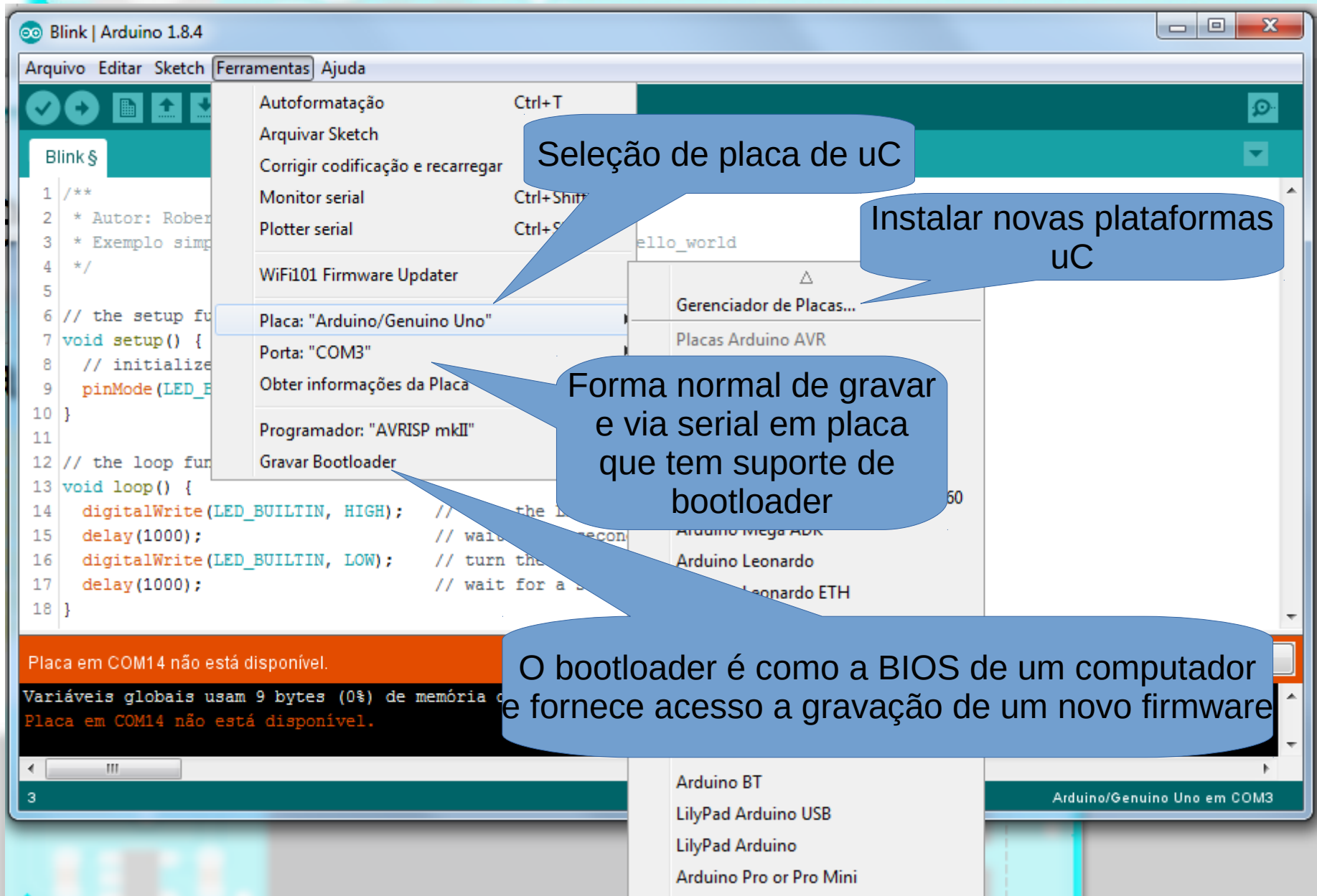
- Confiável
- Solidez
- Ergonomia
- Portátil
- Manutenção
- Eficiente



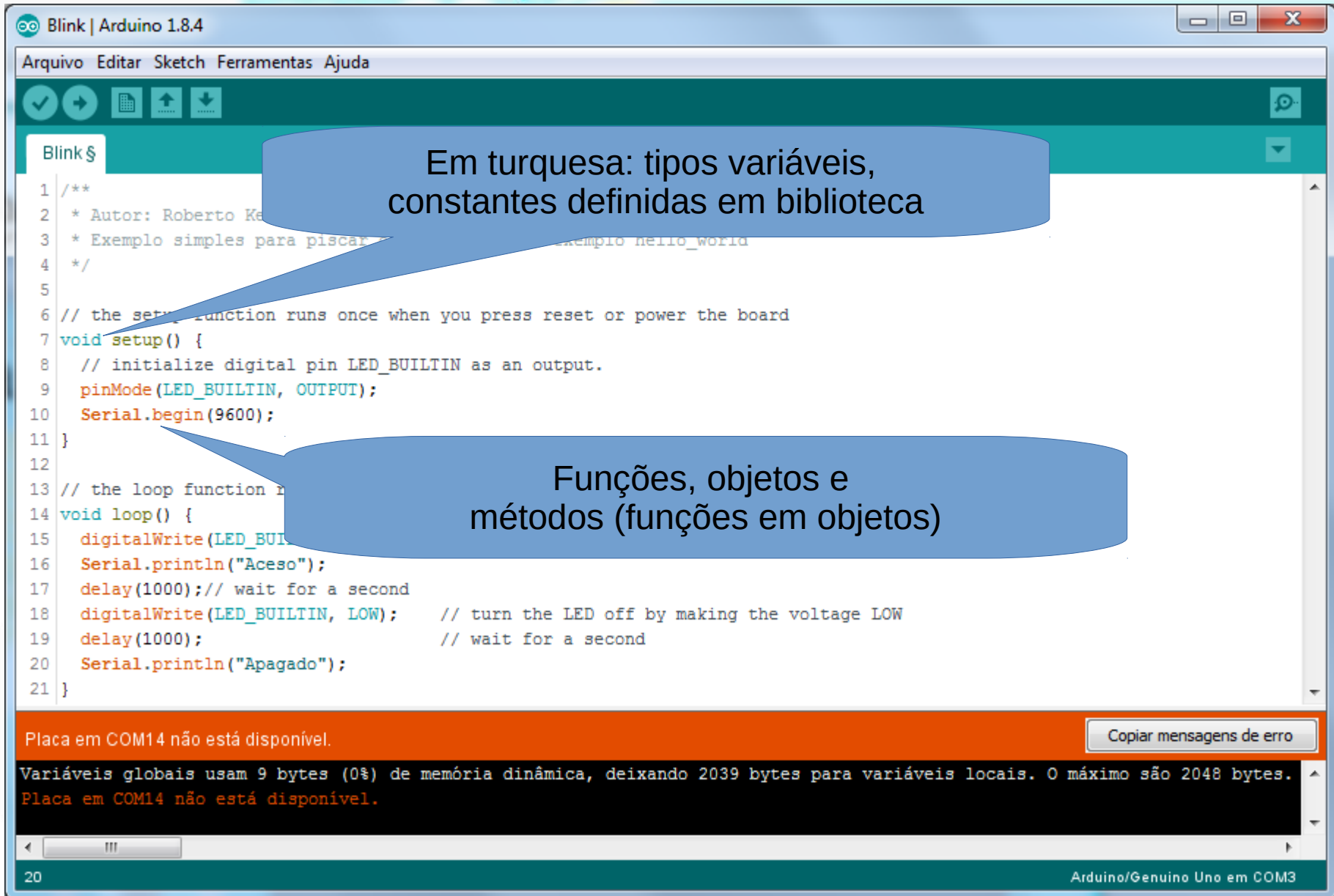
A IDE do Arduino







Realce no código (Highlight)



The image shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is highlighted with colors: blue for keywords, orange for literals and constants, and grey for comments. Two blue callout boxes provide explanations for the highlighting:

- Callout 1 (top):** Em turquesa: tipos variáveis, constantes definidas em biblioteca (In turquoise: variable types, constants defined in library). This points to the turquoise highlighting of `void`, `pinMode`, `Serial`, and `LED_BUILTIN`.
- Callout 2 (bottom):** Funções, objetos e métodos (funções em objetos) (Functions, objects and methods (functions in objects)). This points to the orange highlighting of `setup()`, `loop()`, `digitalWrite`, `Serial.println`, and `delay`.

```
1 /**
2  * Autor: Roberto Ke
3  * Exemplo simples para piscar o LED. Exemplo hello_world
4  */
5
6 // the setup function runs once when you press reset or power the board
7 void setup() {
8   // initialize digital pin LED_BUILTIN as an output.
9   pinMode(LED_BUILTIN, OUTPUT);
10  Serial.begin(9600);
11 }
12
13 // the loop function runs over and over again
14 void loop() {
15   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
16   Serial.println("Aceso");
17   delay(1000); // wait for a second
18   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
19   delay(1000); // wait for a second
20   Serial.println("Apagado");
21 }
```

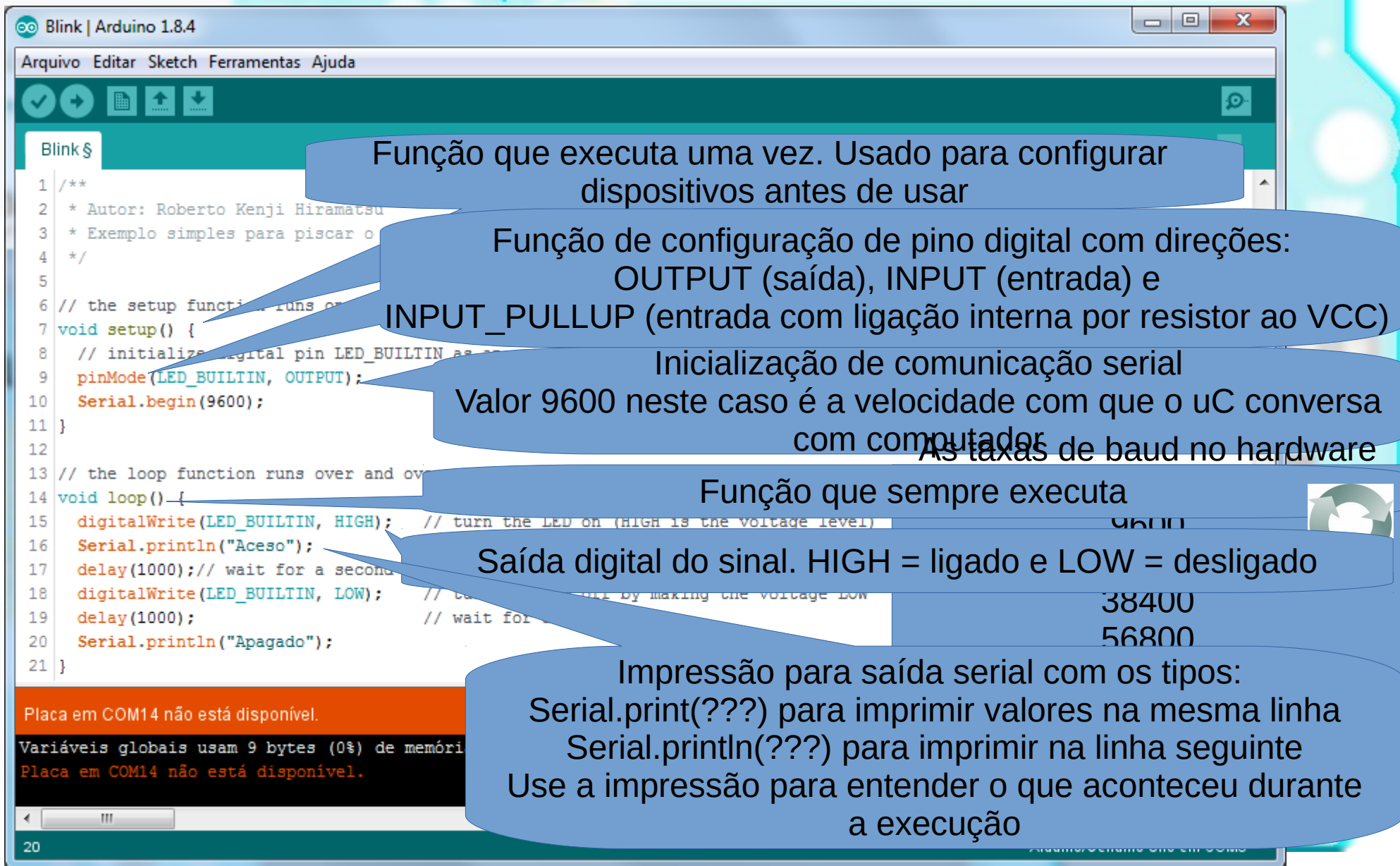
Placa em COM14 não está disponível. Copiar mensagens de erro

Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 2039 bytes para variáveis locais. O máximo são 2048 bytes.

Placa em COM14 não está disponível.

20 Arduino/Genuino Uno em COM3

Anatomia de um programa no Arduino



Blink | Arduino 1.8.4

Arquivo Editar Sketch Ferramentas Ajuda

Blink \$

```
1 /**
2  * Autor: Roberto Kenji Hiramatsu
3  * Exemplo simples para piscar o LED
4  */
5
6 // the setup function runs once when you open the IDE or first power the board
7 void setup() {
8   // initialize digital pin LED_BUILTIN as an output.
9   pinMode(LED_BUILTIN, OUTPUT);
10  Serial.begin(9600);
11 }
12
13 // the loop function runs over and over again
14 void loop() {
15   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
16   Serial.println("Aceso");
17   delay(1000); // wait for a second
18   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
19   delay(1000); // wait for a second
20   Serial.println("Apagado");
21 }
```

Placa em COM14 não está disponível.

Variáveis globais usam 9 bytes (0%) de memória.

Placa em COM14 não está disponível.

20

Função que executa uma vez. Usado para configurar dispositivos antes de usar

Função de configuração de pino digital com direções: OUTPUT (saída), INPUT (entrada) e INPUT_PULLUP (entrada com ligação interna por resistor ao VCC)

Inicialização de comunicação serial

Valor 9600 neste caso é a velocidade com que o uC conversa com computador

As taxas de baud no hardware

Função que sempre executa

Saída digital do sinal. HIGH = ligado e LOW = desligado

Impressão para saída serial com os tipos:
Serial.print(???) para imprimir valores na mesma linha
Serial.println(???) para imprimir na linha seguinte
Use a impressão para entender o que aconteceu durante a execução

Saída serial

Blink | Arduino 1.8.4

Arquivo Editar Sketch Ferramentas Ajuda

Blink \$

```
1 /**
2  * Autor: Roberto Kenji Hiramatsu
3  * Exemplo simples para piscar o led
4  */
5
6 // the setup function runs once when
7 void setup() {
8   // initialize digital pin LED_BUILT
9   pinMode(LED_BUILTIN, OUTPUT);
10  Serial.begin(9600);
11 }
12
13 // the loop function runs over and over
14 void loop() {
15   digitalWrite(LED_BUILTIN, HIGH);
16   Serial.println("Aceso");
17   delay(1000); // wait for a second
18   digitalWrite(LED_BUILTIN, LOW);
19   delay(1000);
20   Serial.println("Apagado");
21 }
```

COM1

Enviar

☒ Auto-rolagem

Ambos, NL e CR 9600 velocidade

Copiar mensagens de erro

Placa em COM14 não está disponível.

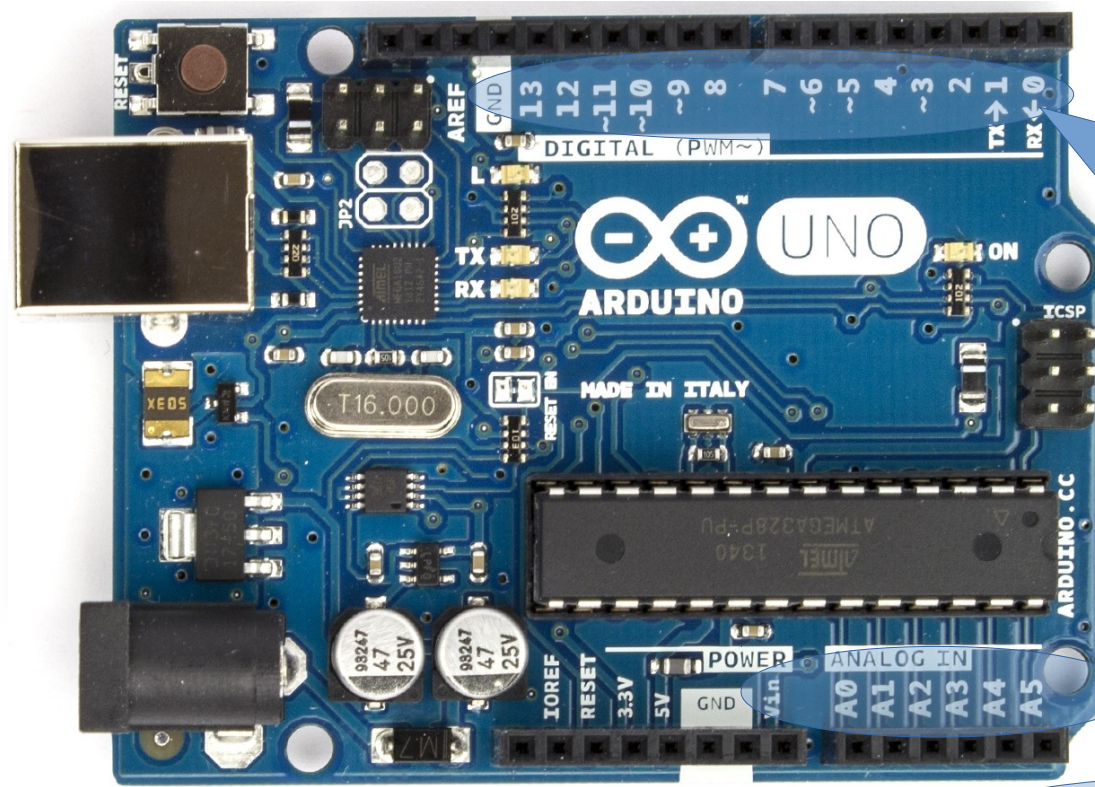
Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 2039 bytes para variáveis locais. O máximo são 2048 bytes.

Placa em COM14 não está disponível.

20

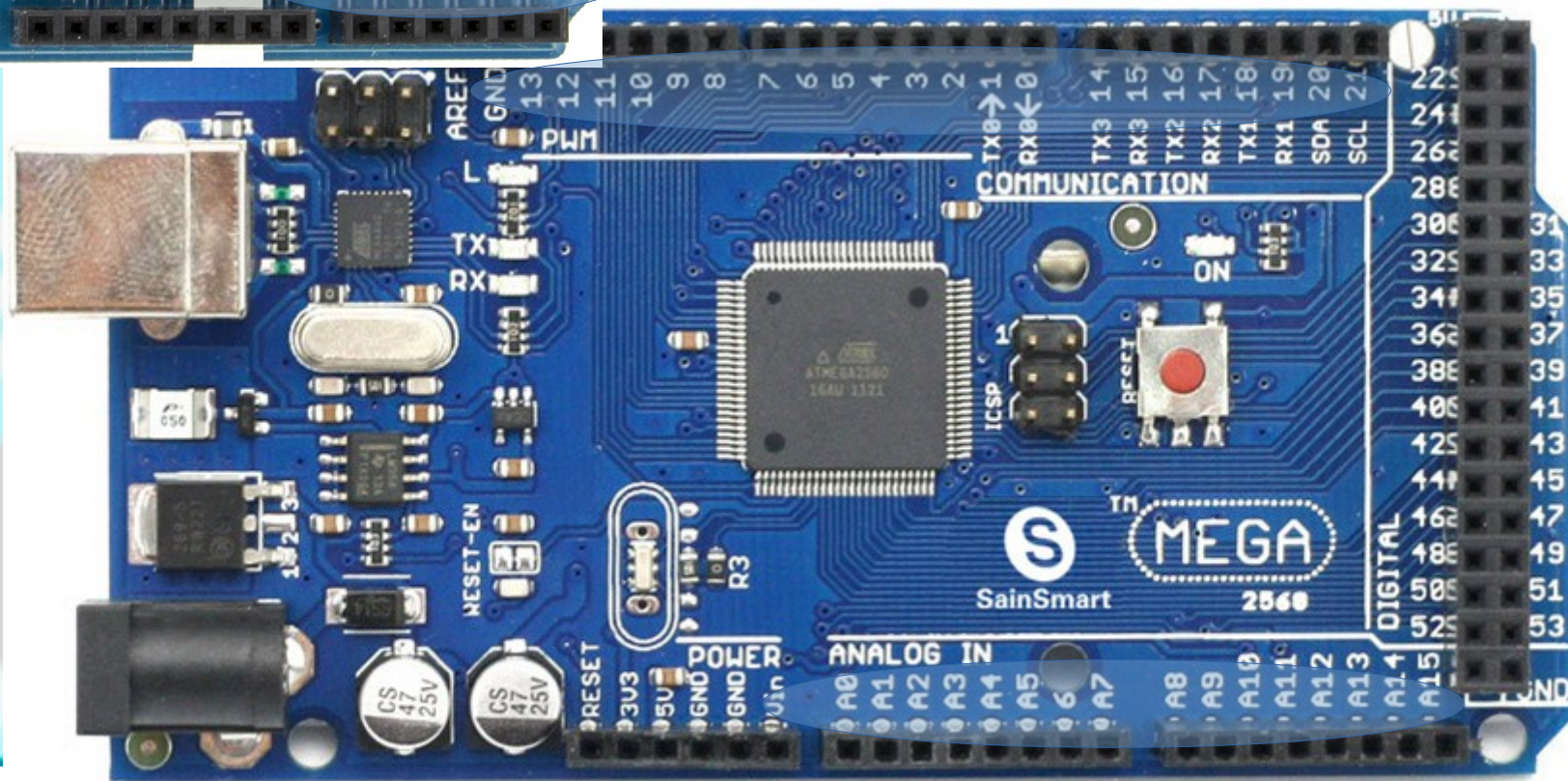
Arduino/Genuino Uno em COM3

Mesma taxa



Nos pinos digitais usa-se diretamente o número do pino digital para acessá-lo definindo no setup o modo com pinMode e depois no loop escrevendo com digitalWrite ou lendo com digitalRead

Nos pinos analógicos usa-se a denominação Ax em que x é o número do pino para as operações como pino digital

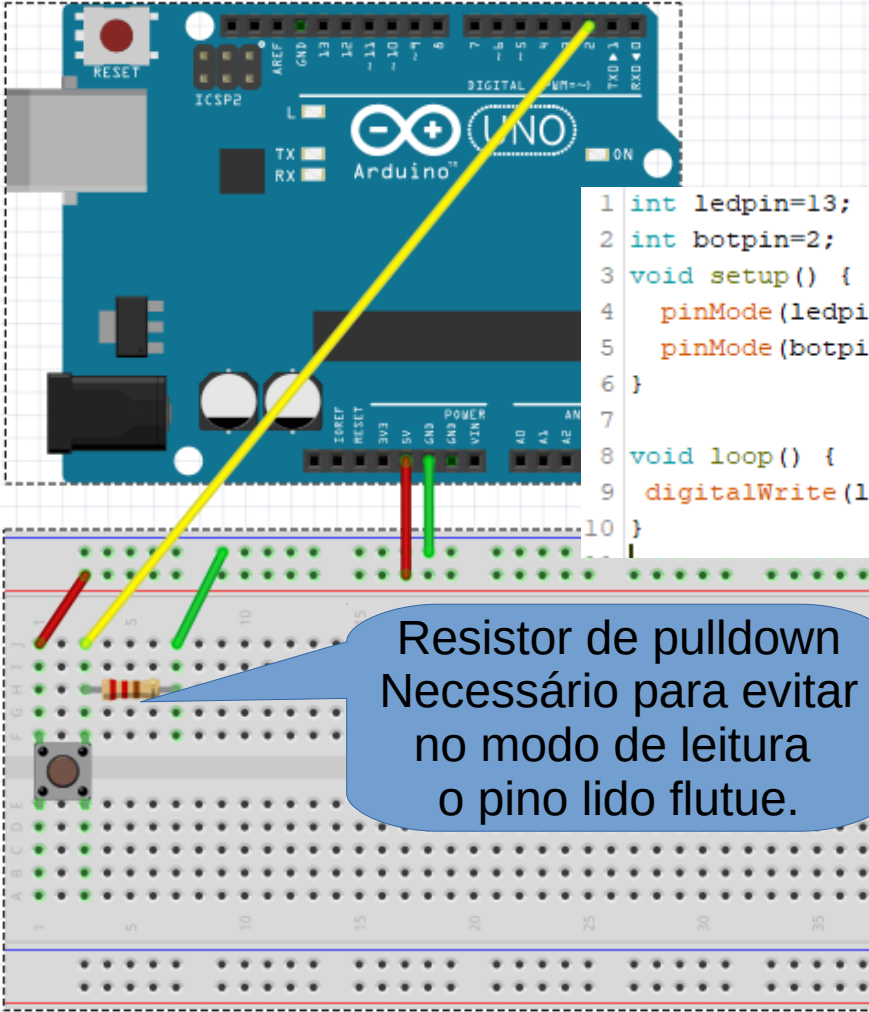


Fritzing

Untitled Sketch.fzz* - Fritzing - [Visão Protoboard]

Arquivo Editar Componente Visão Janela Roteamento Ajuda

Welcome Protoboard Esquemático



Resistor de pulldown
Necessário para evitar
no modo de leitura
o pino lido flutue.

```
1 int ledpin=13;
2 int botpin=2;
3 void setup() {
4   pinMode(ledpin, OUTPUT);
5   pinMode(botpin, INPUT);
6 }
7
8 void loop() {
9   digitalWrite(ledpin, digitalRead(botpin));
10 }
```

Arduino1

3V3 5V Power Vin

RST

D13

D12

D11 PWM

D10 PWM

D9 PWM

D8

D7

D6 PWM

D5 PWM

D4

Digital Input/Output

```
1 int ledpin=13;
2 int botpin=2;
3 void setup() {
4   pinMode(ledpin, OUTPUT);
5   pinMode(botpin, INPUT);
6 }
7
8 void loop() {
9   digitalWrite(ledpin, digitalRead(botpin));
10 }
```

Roteamento

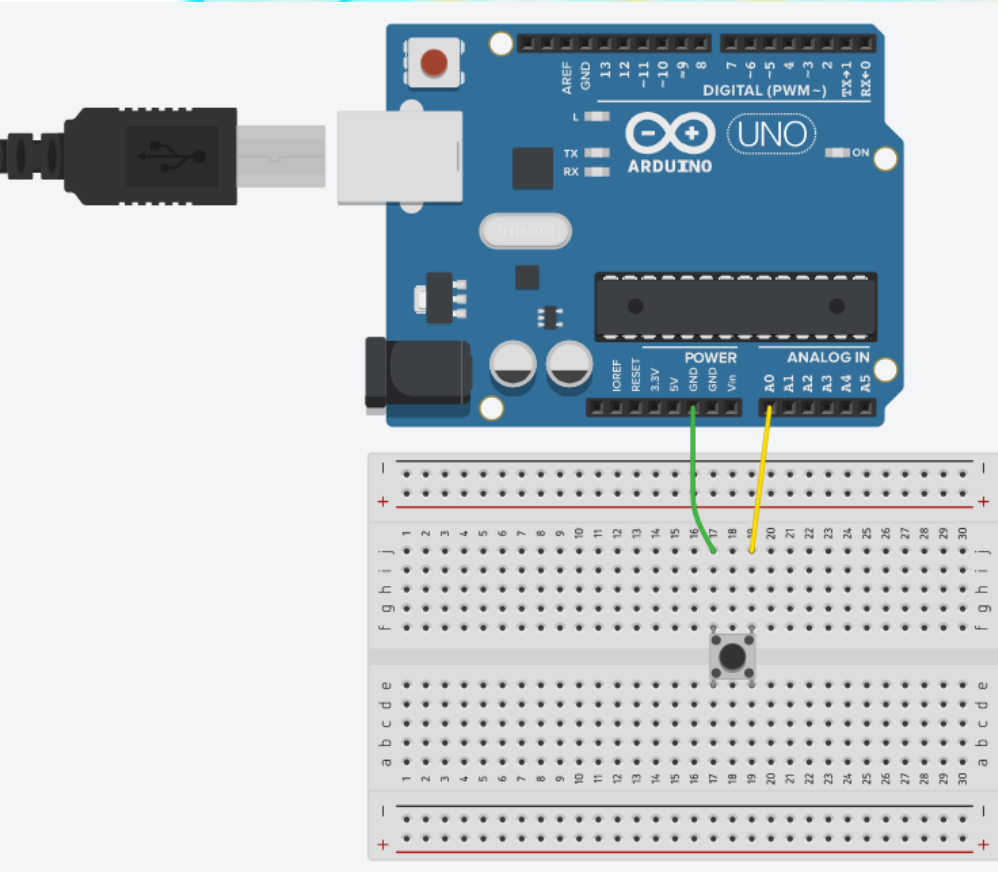
Compartilhar

Componentes

Marcadores

Modo de leitura pullup

- Não precisa de resistor, pois usa resistor interno ao uC ligado ao VCC no pino digital



```
1
2  int ledpin=13;
3  int botpin=A0;
4  void setup()
5  {
6      pinMode(ledpin, OUTPUT);
7      pinMode(botpin, INPUT_PULLUP);
8  }
9
10 void loop()
11 {
12     if(digitalRead(botpin))
13         digitalWrite(ledpin, HIGH);
14     else
15         digitalWrite(ledpin, LOW);
16     delay(10);
17 }
```



roberto.hiramatsu@ufrpe.br

DIALIGHT P/N	EMITTED COLOR	MATERIAL	LENS COLOR	LUMINOUS INTENSITY (mcd)			DOMINANT WAVELENGTH (nm)			FORWARD VOLTAGE (V)			VIEWING ANGLE
				If = 20 ma			If = 20 ma			If = 20 ma			° DEGREES
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
598-8010-107F	RED	AlInGaP	Water Clear	30	40	80	630	635	642	1.7	2.2	2.4	140
598-8020-107F	RED-ORANGE	AlInGaP	Water Clear	120	150	200	620	625	630	1.7	2	2.4	140
598-8030-107F	ORANGE	AlInGaP	Water Clear	70	-	150	600	-	610	1.7	2	2.4	140
598-8040-107F	YELLOW	AlInGaP	Water Clear	100	130	160	590	-	595	1.7	2	2.2	140
598-8050-107F	YELLOW	AlInGaP	Water Clear	100	130	160	583	-	590	1.7	2	2.4	140
598-8060-107F	YELLOW-GREEN	AlInGaP	Water Clear	20	40	60	570	-	575	1.8	2	2.4	140
598-8070-107F	GREEN	GaP	Water Clear	10	20	40	562	-	570	1.8	2	2.4	140
598-8081-107F	GREEN	InGaN	Water Clear	220	300	400	520	523	525	3	3.2	3.5	140
598-8091-107F	BLUE	InGaN	Water Clear	90	140	160	470	473	475	2.8	3.2	3.5	140