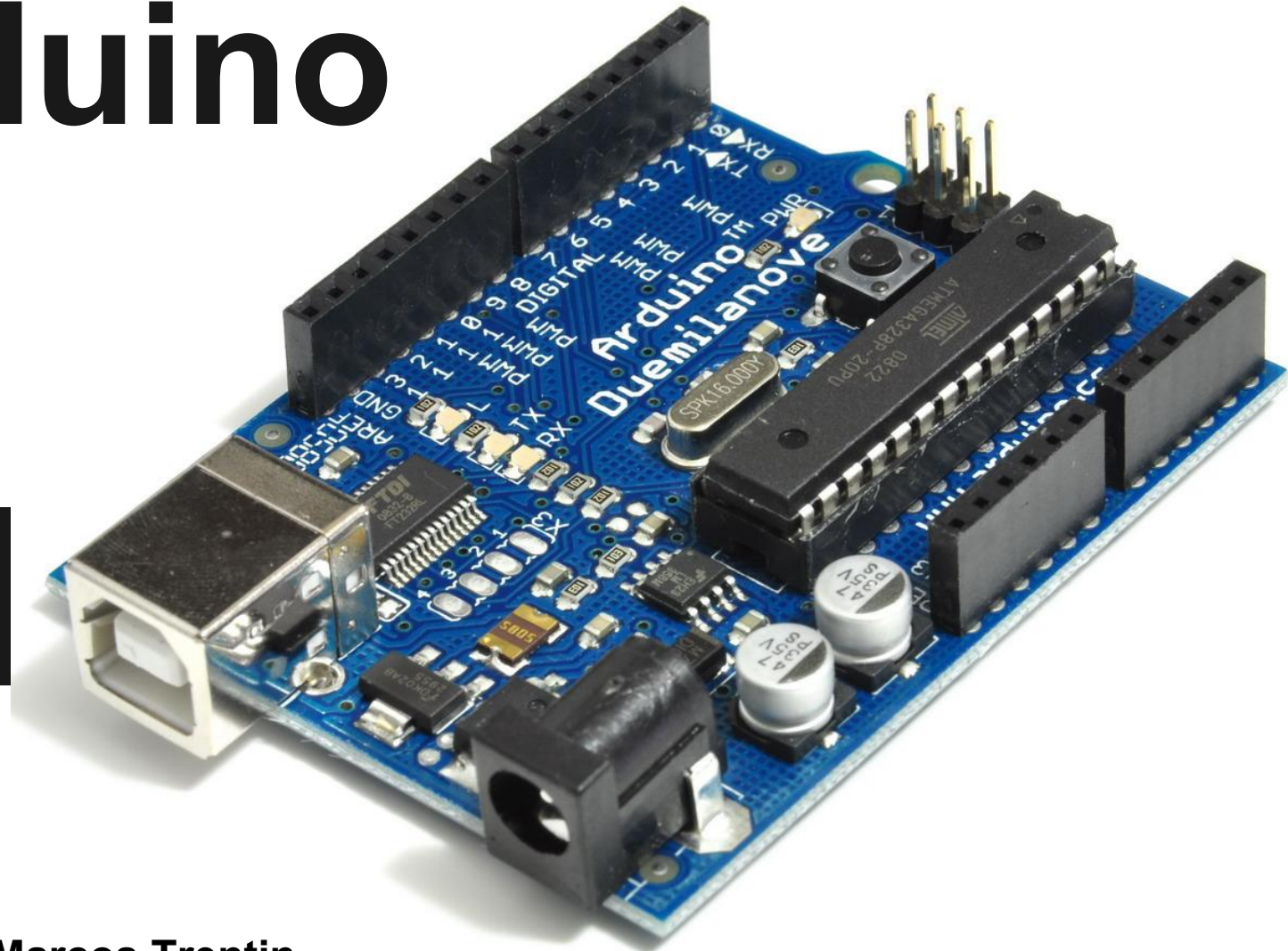


Robótica com Arduino



CDI

Professor: Paulo Marcos Trentin

Apresentações

Professor: Paulo Marcos Trentin

Email: paulo@paulotrentin.com.br

Site: www.paulotrentin.com.br

Graduando 9ª fase do curso de Ciências da
Computação - UNOESC Campus de Videira

Objetivos do Curso

Introduzir o aluno no mundo da programação e da eletrônica, com a ferramenta Arduino.

Objetivos específicos

- Conhecer componentes básicos de eletrônica
- Entender o que é e como funciona o Arduino
- Aprender programação básica em C com a plataforma Arduino
- Montar seu próprio Arduino em uma protoboard
- Desenvolver exercícios propostos envolvendo programação com Arduino em duplas
- Desenvolver um projeto individual (escolhido pelo aluno) usando os conhecimentos adquiridos (Projeto de Conclusão do Curso)

Plano de Ensino p.1

- Apresentação do curso
- Conectando leds ao Arduino, fazer um programa que faça os leds piscar de forma sequencial.
- Entendendo a IDE do Arduino (todos os botões e suas funções). Introdução à comunicação Serial entre o Arduino e o Computador
- Programação em C, o que é? O que é e por que precisamos de um código fonte? Estrutura de um programa em C para Arduino
- Resistores Pull-down e Pull-up, sua função e aplicação

Plano de Ensino p.2

- Matriz de contatos, vantagens e desvantagens. Quando devemos usá-las?
- Entendendo vetores em C. Por que e como usá-los?
- declaração de variáveis e constantes em C, estrutura da programação, tomada de decisão
- Números binários, conversão para hexadecimal, octal
- Comparadores lógicos && || NOT XOR
- Servomotores - Acionar Servo motor de acordo com iluminação
- Acionando cargas maiores com Arduino (motores e relés)

Plano de Ensino p.3

- Display LCD - como conectar, opções do mercado
- Controle de motores com transistores e de lâmpadas (60 W) com relé
- Display 7 segmentos
- Potenciômetro - funcionamento, uso da função analogRead
- Infrared - trabalhando com controles remotos
- Motor de passo - ligar motor de HD
- Entendendo o LDR (Light Dependent Resistor)
- Entendendo o chip 555
- Montando seu próprio Arduino na protoboard

Plano de Ensino p.4

- Trabalhando com HT12D HT12E e RF 433
- Extendendo a aplicação do HT12E/D
- O que é possível fazer com Arduino (levar meu TCC e apresentá-lo)
- Economizando fios com 74HC595 (pg 137),
Conhecendo o shift register
- Criando e entendendo o funcionamento de uma ponte H com CI específico
- Entendendo comunicação serial
- Circuitos flip flops, PWM
- E ainda mais...

O que é Robótica?

Robótica é um ramo da informática que engloba computadores, robôs e computação, que trata de sistemas compostos por partes mecânicas automáticas e controladas por circuitos integrados, tornando sistemas mecânicos motorizados, controlados manualmente ou automaticamente por circuitos eléctricos.

(Fonte: Wikipedia)

Onde é aplicada?

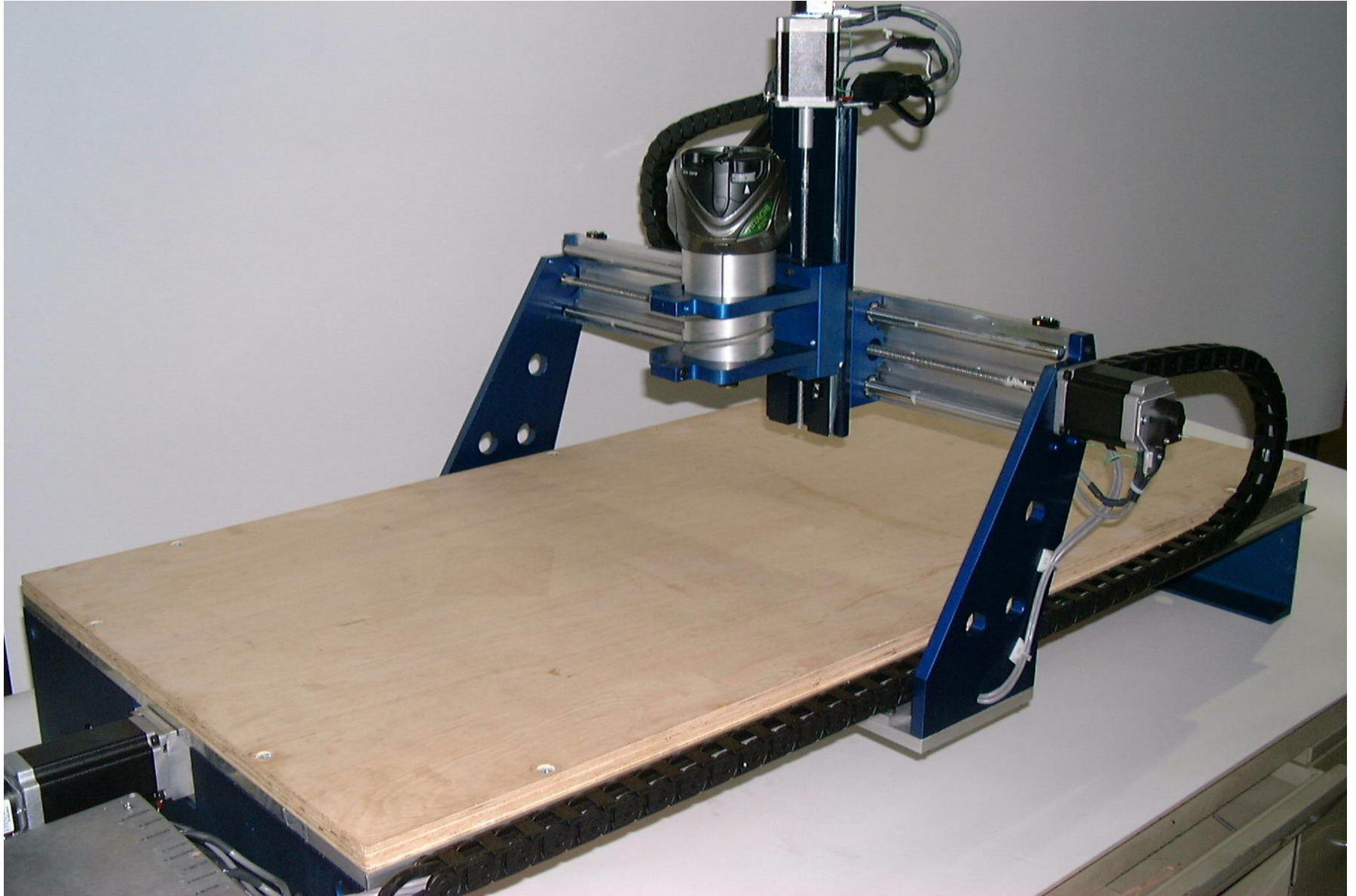
Construção e pintura de carros (motores FIRE da Fiat - Fully Integrated Robotised Engine)



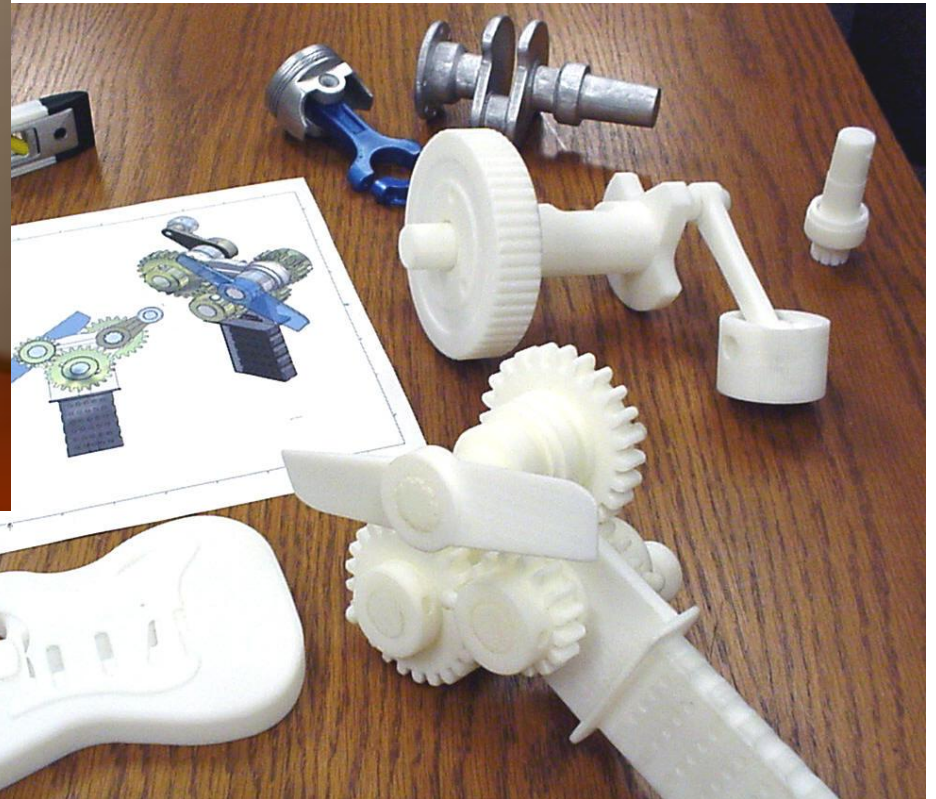
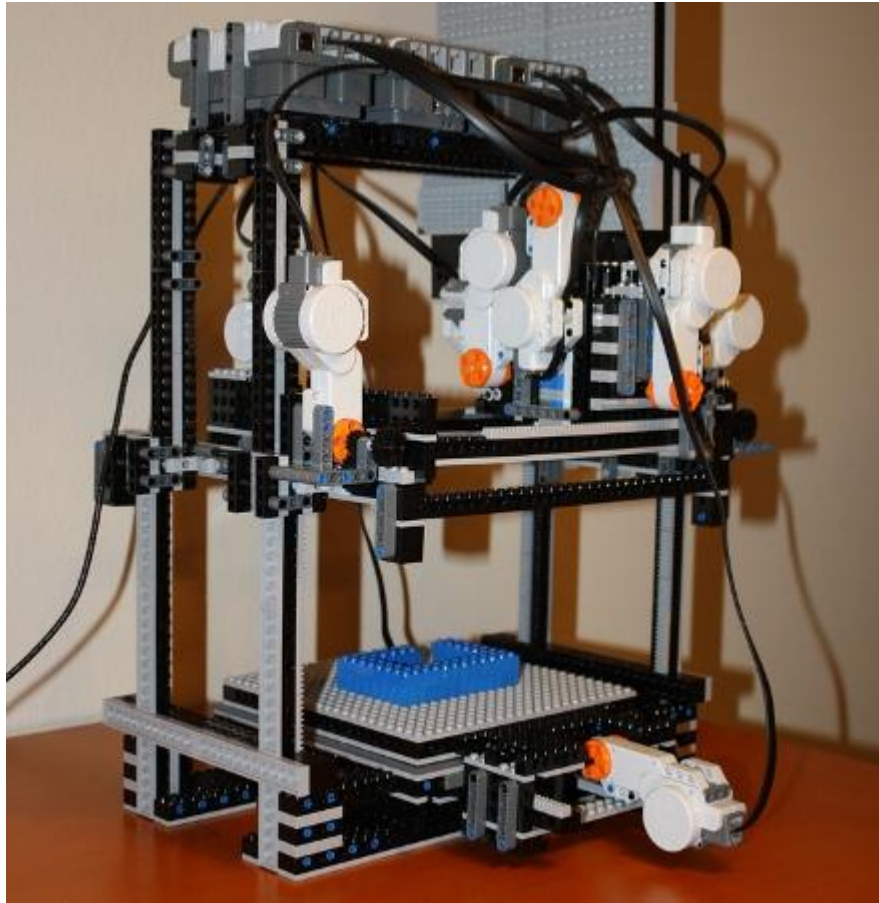
Montagem de Carros



Fresadora CNC



Impressora 3D



O que é Arduino?

"Arduino é uma plataforma de prototipagem eletrônica open-source baseada em hardware e softwares flexíveis e fáceis de usar. Foi projetado para ser usado por artistas, designers, hobbystas e qualquer um que tenha interesse em criar objetos e ambientes interativos."

fonte: www.arduino.cc

Afinal, o que é Arduino?

Em outras palavras: Arduino é uma plataforma de desenvolvimento open-source hardware e software, que com o uso de um microcontrolador programável, permite desenvolvermos programas que interajam com o ambiente e outros objetos.

Existem vários modelos de Arduino: <http://arduino.cc/it/main/boards>

<http://arduino.cc/en/Main/Hardware>

O que é um microcontrolador?

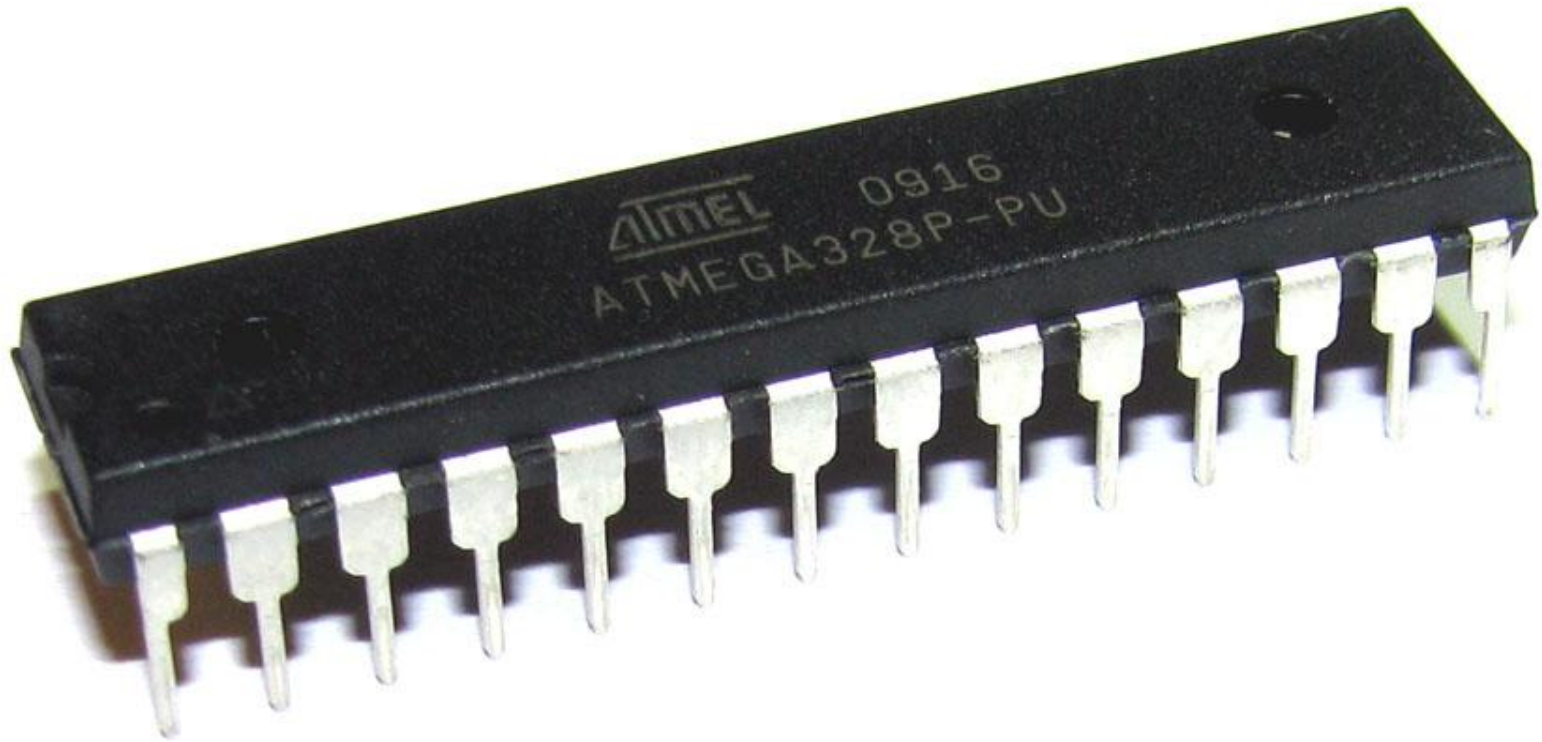
Um microcontrolador é um computador completo dentro de um circuito integrado. “Ele possui em um único encapsulamento, uma unidade central de processamento, memória de programa, memórias auxiliares, sistema de entrada/saída e vários periféricos que variam entre os modelos” (Fonte Microcontroladores AVR)

É “como um pequeno componente eletrônico, dotado de uma inteligência programável, utilizado no controle de processos lógicos” (fonte: Desbravando o PIC - Ampliado e Atualizado para PIC 16F628A)

Microcontrolador sem software

"Um microcontrolador sozinho é somente uma combinação de circuitos eletrônicos digitais montados em um mesmo encapsulamento que não tem nenhuma aplicação se não houver um software que instrue esse hardware no que tem que ser feito" (Fonte: Experimentos com o Arduino, Silveira, p.29)

Microcontrolador Atmega328



O que é um software?

Um software nada mais é que um conjunto de instruções para determinado hardware executar, são também conhecidos como programa de computador. “Estes programas de computador guiam o computador através de conjuntos ordenados de ações especificados por pessoas chamadas de programadores de computador” (Fonte: C++ como programar p. 52).

Exemplo de instruções de um software

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

Por que Arduino?

- Arduino é Fácil de começar. Com uma placa, um cabo USB, e o software correto, já pode-se rodar exemplos.
- É multiplataforma. Roda em qualquer S.O.
- Barato. O Duemilanome custa cerca de 80,00 reais
- É open source software e open source hardware

Um pouco da história

Surgiu em 2005 numa escola de artes interativas e design, localizada em Ivrea, na Itália com o intuito de se ter uma plataforma de baixo custo (o Basic Stamp custava cerca de 100 dólares) e de fácil aprendizagem.

Arduino - o Nome

O Arduino nasceu na cidade Ivrea na Itália.

Nesta mesma cidade nos séculos X e XI houve um outro Arduino (um nobre) que se auto proclamou rei de toda a Itália, obviamente a coisa não funcionou e, como era comum na época, ele foi morto pelos rivais. O fato é que em sua cidade natal ele ainda é muito lembrado

Enquanto viviam lá, os integrantes do time que criou o Arduino (o microcontrolador), depois do expediente iam tomar uma cerveja no Bar Arduino. Assim o nome do Arduino (o microcontrolador) é uma homenagem ao Arduino (o bar) que por sua vez era uma homenagem ao outro Arduino (o nobre).

O que é possível fazer com Arduino?

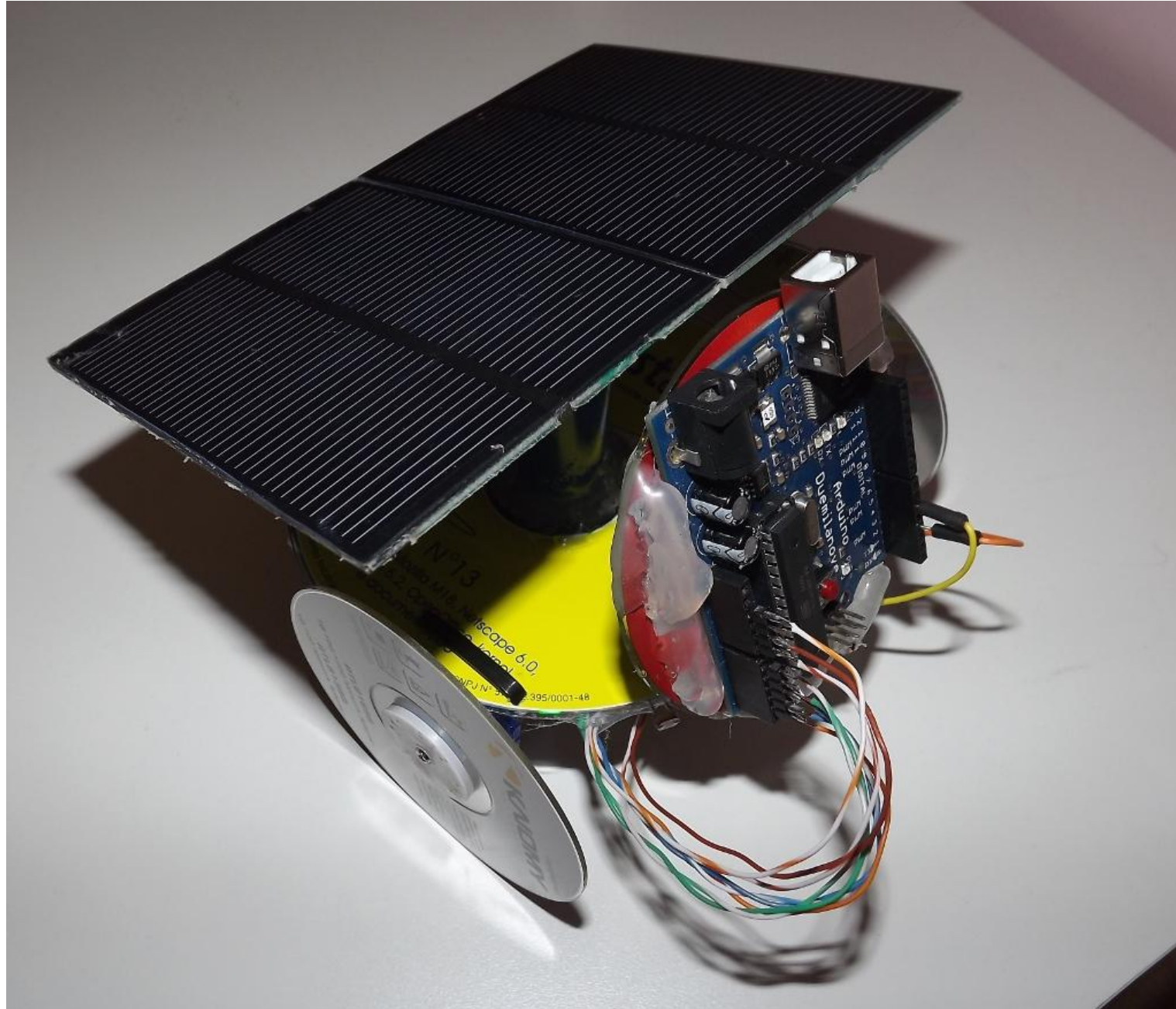
Ao infinito... e além!

<http://info.abril.com.br/noticias/blogs/zonalivre/2011/04/5-projetos-matadores-com-arduino/>

<http://hacknmod.com/hack/top-40-arduino-projects-of-the-web/>

Praticamente qualquer ideia envolvendo eletrônica e programação pode ser feita com Arduino.

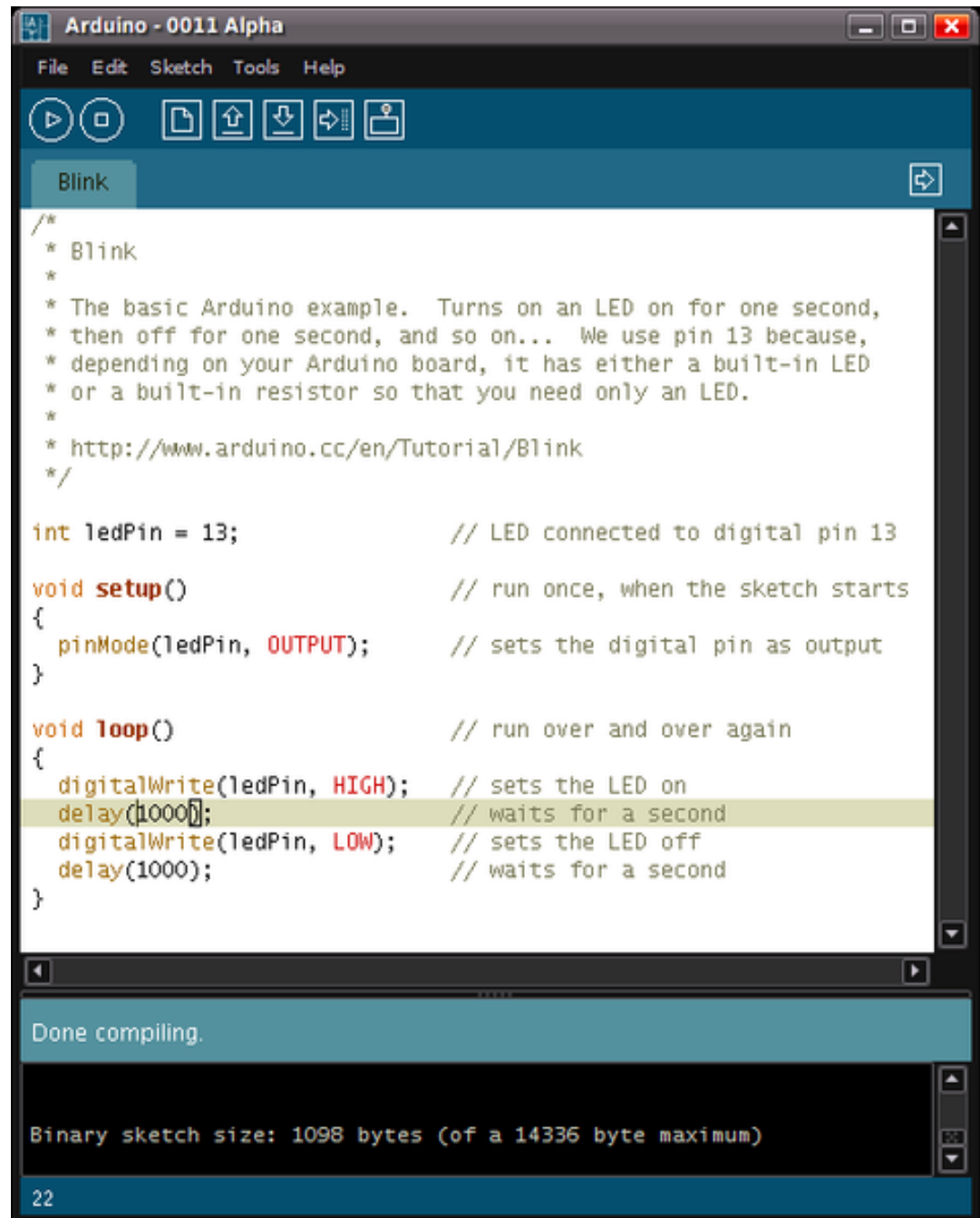
Robô seguidor de Luz



IDE (integrated development environment)

"É um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo." (Fonte: Wikipedia)

IDE do Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, uploading, and other functions. The main editor window displays the "Blink" sketch code. The code is as follows:

```
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()                // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Below the code editor, a status bar indicates "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)".

Exercício prático

- Conectar o Arduino no computador e fazer a instalação dos drivers
- Copiar a IDE para o computador e executá-la
- Criar um novo projeto e copiar o código fonte passado no quadro
- Fazer o Upload do código fonte para o Arduino e verificar seu funcionamento
- Alterar o código e verificar os resultados