

Acionador automático de lâmpadas via RF

Projetista: Leonardo Persike Martins

Email: Leonardo.Persike.Martins@gmail.com

Descrição de funcionamento

O equipamento em questão se trata de um conjunto de dispositivos cujo comportamento baseia-se em um sensor de barreira, utilizado para a detecção de objetos que interrompem um feixe de luz infravermelha. Quando acionado, o dispositivo base envia um sinal de acionamento via rádio frequência para um totem contendo uma lâmpada, acionando assim a mesma por um determinado período de tempo. No caso do sistema em questão, foi planejada a utilização de dois totens, os quais serão acionados de forma aleatória e exclusiva (apenas um totem por vez).

Sumário

Acionador automático de lâmpadas via RF	1
Descrição de funcionamento	1
Sumário	1
Lista de componentes	2
Ferramentas necessárias	3
Diagrama de conexões	4
Sensor de barreira - Receptor	4
Sensor de barreira - Transmissor	4
Totem Luz	4
Instruções adicionais	5
Totem de luz	5
Transmissor e receptor RF	5
Antena	5
Programação do microcontrolador	6
Preparando a interface Arduino	6
Programando	6
Baixando o código	6
Preparação do sistema	7
Inserindo código no relé	7
Clonando código do controle no Arduino	7

Lista de componentes

Os seguintes itens fazem parte de um sistema completo com 2 totens de luz:

Item	Qtd.	Link	Descrição
Placa Arduino Nano	1	Arduino	Módulo contendo microcontrolador que servirá como inteligência do sistema.
Módulo Transmissor RF 433MHz	1	Módulo RF	Módulo RF que opera como um controle.
Antena 433MHz	1	Antena 433MHz	Antena para ser soldada no módulo RF de forma a aumentar o alcance do mesmo.
Módulo Relé 2 (ou mais) canais controlado por RF	2	Relé	Relés conectados aos totens de luz, de forma a receber e interpretar o sinal, acionando e desligando as luzes.
Módulo conversor de tensão 12V para 5V	1	Conversor	Conversor para reduzir a tensão da bateria para 5V, de forma a alimentar os microcontroladores.
Sensor de barreira 12V	1	Sensor Barreira	Sensor de barreira para detectar a passagem do veículo.
Bateria 12V	4	Bateria	Bateria para alimentar os sistemas.
Módulo carregador de bateria 12V	4	Carregador	Módulo para carregamento das baterias.
Fonte 12V	4	Fonte 12V	Fonte para carregar as baterias diretamente pela tomada.
Refletor LED 12V	2	Refletor	Refletores que serão colocados nos totens.
Chave Liga/Desliga	4	Chave on off	Botão para ligar e desligar o sistema.
Conectores rápidos	8	Conector	Conector 2 vias para distribuição de cabos internamente.
Jumpers	2	Jumpers	Jumpers necessários para a interconexão entre componentes.
Cabo PP 2 vias	2	Cabo PP	Cabo 2.5mm (ou mais grosso) para distribuir a alimentação do sistema
Par de garras jacaré	1	Garras	Garras jacaré para conexão da bateria.

Ferramentas necessárias

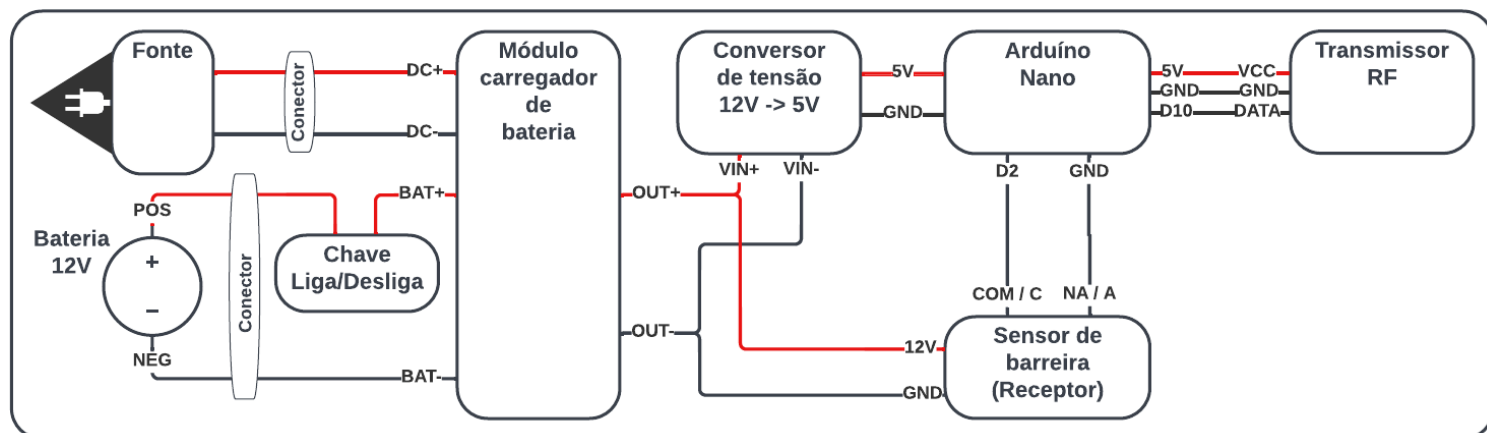
As seguintes ferramentas são necessárias para a montagem dos equipamentos.

Item	Link	Descrição
Ferro de solda	Ferro	Utilizado para soldar as garras jacarés nos cabos que vão para a bateria.
Estanho	Estanho	Utilizado para soldar as garras jacarés nos cabos que vão para a bateria.
Alicate corte	Alicate	Utilizado para descascar os cabos.
Chave philips	Kit	Utilizado para conectar os cabos aos conectores das placas.

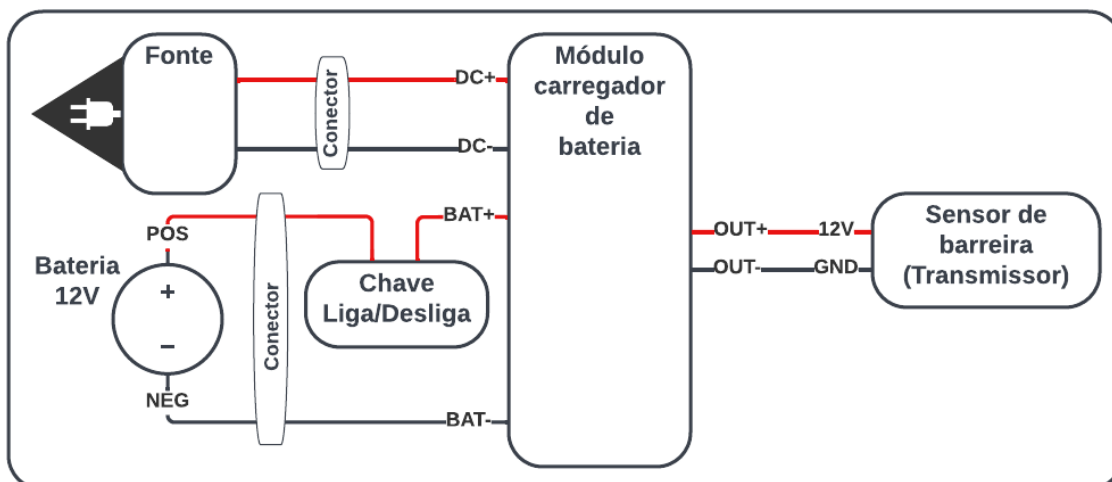
Diagrama de conexões

Os seguintes diagramas podem ser utilizados como referência para a montagem e cabearios dos sistemas. A utilização de conectores é opcional, porém recomendada para facilitar a troca de componentes caso necessário.

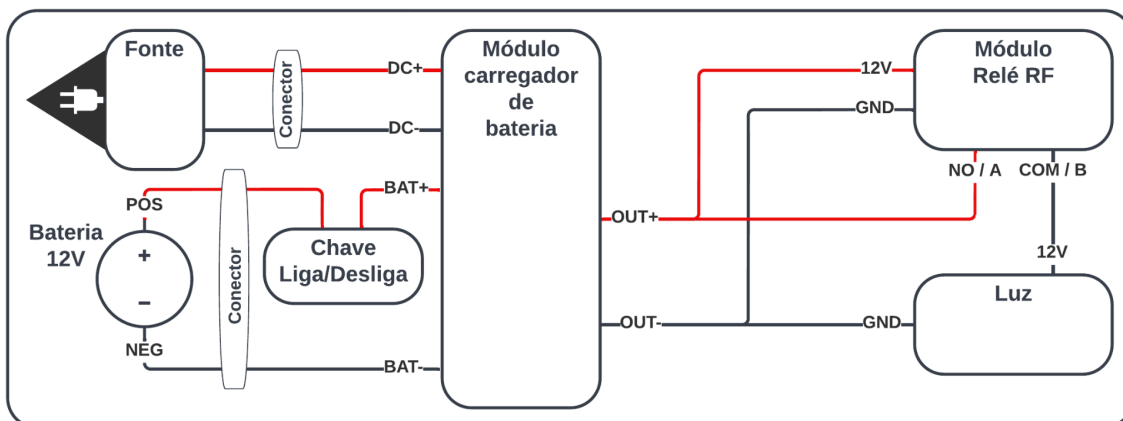
Sensor de barreira - Receptor



Sensor de barreira - Transmissor



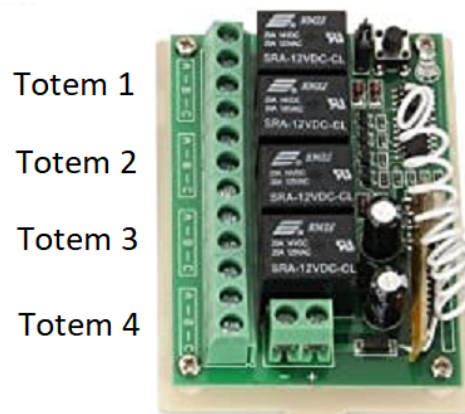
Totem Luz



Instruções adicionais

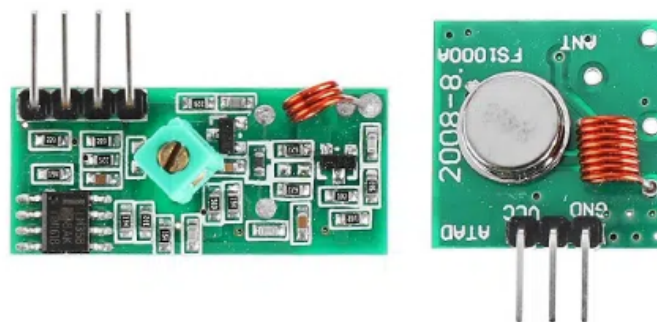
Totem de luz

O sistema tem a capacidade de acionamento de quatro totens de luz diferentes, devido a placa de relé conter 4 relés. Portanto, a conexão dos refletores devem seguir o seguinte diagrama:



Transmissor e receptor RF

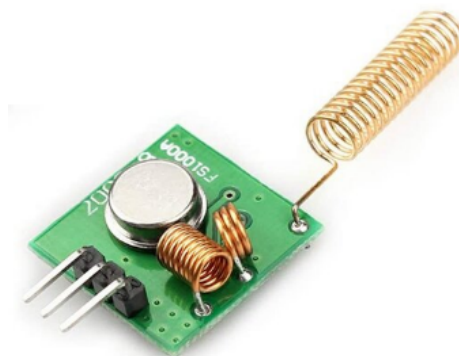
A seguinte imagem identifica os componentes:



Receptor à esquerda. Transmissor à direita.

Antena

De forma a melhorar o alcance do módulo RF, é possível realizar a soldagem de uma antena simples no pino “ANT” do módulo transmissor, conforme indica a imagem a seguir:



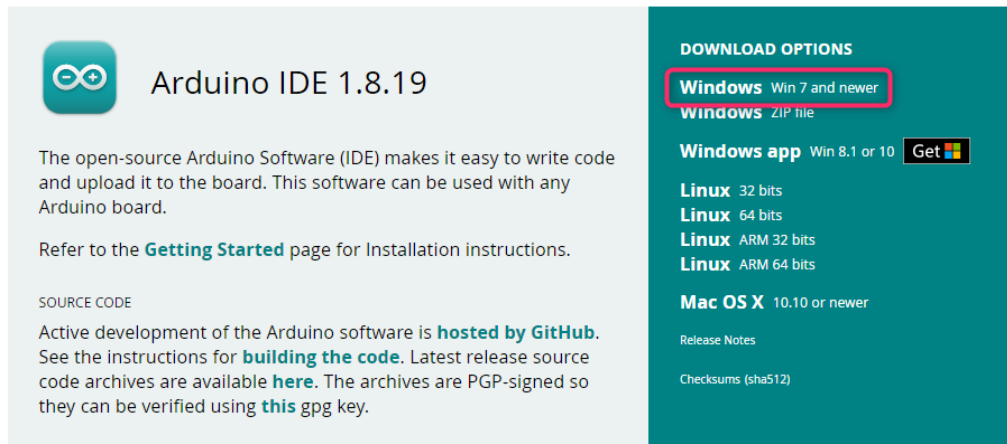
Programação do microcontrolador

Preparando a interface Arduino

Para a programação dos microcontroladores será utilizada a interface de desenvolvimento (IDE) Arduino, que pode ser obtida através do seguinte link:

- <https://www.arduino.cc/en/software>

Downloads



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 [Get](#)
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

Programando

Para programar o microcontrolador é necessário uma conexão USB entre o dispositivo e o computador. Note que a programação pode ser feita antes de montar o sistema, facilitando assim o acesso ao conector USB da placa.

Baixando o código

- Abrir o site: <https://github.com/LeoPersike/ProjetoAcionamentoLampadasRF>
- Faça o download dos códigos clicando em “Code”->”Download Zip”
- Em seguida extraia o arquivo e abra a pasta “acionadorRF”
- Abra o software “Arduino” ou dê dois cliques no arquivo “acionadorRF.ino”

Programando o Arduino

- Conecte o Arduino com o cabo USB no PC
- Selecionar a placa arduino Nano: “Tools” -> “Board: xxx” -> Arduino Nano
- Selecionar a porta em que o dispositivo está conectado: “Tools” -> “Port” -> “COM X”
- Clicar em “Sketch” -> “Upload”
- A seguinte mensagem deverá aparecer:

Done uploading.

Sketch uses 4538 bytes (14%) of program storage space. Maximum is 30720 bytes.

Global variables use 465 bytes (22%) of dynamic memory, leaving 1583 bytes for local variables. Maximum is 2048 bytes.

Preparação do sistema

Inserindo código no relé

Caso o relé forneça a opção de gravar códigos de novos controles, é possível fazer a gravação do código que já se encontra inserido no arduino. Para isto, ligue apenas o sistema “Sensor de barreira - Receptor”, o qual irá ficar simulando o aperto dos botões do controle. Em seguida, realize o procedimento de gravação de novo código no relé. No caso do relé utilizado neste projeto, as instruções são:

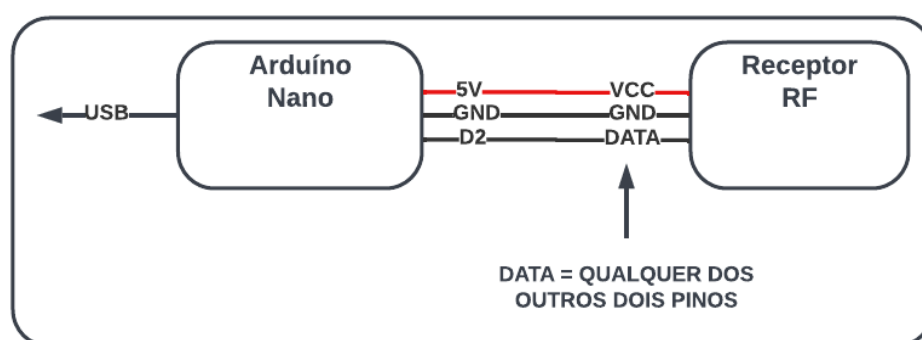
- Pressione o push-button presente na placa até que o led comece a piscar
- Pressione uma tecla do controle remoto
- O led irá piscar 2 vezes, indicando que a tecla foi gravada corretamente e está associada à 1 dos relés da placa
- Refaça o procedimento para as outras teclas do controle remoto
- Para apagar todos os códigos gravados, pressione o push-button por 8 segundos até que o led se apague

Se for utilizado um modelo diferente do apresentado neste documento, procure a informação de como gravar novos códigos no manual do mesmo.

Clonando código do controle no Arduino

Importante: apenas realize esse procedimento caso o relé não aceite gravação de novos códigos! Caso contrário, o arduino já possui um código para ser utilizado.

Caso o relé não ofereça possibilidade de gravação de novos códigos, é possível clonar o controle que já está cadastrado utilizando o arduino. Para isto, você deverá montar o seguinte sistema:



Em seguida, você deverá gravar o arduino com o seguinte arquivo:

- Abrir o site: <https://github.com/LeoPersike/ProjetoAcionamentoLampadasRF>
- Faça o download dos códigos clicando em “Code”->”Download Zip”
- Em seguida extraia o arquivo e abra a pasta “cloneRF”
- Abra o software “Arduino” ou dê dois cliques no arquivo “cloneRF.ino”
- Prosiga com a programação conforme explicado no capítulo anterior

- Em seguida você deverá abrir o monitor serial através do menu “Tools” -> “Serial Monitor”
- Aproxime o controle do módulo RF e aperte o primeiro botão. O código e parâmetros do primeiro deverão aparecer na tela
- Em seguida, aperte o segundo botão
- A imagem a seguir demonstra o resultado esperado:

COM10

```

Decimal: 1452321 (24Bit) Binary: 000101100010100100100001 Tri-State: not applicable PulseLength: 286 microseconds Protocol: 1
Raw data: 8892,276,872,276,880,272,872,860,300,268,876,856,300,848,308,280,868,276,880,276,868,852,304,280,880,844,308,268,880,27
Decimal: 1452322 (24Bit) Binary: 000101100010100100100010 Tri-State: not applicable PulseLength: 285 microseconds Protocol: 1
Raw data: 8896,276,860,312,852,280,868,880,268,288,872,876,276,864,292,296,864,312,840,272,868,864,284,296,872,80,236,84,836,204,

```

Neste caso os parâmetros do controle utilizado foram os seguintes:

- Botão A:
 - Código: 000101100010100100100001
 - Pulse length: 286
 - Protocol: 1
- Botão B:
 - Código: 000101100010100100100010
 - Pulse length: 286
 - Protocol: 1

Descoberta a codificação do controle, é preciso alterar o código do arduino para simular o mesmo. Para isto, abra o arquivo “acionadorRF.ino” e altere as seguintes linhas conforme os parâmetros que foram encontrados:

```

/* Codificacao do controle */
#define BOTAO_1 "100101001110111110100001"
#define BOTAO_2 "100101001110111110100010"
#define PULSE_LENGTH 374
#define PROTOCOL 1

```

Pronto. Agora basta você programar o arduino com a nova codificação e o arduino irá simular o controle que já está cadastrado no relé.