



# Uma introdução à plataforma Arduino

Felipe Kühne  
[fkuhne@pucrs.br](mailto:fkuhne@pucrs.br)

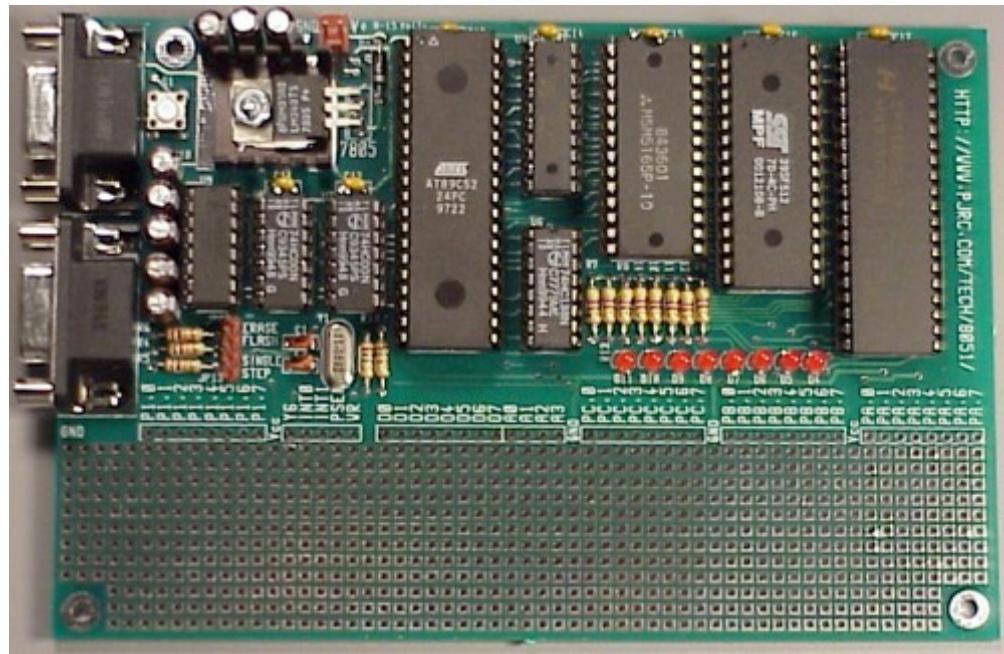
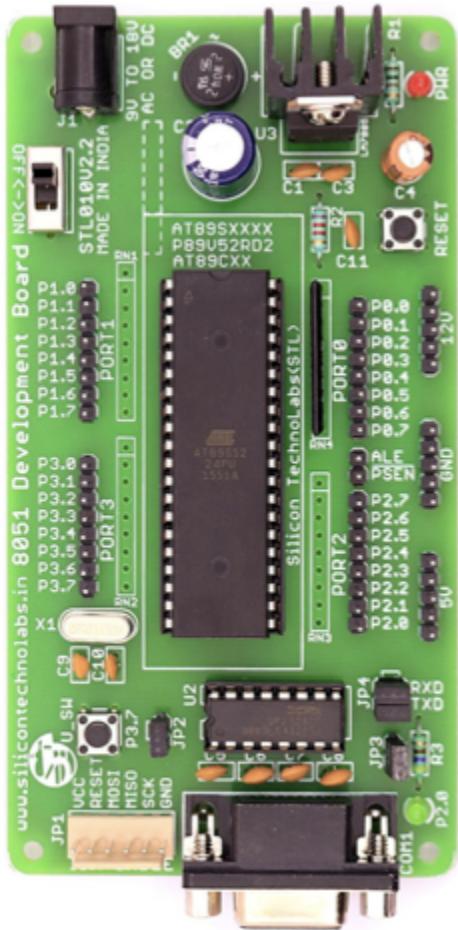
Maio/2018



[github.com/fkuhne/aday18](https://github.com/fkuhne/aday18)



**ANTES DO ARDUINO**



## Assembly: 122 linhas

```

rs    equ   P1.5 ;Reg Select ligado em P1.7
rw    equ   P1.6 ;Read/Write ligado em P1.5
en    equ   P1.7 ;Enable ligado em P1.5
dat   equ   P2 ;Bite de dados em todo P2

org  0000h ;origem no endereçamento 00h da memória
acall delay500ms ;guarda 500ms para estabilizar

início:
    acall lcd_init ;Chama sub rotina de inicialização
    mov dptr,#LCD1 ;Move mensagem para DPTR
    acall send_lcd ;Chama sub rotina para enviar mensagens para LCD
    ajmp $ ;Finaliza programa aqui

lcd_init:
    mov a,#60d ;Sub Rotina para Inicialização do Display
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00011100b para acc
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00001100b para acc
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00001100b para acc
    acall config ;chama sub rotina config
    mov a,#6d ;move literal 00000110b para acc
    acall config ;chama sub rotina config
    ret ;retorna

config:
    ;Sub Rotina de Configuração
    clr en ;limpa pino en
    clr rs ;limpa pino rs
    clr rw ;limpa pino rw
    acall wait ;guarda 55us
    setb en ;seta enable
    acall wait ;guarda 55us
    mov dat,a ;carrega dados em Port P2
    acall wait ;guarda 55us com barramento igual ao valor de acc
    clr en ;limpa pino en
    acall wait ;guarda 55us
    ret ;retorna

send_lcd:
    mov R0,#0d ;Sub Rotina para Enviar dados ao LCD
    ;Move valor 0d para R0

send:
    mov a,R0 ;Move conteúdo de R0 para acc
    inc R0 ;Incrementa acc
    movec a,R0+dptr ;Move o byte relativo de dptr somado
    ;com o valor de acc para accouricou
    acall w_dat ;chama sub rotina para escrita de dados
    cjne R0,#14d,send ;compara R0 com valor de colunas e desvia se for diferente
    ret ;retorna

w_dat:
    ;Sub Rotina para preparar para escrita de mensagem
    clr en ;limpa enable
    setb rs ;seta rs
    clr rw ;limpa rw (escrita)
    acall wait ;guarda 55us
    setb en ;seta enable
    acall wait ;guarda 55us
    mov dat,a ;carrega mensagem
    acall wait ;guarda 55us
    clr en ;limpa enable
    acall wait ;guarda 55us
    ret ;retorna

wait:
    ;Sub Rotina para atraso de 55us

aux0:
    mov R5,#055d ;Carrega 55d em R5
    djns R5,aux0 ;Decrementsa R5. R5 igual a zero? N.o, desvia para aux
    ret ;Sai, retorna

delay500ms:
    ;Sub Rotina para atraso de 500ms
    mov R1,#0fah ;1 x 250 | círculos de máquina do mnemônico call
    ;1 x 250 | move o valor 250 decimal para o registrador R1

aux1:
    mov R2,#0f9h ;1 x 250 | move o valor 249 decimal para o registrador R2
    nop ;1 x 250
    djns R2,aux2 ;2 x 250 x 249 = 622500 | decrementsa o R2 até chegar a zero
    djns R1,aux1 ;2 x 250 | decrementsa o R1 até chegar a zero
    ret ;2 | retorna para a função main
    ;-----Total = 500000 us -- 500 ms = 0,5 seg

LCD1:
    db 'Hello World!' ;Final do programa
    and $ ;Finaliza o bloco

```

## C: 77 linhas

```

#include<xc4000.h>
#include<16f877a.h>

void delay_ms(int);
void send(char);
void display(char);
void custom(void);
void init(void);

char custom_dchar[] = {0x00,0x04,0x0E,0x12,0x2E,0x3D,0x00,
                      0x00,0x04,0x0E,0x12,0x2E,0x3D,0x00,
                      0x00,0x1A,0x15,0x11,0x11,0x1A,0x04,0x00,
                      0x00,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,
                      0x00,0x11,0x10,0x10,0x10,0x10,0x10,0x10,0x10,
                      0x00,0x0E,0x15,0x12,0x12,0x12,0x12,0x12,0x12};

void delay_ms(int d)
{
    unsigned char i;
    for(i=d;i-->0)
        {
            for(j=250,Lcd,i--);
            for(j=250,Lcd,i--);
        }
}

void send(char c)
{
    Lcd;
    xad;
    adr;
    dat;
    delay(5);
    adr;
}

void display(char c)
{
    Lcd;
    xad;
    adr;
    dat;
    delay(5);
    adr;
}

void custom(void)
{
    int i;
    set16(0x00);
    for(j=0;j<16;j++)
        display(custom_dchar[j]),
        end16();
}

void string(char *p)
{
    while(*p)
        {
            display(*p++);
        }
}

void init(void)
{
    cmd16(0x01);
    cmd16(0x01);
    cmd16(0x01);
    cmd16(0x00);
}

void main()
{
    init();
    custom();
    string("Hello World");
    while(1);
}

```

## Assembly: 122 linhas

```
rs    equ   P1.5 ;Reg Select ligado em P1.7
rw    equ   P1.6 ;Read/Write ligado em P1.6
en    equ   P1.7 ;Enable ligado em P1.5
dat   equ   P2      ;Bit de dados em todo P2

org 0000h ;origem no endereço 00h de memória
delay500ms: ;aguarda 500ms para estabilizar

inicio:
    acall lcd_init ;Chama sub rotina de inicializa.c
    mov d0
    acall
    ajmp

lcd_init:
    mov
    acall
    mov
    acall
    mov
    acall
    mov
    acall
    ret

config:
    clr
    clr rs      ;limpa pino rs
    clr rw      ;limpa pino rw
    acall wait
    setb en
    acall
    mov
    acall
    wait
    ;carrega dados
    mov dat,a
    ;guarda 55us
    acall
    wait
    ;guarda dados
    mov dat,a
    ;guarda 55us
    acall
    en      ;limpa pino en
    acall
    wait
    ;guarda 55us
    ret

send_lcd:
    mov R0,#0d ;Sub Rotina para enviar dados ao LCD
    ;Move valor 0d para R0

send:
    mov
    inc
    move
    acall
    cje
    ret

w_dat:
    clr
    setb
    clr
    acall
    setb
    acall
    wait
    ;guarda 55us
    mov dat,a
    ;carrega dados
    acall
    wait
    ;guarda 55us
    clr
    en      ;limpa enable
    acall
    wait
    ;guarda 55us
    ret
    ;retorna

wait:
    ;Sub Rotina para atraso de 55us

aux0:
    mov R5,#055d ;Carrega 55d em R5
    djnx R5,aux0 ;Decrementa R5 se igual a zero? N.o, desvia para aux
    ret

delay500ms:
    mov
    nop
    nop
    nop
    nop
    nop

aux1:
    mov
    nop
    nop
    nop
    nop
    nop
    ; 1 x 250 = 249 = 62250
    nop
    ; 1 x 250 = 249 = 62250
    nop
    ; 1 x 250 = 249 = 62250
    nop
    ; 1 x 250 = 249 = 62250
    nop

aux2:
    nop
    nop
    nop
    ; 2 x 250 = 249 = 124500 | decrementa o R2 ate chegar a zero
    djnx R2,aux2
    ; 2 x 250 = 249 = 124500 | decrementa o R1 ate chegar a zero
    djnx R1,aux1
    ; 2 x 250 = 249 = 124500 | retorna para a função main
    ret
    ; 2
    ; Total = 500000 us -- 500 ms = 0,5 seg

LCD1:
    db 'Hello World'
    and
    ;Final do programa
```

Assembly  
code

Linking

Executable

## C: 77 linhas

```
initializar51.h
#include <xc8.h>
#include <math.h>
#include <string.h>
#include <avr/interrupt.h>

void delay();
void and();
void display();
void carim();
void string();
void init();

unsigned char curm[1] = {(0x00),0x04,0x00,0x00,0x15,0x00,0x00,0x00};

void delay()
{
    unsigned char i;
    for (i=0;i<1000000;i++);
}

void and()
{
    unsigned char i;
    for (i=0;i<1000000;i++);
}

void display()
{
    lcd0,
    xaddr,
    yaddr,
    delay(5),
    end;
}

void curm()
{
    lcd0,
    xaddr,
    yaddr,
    display(),
    end;
}

void carim()
{
    lcd0,
    xaddr,
    yaddr,
    delay(5),
    end;
}

void string()
{
    while (*p)
    {
        display();
        p++;
    }
}

void init()
{
    init();
    curm();
    string("Hello World");
}

void main()
{
    init();
    curm();
    string("Hello World");
}
```

C code

Pre-process

Compilation

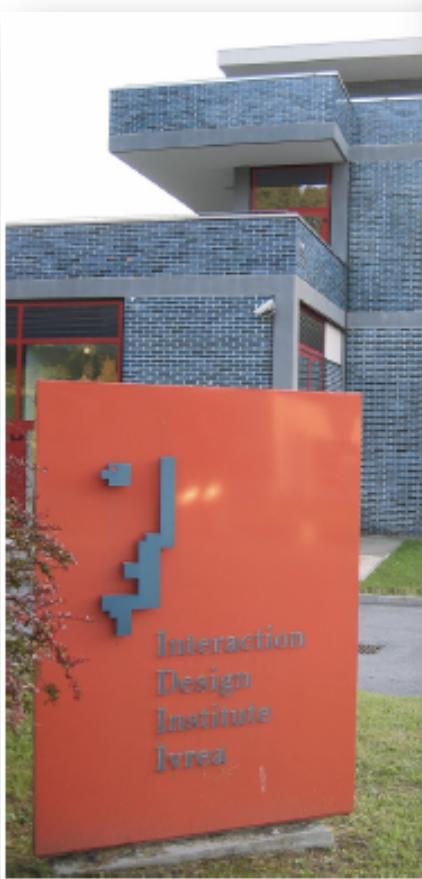
Assembly  
code

Linking

Executable

**2003: AÍ VEM O ARDUINO**

# Interaction Design Institute Ivrea

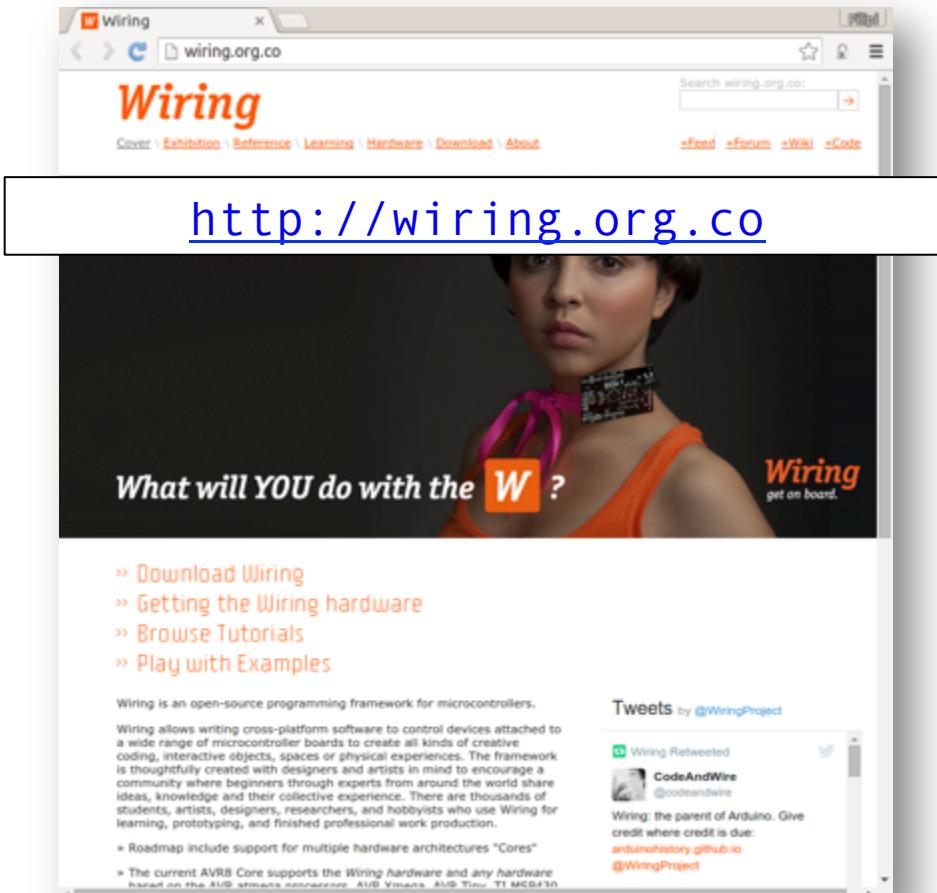


~2000's



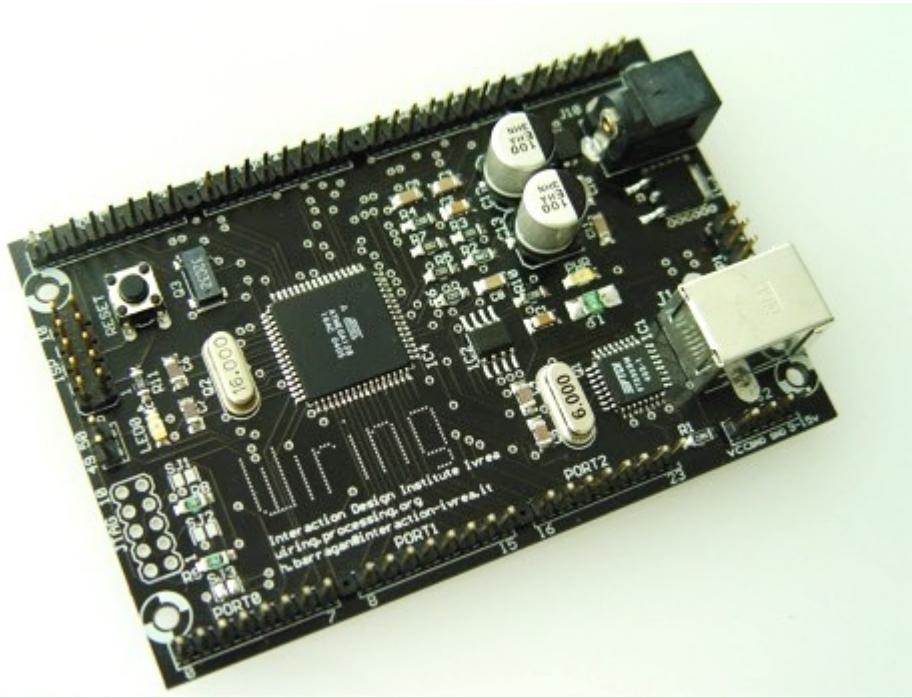
BasicSTAMP  
US\$100

# 2003: Wiring



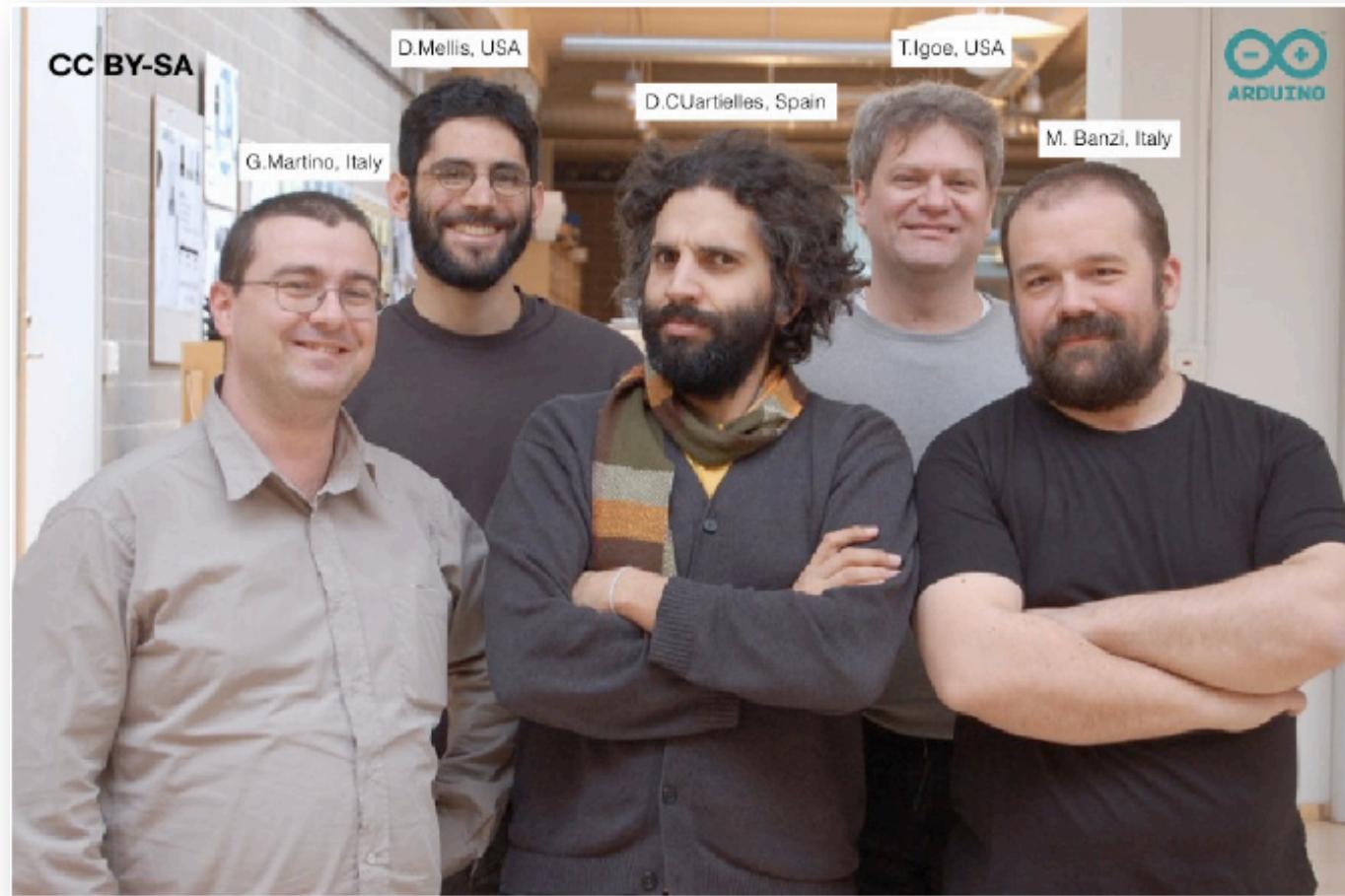
“...to make it easy for artists and designers to work with electronics, by abstracting away the often complicated details of electronics.”

# 2003: Wiring

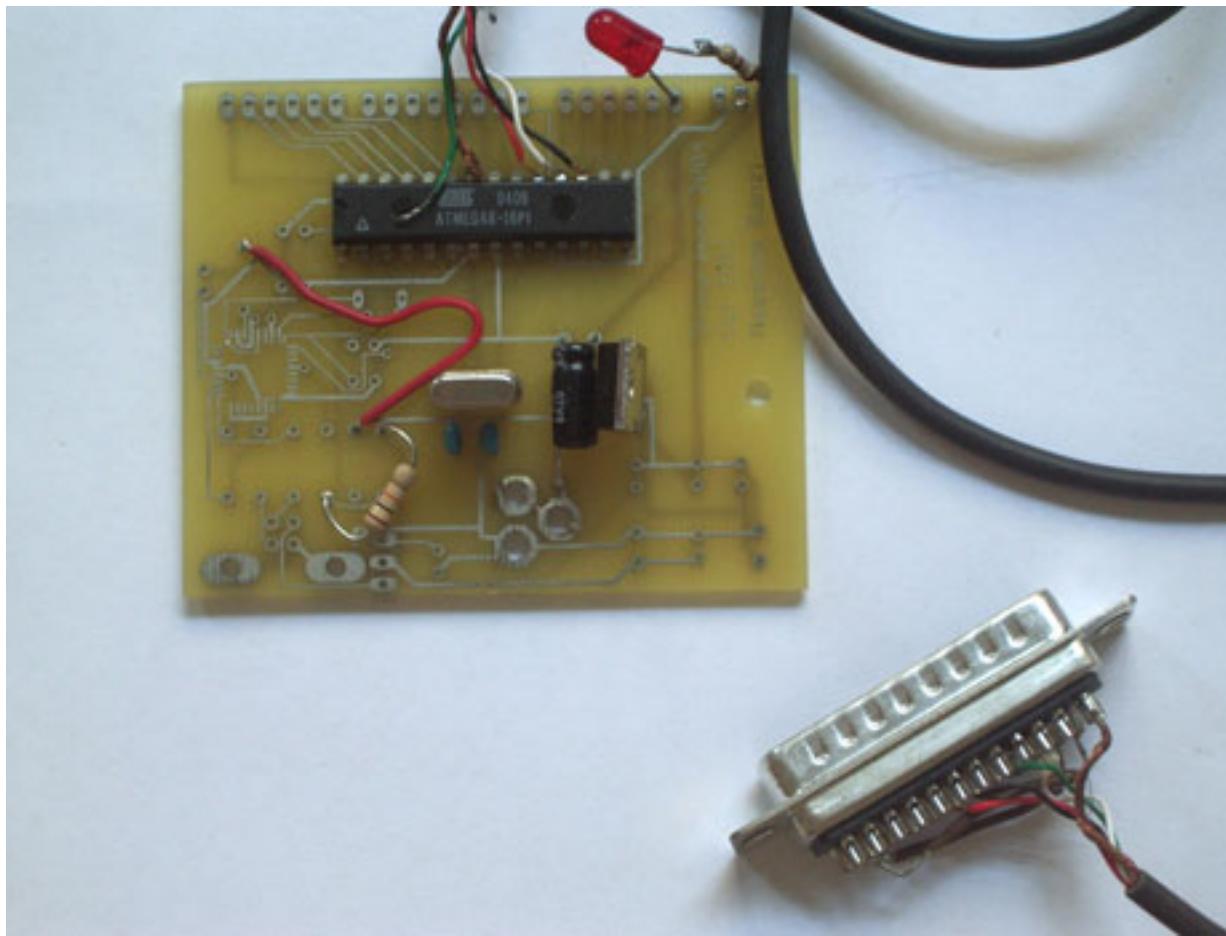


Wiring Board  
**US\$50**

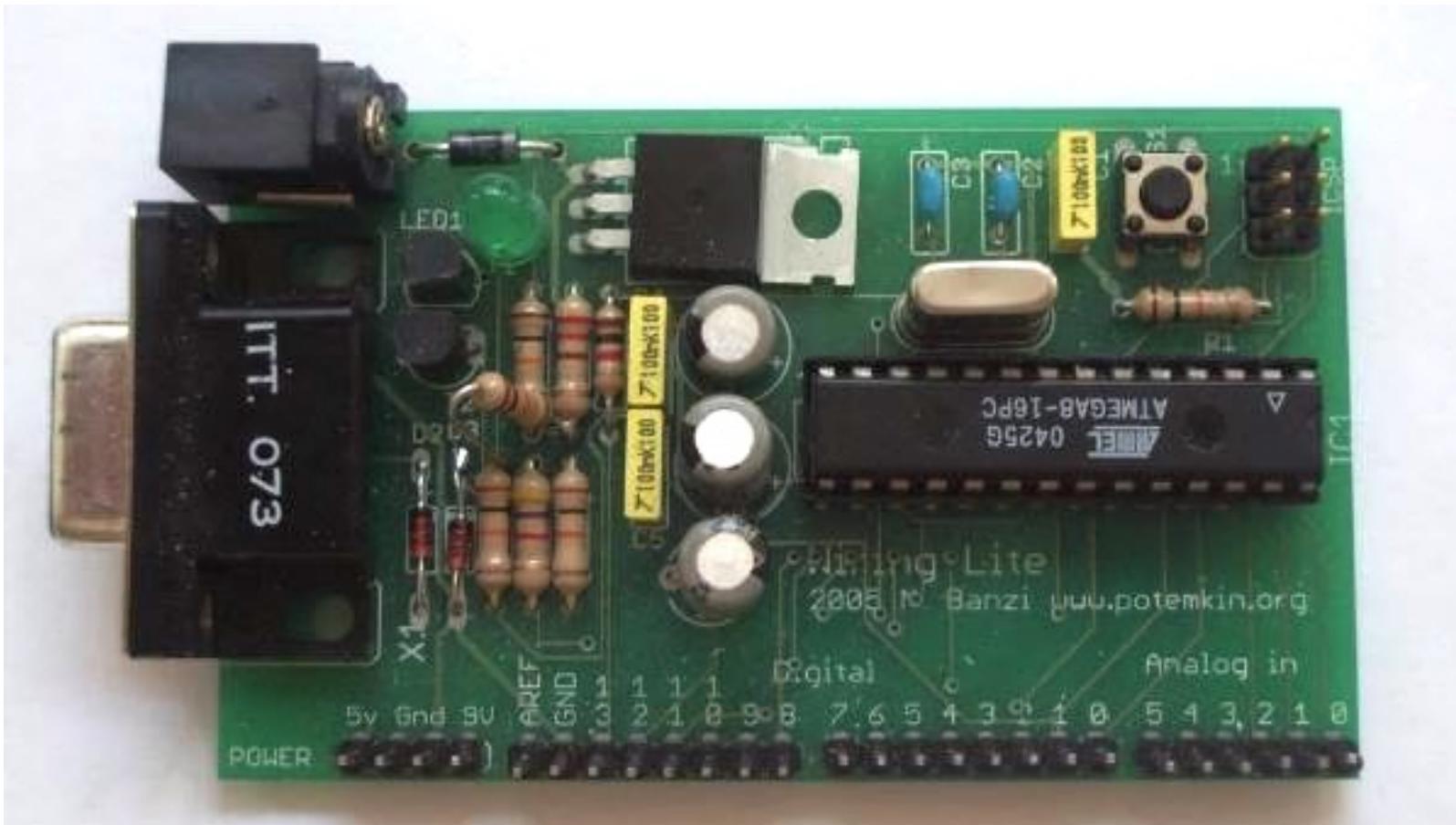
# 2005: Arduino



# 2005: Arduino



# 2005: Arduino



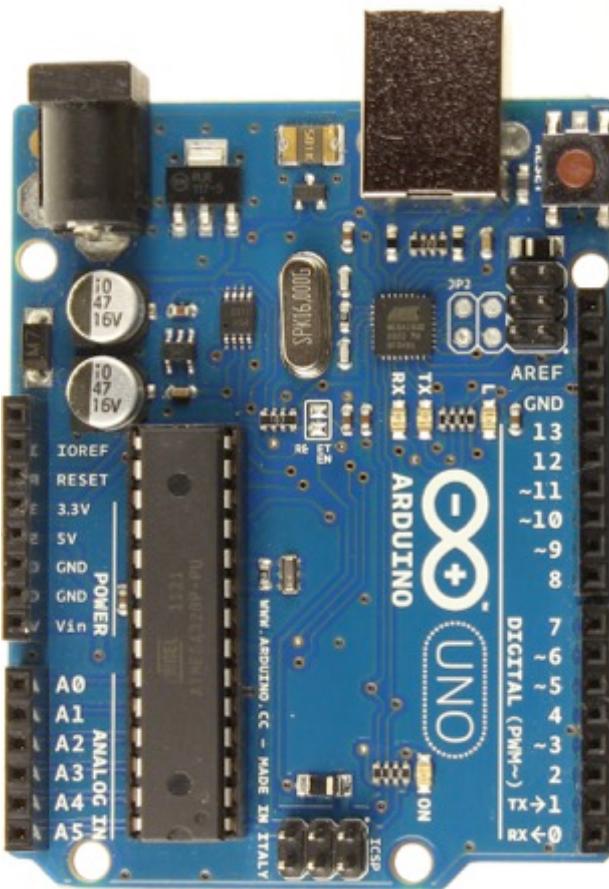
“Arduino é uma plataforma de prototipagem eletrônica de código aberto. É destinado a artistas, designers, hobistas e qualquer interessado em criar objetos ou ambientes interativos.”

“Arduino pode sentir o ambiente recebendo entradas de vários sensores, e pode afetar seu entorno controlando luzes, motores, e outros atuadores.”

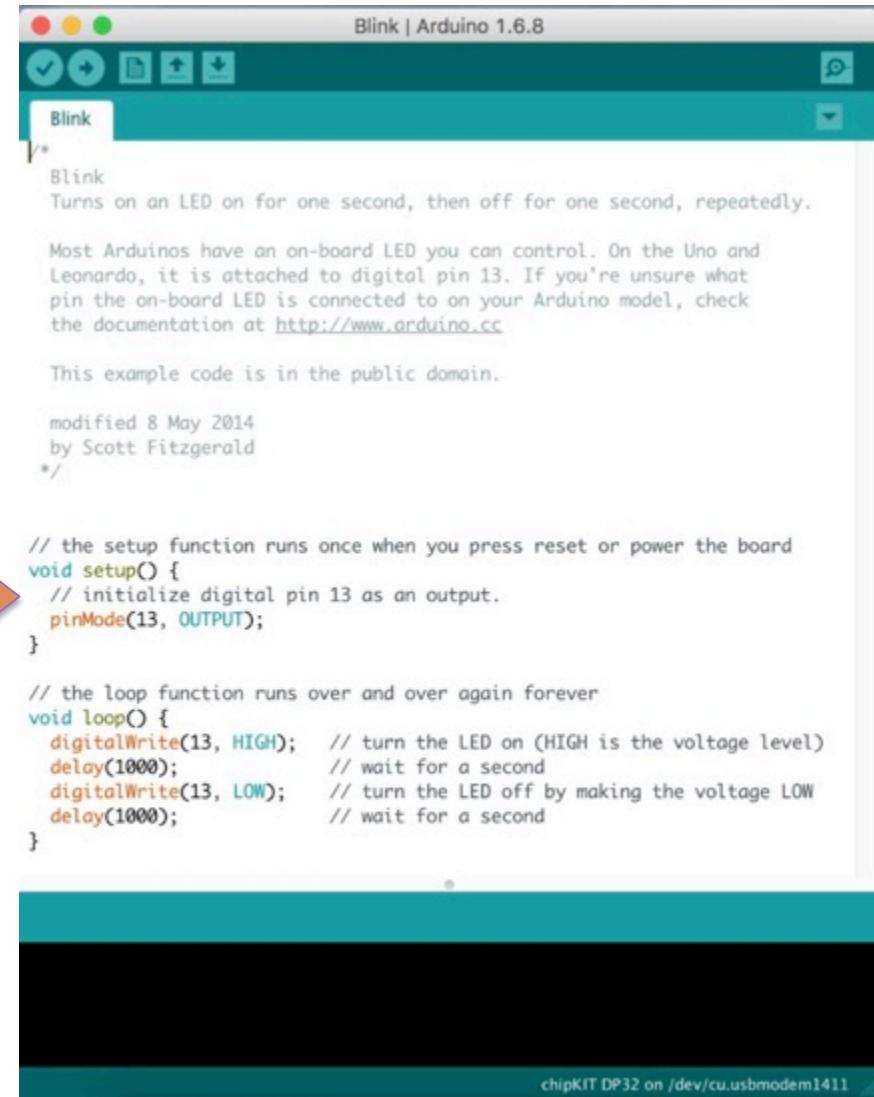
“O microcontrolador da placa é programado usando uma linguagem de programação baseada no framework **Wiring** e um ambiente de desenvolvimento baseado no **Processing**.”

“As placas Arduino podem ser feitas a mão ou compradas pré-montadas; o software pode ser baixado **de graça**. Os projetos estão disponíveis sob uma licença de código aberto, **você é livre** para adaptá-los às suas necessidades.”

**OU SEJA...**



Micro-controlador  
com entradas e  
saídas



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.8". The main area displays the "Blink" example sketch. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

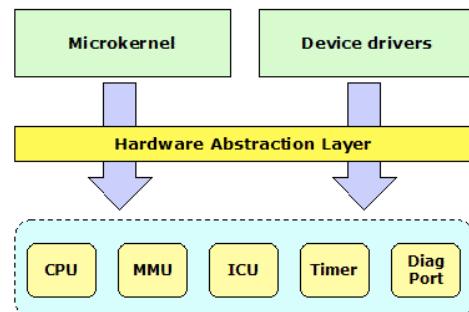
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

At the bottom of the IDE window, a status bar indicates "chipKIT DP32 on /dev/cu.usbmodem1411".

Ambiente de  
programação  
extremamente simples



```
// initialize digital pin 13 as an output.  
pinMode(13, OUTPUT);  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000); // wait for a second  
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
    delay(1000); // wait for a second  
}
```

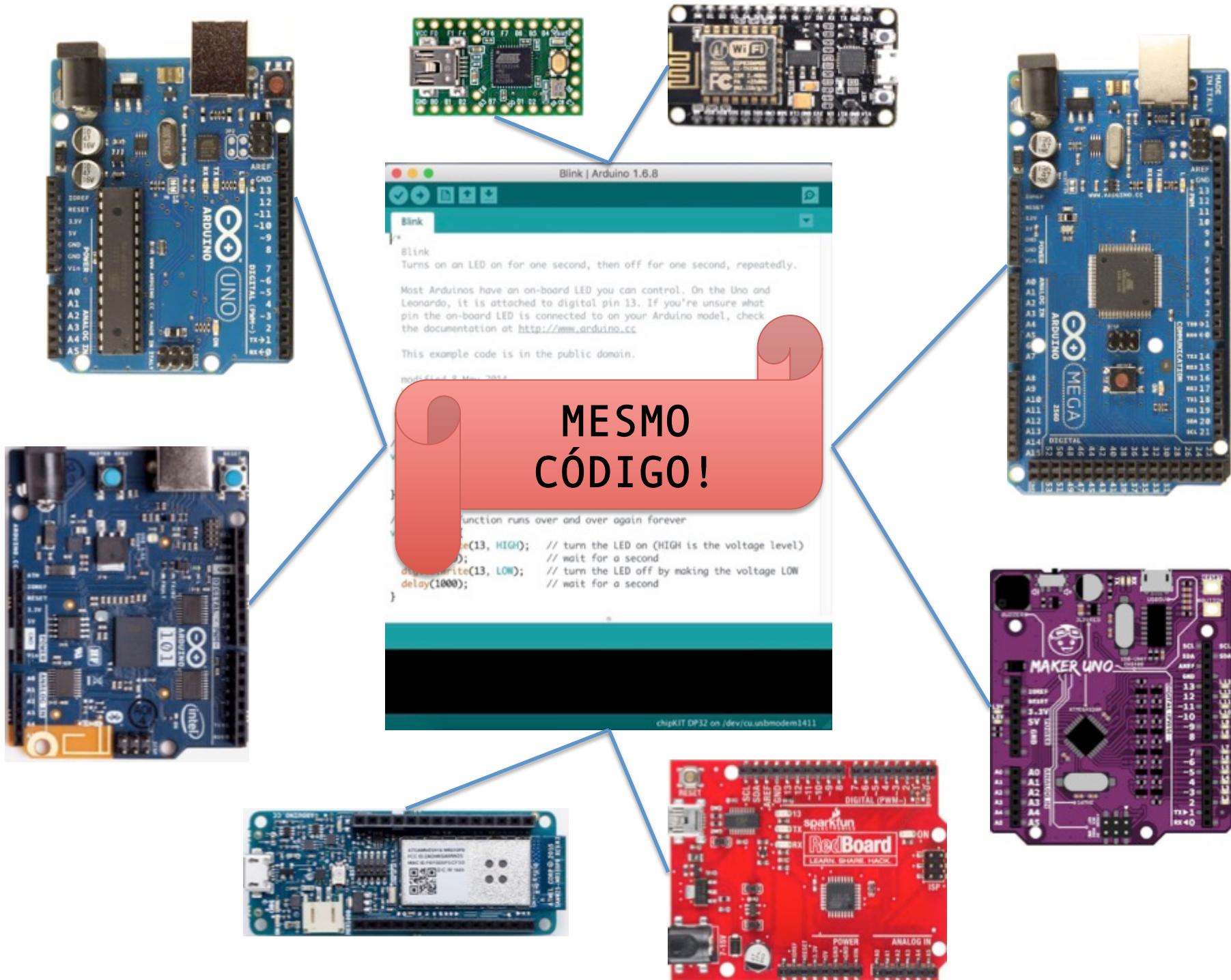


*Hardware  
Abstraction  
Layer*

Vasto número  
de bibliotecas



Comunidade



**RELEMBRANDO...**

## Assembly: 122 linhas

```

rs    equ   P1.5 ;Reg Select ligado em P1.7
rw    equ   P1.6 ;Read/Write ligado em P1.5
en    equ   P1.7 ;Enable ligado em P1.5
dat   equ   P2 ;Bite de dados em todo P2

org  0000h ;origem no endereçamento 00h da memória
acall delay500ms ;guarda 500ms para estabilizar

início:
    acall lcd_init ;Chama sub rotina de inicialização
    mov dptr,#LCD1 ;Move mensagem para DPTR
    acall send_lcd ;Chama sub rotina para enviar mensagens para LCD
    ajmp $ ;Finaliza programa aqui

lcd_init:
    mov a,#60d ;Sub Rotina para Inicialização do Display
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00011100b para acc
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00001100b para acc
    acall config ;chama sub rotina config
    mov a,#14d ;move literal 00001100b para acc
    acall config ;chama sub rotina config
    mov a,#6d ;move literal 00000110b para acc
    acall config ;chama sub rotina config
    ret ;retorna

config:
    ;Sub Rotina de Configuração
    clr en ;limpa pino en
    clr rs ;limpa pino rs
    clr rw ;limpa pino rw
    acall wait ;guarda 55us
    setb en ;seta enable
    acall wait ;guarda 55us
    mov dat,a ;carrega dados em Port P2
    acall wait ;guarda 55us com barramento igual ao valor de acc
    clr en ;limpa pino en
    acall wait ;guarda 55us
    ret ;retorna

send_lcd:
    mov R0,#0d ;Sub Rotina para Enviar dados ao LCD
    ;Move valor 0d para R0

send:
    mov a,R0 ;Move conteúdo de R0 para acc
    inc R0 ;Incrementa acc
    movec a,R0+dptr ;Move o byte relativo de dptr somado
    ;com o valor de acc para accouricou
    acall w_dat ;chama sub rotina para escrita de dados
    cjne R0,#14d,send ;compara R0 com valor de colunas e desvia se for diferente
    ret ;retorna

w_dat:
    ;Sub Rotina para preparar para escrita de mensagem
    clr en ;limpa enable
    setb rs ;seta rs
    clr rw ;limpa rw (escrita)
    acall wait ;guarda 55us
    setb en ;seta enable
    acall wait ;guarda 55us
    mov dat,a ;carrega mensagem
    acall wait ;guarda 55us
    clr en ;limpa enable
    acall wait ;guarda 55us
    ret ;retorna

wait:
    ;Sub Rotina para atraso de 55us

aux0:
    mov R5,#055d ;Carrega 55d em R5
    djns R5,aux0 ;Decrementsa R5. R5 igual a zero? N.o, desvia para aux
    ret ;Sai, retorna

delay500ms:
    ;Sub Rotina para atraso de 500ms
    mov R1,#0fah ;1 x 250 | círculos de máquina do mnemônico call
    ;1 x 250 | move o valor 250 decimal para o registrador R1

aux1:
    mov R2,#0f9h ;1 x 250 | move o valor 249 decimal para o registrador R2
    nop ;1 x 250
    djns R2,aux2 ;2 x 250 x 249 = 622500 | decrementsa o R2 até chegar a zero
    djns R1,aux1 ;2 x 250 | decrementsa o R1 até chegar a zero
    ret ;2 | retorna para a função main
    ;-----Total = 500000 us -- 500 ms = 0,5 seg

LCD1:
    db 'Hello World!' ;Final do programa
    and $ ;Finaliza o bloco

```

## C: 77 linhas

```

#include<xc4000.h>
#include<16f877a.h>

void delay_ms(int);
void send(char);
void display(char);
void custom(void);
void init(void);

char custom_dchar[] = {0x00,0x04,0x0E,0x12,0x2E,0x32,0x00,
                      0x00,0x04,0x0E,0x12,0x2E,0x32,0x00,
                      0x00,0x1A,0x15,0x11,0x11,0x1A,0x04,0x00,
                      0x00,0x11,0x11,0x11,0x11,0x11,0x11,0x11,
                      0x00,0x10,0x10,0x10,0x10,0x10,0x10,0x10,
                      0x00,0x0E,0x15,0x12,0x12,0x12,0x12,0x00};

void delay_ms(int d)
{
    unsigned char i;
    for(i=d;i-->0)
        {
            for(j=250,Lcd,i--);
            for(j=250,Lcd,i--);
        }
}

void send(char c)
{
    Lcd;
    xrd;
    wrt;
    delay(5);
    wrt;
}

void display(char c)
{
    Lcd;
    xrd;
    wrt;
    delay(5);
    wrt;
}

void custom(void)
{
    int i;
    custom_dchar;
    for(j=0;j<8;j++)
        display(custom_dchar[j]),
        end(0x00);
}

void string(char *p)
{
    while(*p)
        {
            display(*p++);
        }
}

void init(void)
{
    cmd(0x30);
    cmd(0x30);
    cmd(0x31);
    cmd(0x30);
}

void main()
{
    init();
    custom();
    string("Hello World");
    while(1);
}

```

# Arduino (Wiring)

11  
linhas →  


The screenshot shows the Arduino IDE interface with the title bar "HelloWorld | Arduino 1.8.1". The main window displays the following C++ code for a LiquidCrystal display:

```
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
4
5 void setup() {
6     lcd.begin(16, 2);
7     lcd.print("Hello World!");
8 }
9
10 void loop() {
11 }
```

The code includes the `LiquidCrystal.h` header, initializes a `LiquidCrystal` object named `lcd`, sets the display to 16 columns and 2 rows, prints "Hello World!" to the screen, and loops forever.

## Assembly: 122 linhas

```
rs    equ    P1.5 ;Reg Select ligado em P1.7
rw    equ    P1.6 ;Read/Write ligado em P1.6
en    equ    P1.7 ;Enable ligado em P1.5
dat   equ    P2 ;Bit de dados em todo P2

org 0000h ;origem no endereço 00h de memória
delay500ms: ;aguarda 500ms para estabilizar
    .org 0000h
    .delay 500ms

inicio:
    ldd init ;Chama sub rotina de inicializa.c
    mov d0
    acall
    ajmp

lod_init:
    mov
    acall
    mov
    acall
    mov
    acall
    mov
    acall
    ret

config:
    clr
    clr rs ;limpa pino rs
    clr rw ;limpa pino rw
    acall wait
    setb en ;ativa enable
    acall
    mov dat,a ;carrega dados
    mov a0
    acall
    mov dat,a ;carrega dados Port P2
    mov a0
    acall
    mov en ;limpa pino en
    acall
    mov a0
    acall
    ret

send_lod:
    mov R0,#0d ;Sub Rotina para enviar dados ao LCD
    mov R0,#0d ;Move valor 0d para R0

send:
    mov
    inc
    move
    acall
    cje a, r0, neq ;verifica se os valores são diferentes
    ret

w_dat:
    clr
    setb
    clr
    acall
    setb
    acall
    wait
    ;aguarda 55us
    mov dat,a ;carrega dados
    acall
    mov a0
    acall
    clr en ;limpa enable
    acall
    wait
    ;aguarda 55us
    ret

wait:
    ;Sub Rotina para atraso de 55us

aux0:
    mov R5,#055d ;Carrega 55d em R5
    djnx R5,aux0 ;Decrementa R5 se igual a zero? N.o, desvia para aux
    ret

delay500ms:
    mov
    nop
    nop
    aux1:
    mov
    nop
    nop
    nop
    nop
    aux2:
    nop
    nop
    nop
    nop
    nop
    ; 1 x 250 = 249 = 62250
    nop
    ; 1 x 250 = 249 = 62250
    nop
    ; 1 x 250 = 249 = 62250
    nop
    djns R2,aux2 ; 2 x 250 x 249 = 124500 | decrementa o R2 ate chegar a zero
    djns R1,aux1 ; 2 x 250 | decrementa o R1 ate chegar a zero
    ret
    ; 2 | retorna para a função main
    ;-----|-----|
    ; Total = 500000 us -- 500 ms = 0,5 seg

LCD1:
    db 'Hello World! '
    .end ;Final do programa
```

Assembly  
code

Linking

Executable

## C: 77 linhas

```
#include<xc4000.h>
#include<lcd.h>
shift xw20
shift sw20

void delay();
void and();
void display (char);
void carim (char);
void string (char);
void init ();

unsigned char curm_dado[] = {0x00,0x04,0x00,0x00,0x15,0x15,0x00,0x00,
                             0x00,0x04,0x00,0x00,0x15,0x15,0x00,0x00,
                             0x00,0x04,0x00,0x00,0x15,0x15,0x00,0x00,
                             0x00,0x04,0x00,0x00,0x15,0x15,0x00,0x00};

void delay()
{
    unsigned char i;
    for (i=0;i<1000000;i++);
}

void and (char d)
{
    xw20;
    sw20;
}

void carim (char d)
{
    xw20;
    sw20;
}

void curm ()
{
    xw20;
    sw20;
    delay(5);
    and(d);
}

void display (char d)
{
    xw20;
    sw20;
    carim(d);
}

void string (char *p)
{
    while (*p)
    {
        display(*p);
        p++;
    }
}

void init ()
{
    and();
    curm();
    string("Hello World");
}

void main()
{
    init();
    curm();
    string("Hello World");
}
```

C code

Pre-process

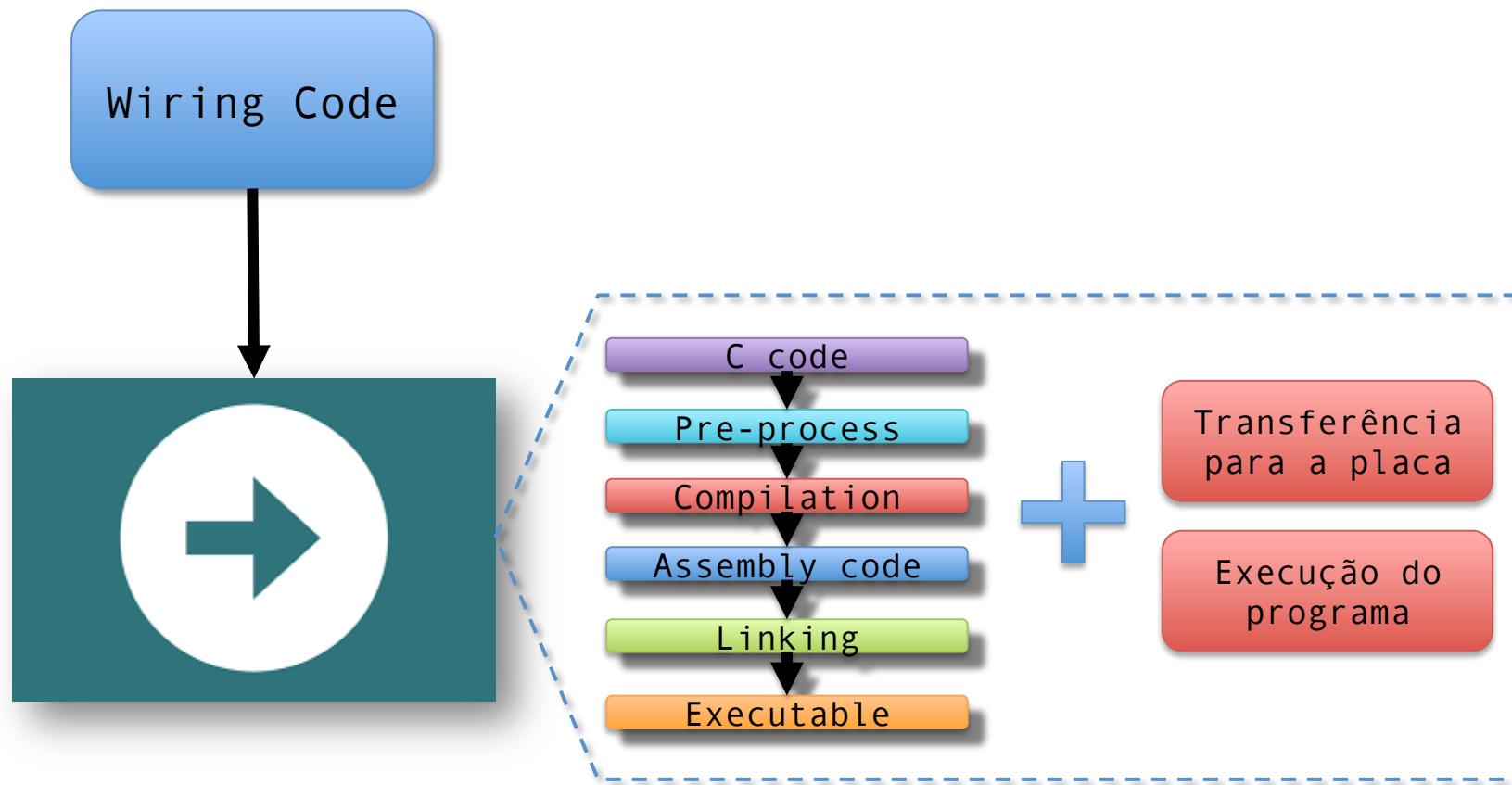
Compilation

Assembly  
code

Linking

Executable

# Arduino (ícone de “Upload”)



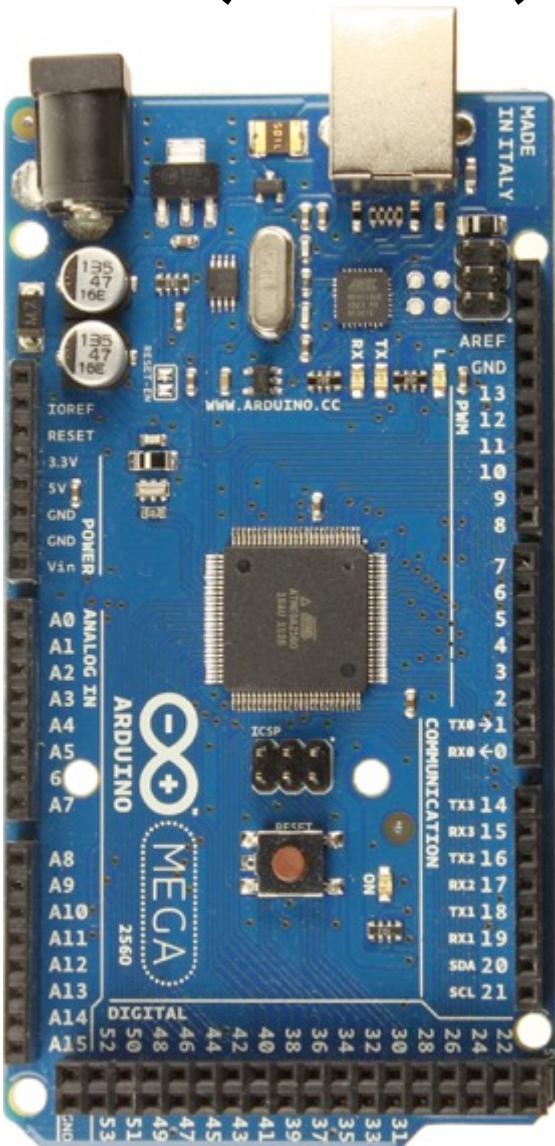
# **PLACAS**

# Uno R3 (2010)



- ATmega328P
- 8-bit
- 16MHz
- 14 E/Ss digitais
- 6 entradas analógicas
- 6 saídas analógicas
- 32kB Flash
- 2kB RAM

# MEGA (2010) - Mais pinos!



- ATmega2560
- 8-bit
- 16MHz
- 54 E/Ss digitais
- 16 entradas analógicas
- 15 saídas analógicas
- 256kB Flash
- 8kB RAM

# DUE (2012) - Mais processamento!



- SAM3X8E ARM Cortex-M3
- **32-bit**
- **84MHz**
- 54 E/Ss digitais
- 12 entradas analógicas
- 12 saídas analógicas
- 512kB Flash
- **96kB RAM**

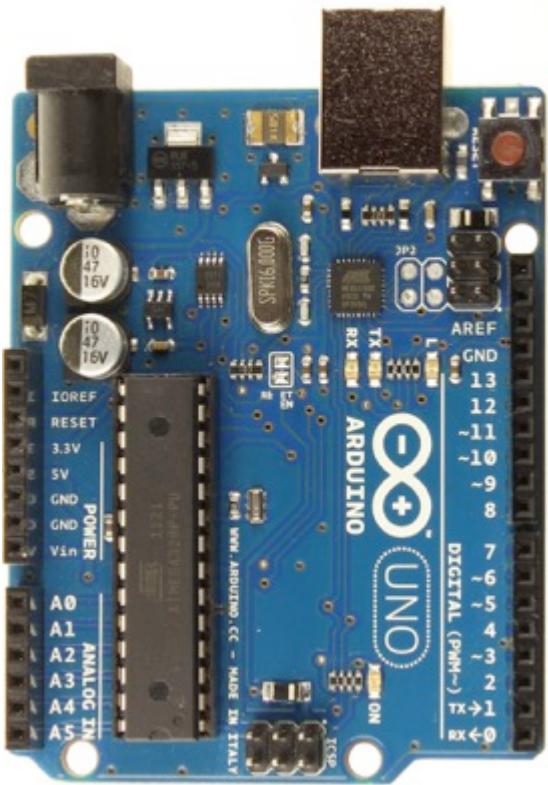
# MKR1000 (2016) - IoT



- SAMD21 ARM Cortex M0+
- 32-bit
- 48MHz
- 8 E/Ss digitais
- 7 entradas analógicas
- 12 saídas analógicas
- 256kB Flash
- 32kB RAM
- **Wi-Fi**
- **Li-Po battery**

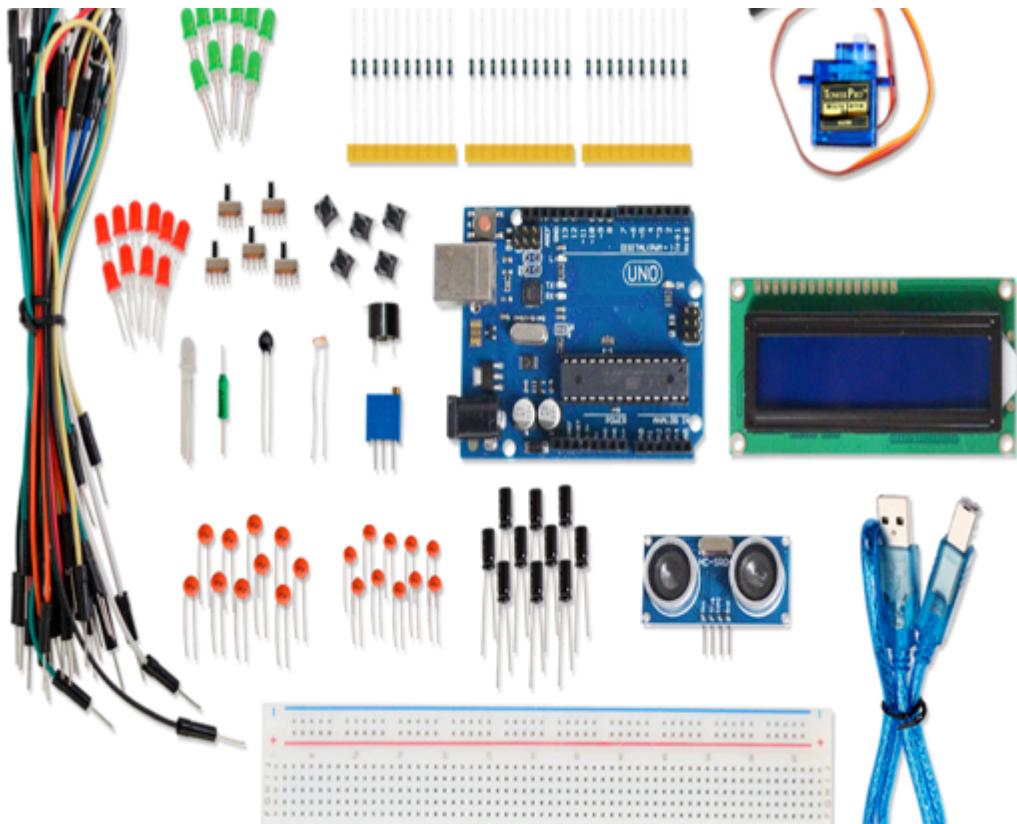
**COMO COMEÇAR?**

# 1) Comprar uma placa



- Webtronico: R\$ 49
- eBay: R\$ 20

# 1) Ou um kit



- FilipeFlop: R\$ 120
- eBay: R\$ 50

## 2) Instalar a IDE



The screenshot shows a web browser window titled "Arduino - Software". The URL in the address bar is <https://www.arduino.cc/en/Main/Software>. The page content is as follows:

**ARDUINO 1.8.5**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
[Get](#)

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

**HOURLY BUILDS** LAST UPDATE 8 May 2018 12:13:3 GMT

Download a **preview of the incoming release** with the most updated features and bugfixes.

**BETA BUILDS** **BETA**

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be

# 3) Rodar o “Blink”

The screenshot shows the Arduino IDE interface on a Mac OS X system. The title bar reads "Blink | Arduino 1.8.1". The left sidebar has icons for File, Sketch, Tools, Board, and Help. The main window displays the "Blink" sketch:

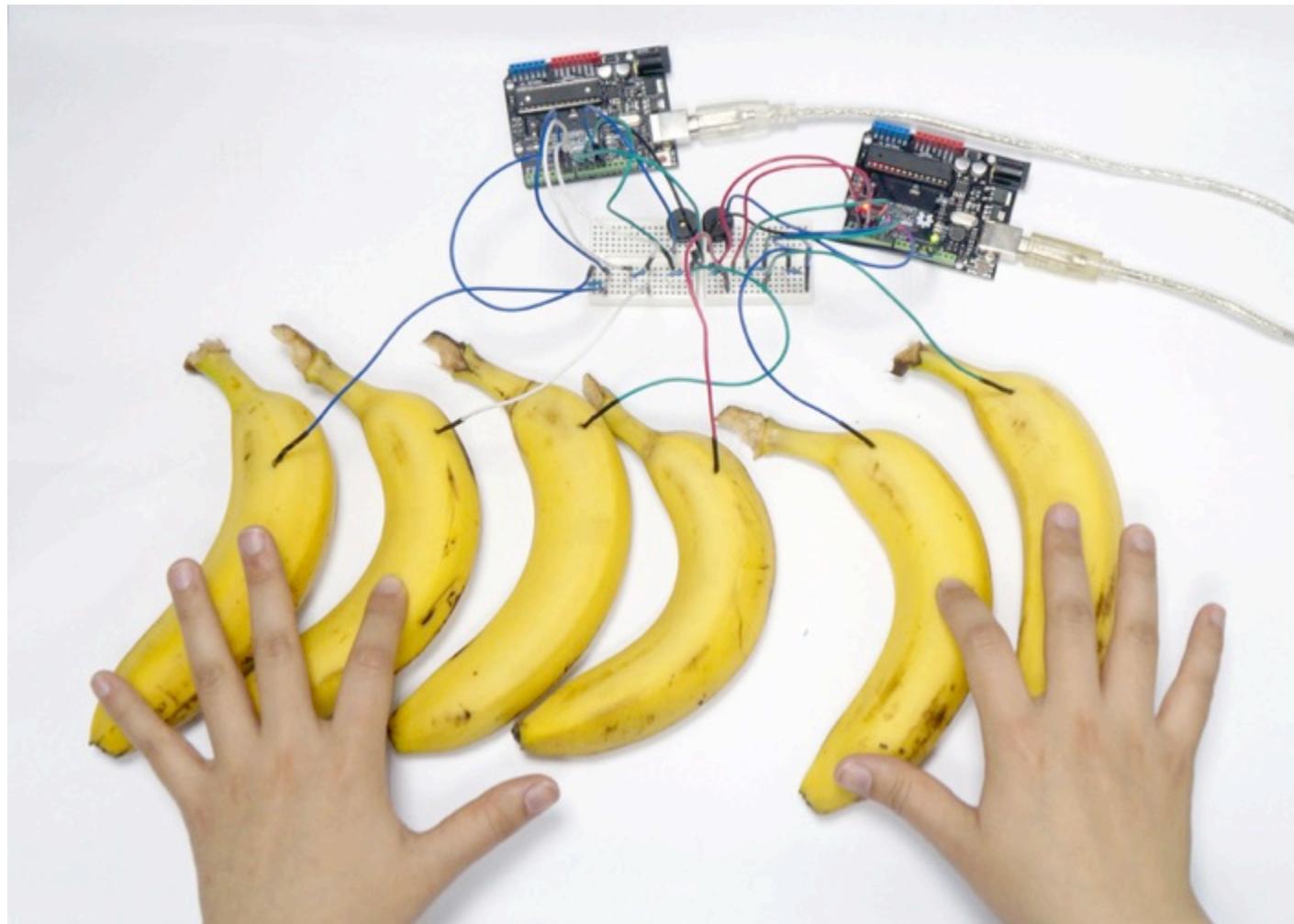
```
// the setup function runs once when you press reset or void setup() { // initialize digital pin LED_BUILTIN as an output. pinMode(LED_BUILTIN, OUTPUT); } // the loop function runs over and over again forever void loop() { digitalWrite(LED_BUILTIN, HIGH); // turn the LED on delay(1000); // wait for a second digitalWrite(LED_BUILTIN, LOW); // turn the LED off delay(1000); // wait for a second }
```

A context menu is open over the word "Serial" in the "readSerial()" function call. The menu items visible are "gReadSerial", "Minimum", "IRReadSerial", and "AnalogVoltage". At the bottom of the screen, the status bar shows "Arduino Nano, ATmega328 on /dev/cu.usbmodem1421".

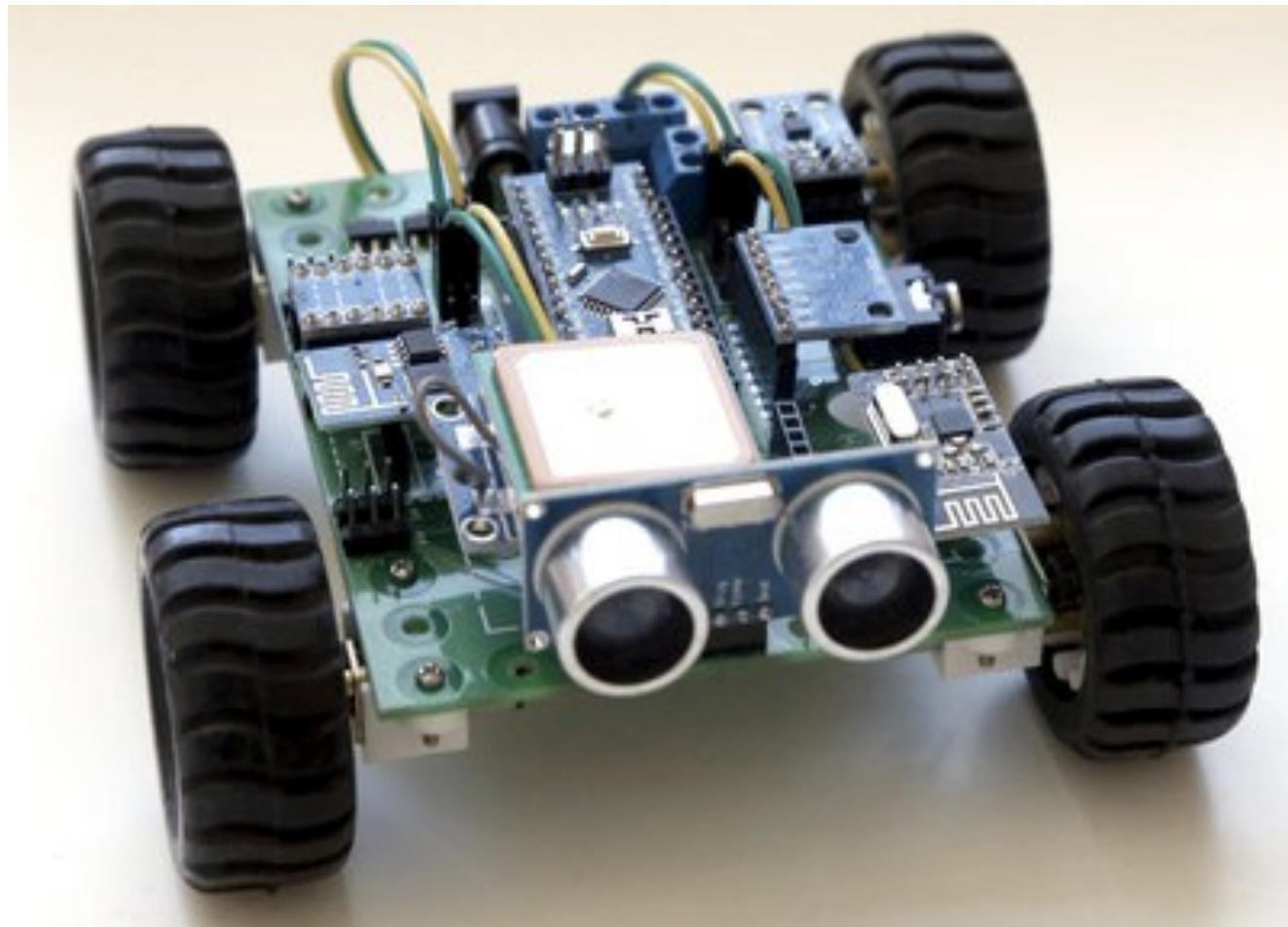
# 4) Estudar as funções básicas através de exemplos

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`
- `delay()`
- `analogWrite()`
- `analogRead()`
- Classe Serial

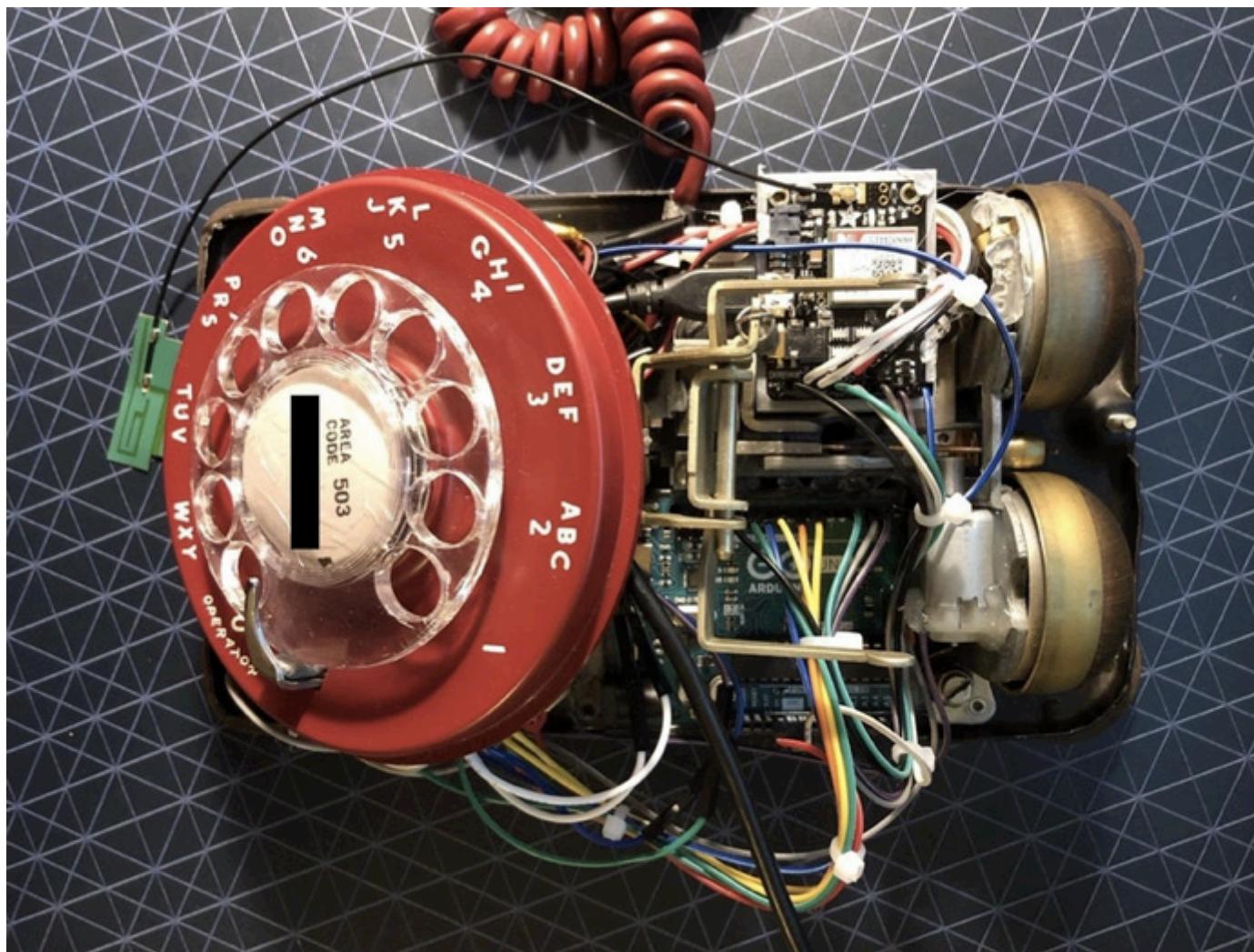
# 5) Construir seus próprios circuitos!



5) Construir seus próprios circuitos!



# 5) Construir seus próprios circuitos!



# **FONTES**

# Blog FilipeFlop

Arquivos Arduino - FilipeFlop X Felipe

<http://www.filipeflop.com/blog/category/arduino>

≡ Q FILIPEFLOP ⚙️ 🔍

Início > Arduino

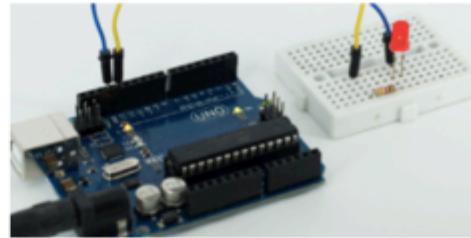


**WORKSHOP**  
Arduino para iniciantes

Workshop Arduino para iniciantes – Arduino Day 2018

Arduino • 8 de maio de 2018 • Pedro Bertoleti

A loja FilipeFlop, através do articulista Pedro Bertoleti, ministrará o workshop “Arduino para



Primeiros passos com Arduino

Arduino • 27 de abril de 2018 • Giovanni Bauermeister

Neste post iremos apresentar os primeiros passos com arduino, ou seja, tudo o que é necessário para

# Portal Embarcados

The screenshot shows a web browser window with the title bar "Arduino - Embarcados - Sua fo..." and a user profile "Felipe". The address bar contains the URL <https://www.embarcados.com.br/arduino>. The page content is a list of four articles under the heading "ARDUINO".

- Introdução ao Blynk App**  
Tiago Medicci Serrano - 08/05/2018
- Lendo e Escrevendo nos pinos do Arduino com linguagem C/C++**  
Pedro Igor Borçatti - 07/05/2018
- Medindo temperatura e umidade com Arduino UNO utilizando Si7021 e LCD**  
Giovanni Cerqueira - 03/05/2018
- Tinkercad e PQDB: Teclado matricial**  
Thiago Pereira do Prado - 02/05/2018

# Vida de Silício

A screenshot of a computer monitor displaying a website for "Vida de Silício". The browser window title is "Apostila Arduino Básico | Ebook". A user profile "Felipe" is visible in the top right corner. The URL in the address bar is <https://portal.vidadesilicio.com.br/apostila-arduino-basico>. The website header includes the logo "VIDA DE SILÍCIO", navigation links for "INÍCIO", "TUTORIAIS", "APOSTILA ARDUINO BÁSICO", and "SEJA UM AUTOR", and a search bar labeled "LOJA VIDA DE SILÍCIO". The main content area features a dark background with a large, semi-transparent watermark of the word "ARUINO". It contains the text "Aprenda a usar seu Arduino de forma simples e prática" and a paragraph explaining the series of free Arduino tutorial documents. A button at the bottom left says "BAIXAR APOSTILA ARDUINO BÁSICO VOL. 1". To the right, there's a photograph of a smartphone displaying the "Apostila Arduino Básico" app interface.

<https://portal.vidadesilicio.com.br/apostila-arduino-basico>

INÍCIO TUTORIAIS APOSTILA ARDUINO BÁSICO SEJA UM AUTOR LOJA VIDA DE SILÍCIO

Aprenda a usar seu Arduino de forma simples e prática

Conheça a série de apostilas gratuitas que explica os conceitos básicos sobre Arduino. As apostilas já foram baixadas milhares de vezes e são usadas por muitos professores em todo o Brasil e valem por um livro ou ebook Arduino.

BAIXAR APOSTILA ARDUINO BÁSICO VOL. 1

Apostila Arduino Básico

# Referências

- <https://www.arduino.cc>
- <https://en.wikipedia.org/wiki/Arduino>
- <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- <https://arduinohistory.github.io>
- <http://wiring.org.co>
- <https://processing.org>
- [https://www.ted.com/talks/massimo\\_banzi\\_how\\_arduino\\_is\\_open\\_sourcing\\_imagination](https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination)
- <https://vimeo.com/31389230>
- <https://www.webtronico.com>
- <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- <https://blog.arduino.cc/2018/04/23/1961-rotary-phone-gets-a-2018-cellular-upgrade>
- <http://www.filipeflop.com/blog/category/arduino>
- <https://www.embarcados.com.br/arduino>
- <https://portal.vidadesilicio.com.br/apostila-arduino-basico>
- <https://learn.adafruit.com/category/learn-arduino>

[github.com/fkuhne/aday18](https://github.com/fkuhne/aday18)

