

SISTEMAS EMBARCADOS: O ARDUINO E A ELETRÔNICA

Natan Ogliari

Resumo

Este artigo tem como objetivo a história e a influência no aprendizado da eletrônica em plataforma de desenvolvimento, no caso o Arduino. Em outras palavras, as mudanças no aprendizado em torno da eletrônica com o Arduino e suas influências no uso dos sistemas embarcados, é conhecido que para o desenvolvimento de um projeto consiste em dois subprojetos em paralelo ao do software e do hardware. Em suma fica complexo o teste dos dois subprojetos juntos sem que tenha certeza do funcionamento correto de ambos. Em muitos casos, uma plataforma de prototipagem viabiliza o desenvolvimento de um produto final. Podemos afirmar que, após o ano de 2005, o movimento maker começou a ser difundido pelo mundo, tendo em vista que, a partir daquele ano, a prototipagem para projetos pequenos, tornou-se mais fácil e barata, pois não necessitavam mais do projeto do hardware, mas apenas do software. Com isso, a eletrônica começa a caminhar nos projetos de garagem (desenvolvedores maker) com maior força e, com isso, surgem a necessidade dos shields. Shield é um hardware pré-projetado para aplicações específicas em paralelo com o Arduino. Com isso facilita-se o desenvolvimento de um projeto.

Palavras-chaves: Arduino. Sistemas Embarcados. Plataforma de Desenvolvimento.

1 INTRODUÇÃO

Atualmente, temos uma inserção grande dos sistemas embarcados em nosso cotidiano e, como isso, pode mudar o cotidiano das pessoas, desde um simples acionamento de lâmpadas por controle remoto, até um sistema

mais complexo como, por exemplo, controle total de residências por meio da internet das coisas.

É conclusivo que as plataformas de desenvolvimento, neste caso o Arduino veio a prevalecer a eletrônica em nossas vidas. Atualmente, não se faz necessário um conhecimento complexo para que um hobista (projeto de garagem) possa automatizar sua residência por meio da plataforma Arduino e por consequência acaba-se estudando um pouco sobre a área de eletrônica. Todavia, uma área que nos traz cada vez mais conforto e segurança. No final deste artigo de revisão bibliográfica, espera-se que se tenha uma visão geral da plataforma Arduino como aprendizado para a eletrônica e pequenos projetos.

2 DESENVOLVIMENTO

A eletrônica vem desde a antiguidade nos proporcionando maior conforto e desenvolvimento tecnológico. Nos projetos da atualidade e do passado o problema maior é o hardware. Geralmente o hardware é responsável pela maior parcela dos custos finais dos produtos eletrônicos. Como solução deste problema, desenvolvem-se plataformas de desenvolvimento genéricas para aplicações. Algumas destas com uma vasta capacidade de processamento e outras com baixa. Entretanto, vale ressaltar que não existe um melhor do que o outro. Existe aquele que satisfaz a necessidade e que tenha um custo benefício baixo.

No passado, usavam-se circuitos digitais ou analógicos para realizar determinadas operações, baseados em portas lógicas, tornando um circuito com vários componentes e geralmente suscetíveis a ruídos. A partir do ano de 1969, a empresa BUSICOM, desenvolveu uma calculadora eletrônica Souza (2006). Com este avanço na tecnologia, foi possível reduzir e otimizar circuitos, tornando-os mais eficientes e mais modelados. Após este protótipo a empresa acabou vendendo a patente para a Intel, ao qual lançou o primeiro microprocessador o Intel 4004, isso em 1971 com apenas 4 bits.

2.1 A plataforma Arduino

A ideia da plataforma de desenvolvimento Arduino, surgiu como solução de uma necessidade de alguns professores da Itália, no qual os estudantes que quisessem, desenvolver algum protótipo em plataformas de desenvolvimentos deveriam comprar como base, a plataforma BASIC stamp (Figura 1).

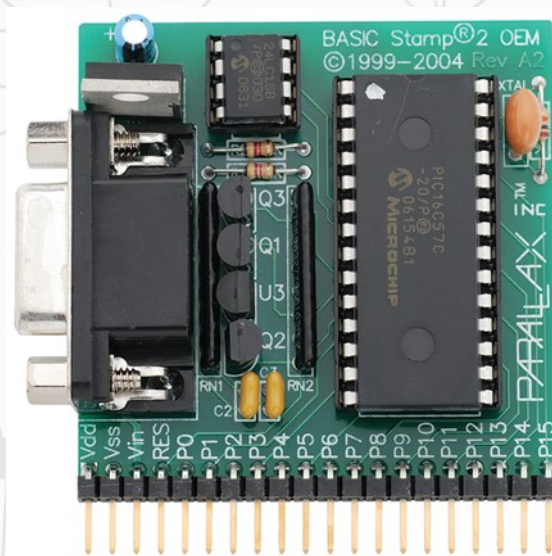


Figura 1 – Basic stamp (Todbot, 2006).

Conforme pode ser visto na Figura 1, a plataforma BASIC stamp, com um microcontrolador (μC), da empresa Microchip. No entanto, o preço desta plataforma girava em torno de €65,00 a €120,00, em 2005.

Como solução, nasce o Arduino, em meados de 2005 na Itália (Figura 2).



Figura 2 – Arduino vs BASIC stamp (Todbot, 2006)

Como pode ser visto na figura 2, podemos fazer um comparativo visual e de preço da plataforma Arduino com a BASIC stamp. Na mesma figura, à esquerda pode-se ver a plataforma Arduino com seu respectivo custo. À direita, podemos ver a plataforma BASIC stamp com seu respectivo preço. Além da diferença de preço, verifica-se o tamanho de uma plataforma em comparação com a outra. Com esta redução no preço, começava na época o estímulo ao desenvolvimento de pequenos projetos. Na atualidade, este passo dado pelos desenvolvedores do Arduino, possibilitou o espírito hobista.

Com a disseminação da plataforma Arduino e juntamente com o espírito hobista, cria-se a necessidade do projeto dos shields. Shields são hardware, dimensionados para uma determinada aplicação com reles, teclados e displays. Os shields são estruturas construídas para uso exclusivo em paralelo com o Arduino, sendo que cada uma contém componentes voltados para uma determinada aplicação, o que ampliava ainda mais o horizonte de expectativas para projetos desenvolvidos.

Shields mais simples, como os para controle de motores, sensores de distância ou de presença, transformaram o Arduino em um caminho para novos estudantes e amantes da eletrônica. A Figura 3 mostra alguns shields.



Figura 3 – Shields (Randomnerdtutorial, 2016)

A Figura 3, demonstra a forma de alguns shields, comercializados para uso em plataformas de desenvolvimento.

2.1.1 O Hardware

O Arduino é composto por uma estrutura relativamente simples. Ele possui um microprocessador de 8 bits com uma velocidade de clock de 16 MHz. As principais características do Arduino são: é um microcontrolador (μC) de 8 bits, ADC de 8 canais com 10 bits, com uma resolução de ADC de 1024 (10 bits), frequência máxima de operação de 20Mhz, 6 canal para PWM (Pulse Width Modulation), tensão de operação 1,5 volts á 5,5 volts, memória flash 32kBytes EEPROM 1kBytes e RAM 2kBytes. É possível obter maiores informações através do datasheet do Atmega328p (ATMEL, 2015).

Para comunicação com o software no computador, há um ATmega16u que possibilita esta conexão através do USB, via protocolo USART e o mesmo ATmega16u é responsável pela gravação do código no atmega328p.

A placa contém 14 pinos de entrada e saída digital, com tensão de 5V e corrente máxima de 40mA por pino e 200mA totais. Porém, estas também têm resistores pull-up que podem ser utilizados quando for realizar uma ou mais leitura digital em níveis lógico de tensão em alto e baixo. Sendo que destas, as entradas 3, 5, 6, 9, 10 e 11 têm a opção de serem usadas como PWM de 8 bits. Há um barramento de entradas analógicas. Além disso, há um outro onde há opções de tensão 5V, 3,3V regulada e, também, um pino para alimentação, além de duas referências (gnd). A Figura 4 mostra a parte frontal e a parte de trás da plataforma Arduino.

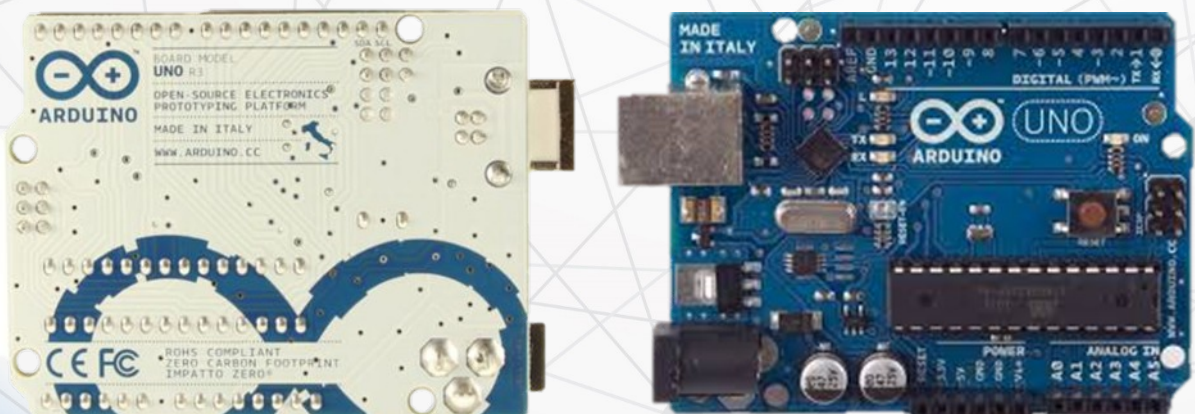


Figura 4 – O Arduino (Arduino, 2015)

A Figura 4, demonstra a fotografia da plataforma de desenvolvimento Arduino, a face frontal e traseira, na frontal fica localizado as os pinos de entradas e saídas, localiza-se o μ C.

2.1.2 O Software

O Arduino Possui como software padrão uma IDE (Integrated Development Environment) chamada de Arduino. Este software possui interface de fácil interação, como apresenta a Figura 5.

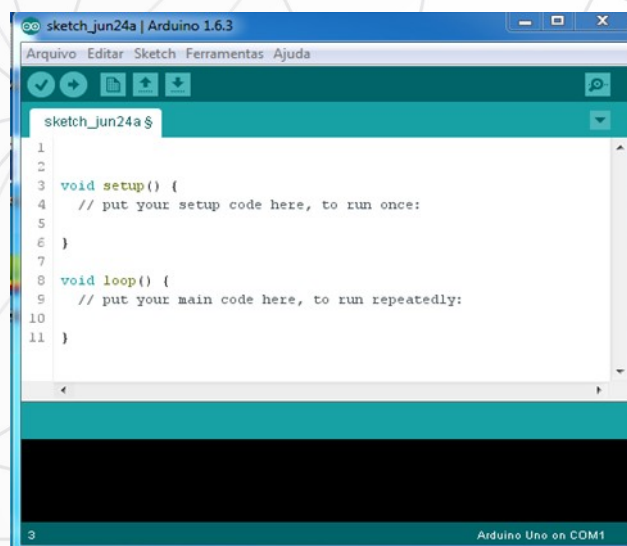


Figura 5 – Ide Arduino, (o autor).

A IDE pode ser dividida em três seções: (1) primeira como declarações de variáveis; bibliotecas, a (2) segunda e para setar configurações que serão usadas no código, como definir se um pino é entrada ou saída e (3) terceira, a função principal, é onde o seu código irá desenvolver as atividades pré-programadas.

Com o desenvolvimento da plataforma Arduino, populariza-se a linguagem wiring de alto nível. Anteriormente, a mais popularizado era a linguagem basic. A ide do Arduino é disponibilizado no sitio

<<https://www.arduino.cc/>>, sem nenhum custo adicional. Neste mesmo sítio, há vários exemplos de suporte, como se trata de um projeto open source (código aberto), fica acessível ademais dá uma vasta gama de sítios e blogs abordando o assunto, geralmente focado ao público maker. Possui também vários fóruns online mais voltado ao desenvolvimento científico com dúvidas mais pontuais, como a AVRfreaks, que pode ser acessada neste link: <<http://www.avrfreaks.net/>>.

Na Tabela 1, é possível fazer uma comparação de código em três tipos de linguagem de programação.

Tabela 1 - Comparação entre código, (o autor)

BASIC	Arduino	C
<pre>{ \$ SELO BS2 } { \$ PBASIC 2.5 }</pre>	<pre>int ledPin = 13; void setup() { pinMode(ledPin, OUTPUT); }</pre>	<pre>#include <avr/io.h> #include <util/delay.h> #define BLINK_DELAY_MS 1000 int main (void) { DDRB = _BV(DDB5); while(1) { PORTB = _BV(PORTB5); _delay_ms(BLINK_DELAY_MS); PORTB &= ~_BV(PORTB5); _delay_ms(BLINK_DELAY_MS); } }</pre>
<pre>OUTPUT 14 FAZ HIGH 14 PAUSE 1000 LOW 14 PAUSE 1000 LOOP</pre>	<pre>void main () { digitalWrite (ledPin, HIGH); delay(1000); digitalWrite (ledPin, LOW); delay (1000); }</pre>	

Na primeira coluna da Tabela 01, é possível verificar um código pisca LED (Light-emitting diode). Em linguagem Basic, utilizado na plataforma Basic stamp. A segunda coluna demonstra o mesmo exemplo na IDE arduino em linguagem wiring. Na terceira coluna verifica-se a mesma aplicação em linguagem C. Com estes três exemplos, fica mais fácil e coerente a

comparação entre as linguagem, Podemos verificar que a linguagem wiring (Arduino) é de mais fácil entendimento.

A ide arduino mantém algumas palavras reservadas no sistema, como demonstrado na Tabela 2.

Tabela 2 – palavras reservadas da linguagem C, (Lima e Villaça, 2012).

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while
byte	word		

Na tabela 2, foram expostas as palavras reservadas pela linguagem C que executam algum comando. Quando for desenvolver o próprio código, não é possível utilizar estas palavras com variáveis locais ou globais. (Lima e Villaça, 2012). Além das palavras reservadas pela linguagem C, a ide arduino possui algumas palavras próprias, como expõe a Tabela 3.

Tabela 3 – Palavras reservadas da ide Arduino, (o autor).

True	false	HIGH	LOW	INPUT	OUTPUT
pinMode()	digitalRead()	digitalWrite()	analogRead	analogWrite()	delay()
setup()	Serial.begin()	Serial.print()	Serial.available()	Serial.read()	

Na tabela 3, verifica-se o aparecimento de algumas palavras do tipo “palavra()”, isso significa que esta palavra é uma função, a qual, dentro dos parênteses o desenvolvedor irá passar um ou mais parâmetros. Um exemplo é a função “Serial.begin(valor da transmissão)” que, dentro dos parênteses, informa o valor de transmissão de dados. Além destas palavras reservadas, poderão ocorrer de aparecer outras no seu código que irá depender das bibliotecas que estão sendo utilizada no seu código, para saber deverá

analisar as bibliotecas incluídas. Ocorrem também as palavras reservadas das variantes das reservadas. Um exemplo é a palavra reservada "delay()" que possui com uma de suas variantes a palavra "_delay_ms()".

2.2 Sistemas embarcados

Sistemas embarcados é um conjunto de software com hardware, que irá desenvolver uma atividade específica. Atualmente, um sistema embarcado é capaz de salvar uma vida, como o uso de um marca-passo. Na Figura 6, é demonstrado um diagrama de funcionamento básico de um sistema embarcado.

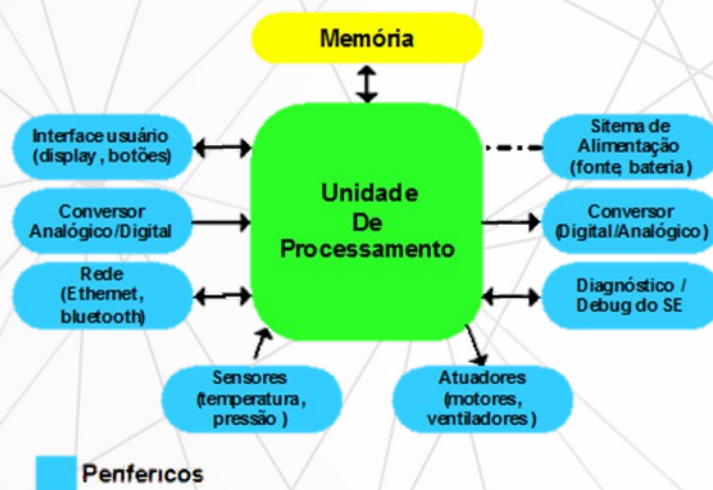


Figura 6 - diagrama de um sistema embarcado, (Todbot, 2006).

A Figura 6, demonstra um breve diagrama de sistema embarcado simples, conforme aumenta a complexidade do sistema embarcado, vai aumentando os sensores, conversores e as interfaces.

Basicamente, os sistemas embarcados possuem um núcleo de processamento, a qual, são tomadas as decisões; uma memória que irá armazenar estados; e os periféricos, aos quais geralmente são responsáveis pela interação homem máquina e pela coleta dos parâmetros que estão sendo monitorados.

O primeiro sistema embarcado foi o Apollo Guidance Computer, responsável pelo controle da espaçonave apollo. Na época não possuía μC , o mesmo foi construído usando portas lógicas do tipo NOR.

Atualmente, os sistemas embarcados estão cada vez mais com um custo-benefício reduzido e, com isso, acaba impactando no seu uso.

2.2.1 A plataforma Arduino como sistemas embarcados

A plataforma Arduino vem cada vez mais se popularizado devido a sua facilidade em sua programação. O arduino, como sistema embarcado, vem sendo vastamente utilizado por hobistas e estudantes de eletrônica, para pôr em prática o que é adquirido em sala de aula, e para os hobistas, como desafogamento para o seu dia a dia, como por exemplo, o controle de da residência, ou até mesmo o controle da comida do seu animal de estimação, com estas aplicações vem o conceito da internet das coisas, um protocolo vastamente utilizado nas indústrias, para controle dos equipamentos, para nos hobista/estudantes, podemos aplicar o conceito da internet das coisas para verificar o que o seu animal de estimação está fazendo em casa, sem precisar sair do escritório, ou um controle de acesso, via RFID , equipamento disponibilizado como shield, para a plataforma de desenvolvimento Arduino.

2.3 O Arduino no aprendizado da eletrônica

A plataforma Arduino torna-se interessante no aprendizado de eletrônica, pelo fato de ser simples e prático. Muitas vezes o estudante ou hobista coloca em prática um projeto com um microcontrolador (μC) e a dificuldade de saber se o software está mal projetado ou se o hardware está com mal funcionamento é muito maior nos testes de validação. Neste momento, as plataformas de desenvolvimento entram, pois, desta forma, tem-se a certeza de que o problema não está no hardware, mas sim no software. Após isso, é possível desenvolver o hardware. Mesmo que não tivesse domínio nem um sobre hardware e quiser controlar uma lâmpada por um sensor de luminosidade LDR (Light Dependent Resistor), com o Arduino fica fácil de realizar esta aplicação, pois conta com uma ide simples, como

será demonstrado adiante. Durante o desenvolvimento do hardware, existe um tempo gasto e, em muitas situações, quando o projetista não precisa se

preocupar com tal etapa, torna a criação de pequenos e grandes projetos mais acessíveis, o que acaba instigando cada vez mais os estudante e os hobistas a irem em busca dos conhecimentos e entendimentos da eletrônica.

Vejamos o exemplo do LDR para comprovar a facilidade na implementação. Para esta aplicação, faz-se necessário três componentes: o (1) Arduino uno, um (2) shield de LDR e um (3) shield de relê. A Figura 7 mostra dois shields,



Figura 7 – shields por exemplo, (o autor).

A Figura 7 demonstra, à esquerda o shield de LDR e o da direita o shield de relê. O LDR vai realizar o levantamento da luminosidade do ambiente, já o relê irá acionar a lâmpada, caso seja necessário.

Após ter os componentes em mãos, as instruções para o Arduino uno via ide são passadas. Na Tabela 4, podemos ver o exemplo de um código pisca led.

Tabela 4 – código exemplo, (o autor).

<pre>//comentário simples // Programa : LDR - Sensor de Iluminação int portaLed = 10; //Porta a ser utilizada para ligar o led int portaLDR = A5; //Porta analógica utilizada pelo LDR int valAcio = 800; // ajuste o valor conforme necessario void setup() { pinMode(portaLed, OUTPUT); //Define a porta do Led como saída }</pre>	<pre>void loop() { int estado = analogRead(portaLDR); //Lê o valor fornecido pelo LDR // Caso o valor lido na porta analógica seja maior do que // valAcio, acende o LED if (estado > valAcio) { digitalWrite(portaLed, HIGH); } else //Caso contrário, apaga o led { digitalWrite(portaLed, LOW); } }</pre>
--	---

A Tabela 4, expõe um código que irá acionar uma lâmpada, caso a luminosidade do ambiente esteja abaixo da programada pelo projetista.

Após estes procedimentos, ligue os shields no Arduino, a saída analógica do LDR ligue na entrada analógica do Arduino, neste caso a porta "A5", posteriormente ligue a entrada do relê na saída digital do Arduino, neste caso a porta "10", faça a ligação da alimentação dos shields, e pronto tem um sistema de controle por LDR funcionando.

Geralmente, os hobistas ou makers, não possuem suporte para o desenvolvimento e confecção de um hardware. Neste contexto, também são aplicadas as plataformas de desenvolvimento e os famosos shields, atualmente possui, um grande aceite em automação residencial, com o Arduino e seus shields, viabiliza estas aplicações, desde o seu hardware até o seu software, que é de fácil manuseio, como demonstrado no exemplo do LDR, não precisou em nem um momento projetar um hardware para este exemplo, o qual fosse necessário tomaria um tempo maior e dedicação, pois

teríamos que desenhar o esquemático em um software específico e posteriormente realizar a confecção da Placa de Circuito Impresso (PCB).

3 METODOLOGIA

Surge um desafio, ao qual se dispõe trazer um artigo de referência bibliográfica sobre a interação no aprendizado na eletrônica com o auxílio de plataformas de desenvolvimento. Tendo em vista que nos últimos anos a eletrônica ficou vastamente difundida com seu conhecimentos gerais. Em seguida o próximo passo seria a escolha de um tema geral. O tema principal escolhido foi sobre sistemas embarcado. Posteriormente, foram levantados vários tópicos a serem abordados. Com estes tópicos documentados, consistir em solicitar algumas opiniões para alguns colegas/amigos. Vários dos tópicos levantados pelo autor foi banido, devido a sua alta complexidades para o momento, o autor decidiu o tema específico que foi o Arduino no contexto da eletrônica, após estes passos iniciais, foi assistido um documentário chamado de “Arduino: The Documentary”, disponível no link: <https://www.youtube.com/watch?v=SiTmTP_F6so>. Foram definidos algumas palavras-chave tais como: Arduino, Sistemas embarcados e plataforma de desenvolvimento para a busca de artigos de referência e com isso passa a estudar a plataforma Arduino, a ide, e os shields. Após esta etapa, inicia-se a próxima etapa que foi a leitura de alguns artigos que tem como base o uso da plataforma Arduino no desenvolvimento de protótipos. Posteriormente deu-se início ao desenvolvimento do artigo de revisão bibliográfica.

3 CONCLUSÃO

O autor tinham como proposta inicial de dissertar sobre a influência dos sistemas embarcados no aprendizado da eletrônica. No entanto, voltaram-se para uma visão mais sucinta, com a intenção de mostrar o quão interessante e produtiva é a inserção das plataformas de prototipagem, no

caso o Arduino, no contexto da eletrônica. A criação de protótipos para iniciantes estimula o desenvolvimento e a ampliação do interesse por tais assuntos, devido à visualização dos conteúdos vistos em sala, para estudantes, ou até mesmo para hobistas que leem sobre o meio da eletrônica.

Tais facilidades, propostas pela plataforma, influenciam de forma positiva o aprendizado da eletrônica, e fica claro a vantagem de uma plataforma de desenvolvimento como demonstrado no exemplo do LDR, ao qual economiza tempo na confecção da PCB. Com o Arduino, é possível desenvolver projetos, sem que aja uma infraestrutura significativa, pois algumas partes do seu projeto não se faz necessário seu desenvolvimento.

REFERÊNCIAS

- ARDUINO (Itália). Arduino. 2015. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 06 set. 2026.
- ATMEL (Eua). Datasheet atmega328p. 2015. Disponível em: <http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>. Acesso em: 15 set. 2026.
- BORGES, Rodrigo Weissmann. Aplicabilidade de Sistemas Operacionais de Tempo Real (RTOS) para sistemas embarcados de baixo custo e pequeno porte. 2011. 152 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade de São Paulo, Carlos, 2011.
- ELECTRONICS LAB. Shields. 2016. Disponível em: <<http://www.electronics-lab.com/25-useful-arduino-shields-that-you-might-need-to-get/>>. Acesso em: 16 set. 2026.
- LIMA, Charles Borges de; VILLAÇA, Marco Valério Miorim. Avr e arduino: técnicas de projeto. 2. ed. Florianópolis: dos Autores, 2012. 632 p.
- SOUZA, Vitor Amadeu. Programação em BASIC para o microcontrolador PIC18F1220. São Paulo: Erica, 2006. 264 p.
- TODBOT. Arduino, the Basic Stamp killer. 2006. Disponível em: <<https://todbot.com/blog/2006/09/25/arduino-the-basic-stamp-killer/>>. Acesso em: 16 set. 2026.

Sobre o(s) autor(es)

Discente Ciência da Computação – campus Maravilha, natanogliari@gmail.com