
MINISTÉRIO DA EDUCAÇÃO – MEC
SECRETARIA DE EDUCAÇÃO SUPERIOR – SESU
PROGRAMA DE EDUCAÇÃO TUTORIAL – PET

UNIVERSIDADE FEDERAL FLUMINENSE – UFF
ESCOLA DE ENGENHARIA – TCE
GRUPO PET DO CURSO DE ENG. DE TELECOMUNICAÇÕES – PET-TELE

Dicas PET-Tele

FAQ sobre Arduino
(Versão: A2021M07D04)

Autores: Brenda de Farias Albuquerque

Luisa Nunes Azevedo

Professor: Alexandre Santos de la Vega

Niterói-RJ
Julho / 2021

Sumário

1	Perguntas Frequentes	2
1.1	O que é um <i>software</i> Livre?	2
1.2	Instalando os <i>drivers</i>	2
1.2.1	Instalando <i>drivers</i> para o Arduino Uno ou Arduino Mega 2560 no Windows7, Vista, ou XP	2
1.2.2	Instalando <i>drivers</i> para o Arduino Duemilanove, Nano, Diecimila ou Ale-vino com Nanoshield USB no Windows7, Vista, ou XP	3
1.3	Posso rodar a IDE do Arduino no Mac OS?	3
1.4	Posso rodar a IDE do Arduino no Linux?	3
1.5	Como instalar a IDE do Arduino no Linux?	3
1.5.1	Pela central de programas	3
1.5.2	Pelo terminal	4
1.6	Posso usar uma IDE diferente?	4
1.7	Posso usar a placa do Arduino sem utilizar o <i>software</i> do Arduino?	4
1.8	Posso programar em C/C++?	4
1.9	Posso usar o <i>software</i> do Arduino com outras placas AVR?	4
1.10	Onde encontrar bibliotecas para Arduino?	4
1.11	Quais são as placas oficiais do Arduino?	4
1.12	Simuladores de Arduino	5
1.13	Selecionar sua placa	5
1.14	Selecionar a porta serial	6
1.15	Fazer <i>upload</i> do programa	7
1.16	Erros de sintaxe	8
2	Bootloader	9
2.1	Como usar a placa oficial do Arduino para gravar o <i>bootloader</i> ?	9
2.2	Preciso gravar o <i>bootloader</i> no microcontrolador Atmega?	10
2.3	Posso usar o microcontrolador Atmega fora da placa do Arduino?	10
3	Apêndice	11
3.1	Apêndice 1 - Estrutura do código	11
3.1.1	Estruturas de controle	11
3.1.2	Comentários	11
3.1.3	Operadores	12
3.2	Apêndice 2 - Funções	13
3.2.1	Entrada e saída digitais	13
3.2.2	Entrada e saída analógicas	13
3.2.3	Entrada e saída avançadas	13
3.2.4	Tempo	14
3.2.5	Funções matemáticas	14
3.2.6	Números aleatórios	14
3.2.7	Comunicação serial	15

Capítulo 1

Perguntas Frequentes

1.1 O que é um *software* Livre?

Para que um *software* seja considerado livre este precisa respeitar quatro liberdades essenciais definidas pela Free Software Foundation:

- Executar o programa para qualquer propósito;
- Adaptá-los às necessidades dos usuários, fornecendo o acesso ao código fonte;
- Redistribuir cópias para outros usuários;
- Liberdade para distribuir cópias de suas versões modificadas a outros usuários, facilitando, assim o acesso da comunidade aos benefícios das mudanças.

1.2 Instalando os *drivers*

1.2.1 Instalando *drivers* para o Arduino Uno ou Arduino Mega 2560 no Windows7, Vista, ou XP

1- Conecte a placa e aguarde até que o Windows inicie o processo de instalação de *drivers*. Caso a instalação não obtenha êxito, siga os passos abaixo;

2- Clique no menu iniciar e vá até o Painel de Controle;

3- Selecione “Sistema e Segurança”, em seguida selecione “Sistema”, e após escolha “Gerenciador de Dispositivos”;

4- Procure na seção “Portas (COM e LPT)”. Você deverá encontrar uma porta aberta chamada “Arduino UNO (COMetc)”;

5- Clique com o botão direito na porta “Arduino UNO (COMetc)” e selecione “Atualizar *Driver*.”;

6- Depois selecione a opção “Procurar *software* de *driver* no computador”;

7- Finalmente, navegue até a pasta “*drivers*” dentro do aplicativo do Arduino, na pasta que foi descompactada anteriormente, e selecione o arquivo “arduino.inf”;

8- A partir daí, o Windows completará a instalação dos *drivers*.

1.2.2 Instalando *drivers* para o Arduino Duemilanove, Nano, Diecimila ou Alevino com Nanoshield USB no Windows7, Vista, ou XP

1- Quando conectar a placa, o Windows deve iniciar o processo de instalação dos *drivers*, caso você ainda não tenha usado o computador com um Arduino antes;

2- A janela “Adicionar novo *hardware*” irá aparecer: Nessa janela, selecione “Não, não agora” e clique em avançar;

3- Selecione “Instalar o *driver* de uma lista ou um lugar específico (avançado)” e clique em avançar;

4- Verifique que “Procurar o melhor *driver* nesses lugares específicos” está selecionado; Desmarque “Procurar em mídias removíveis”; Selecione “Incluir este lugar na procura” e vá até o diretório “*drivers*” do Arduino, dentro da pasta que você descompactou anteriormente (a versão mais atual dos *drivers* pode ser encontrada no site da FTDI (<http://www.ftdichip.com/FTDrivers.htm>)). Depois, clique em avançar;

5- O Windows irá procurar pelo *driver* e dizer que um “USB Serial Converter” foi encontrado. Clique em finalizar;

6- A janela “Adicionar novo *hardware*” aparecerá novamente. Siga os mesmos passos, selecionando as mesmas opções e o mesmo diretório. Dessa vez, uma “USB Serial Port” será encontrada;

7- Você pode verificar que os *drivers* foram instalados clicando no menu iniciar e depois em Painel de Controle > Sistema e Segurança > Sistema > Gerenciador de Dispositivos. Procure por uma “USB Serial Port” na seção “Portas (COM e LPT)”.

1.3 Posso rodar a IDE do Arduino no Mac OS?

A IDE está disponível para Mac OS, sistemas com versão *10.7 Lion* ou superiores. As versões do *software* estão disponíveis no *website* da plataforma (<https://www.arduino.cc/en/Main/Software>).

1.4 Posso rodar a IDE do Arduino no Linux?

A IDE está disponível para Linux, tanto para sistemas de 32 quanto para 64 *bits*. As versões do *software* estão disponíveis no *website* da plataforma (<https://www.arduino.cc/en/Main/Software>).

1.5 Como instalar a IDE do Arduino no Linux?

Antes de realizar esse procedimento, é necessário ter Java instalado no computador para que a IDE funcione corretamente. Caso não possua Java, instale-o antes de prosseguir. Feito isso, a instalação da IDE pode ser feita de duas formas: pela central de programas ou pelo terminal.

1.5.1 Pela central de programas

1- Acesse a central de programas do Ubuntu;

2- Pesquise por Arduino;

3- Clique no ícone da IDE;

4- Clique em Instalar e insira a senha de administrador;

5- Clique em proceder e aguarde o fim do *download*.

1.5.2 Pelo terminal

1- Acesse o Terminal do Linux, e utilize o comando *apt-get install Arduino*.

1.6 Posso usar uma IDE diferente?

Sim. É possível compilar programas para o Arduino utilizando outras plataformas. Algumas configurações deverão ser feitas para que a ligação com as bibliotecas do Arduino ocorra de forma apropriada. Vale ressaltar que, para projetos iniciantes, a IDE própria do Arduino cumpre as funcionalidades e é de fácil utilização.

Outras alternativas à IDE:

- Sublime Text 3 (<https://www.sublimetext.com/3>) + extensão Stino (<https://robot-will.github.io/Stino/>)
- Atmel Studio (<http://www.atmel.com/Microsite/atmel-studio/software.aspx>) + extensão Visual Micro (<http://www.visualmicro.com/>)
- Visual Studio (<https://www.visualstudio.com>) + extensão Visual Micro (<http://www.visualmicro.com/>)
- Arduino Eclipse (<http://www.baeyens.it/eclipse/>)

1.7 Posso usar a placa do Arduino sem utilizar o *software* do Arduino?

Sim. A placa do Arduino é apenas uma placa de desenvolvimento AVR. A IDE utilizada não irá interferir do desenvolvimento do programa.

1.8 Posso programar em C/C++?

A linguagem utilizada no Arduino é C/C++, com diminutas modificações. Tais modificações são referentes aos tipos de biblioteca e funções específicos para o hardware. Portanto, ao programar o Arduino, há a obrigatoriedade do emprego da linguagem C/C++.

1.9 Posso usar o *software* do Arduino com outras placas AVR?

Sim, contudo algumas modificações podem ser necessárias nas bibliotecas do Arduino.

1.10 Onde encontrar bibliotecas para Arduino?

As bibliotecas compatíveis estão disponíveis, separadas por categorias e especificidades no *website* da plataforma (<https://www.arduino.cc/en/Reference/Libraries>).

1.11 Quais são as placas oficiais do Arduino?

Para consultar se a placa utilizada é oficial, basta consultar o *link* <https://www.arduino.cc/en/Main/Products?from=Main.Hardware>. Uma vez que, está página contém uma lista com todos os produtos incorporados pela plataforma e, portanto, considerados oficiais. Vale salientar que, apenas produtos oficiais podem utilizar a palavra “Arduino” em seu nome.

1.12 Simuladores de Arduino

Há momentos em que é necessário montar um circuito e efetuar a simulação para verificar o funcionamento do código, porém nem todos os componentes necessários estão disponíveis. Neste caso, um simulador pode ser utilizado como suporte. A seguir, listamos alguns dos que se encontram disponíveis na internet:

- **Virtual BreadBoard**

O Virtual Breadboard é um ambiente virtual para simulação de circuitos eletrônicos, juntamente com protoboards e microcontroladores. O ambiente é dotado de vários componentes e instrumentos virtuais, representados de forma realista na área de trabalho (JANELA) do simulador. Apesar de fazer simulação como diferentes tipos de microcontroladores, ele possui um kit especialmente voltado para Arduino, disponível apenas para Windows.

No *website* do fabricante (<http://www.virtualbreadboard.com/>) é possível tanto fazer *download* do Virtual Breadboard, quanto acessar a *Market Place*, para adquirir a licença do Arduino Toolkit.

- **123D Circuits/TinkerCAD**

O 123D Circuits é um *software online* que viabiliza a criação de projetos Arduino, bem como a simulação de seu funcionamento. Existem as opções de montar o circuito utilizando tanto o *layout* de protoboard, quanto o de PCB, e, para a segunda opção, há a possibilidade de imprimir o circuito elaborado diretamente do *website*, utilizando uma impressora 3D. Como é permitido que cada usuário compartilhe seu projeto, a ferramenta possui uma vasta biblioteca que pode ser consultada.

É oferecido em versão *free*, ou em versão paga, que possui uma série de vantagens em relação à versão gratuita.

Para utilizar a ferramenta, basta acessar o *website* (<https://www.circuits.io/>) e criar uma conta. Caso queira usar somente o ambiente de simulação, basta acessar o *website TinkerCAD*, (<https://www.tinkercad.com>), responsável pelo aplicativo de laboratório de eletrônica.

- **Proteus**

O Proteus é um ambiente que permite montar, simular e criar circuitos impressos de projetos eletrônicos. Apesar de não ser um *software* gratuito, seu fabricante oferece uma versão demo, que pode ser baixada gratuitamente na página da empresa. É considerado uma das melhores opções de simuladores, pois possui uma vasta quantidade de componentes, que podem auxiliar na criação de projetos.

A ferramenta está disponível em <https://www.labcenter.com/downloads/>

Observação: Para realizar somente a modelagem de circuitos, há o **Fritzing**, que é um *software* livre. Ele fornece o diagrama elétrico e o *layout* PCB do projeto e, ainda oferece a possibilidade de fazer a encomenda do circuito impresso no próprio site do fabricante. O *download* deve ser feito acessando o *website* (<http://fritzing.org/download/>) e escolhendo o pacote que melhor se adequa ao sistema operacional utilizado.

1.13 Selecionar sua placa

Ao usar a IDE do Arduino, um dos mais básicos erros e com simples solução é *selecionar a placa*. Por vezes, o processo de reconhecimento da mesma é feito automaticamente. Caso isto não ocorra, será exibida na tela a seguinte mensagem de erro ao carregar o seu código:

Problema ao carregar para a placa.

Veja <http://www.arduino.cc/en/Guide/Troubleshooting\#upload> para sugestões.

Para solucionar este erro, selecione a opção do menu Ferramentas , e em seguida selecione com o cursor a placa que corresponde ao seu tipo de Arduino.

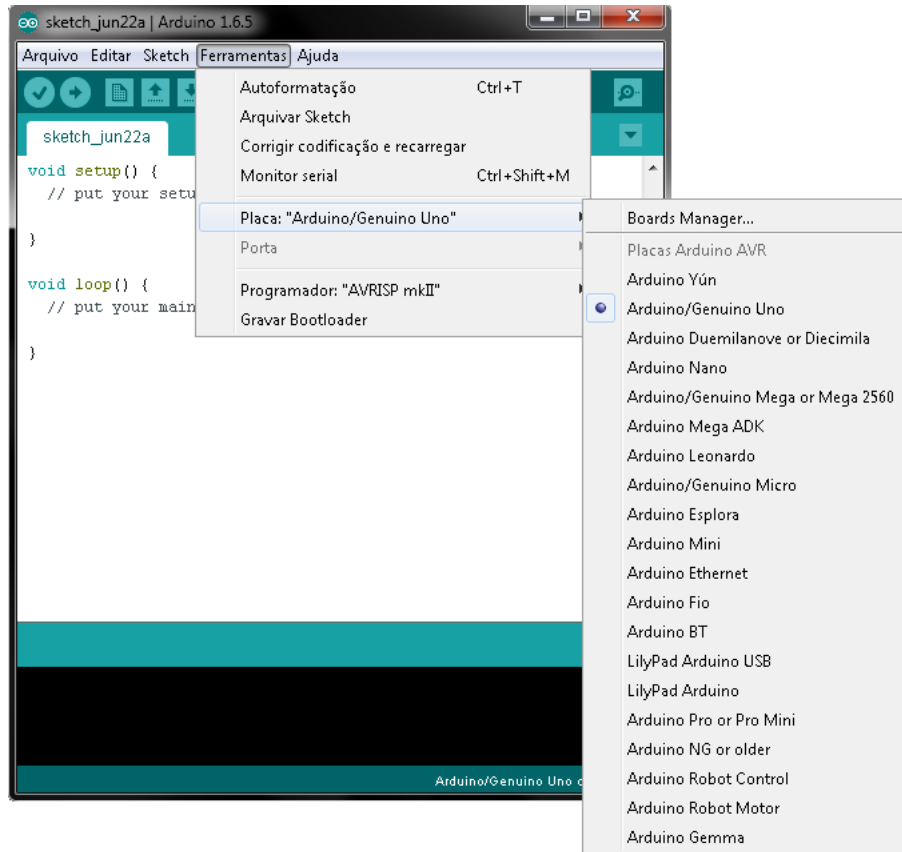


Figura 1.1: Sequência para selecionar a placa do Arduino.

1.14 Selecionar a porta serial

Ao conectarmos o Arduino ao computador, nem sempre a IDE reconhece automaticamente em qual porta serial o mesmo foi conectado. Caso isto aconteça, será exibida na tela a seguinte mensagem de erro ao carregar o código:

Ocorreu um erro enquanto o sketch era carregado

Caso esta mensagem apareça, basta selecionar o dispositivo serial correspondente à placa do Arduino no menu Ferramentas > Porta Serial.

É provável que ele seja COM3 ou maior (COM1 e COM2 são normalmente reservados para portas seriais nativas, e não por USB).

Para descobrir a COM correspondente à sua placa, você pode desconectar o seu Arduino e reabrir o menu; a opção que desaparecer deve ser a da placa do Arduino. Reconecte a placa e selecione a porta serial.

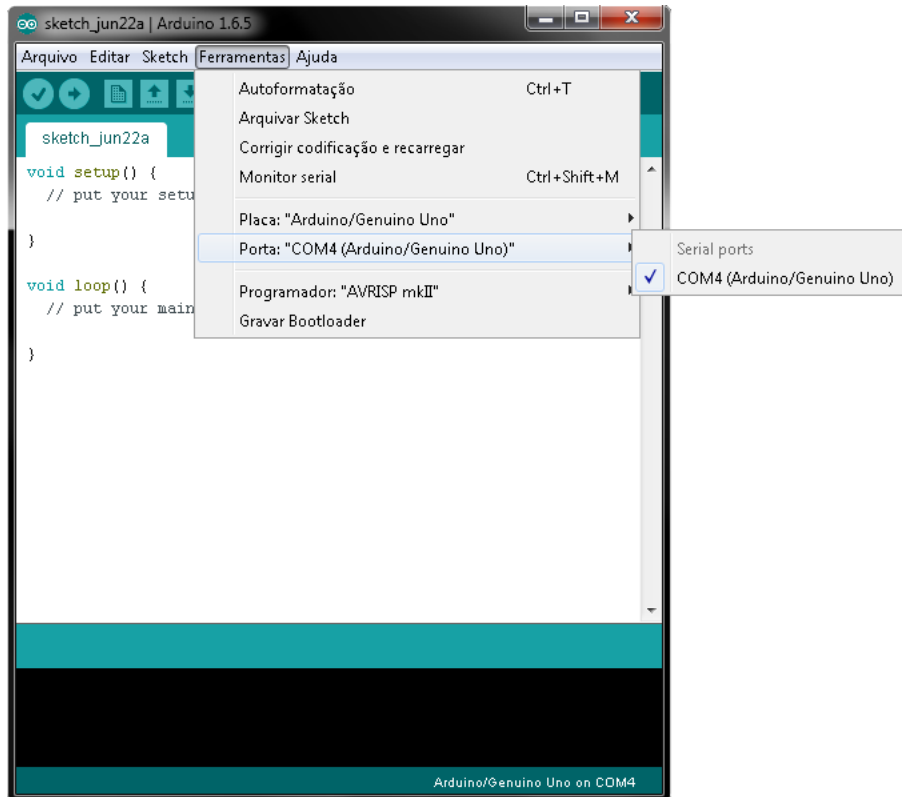


Figura 1.2: Selecionando a porta referente ao Arduino.

1.15 Fazer *upload* do programa

No menu principal, vá até a opção “*Sketch*” e clique no botão “*Carregar*” . Espere alguns segundos - você deve ver os LEDs RX e TX na placa piscando. Se o *upload* for bem sucedido, a mensagem “Transferência concluída” aparecerá na barra de *status*.

Caso, após realizar os passos descritos acima, apareça a seguinte mensagem de erro:

Ocorreu um erro enquanto o sketch era carregado

verifique, primeiramente, se a porta selecionada é a correta, conforme a seção 1.14. Caso ainda assim a mensagem seja exibida, nas versões mais recentes da IDE do Arduino, esse erro pode ocorrer ao abrir a IDE através do arquivo de extensão *.ino*. Uma solução para tal é

- Abra a IDE;
- Vá na aba Arquivo;
- Clique em Abrir;
- Selecione o arquivo *.ino* desejado.

Importante: Se você possui um Arduino Mini, NG, ou outra placa não mencionada aqui, você pode ter que pressionar fisicamente o botão de *reset* na placa imediatamente antes de apertar o botão de *upload*.

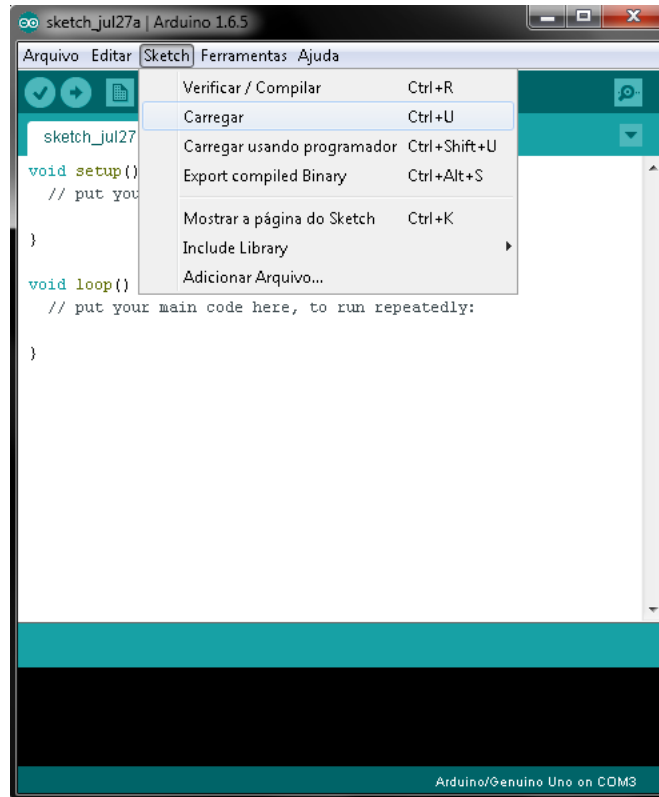


Figura 1.3: Carregando o *Sketch* para a placa.

1.16 Erros de sintaxe

A sintaxe escrita de forma inadequada não permitirá que o código seja carregado à placa. Como a linguagem utilizada no Arduino é C/C++, com suas devidas modificações, alguns erros básicos como a falta de ; ou {} podem retornar mensagens de erro como a seguinte:

```
exit status 1
expected ';' before '}' token
```

Caso isso ocorra, basta verificar a sintaxe do código escrito, procurando funções escritas inadequadamente, ausência de elementos como ;, {, } ou declaração de variáveis.

Capítulo 2

Bootloader

2.1 Como usar a placa oficial do Arduino para gravar o *bootloader*?

- 1- Conecte a placa do Arduino em seu computador e abra a IDE.
- 2- Em seguida, selecione Arquivo > Exemplos > ArduinoISP;
- 3- Selecione a placa e a porta serial correspondentes ao seu Arduino. Tais procedimentos estão descrito nas seções 1.13 e 1.14 deste documento;
- 4- Faça *upload* do programa na placa;
- 5- Conecte sua placa ao dispositivo onde deseja gravar o *bootloader*, de acordo com a imagem abaixo;

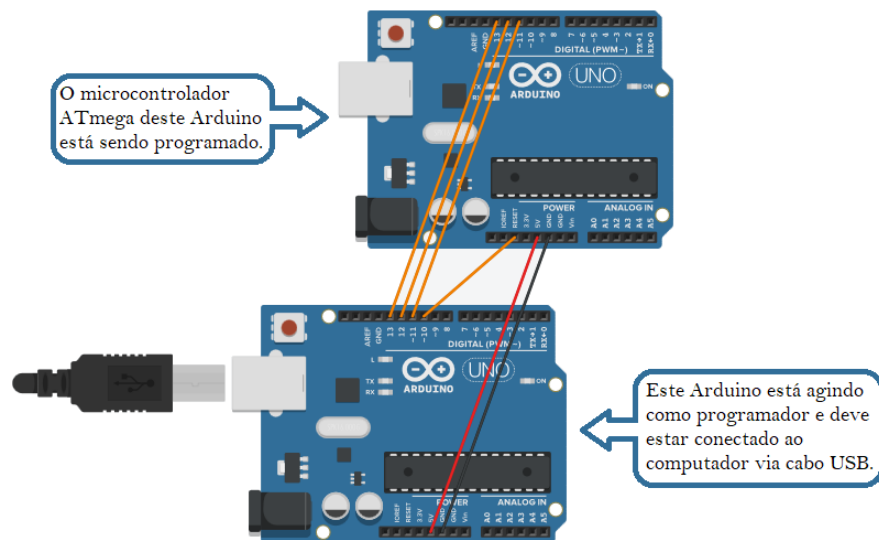


Figura 2.1: Carregando o *bootloader* no ATmega da placa do Arduino.

- 6- Do mesmo modo em que foi selecionada a placa onde o ArduinoISP foi carregado, selecione o

modelo da placa onde o *bootloader* será gravado;

7- Em Ferramentas > Programador, clique em *Arduino as ISP*;

8- Ainda em Ferramentas, clique em Gravar *bootloader*

2.2 Preciso gravar o *bootloader* no microcontrolador Atmega?

Não. Com o software Arduino IDE 0011, você possui a opção de carregar programas no ATmega sem ter o *bootloader* gravado inicialmente. Desta forma, economiza-se espaço de memória, pois não há *bootloader* ocupando espaço em memória, e tempo, pois se não há *bootloader*, o *sketch* é a primeira e única rotina a rodar.

2.3 Posso usar o microcontrolador Atmega fora da placa do Arduino?

Sim. Uma opção é conectar o microcontrolador à uma *protoboard* e carregar diretamente *sketch* desejado ou *bootloader*.

A opção de programar o microcontrolador na placa do Arduino é mais prática, se comparada com a anterior, já que a pinagem da placa facilita a montagem do circuito.

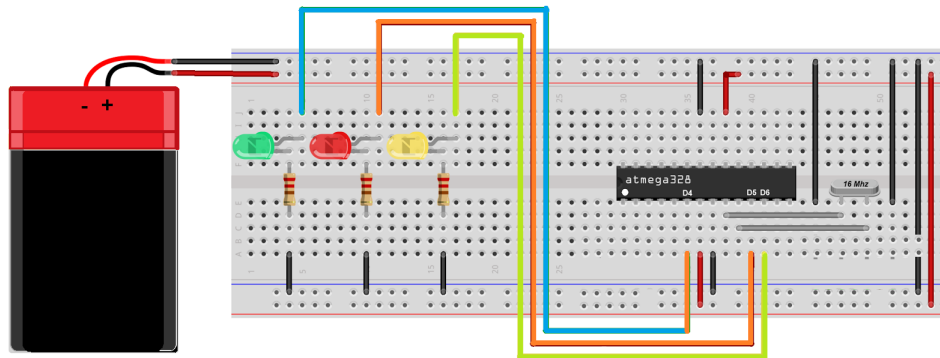


Figura 2.2: Exemplo de circuito envolvendo o ATmega.

Capítulo 3

Apêndice

3.1 Apêndice 1 - Estrutura do código

3.1.1 Estruturas de controle

if
if... else
for
switch case
while
do...while
break
continue
return
goto

3.1.2 Comentários

//	Única linha de comentário
/* */	Conjunto de comentários

3.1.3 Operadores

=	Igualdade
+	Adição
-	Subtração
/	Divisão
*	Multiplificação
==	Comparação
!=	Diferente de
<	menor que
>	maior que
&&	e
	ou
!	não
&)	Operador de <i>bits</i> AND
	Operador de <i>bits</i> OR
^	Operador de <i>bits</i> XOR
~	Operador de <i>bits</i> NOT
<<	Desvio à esquerda
>>	Desvio à direita
&=	De <i>bits</i> compostos AND
=	De <i>bits</i> compostos OR
++	Incremento
--	Decremento
+=	Adição composta
-=	Subtração composta
*=	Multiplificação composta
/=	Divisão composta

3.2 Apêndice 2 - Funções

3.2.1 Entrada e saída digitais

pinMode (pin,modo)
digitalWrite (pin,valor)
digitalRead (pin)

3.2.2 Entrada e saída analógicas

analogRead (pin)
analogWrite (pin,valor)

3.2.3 Entrada e saída avançadas

shiftOut (dataPin, clockPin, bitOrder, valor)
pulseIn (pin, valor)

3.2.4 Tempo

millis ()
macros ()
delay ()
delayMicroseconds ()

3.2.5 Funções matemáticas

min (x, y)	Valor mínimo
max (x, y)	Valor máximo
abs (x)	Valor absoluto
constrain (x, a, b)	Limita o valor de x entre a e b
map (value, fromLow, fromHigh, toLow, toHigh)	Escala
pow (base, exponent)	Potência
sq (x)	Eleva o valor de x ao quadrado
sqrt (x)	Raiz quadrada de x
sin (rad)	Seno
cos (rad)	Cosseno
tan (rad)	Tangente

3.2.6 Números aleatórios

random (max)	Especificando um valor máximo
random (min,max)	Especificando um intervalo
randomSeed ()	Sequência de números aleatórios

3.2.7 Comunicação serial

Serial.begin ()	Inicia a comunicação com a Serial
Serial.available ()	Verifica se a Serial está recebendo dados
Serial.read ()	Leitura de dados pela serial
Serial.flush ()	Esvazia o <i>buffer</i> de entrada da porta serial
Serial.print ()	Imprime dados
Serial.println ()	Imprime cada dado em uma linha diferente

Referências Bibliográficas

- [1] Arduino forum. <https://forum.arduino.cc/index.php?topic=28835.0>.
- [2] Ferramentas para design de circuitos eletrônicos. <https://www.embarcados.com.br/ferramentas-para-design-de-circuitos-eletronicos/>.
- [3] Frequently asked questions. <https://www.arduino.cc/en/Main/FAQ>.
- [4] Simulador on-line para arduino. <http://tecnicolinux.blogspot.com.br/2014/04/simulador-on-line-para-arduino.html>.
- [5] Tinkercad. <https://www.tinkercad.com/learn/>.
- [6] Using an arduino as an avr isp (in-system programmer). <https://www.arduino.cc/en/Tutorial/ArduinoISP>.