

# VISÃO GERAL SOBRE MICROCONTROLADORES E PROTOTIPAGEM COM ARDUINO

## OVERVIEW ON MICROCONTROLLERS AND ARDUINO PROTOTYPING

Warley Monteiro Araujo,  
Maxwell Machado Cavalcante,  
Rogério Oliveira da Silva

### RESUMO

Neste artigo apresentaremos uma visão geral sobre os microcontroladores com ênfase na fabricante *ATmel*, embarcados nas diversas versões do *Arduino*, que impulsionou e popularizou a prototipagem e automação para fins acadêmicos, hobby, profissional, dentre outros.

**Palavras-chave:** *ATmel*; *ATmega*; *Arduino*; Automação, Sistema Embarcado, Robótica.

### ABSTRACT

*In this article we will present an overview of microcontrollers with emphasis on the manufacturer ATmel, embedded in the various versions of Arduino, which has promoted and popularized prototyping and automation for academic, hobby and professional purposes, among others.*

**Keywords:** *ATmel*; *ATmega*; *Arduino*; Automation, Embedded system, Robotic.

### INTRODUÇÃO

É comum encontrarmos diversos conteúdos desde livros, blog, fóruns e vídeos demonstrando as infinitas possibilidades de uso de uma interface *Arduino*. No entanto, devido a popularização do *Arduino*, desde então seu nome se sobrepôs ao seu microcontrolador, responsável por basicamente todas as suas funcionalidades, já que depois de programado, ele passa a ser completamente independente da sua interface.

Ainda hoje muitos confundem suas funções dentro da prototipagem e no decorrer deste artigo iremos desmistificar esse conceito, apresentando os diferentes tipos de microcontroladores, suas características e funcionamento, além de apresentar a *ATmel*, fabricante dos microcontroladores embarcados no *Arduino*, o quão poderoso essa linha de chip pode ser, apresentar as possibilidades de projetos que podem ser executados, e o conceitos e mercado da automação.

## SISTEMAS EMBARCADOS

Antes de falar de Arduino e microcontroladores precisamos definir o conceito de sistema embarcado e como eles são importantes para a sociedade moderna. Um sistema embarcado trata-se de um sistema computacional completamente independente, geralmente fabricado para um propósito único, e com o objetivo de executar tarefas específicas. Esses sistemas possuem seus próprios recursos computacionais como processamento, memória e portas de comunicação de entrada e saída.

Ao contrário de um computador pessoal, por exemplo, que depois de pronto pode ser programado e reprogramado quantas vezes for necessário, a atender os propósitos mais variados possíveis, os sistemas embarcados são desenvolvidos para executar tarefas predefinidas e geralmente de uso limitadas. Essa limitação permite o desenvolvimento de circuitos menores, otimizando recursos computacionais a necessidade de cada dispositivo, assim reduzindo significativamente o custo do produto.

A programação das instruções de um sistema embarcado é feita através de uma codificação normalmente chamada de *firmware*, é através dela que seus desenvolvedores definem as funcionalidades dos dispositivos. O *firmware* normalmente é armazenado em uma memória *ROM* ou memória *FLASH* diferentemente de um computador pessoal no qual o armazenamento é feito por um disco rígido. Existe um cuidado maior em desenvolver *softwares* para sistemas embarcados, pois esperam que esses dispositivos possam trabalhar ininterruptamente por longos períodos sem a incidência de erros, e se acontecer, ser possível a auto recuperação do mesmo. Isso porque uma vez gravado o *firmware*, a sua atualização seria inviável para o usuário, ou até mesmo impossível.

Com recursos computacionais limitados muitas vezes dispositivos embarcados não possuem nenhuma, ou quase nenhuma, interação com o utilizador, executando a tarefa a qual foi desenvolvida assim que ligado. Os que permitem uma pequena participação do usuário utilizam de botões, luzes e pequenos visores com poucos caracteres para exibir informações ou ajustar suas funcionalidades. Mas com o aumento da capacidade computacional dos microcontroladores com o passar do

tempo, permitiu uma maior interação do usuário, essa interação pode ser feita através de interfaces como teclados, joysticks e até mesmo por telas gráficas e/ou táteis. Também alguns dispositivos têm a capacidade de se comunicar com redes computacionais, desde redes internas, ou até mesmo com grandes redes como a internet.

Como exemplos de sistemas embarcados podemos citar: Controle remotos, eletrodomésticos, player de música, celulares entre outros dispositivos que executam tarefas independentes de outros dispositivos.

## **MICROCONTROLADORES**

No início do desenvolvimento de circuitos eletrônicos, esses eram produzidos a partir de componentes individuais como resistores, capacitores, diodos transistores entre outros. Porém, o uso desses componentes tornavam circuitos mais complexos muito grandes, o que tornavam os projetos eletrônicos caros e até mesmo poderiam inviabilizar alguns projetos.

Com o surgimento dos circuitos integrados isso mudou. Circuito integrado é um circuito eletrônico miniaturizado produzido com semicondutores que substitui os componentes antes mencionados, encapsulado sobre o substrato de um material muito fino, normalmente o silício. Com isso foi possível reduzir significativamente o tamanho dos equipamentos eletrônicos.

Mas normalmente os circuitos integrados são desenvolvidos para executar funções únicas com um objetivo bem específico, como portas lógicas do tipo *AND*, *OR* e *NOT*, ou com funções mais elaboradas como temporizadores, decodificadores de sinal e uma infinidade de tarefas. Mas as capacidades dos circuitos integrados não param por aí, existem circuitos integrados muito poderosos com a capacidade de executar uma infinidade de tarefas em um único chip, como por exemplo os microprocessadores e microcontroladores.

## **DIFERENÇAS ENTRE MICROPROCESSADORES E MICROCONTROLADORES**

Microprocessadores ou apenas processador, é um circuito integrado dedicado a realização de cálculos execução de instruções e tomadas de decisões, considerado o cérebro de um sistema computacional. Mas para ele funcionar ele precisa de alguns componentes como memória *RAM* para armazenamento

temporário das informações que estão sendo executadas, um dispositivo para armazenar o sistema que vai gerenciar o seu funcionamento, dispositivos de entrada e saída, entre outros circuitos que o auxiliam para promover a interação com o usuário.

Já o microcontrolador por outro lado é considerado um computador completo em miniatura desenvolvido em um único circuito integrado. Dentro de um microcontrolador temos uma unidade de processamento e os componentes necessários para que ele possa executar tarefas de forma autônoma, como memórias de leitura, escrita de dados e armazenamento da firmware, conversores de sinais analógico/digital e digital/analógico, e portas programáveis de entrada e saída para diversas funcionalidade como controlar outros dispositivos, fornecer uma interação com o utilizador entre outras.

## **HISTÓRIA DOS MICROCONTROLADORES**

Em 1971 em paralelo ao primeiro microprocessador lançado pela *Intel*, surgiu o primeiro microcontrolador, o *TMS1000* idealizado por *Gary Boone* e *Michael Cochram*, dois engenheiros da Texas Instruments, usado para atender a demanda de suas calculadoras até 1974, quando foi ofertado para a indústria eletrônica. O *TMS1000* continha um microprocessador de 4 *bits* e integrava memórias de leitura e escrita, uma apenas de leitura, um relógio e portas de entrada e saída de informações, tudo isso em um único chip. O *TMS1000* foi aprimorado com o passar dos anos e estava disponível em configurações de memórias variadas. Até o término da sua fabricação, estima-se que foram vendidos mais de 100 milhões de unidades por todo o planeta.

Já em 1974, ostentando o título de ser a primeira fabricante de microprocessadores, a *Intel* resolveu entrar na briga produzindo seu primeiro microcontrolador, o *8048* embarcado em teclados e videogames da época. Com o sucesso do *8048*, em 1980 a *Intel* lança o microcontrolador *8051* com 4 *Kilobytes* de memória programável e 128 *Bytes* de memória de dados, com um processador de 8 *bits*. Considerado o microcontrolador mais bem sucedido e um dos projetos eletrônicos mais longos da história, sendo fabricado até hoje.

Os microcontroladores até então dependiam de muito conhecimento técnico e ferramentas específicas proprietárias para efetuar sua programação, além de só

poder ser programado uma única vez, o que inviabilizavam o uso desses para fins doméstico e pequenos projetos. Mas tudo mudou a partir dos anos de 1990 quando surgiram os primeiros microcontroladores com memórias eletricamente programáveis, as *EEPROM* (*Electrically-Erasable Programmable Read-Only Memory*), com isso passou a ser possível reprogramar um microcontrolador para usá-los em outros projetos ou até mesmo ter seu firmware atualizado sempre que necessário sem a necessidade de removê-lo do circuito eletrônico.

Atualmente os microcontroladores com tecnologias mais modernas usam memórias *FLASH* baseadas nas memórias *EEPROM*, porém com uma maior facilidade na sua programação podendo ser apagados e escritos com velocidades muito altas.

## **MICROCONTROLADORES ATMEL**

Os microcontroladores Atmega da família AVR composta pelos microcontroladores *TinyAVR*, *MegaAVR*, *XmegaAVR* e *Atmel At94k* foram desenvolvidos inicialmente por dois estudantes da Universidade Norueguesa de Tecnologia e Ciência, chamados *Alf-Egil Bogen* e *Vegard Wollan* que durante sua formação acadêmica desenvolveram o microcontrolador de arquitetura *RISC* com memória de programação *Flash*, confiantes com o designer inovador e competitivo por se tratar de um microcontrolador de baixo custo, alto desempenho, e baixíssimo consumo de energia, os criadores do AVR dedicaram alguns anos no aprimoramento do projeto, adicionando periféricos à arquitetura do microcontrolador.

Com o projeto pronto para ser fabricado industrialmente, *Alf-Egil Bogen* e *Vegard Wollan*, como muitos outros inventores precisavam agora de um parceiro para produzir o microcontrolador inovador. Partiram da Noruega para San José na Califórnia, para tentarem uma parceria de produção com a empresa *Atmel*, que atualmente pertence a empresa Norte Americana *Microchip Technology Inc.* Felizmente a negociação foi bem sucedida e *Atmel* produziu o primeiro microcontrolador AVR de 8 bits, hoje a família AVR evoluiu para os novos modelos, abaixo temos a descrição dos grupos de microcontroladores.

TinyAVR	MegaAVR	XMEGA	Atmel At94k
Série ATtiny	Série ATmega	Série ATxmega	Série FPLIC (Circuito Integrado com Campo de Nível de Sistema Programável)
<ul style="list-style-type: none"> <li>• Memória do programa de 0.5 a 8 kB</li> <li>• 6 ou 32 pinos</li> <li>• Conjunto limitado de periféricos</li> </ul>	<ul style="list-style-type: none"> <li>• Memória do programa de 4 a 256 kB</li> <li>• 28 ou 100 pinos</li> <li>• Conjunto de instruções estendidas (múltiplas instruções e instruções para gerenciamento de programas com grandes memórias)</li> <li>• Conjunto extensivo de periféricos</li> </ul>	<ul style="list-style-type: none"> <li>• Memória do programa de 16 a 384 kB</li> <li>• 44, 64 ou 100 pinos (A4, A3, A1)</li> <li>• Maior desempenho, tal como DMA, "Evento de Sistema"</li> <li>• Suporte a <a href="#">criptografia</a></li> <li>• Conjunto extensivo de periféricos com DACs</li> </ul>	<ul style="list-style-type: none"> <li>• Núcleo baseado em <a href="#">FPGA</a></li> <li>• Utiliza SRAM para o código do programa, diferente dos restantes AVR's</li> <li>• Em parte, devido a diferença de velocidade relativa entre SRAM e Flash, este núcleo pode executar velocidades acima de 50 megahertz</li> </ul>

## ARDUINO

Em 2005, na Itália, iniciou-se um projeto chamado *Arduino*, com o objetivo de criar um circuito de prototipagem de baixo custo e que tivesse uma interface simples para ser aplicada na educação em escolas públicas. Essa interface foi projetada para fornecer uma comunicação completa e fácil com o microcontrolador ATmel AVR.

*Arduino* surgiu como uma solução de desenvolvimento e prototipagem, com possibilidades que vão desde um simples sistema de acender e apagar de uma luz, à projetos sofisticados e profissionais, que dependem apenas do conhecimento que o usuário de *Arduino* tem de programação e eletrônica.

Programar para *Arduino* não é uma tarefa difícil, sua programação é muito parecida com a linguagem C/C++, além disso a plataforma *Arduino* disponibiliza uma *IDE* do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado que é uma ferramenta de desenvolvimento escrita na

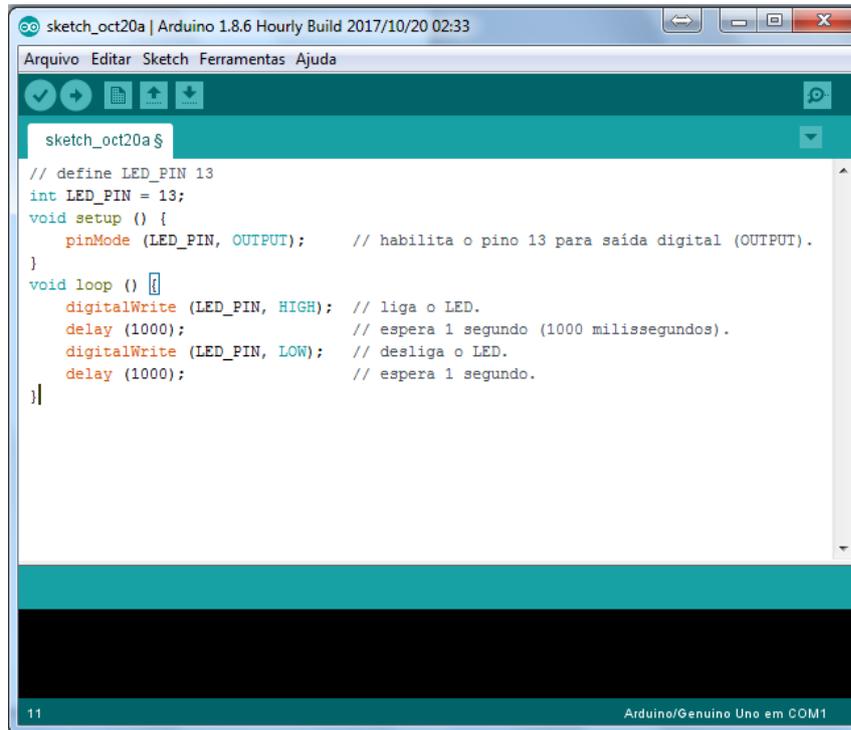
linguagem Java e implementa a linguagem baseada em *Processing/Wring*, que auxilia os usuários da ferramenta desenvolver e programar suas aplicações. Com um programa escrito para *Arduino* o usuário pode facilmente criar comandos de entrada e saída com poucos comandos, tendo que implementar apenas dois comando básicos para executar um programa funcional, os comandos *setup()* – Inserida no início, serve para configurar, inicializar e definir as portas digitais e lógicas, e por fim o comando *loop()* – Que é uma chamada para repetir um bloco de instruções ou esperar até que seja desligado.

Os programas escritos usando uma *IDE Arduino* são chamados de *Sketch* ou esboço, escrito com a ajuda de um editor de texto e salvo com a extensão *.ino*. Todo programa para *Arduino* tem uma estrutura básica para que o programa execute corretamente, o código abaixo demonstra um *Sketch* com um programa básico de exemplo que faz com que um Led pisque na porta digital 13 da placa *Arduino*.

```
// define LED_PIN 13
int LED_PIN = 13;
void setup () {
  pinMode (LED_PIN, OUTPUT); // habilita o pino 13 para saída digital
  (OUTPUT).
}
void loop () {
  digitalWrite (LED_PIN, HIGH); // liga o LED.
  delay (1000); // espera 1 segundo (1000 milissegundos).
  digitalWrite (LED_PIN, LOW); // desliga o LED.
  delay (1000); // espera 1 segundo.
}
```

A *IDE* de desenvolvimento *Arduino* é *open-source*, atualmente está na versão 1.8.6 disponível para *download* no site oficial do *Arduino* no endereço: site [www.arduino.cc](http://www.arduino.cc), após o *download* da ferramenta o usuário será capaz de criar e executar seus próprios *Sketch* e contar ainda com vários exemplos já disponíveis na própria *IDE*.

Figura 1 - Screenshot IDE Arduino 1.8.6.



```
sketch_oct20a $
// define LED_PIN 13
int LED_PIN = 13;
void setup () {
  pinMode (LED_PIN, OUTPUT); // habilita o pino 13 para saída digital (OUTPUT).
}
void loop () {
  digitalWrite (LED_PIN, HIGH); // liga o LED.
  delay (1000); // espera 1 segundo (1000 milissegundos).
  digitalWrite (LED_PIN, LOW); // desliga o LED.
  delay (1000); // espera 1 segundo.
}
```

O *software* do *Arduino* também possui uma saída serial, que permite o monitoramento via serial de instruções como escrever na tela ou receber dados via teclado fornecidos pelo usuário, o que facilita no desenvolvimento onde o desenvolvedor precise interagir com o *software*, o circuito serial do *hardware* do *Arduino* também possui um *led* integrado na placa que pisca quando o circuito recebe dados em sua porta serial, facilitando visualmente o tráfego de dados entre o *software* e o *hardware*.

O *Arduino* tem várias versões de *Hardware* para atender diversos tipos de projetos, desde projetos para serem acomodados em pequenos espaços como *Gadgets*, *Vestimentas*, *Robôs*, *Automação*, *Veículos* etc. Cada um dos *Arduinos* disponíveis no mercado possuem suas próprias características, cada um desenvolvido com uma gama de possibilidades de utilização.

Um dos *Arduinos* mais conhecidos e utilizados por alunos, hobbistas e entusiastas são as placas *Arduino Uno*, *Arduino Mega*, *Diecimila* e *Arduino Nano* e outros, a facilidade de se usar essas placas de prototipagem e o preço acessível fez com que essas placas ganhassem muito espaço no mundo da prototipação e a preferência por usar esses modelos de *Arduino*.

## **O CRESCENTE MERCADO DA AUTOMAÇÃO**

Automação é um sistema que opera de forma autônoma um processo sem ou com pouca interferência do ser humano. Diante dessa definição podemos afirmar que sempre foi da natureza humana produzir ferramentas e equipamentos que facilitem o seu trabalho nas mais variadas tarefas.

Com a eletrônica e a computação o processo de automação teve um salto gigantesco, que permitiu que cada vez mais pessoas pudessem ter acesso equipamentos que automatiza suas rotinas. E isso criou uma certa dependência e cada dia que passa, um desejo maior de tornar suas vidas cada vez automáticas.

Aproveitando essa dependência, surgiu no mercado tecnologias de automação voltada para o “Faça você mesmo”, com isso uma pessoa que tem pouco conhecimento em eletrônica, ou quase nenhum, possa fazer protótipos de automação e aprender a construir desde equipamentos para facilitar sua vida, até robôs.

## **CONCLUSÃO**

Vimos que os microcontroladores foram responsáveis pelo grande crescimento da tecnologia nos últimos quase 50 anos, revolucionando o mercado de sistemas embarcados, eletrônica e principalmente da computação.

Mas não só isso, com a evolução desses microcontroladores, e o surgimento de interfaces de prototipagens, como o *Arduino*, permitiu uma popularização da prototipagem eletrônica e automação nunca visto antes, permitindo os mais variados projetos voltados tanto para educação, como para soluções pessoais e *Hobby*.

## REFERÊNCIAS

MONK, Simon. **30 projetos com arduino**. 2. Porto Alegre, RS: Bookman, 2014.

MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: AMGH, 2013.

SOUZA, Vitor Amadeu. **Programação em BASIC para o microcontrolador PIC18F1220**. Editora Érica, 2006.

PT Computador, **A História do microcontrolador**. Disponível em: <<http://ptcomputador.com/Ferragens/computer-drives-storage/47399.html>>. Acesso em: 14 out. 2017.

Arduino, **Arduino**. Disponível em: <<https://www.arduino.cc>>. Acesso em: 30 ago. 2017.

Esse conhece, **Arduino, plataforma de prototipagem eletrônica**. Disponível em: <<http://esseconhece.com.br/arduino-plataforma-de-prototipagem-eletronica/>>. Acesso em: 19 out. 2017.

Eletrônica Progressiva, **Microcontroladores, O que são, para que servem e onde são usados**. Disponível em: <<http://www.eletronicaprogressiva.net/2014/08/Microcontroladores-O-que-sao-Para-que-servem-Onde-sao-usados.html>>. Acesso em: 14 out. 2017.

Robolivre, **Microcontroladores**. Disponível em: <<http://www.roboliv.re/conteudo/microcontroladores>>. Acesso em: 14 out. 2017.

Newton C. Braga, **O básico sobre os Microcontroladores**. Disponível em: <<http://www.newtoncbraga.com.br/index.php/electronica/52-artigos-diversos/13263-o-basico-sobre-os-microcontroladores-parte-1-mic139>>. Acesso em: 14 out. 2017.

Wikipedia, **ATmel AVR**. Disponível em: <[https://en.wikipedia.org/wiki/Atmel\\_AVR](https://en.wikipedia.org/wiki/Atmel_AVR)>. Acesso em: 15 out. 2017.

Wikipedia, **Arduino**. Disponível em: <<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em: 29 ago. 2017.

Wikipedia, **Sistema Embarcado**. Disponível em: <[https://pt.wikipedia.org/wiki/Sistema\\_embarcado](https://pt.wikipedia.org/wiki/Sistema_embarcado)>. Acesso em: 30 ago. 2017.

Programar Fácil, **La vida antes de Arduino, microcontroladores antiguos**. Disponível em: <<https://programarfácil.com/blog/la-vida-antes-de-arduino>>. Acesso em: 14 out. 2017.

Wikipedia, **Intel Corporation**. Disponível em: <[https://pt.wikipedia.org/wiki/Intel\\_Corporation](https://pt.wikipedia.org/wiki/Intel_Corporation)>. Acesso em: 15 out. 2017.

Wikipedia, **Memória Flash** . Disponível em:  
<[https://pt.wikipedia.org/wiki/Mem%C3%B3ria\\_flash](https://pt.wikipedia.org/wiki/Mem%C3%B3ria_flash)>. Acesso em: 15 out. 2017.

Wikipedia, **EEPROM**. Disponível em: <<https://pt.wikipedia.org/wiki/EEPROM>>.  
Acesso em: 15 out. 2017.

Wikipedia, **Circuito Integrado**. Disponível em:  
<[https://pt.wikipedia.org/wiki/Circuito\\_integrado](https://pt.wikipedia.org/wiki/Circuito_integrado)>. Acesso em: 14 out. 2017.

Wikipedia, **Intel 8048**. Disponível em: <[https://pt.wikipedia.org/wiki/Intel\\_8048](https://pt.wikipedia.org/wiki/Intel_8048)>.  
Acesso em: 15 out. 2017.

Wikipedia, **Intel 8051**. Disponível em: <[https://pt.wikipedia.org/wiki/Intel\\_8051](https://pt.wikipedia.org/wiki/Intel_8051)>.  
Acesso em: 15 out. 2017.

Wikipedia, **Microcontrolador**. Disponível em:  
<<https://pt.wikipedia.org/wiki/Microcontrolador>>. Acesso em: 14 out. 2017.