



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**ESCOLA DE MINAS**  
**DEPARTAMENTO DO CURSO DE ENGENHARIA**  
**DE CONTROLE E AUTOMAÇÃO - CEAU**



**FILIPPE HENRIQUE MOREIRA GUERRA**

**AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM PROTOCOLO X10 E  
ESP8266**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

**Ouro Preto, 2016**

FILIPÉ HENRIQUE MOREIRA GUERRA

**AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM PROTOCOLO X10 E  
ESP8266**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Escola de Minas da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Mestre Ronaldo Silva

Trindade

Ouro Preto

Escola de Minas – UFOP

Abril/2016

G934a Guerra, Filipe H. Moreira.

Automação residencial de baixo custo com protocolo X10 e ESP8266 [manuscrito] / Filipe H. Moreira Guerra. - 2016.

57f.: il.: color; grafs; tabs.

Orientador: Prof. Dr. Ronaldo S Trindade.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Dispositivo. 2. Automação residencial. 3. Wireless. I. Trindade, Ronaldo S. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 681.5

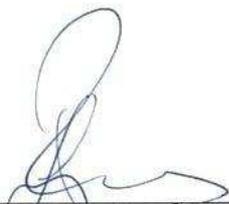
Catálogo: [ficha@sisbin.ufop.br](mailto:ficha@sisbin.ufop.br)

Monografia defendida e aprovada, em 02 de junho de 2016, pela comissão avaliadora constituída pelos professores:



---

Prof. M. Sc. Ronaldo Silva Trindade - Orientador



---

Prof. Dr. Luiz Fernando Rispoli Alves – Professor Convidado



---

Prof. Dr. Henor Artur de Souza – Professor Convidado

## RESUMO

Ao longo do desenvolvimento humano a qualidade de vida tem aumentado muito devido ao desenvolvimento tecnológico. Nesse contexto, a aplicação automação residencial ainda deixa a desejar em relação ao avanço tecnológico já alcançado em uma residência tradicional no Brasil. Se em outras áreas tecnológicas as pessoas utilizam tecnologia de ponta, em automação residencial esses sistemas são pouco presentes, se limitando a poucos utensílios sem integração entre si e com custo relativamente alto. Neste trabalho apresenta-se uma alternativa de automação residencial, utilizando sistema de baixo custo, com comunicação realizada via rede elétrica, com o auxílio do Protocolo X10, e utilizando o ESP8266, que realiza comunicação wireless. Para isso, foram desenvolvidos projetos para a implementação do Protocolo X10, construindo toda sua estrutura de comunicação e envios de dados pela rede elétrica, e para aplicação do ESP8266 em automação residencial. Os resultados foram satisfatórios, no entanto o protocolo X10 apresentou instabilidade na aplicação devido às interferências de fontes de alimentação chaveadas presentes nos equipamentos modernos. Sua aplicação fica condicionada então à existência de uma rede elétrica com menos ruídos. O ESP8266 apresentou resultados satisfatórios com uma comunicação eficaz e com boa resposta.

**Palavras-chave:** Protocolo X10, automação residencial, ESP8266.

## ABSTRACT

Throughout human development, the quality of life has greatly increased due to technological development. In this context, the application of home automation still leaves to be desired in relation to technological advances already achieved in a traditional residence in Brazil. In other technological areas people use the latest technology in home automation these systems are little present, limited to few utensils without integration with each other and relatively high cost. In this paper presents an alternative to home automation, using low-cost system with communication by means of the grid, with the help of the X10 protocol, and using the ESP8266, which performs wireless communication. For this, projects has been developed for the implementation of the Protocol X10, building its entire communication structure and uploads data by the electrical network, and application of ESP8266 in home automation. The results were satisfactory, however the X10 protocol showed instability in the application due to interference of switching power supplies present in modern equipment. Your application is then dependent on the existence of an electricity grid with less noise. The ESP8266 presented satisfactory results with effective communication and responsive.

**Keywords:** X10 Protocol, home automation, ESP8266.

“Se tiver que amar, ame hoje.

Se tiver que sorrir, sorria hoje.

Se tiver que chorar, chore hoje.

Pois o importante é viver hoje.

O ontem já foi e o amanhã talvez não venha. ”

André Luis (por Chico Xavier)

## SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUÇÃO .....</b>                       | <b>10</b> |
| 1.1      | OBJETIVOS DO TRABALHO .....                   | 10        |
| 1.2      | METODOLOGIA .....                             | 11        |
| 1.3      | MOTIVAÇÃO .....                               | 12        |
| 1.4      | ESTRUTURA DO TRABALHO.....                    | 12        |
| <b>2</b> | <b>REFERENCIAL TEÓRICO .....</b>              | <b>13</b> |
| 2.1      | PROTOCOLO DE COMUNICAÇÃO.....                 | 13        |
| 2.1.1    | <i>Protocolo X10.....</i>                     | <i>13</i> |
| 2.1.2    | <i>Padrão 802.11 .....</i>                    | <i>16</i> |
| 2.2      | ALGUNS FUNDAMENTOS DA ELETRÔNICA .....        | 18        |
| 2.2.1    | <i>Amplificadores operacional .....</i>       | <i>18</i> |
| 2.2.2    | <i>Ruído.....</i>                             | <i>19</i> |
| 2.2.3    | <i>Filtros .....</i>                          | <i>19</i> |
| 2.2.4    | <i>Semicondutores.....</i>                    | <i>21</i> |
| 2.2.5    | <i>Transistores .....</i>                     | <i>21</i> |
| 2.3      | COMPONENTES UTILIZADOS.....                   | 21        |
| 2.3.1    | <i>Arduino .....</i>                          | <i>22</i> |
| 2.3.2    | <i>Timer LM555 .....</i>                      | <i>23</i> |
| 2.3.3    | <i>Optoacoplador.....</i>                     | <i>23</i> |
| 2.3.4    | <i>Relé e triac.....</i>                      | <i>24</i> |
| 2.3.5    | <i>ESP8266 .....</i>                          | <i>26</i> |
| 2.3.6    | <i>Buffer não inversor.....</i>               | <i>28</i> |
| 2.3.7    | <i>Regulador de tensão .....</i>              | <i>29</i> |
| 2.4      | REVISÃO DE PROCEDIMENTOS DE PROGRAMAÇÃO ..... | 30        |
| 2.4.1    | <i>Operadores de condição/repetição .....</i> | <i>30</i> |

|          |   |           |
|----------|---|-----------|
| 2.4.2    | <i>Estrutura if</i> .....   | 31        |
| 2.4.3    | <i>Estrutura if-else</i> .....  | 31        |
| 2.4.4    | <i>Estrutura if-else if-else</i> .....                                  | 32        |
| 2.4.5    | <i>Estrutura while</i> .....  | 34        |
| 2.4.6    | <i>Estrutura for</i> .....  | 35        |
| 2.5      | MIT APP INVENTOR .....  | 36        |
| <b>3</b> | <b>DESENVOLVIMENTO DO TRABALHO</b> .....                                | <b>37</b> |
| 3.1      | CIRCUITO PARA IMPLEMENTAÇÃO DO PROTOCOLO X10.....                       | 38        |
| 3.1.1    | <i>Detecção de passagem por zero</i> .....                              | 38        |
| 3.1.2    | <i>Implementação do circuito transmissor</i> .....                      | 40        |
| 3.1.3    | <i>Implementação do circuito receptor</i> .....                         | 42        |
| 3.2      | ESP8266 .....   | 44        |
| 3.2.1    | <i>Comunicação do ESP8266 com o arduino</i> .....                       | 44        |
| 3.2.2    | <i>Configurando ESP8266 para se conectar a um ponto de acesso</i> ..... | 45        |
| 3.2.3    | <i>Configurando o ESP8266 no roteador com IP fixo</i> .....             | 46        |
| 3.2.4    | <i>Programação para Android com MIT App Inventor</i> .....              | 46        |
| <b>4</b> | <b>RESULTADOS E ANÁLISES</b> .....                                      | <b>49</b> |
| 4.1      | PROTOCOLO X10 .....   | 49        |
| 4.2      | ESP8266 .....   | 51        |
| <b>5</b> | <b>CONSIDERAÇÕES FINAIS</b> .....                                       | <b>53</b> |
|          | <b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....                                 | <b>54</b> |

## LISTA DE SIGLAS E ABREVIATURAS

|        |   |
|--------|---|
| Amp-op | Amplificador operacional  |
| Bps    | Bits por segundo  |
| DHCP   | Dynamic Host Configuration Protocol   |
| IEEE   | Maior organização técnica profissional do mundo dedicada a tecnologia avançada.         |
| IP     | Internet Protocol   |
| Led    | Light emitting diode  |
| Pwm    | Pulse width modulation  |
| Wifi   | Tecnologia de rede que permite que equipamentos se comuniquem através de sinal sem fio. |

## LISTA DE FIGURAS

|   |    |
|---|----|
| FIGURA 1 - TRANSMISSÕES SINCRONIZADAS COM PASSAGEM POR ZERO. ....   | 14 |
| FIGURA 2 - TRANSMISSÕES SINCRONIZADAS COM PASSAGEM POR ZERO EM REDE<br>TRIFÁSICA.....                                   | 14 |
| FIGURA 3 - REPRESENTAÇÃO DE UMA TRANSMISSÃO NO PADRÃO X-10 .....  | 15 |
| FIGURA 4 - CÓDIGOS DE COMANDOS NO PADRÃO X-10 .....   | 16 |
| FIGURA 5 - SOBREPOSIÇÃO DOS CANAIS DO PADRÃO IEEE 802.11G.....  | 17 |
| FIGURA 6 - ESQUEMA SIMPLES DE UM AMP-OP.....  | 18 |
| FIGURA 7 - AMP-OP INVERSOR.....   | 18 |
| FIGURA 8 - AMP-OP NÃO INVERSOR. ....  | 19 |
| FIGURA 9 - FITRO R-C PASSA-BAIXA .....  | 20 |
| FIGURA 10 - FITRO R-C PASSA-ALTA.....   | 20 |
| FIGURA 11 -DIAGRAMA DE BLOCOS DO LM555.....   | 23 |
| FIGURA 12 - CIRCUITO OPTOACOPLADOR .....  | 23 |
| FIGURA 13 - ESTRUTURA SIMPLIFICADA DE UM RELÉ. ....   | 24 |
| FIGURA 14 - COM A ATRAÇÃO, OS CONTATOS SE TOCAM. ASSIM, A CORRENTE PODE<br>PASSAR PELO CIRCUITO CONTROLADO. ....        | 24 |
| FIGURA 15 – TIRISTORES .....  | 25 |
| FIGURA 16 – FUNCIONAMENTO DO TRIAC.....   | 26 |
| FIGURA 17 - ESP8266-01 .....  | 26 |
| FIGURA 18 - ESQUEMA DO ESP8266 .....  | 27 |
| FIGURA 19 - BUFFER NÃO INVERSOR HEXADECIMAL CD4050BC.....   | 29 |
| FIGURA 20 - VISTA SUPERIOR DO REGULADOR DE TENSÃO AMS1117 SOT-223 .....   | 29 |
| FIGURA 21 - FLUXOGRAMA DA FUNÇÃO IF.....  | 31 |
| FIGURA 22 - FLUXOGRAMA DA FUNÇÃO IF-ELSE.....   | 32 |
| FIGURA 23 - FLUXOGRAMA DA FUNÇÃO IF-ELSEIF-ELSE. ....   | 34 |
| FIGURA 24 - FLUXOGRAMA DA FUNÇÃO WHILE. ....  | 35 |
| FIGURA 25 - FLUXOGRAMA DA FUNÇÃO FOR. ....  | 36 |
| FIGURA 26 - CIRCUITO DE ACIONAMENTO DE UMA LÂMPADA.....   | 37 |
| FIGURA 27 - CIRCUITO DETECTOR DE PASSAGEM POR ZERO.....   | 38 |
| FIGURA 28 – DETECÇÃO E O ACIONAMENTO DO ENVIO DOS BITS DE COMUNICAÇÃO NO<br>MOMENTO EM QUE ACONTECE A INTERRUPÇÃO. .... | 39 |
| FIGURA 29 – SINAL DE ALTA FREQUÊNCIA GERADO NO LM555 INDICADO PELA SETA.....  | 40 |

|   |    |
|---|----|
| FIGURA 30 - ENVIO DO BIT "1".....   | 41 |
| FIGURA 31 - CIRCUITO TRANSMISSOR.....   | 41 |
| FIGURA 32 - FILTRO DO CIRCUITO RECEPTOR.....  | 42 |
| FIGURA 33 – SAÍDA DO PRIMEIRO FILTRO INDICADO PELA SETA. ....   | 43 |
| FIGURA 34 – SAÍDA DO SEGUNDO FILTRO INDICADO PELA SETA. ....  | 43 |
| FIGURA 35 – SAÍDA DO TERCEIRO FILTRO INDICADO PELA SETA. ....   | 44 |
| FIGURA 36 - SAÍDA DOS AMPLIFICADORES INDICADO PELA SETA.....  | 44 |
| FIGURA 37 - CONFIGURAÇÃO DE UM ENDEREÇO DE IP PARA UM ESP8266 NO ROTEADOR<br>TP-LINK MODELO NO TL-WR340G/TL-WR340GD ..... | 46 |
| FIGURA 38 - INTERFACE DO MIT APP INVENTOR - DESIGNER .....  | 47 |
| FIGURA 39 - INTERFACE DO MIT APP INVENTOR - BLOCKS.....   | 47 |
| FIGURA 40 - PROGRAMAÇÃO DA LÓGICA EXECUTADA APÓS CLICAR NUM BOTÃO NO<br>APLICATIVO ANDROID .....                          | 48 |
| FIGURA 41 – MONTAGEM DOS CIRCUITOS DE COMUNICAÇÃO POR PROTOCOLO X10.....  | 49 |
| FIGURA 42 – FLUXOGRAMA DA FUNÇÃO DE ENVIO DE BITS PELA REDE ELÉTRICA. ....  | 50 |
| FIGURA 43 - MONTAGEM DOS CIRCUITOS DE COMUNICAÇÃO COM ESP8266.....  | 51 |
| FIGURA 44 – FLUXOGRAMA DA PROGRAMAÇÃO DO ARDUÍNO .....  | 52 |

## LISTA DE TABELAS

|   |    |
|---|----|
| TABELA 1 - CARACTERÍSTICAS PRINCIPAIS DOS PADRÕES DE WLAN.....                      | 17 |
| TABELA 2 - ALGUNS EXEMPLOS DE COMANDOS DO ESP8266, SUAS FUNÇÕES E<br>RESPOSTAS..... | 28 |
| TABELA 3 - OPERADORES DE COMPARAÇÃO.....  | 30 |
| TABELA 4- OPERADORES DE RELACIONAIS.....  | 30 |

# 1 INTRODUÇÃO

Ao longo de sua evolução, o homem sempre buscou melhorar sua condição de vida, seja por meio de praticidade, conforto, segurança e até mesmo no gerenciamento de seus recursos. Uma ferramenta atual que proporciona uma melhora nessas condições é a criação de casas inteligentes.

Casas inteligentes são construções com aplicações e sistemas eletrônicos integrados, capaz de oferecer controle remoto e/ou centralizado de funcionalidades e serviços (JUNESTRAND, 2004). Uma das tecnologias amplamente usadas por sistema de controle residencial para conectar o controlador da casa com os aparelhos, equipamentos e dispositivos, é o protocolo X10, que usa linhas de energia para transmissão de dados (CHUNDURU; SUBRAMANIAN, 2006).

O protocolo X10 transmite informações através da rede elétrica, enviando dados digitais entre dispositivos que utilizam este protocolo. Esta informação digital é codificada numa frequência de 120 kHz, que é transmitida durante os cruzamentos de zero das frequências de 50 ou 60Hz da rede de corrente alternada. Um bit é transmitido em cada cruzamento de zero. Os dados digitais consistem em um endereço e um comando enviado a partir de um controlador para um dispositivo controlado. Controladores podem consultar dispositivos para responder com seu status. Esse status pode ser tão simples como "on" ou "off", ou como o nível de intensidade de corrente, até mesmo a temperatura ou leitura de um sensor (CHUNDURU; SUBRAMANIAN, 2006).

Uma outra maneira prática realizar comunicação de dispositivos é utilizando uma rede de internet wireless. Vários dispositivos conectados em um ponto de acesso, se corretamente configurados, podem trocar informações pela rede local e até mesmo, pela internet. Um desses dispositivos é o ESP8266, que é um módulo Wifi de baixo custo, podendo ser encontrado com custo inferior a 2 dólares, que trabalha tanto como uma estação, que se conecta a um ponto de acesso, como ponto de acesso.

## 1.1 Objetivos do trabalho

Este trabalho tem como objetivo a construção de um sistema de automação residencial de baixo custo, sem a necessidade de instalação de uma nova infraestrutura física,

comunicando dois ou mais dispositivos utilizando o protocolo X10 e o módulo Wifi ESP8266.

## **1.2 Metodologia**

Para aplicação do protocolo X10 devem ser realizadas as seguintes etapas:

- 1 - Realizar a detecção da passagem pelo zero da onda da corrente AC em uma residência.
- 2 - Implementar um circuito oscilador de 120kHz para realizar a comunicação na rede.
- 3 - Programar o circuito transmissor para o envio das informações desejadas.
- 4 - Implementar o circuito receptor para a recepção do sinal de 120kHz, bem como a execução do comando desejado.
- 5 - Realizar testes práticos do sistema criado enviando comandos para desligar e ligar dispositivos em rede elétrica doméstica.
- 6 - Analisar os resultados.

Para a criação de um sistema de automação utilizando o ESP8266 devem ser realizadas as seguintes etapas:

- 1 - Realizar a comunicação do ESP8266 com o arduíno.
- 2 - Configurar o ESP8266, o conectando a um ponto de acesso.
- 3 - Programar o arduíno para interpretar os comandos recebidos através do ESP8266.
- 4 - Programar um dispositivo para enviar comandos para o arduíno através da rede wifi doméstica.
- 5 - Realizar testes práticos do sistema criado enviando comandos para desligar e ligar dispositivos em um rede wifi doméstica.
- 6 - Analisar os resultados.

### **1.3 Motivação**

Implementar um sistema de automação residencial de baixo custo, com o auxílio do protocolo X10 e do ESP8266, gerando benefícios como: conforto para o usuário, praticidade, segurança e economia.

### **1.4 Estrutura do trabalho**

Este trabalho foi dividido em cinco capítulos. O primeiro apresenta um contexto introdutório sobre o trabalho. No capítulo 2 apresentam-se referenciais teóricos gerais sobre os protocolos de comunicação utilizados, alguns fundamentos da eletrônica, componentes utilizados para construção dos circuitos do projeto e uma revisão de programação. No capítulo 3 é apresentado o desenvolvimento do trabalho, explicando como foram realizadas as construções dos circuitos, suas programações e configurações. No capítulo 4, são apresentadas as simulações e os resultados obtidos. Por fim, no último capítulo são tratadas as considerações finais.

## **2 REFERENCIAL TEÓRICO**

### **2.1 Protocolo de comunicação**

Protocolo são regras ou padrões cuja finalidade é permitir que os computadores se interliguem e troquem informações com o menor número de erros possível (ROSETTO, 1997).

O protocolo de comunicação pode ser comparado a um idioma. Duas pessoas que falam o mesmo idioma conseguem se comunicar com sucesso, mas para isso elas existem condições que precisam ser satisfeitas para possibilitar que isso aconteça. Se a distância entre elas for muito grande a ponto que o som da voz de uma não chegue à outra, ou se o ambiente tiver com muito ruído se sobrepondo à voz na comunicação, são exemplos de situações que podem atrapalhar a comunicação. Por isso, o protocolo também pode determinar o meio físico e as condições para a realização dessa comunicação, como por exemplo uma comunicação por sinais, onde duas pessoas se comunicam utilizando gestos, ambas as pessoas têm que ter um entendimento comum do significado dos sinais e elas tem que estar a uma distância onde possam observar os gestos das outras, para que a comunicação aconteça com sucesso.

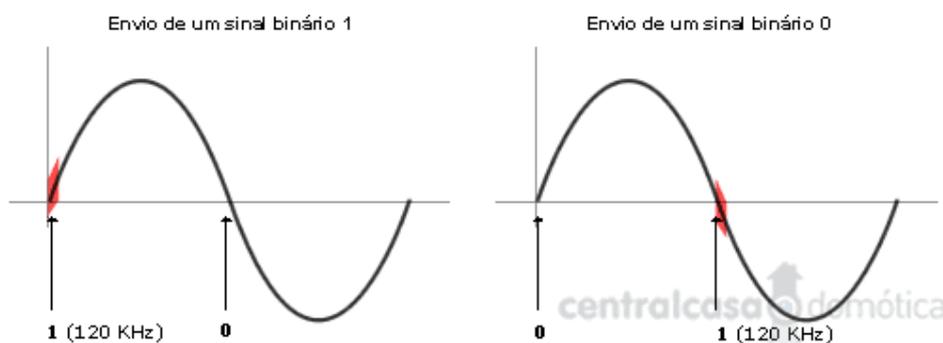
#### **2.1.1 Protocolo X10**

O protocolo X10 foi desenvolvido na década 70 (1976-1978), pela empresa escocesa, Pico Electronics em Glenrothes. O seu nome deve-se ao fato deste sistema domótico ter sido o décimo projeto que a Pico Electronics desenvolveu (BURROUGHS, 2002). É uma tecnologia usada para realizar o controle de um sistema residencial. Ele realiza a comunicação entre o transmissor e o receptor, enviando e recebendo sinais na rede elétrica residencial. Este protocolo tem muitas vantagens, incluindo ser barato, não é necessário a instalação de nova fiação, simples de instalar, compatível com muitos produtos, controla até 256 dispositivos (CHUNDURU; SUBRAMANIAN, 2006).

A patente do protocolo foi posteriormente adquirida pela empresa X10 Ltd e mantida até ter expirado em 1997. O X10 é atualmente um protocolo aberto, sendo várias as empresas que comercializam produtos baseados nesta tecnologia. A enorme variedade de módulos e interfaces X10, a utilização da rede elétrica existente como meio de transmissão, os

preços baixos e a facilidade na instalação (a maior parte dos módulos são Plug&Play) contribuíram para o rápido sucesso do X10 nos EUA e Europa (EUROX10..., 2015).

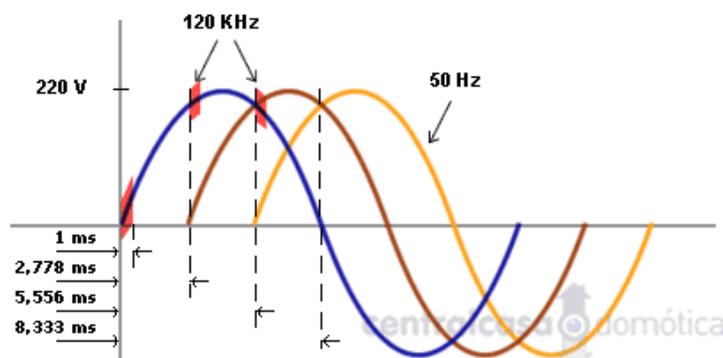
Segundo Chundururu e Subramanian (2006), na comunicação entre dispositivos X10 a informação digital é codificada numa frequência de 120 kHz e é transmitida durante o cruzamento pelo zero da onda de corrente alternada, 50 ou 60Hz, existente numa residência. Um bit é transmitido em cada cruzamento de zero, como pode ser observado na Figura 1.



**Figura 1 - Transmissões sincronizadas com passagem por zero.**

**Fonte: EuroX10..., 2015.**

Para permitir o uso em instalações elétricas trifásicas, os sinais de 120kHz são emitidos três vezes em cada ciclo, em instantes que coincidem com a passagem por zero da tensão de cada uma das fases. Deste modo, e recorrendo a acopladores próprios, torna-se possível comunicar com qualquer dispositivo, independentemente da fase em que esteja instalado. As curvas são mostradas na Figura 2.



**Figura 2 - Transmissões sincronizadas com passagem por zero em rede trifásica.**

**Fonte: EuroX10..., 2015.**

Dado que o meio de distribuição de energia é muito ruidoso eletricamente, foi adotada uma política em que um bit nunca é enviado isoladamente, sendo sempre enviado o bit e o seu complemento. Na prática isto significa que, sempre que se pretende enviar o bit 1, isso corresponde a enviar um 1 (sinal de 120kHz na origem) seguido de um 0 (ausência de sinal). O envio do bit 0 corresponde a enviar um 0 (ausência de sinal) seguido de um 1 (frequência de 120kHz na origem). Este cuidado visa minimizar a probabilidade do ruído elétrico poder ser confundido com um sinal válido. Contudo, tem como aspecto negativo reduzir o ritmo de transmissão que fica assim restrito a uns meros 60 bps (é enviado um bit por cada ciclo da rede elétrica) (EUROX10, 2015).

Uma mensagem completa do protocolo X10 é composta de um código de início (1110), seguido por um código de casa que, por sua vez, é seguido por um código chave. O código chave pode ser um endereço de um dispositivo ou um código de função (BURROUGHS, 2002), Figura 3.

| Initiation Signal   |   |   |   |                       |   |   |   |                        |   |   |   |   |
|---------------------|---|---|---|-----------------------|---|---|---|------------------------|---|---|---|---|
| 1                   | 1 | 1 | 0 | 0                     | 1 | 1 | 0 | 1                      | 1 | 1 | 0 | 0 |
| <i>Start Signal</i> |   |   |   | <i>House Code "A"</i> |   |   |   | <i>Device Code "2"</i> |   |   |   |   |

| Command Codes |                       |                |
|---------------|-----------------------|----------------|
| On = 00101    | All Lights On = 00011 | Bright = 01011 |
| Off = 00111   | All Units Off = 00001 | Dim = 01001    |

**Figura 3 - Representação de uma transmissão no padrão X-10**

**Fonte: CHUNDURU; SUBRAMANIAN, 2006.**

O Protocolo X10 especifica um total de 256 endereços diferentes: 16 códigos de unidade (1-16) para cada um dos 16 códigos de casa (A-P), como pode ser observado na Figura 4. Normalmente um transmissor é ajustado para um certo código de casa e então pode controlar até 16 códigos de unidade. Não há restrição de se usar múltiplos transmissores, cada um ajustado para um código diferente de casa, na mesma fiação. Permite-se, ainda, que os receptores sejam ajustados para o mesmo código de casa e o mesmo código de unidade, o que possibilita que um mesmo comando emitido por um transmissor X-10 possa controlar múltiplos receptores em paralelo (BURROUGHS, 2002).

| House Codes |    |    |    |    | Unit/Function Codes |    |    |    |    |   |
|-------------|----|----|----|----|---------------------|----|----|----|----|---|
|             | H8 | H4 | H2 | H1 |                     | D8 | D4 | D2 | D1 | F |
| A           | 0  | 1  | 1  | 0  | 1                   | 0  | 1  | 1  | 0  | 0 |
| B           | 1  | 1  | 1  | 0  | 2                   | 1  | 1  | 1  | 0  | 0 |
| C           | 0  | 0  | 1  | 0  | 3                   | 0  | 0  | 1  | 0  | 0 |
| D           | 1  | 0  | 1  | 0  | 4                   | 1  | 0  | 1  | 0  | 0 |
| E           | 0  | 0  | 0  | 1  | 5                   | 0  | 0  | 0  | 1  | 0 |
| F           | 1  | 0  | 0  | 1  | 6                   | 1  | 0  | 0  | 1  | 0 |
| G           | 0  | 1  | 0  | 1  | 7                   | 0  | 1  | 0  | 1  | 0 |
| H           | 1  | 1  | 0  | 1  | 8                   | 1  | 1  | 0  | 1  | 0 |
| I           | 0  | 1  | 1  | 1  | 9                   | 0  | 1  | 1  | 1  | 0 |
| J           | 1  | 1  | 1  | 1  | 10                  | 1  | 1  | 1  | 1  | 0 |
| K           | 0  | 0  | 1  | 1  | 11                  | 0  | 0  | 1  | 1  | 0 |
| L           | 1  | 0  | 1  | 1  | 12                  | 1  | 0  | 1  | 1  | 0 |
| M           | 0  | 0  | 0  | 0  | 13                  | 0  | 0  | 0  | 0  | 0 |
| N           | 1  | 0  | 0  | 0  | 14                  | 1  | 0  | 0  | 0  | 0 |
| O           | 0  | 1  | 0  | 0  | 15                  | 0  | 1  | 0  | 0  | 0 |
| P           | 1  | 1  | 0  | 0  | 16                  | 1  | 1  | 0  | 0  | 0 |
|             |    |    |    |    | All Units Off       | 0  | 0  | 0  | 0  | 1 |
|             |    |    |    |    | All Units On        | 0  | 0  | 0  | 1  | 1 |
|             |    |    |    |    | On                  | 0  | 0  | 1  | 0  | 1 |
|             |    |    |    |    | Off                 | 0  | 0  | 1  | 1  | 1 |
|             |    |    |    |    | Dim                 | 0  | 1  | 0  | 0  | 1 |
|             |    |    |    |    | Bright              | 0  | 1  | 0  | 1  | 1 |
|             |    |    |    |    | All Lights Off      | 0  | 1  | 1  | 0  | 1 |
|             |    |    |    |    | Extended Code       | 0  | 1  | 1  | 1  | 1 |
|             |    |    |    |    | Hail Request        | 1  | 0  | 0  | 0  | 1 |
|             |    |    |    |    | Hail Acknowledge    | 1  | 0  | 0  | 1  | 1 |
|             |    |    |    |    | Pre-Set Dim         | 1  | 0  | 1  | X  | 1 |
|             |    |    |    |    | Extended Data       | 1  | 1  | 0  | 0  | 1 |
|             |    |    |    |    | Status is On        | 1  | 1  | 0  | 1  | 1 |
|             |    |    |    |    | Status is Off       | 1  | 1  | 1  | 0  | 1 |
|             |    |    |    |    | Status Request      | 1  | 1  | 1  | 1  | 1 |

Figura 4 - Códigos de comandos no padrão X-10

Fonte: BURROUGHS, 2002.

### 2.1.2 Padrão 802.11

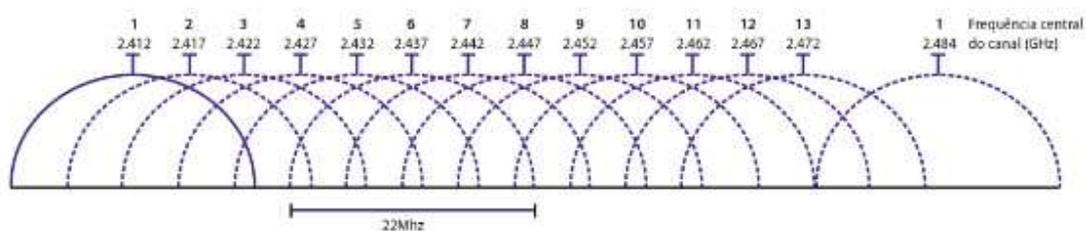
Os padrões 802.11a, 802.11b, 802.11g, 802.11n são padrões de rede wireless estabelecido pelo IEEE de fácil implementação e uso. O primeiro padrão citado é pouco utilizado, e os dispositivos modernos costumam opera nos dois últimos. Esses padrões apresentam algumas diferenças, como por exemplo, velocidade máxima de transmissão. Algumas dessas diferenças são apresentadas na Tabela 1. Os dispositivos mais modernos costumam operar nos padrões 802.11b ou 802.11n, que são mais rápidos. (LOUREIRO et al., 2014).

**Tabela 1 - Características principais dos padrões de WLAN.**

|                                    | <b>802.11a</b> | <b>802.11b</b>                       | <b>802.11g</b>        | <b>802.11n</b>   |
|------------------------------------|----------------|--------------------------------------|-----------------------|--|
| <b>Velocidade máxima</b>           | 54Mbps         | 11Mbps                               | 54Mbps                | 150-600Mbps  |
| <b>Quantidade de canais</b>        | 23             | Mundo - 13<br>EUA - 11<br>Japão - 14 | Mundo -13<br>EUA - 11 | Mundo - 13 (2,4GHz - 20MHz)<br>EUA - 11 (2,4GHz - 20MHz)<br>24 (5GHz - 20MHz)<br>12 (2,4GHz - 20MHz) |
| <b>Canais que não se sobrepõem</b> | -              | 1, 6, 11, 14                         | 1, 6, 11              | 1, 6, 11 (2,4GHz - 20MHz)  |
| <b>Sinalização</b>                 | OFDM           | HR-DSS                               | OFDM                  | MIMO-OFDM  |

**Fonte: Adaptado de LOUREIRO et al., 2014.**

As WLANs costumam alcançar distâncias de 35 a 140 metros, dependendo da quantidade de paredes entre os pontos. As redes que operam na frequência de 2,4GHz sofrem interferência de fornos de micro-ondas, telefones sem fio e equipamentos com tecnologia bluetooth. A frequência de 5GHz apresenta maior dificuldade no que diz respeito a distâncias (LOUREIRO et al., 2014). A sobreposição de canais podem ser observadas na Figura 5.



**Figura 5 - Sobreposição dos canais do padrão IEEE 802.11g.**

**Fonte: LOUREIRO et al., 2014.**

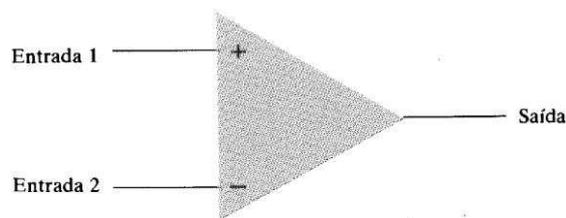
Os dispositivos rede possuem um endereço único chamado MAC address (Media Access Control Address), que é composto por um número de 48 bits. Os 24 bits iniciais podem ser usados para identificação do fabricante da placa de rede. O MAC é hexadecimal, ou seja, é representado por 16 símbolos, sendo eles números de 0 a 9 e letras de A a F (CUNCHE, 2013).

## 2.2 Alguns fundamentos da eletrônica

Para o desenvolvimento dos circuitos foram necessários alguns conceitos de eletrônica como amplificadores operacionais, ruído, filtros, optoacoplador, relé, triac, entre outros que serão exibidos adiante.

### 2.2.1 Amplificadores operacional

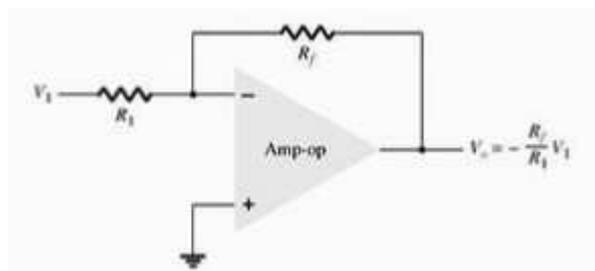
Um amplificador operacional (amp-op) é um amplificador diferencial de ganho muito alto com impedância de entrada muito alta e impedância de saída baixa. É comum o amplificador operacional ser utilizado para a obtenção de variações na tensão (amplitude e polaridade), em osciladores, filtros e diversos tipos de circuitos de instrumentação. (BOYLESTAD; NASHELSKY, 1998). A simbologia de um amplificador operacional pode ser vista na Figura 6.



**Figura 6 - Esquema simples de um amp-op.**

**Fonte: BOYLESTAD; NASHELSKY, 1998.**

Um circuito amplificador operacional básico, Figura 7, opera como um multiplicador de ganho constante. Um sinal de entrada  $V_i$  é aplicado através do resistor  $R_1$  à entrada negativa. A saída é, então conectada de volta à entrada negativa através do resistor  $R_f$ . A entrada positiva é conectada em GND.



**Figura 7 - Amp-op inversor.**

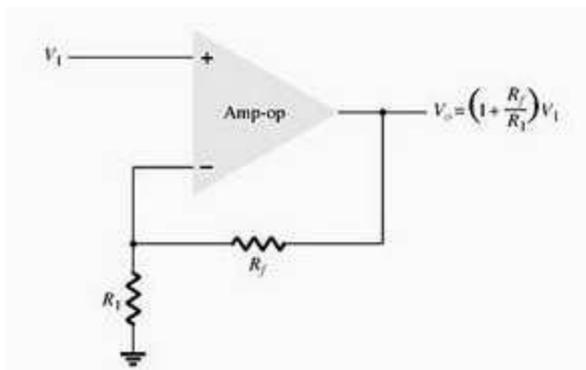
**Fonte: BOYLESTAD; NASHELSKY, 1998.**

Como o sinal  $V_i$  é aplicado exclusivamente à entrada negativa, a saída resultante é oposta em fase ao sinal de entrada, ou seja, esse modo de operação é chamado de amplificador inversor, o valor de sua saída pode ser calculado pela equação:

$$V_o = - \frac{R_f}{R_1} V_i \quad (\text{Equação 1})$$

Já o amplificador não inversor ou multiplicador de ganho constante, Figura 8, seu ganho é dado pela equação:

$$\frac{V_o}{V_i} = \frac{R_1 + R_f}{R_1} \quad (\text{Equação 2})$$



**Figura 8 - Amp-op não inversor.**

**Fonte: BOYLESTAD; NASHELSKY, 1998.**

### 2.2.2 Ruído

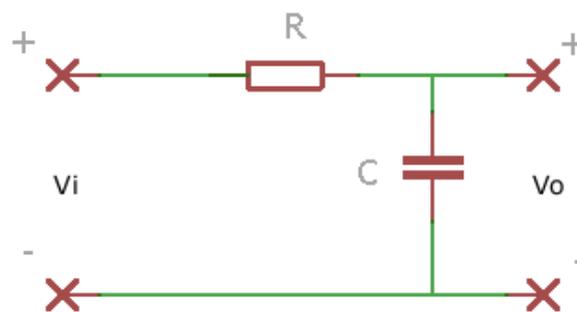
Ruído é definido como qualquer tipo de sinal indesejado que não provém do sinal de entrada nem está harmonicamente relacionado a ele (MALVINO; BATES, 1987). O ruído deve ser evitado no projeto pois ele pode interferir na comunicação, causando uma comunicação incompreensível ou até mesmo distorcida.

### 2.2.3 Filtros

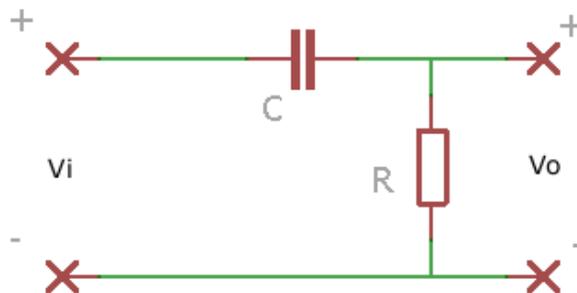
Qualquer combinação de dispositivos passivos (resistor, indutor e capacitor) e/ou ativos (transistores e amplificadores operacionais) projetada para selecionar ou rejeitar uma faixa de frequências é denominada filtro. Nos sistemas de comunicações, os filtros são usados para deixar passar as frequências que contêm as informações desejadas e rejeitar as frequências restantes. Os filtros também são usados para eliminar frequências

indesejáveis, denominadas ruídos, gerados por alguns componentes eletrônicos com características não lineares ou captadas do ambiente. Em geral, os filtros podem ser classificados em quatro grandes categorias: passa-baixas, passa-altas, passa-faixa e banda de atenuação.

O filtro R-C, resistor-capacitor, é extremamente simples, pode ser usado como filtro passa-baixa ou como filtro passa-alta. Se a saída for obtida a partir do capacitor ( $V_o$ ), Figura 9, o circuito se comportará como passa-baixa. Caso a saída seja obtida a partir do resistor ( $V_o$ ), Figura 10, o circuito se comportará como passa-alta. As frequências incluídas na banda de passagem passam para o estágio seguinte do filtro com pelo menos 70,7 por cento da tensão de entrada (BOYLESTAD, 2012).



**Figura 9 - Filtro R-C passa-baixa**



**Figura 10 - Filtro R-C passa-alta**

A frequência de corte para os filtros passa-baixa e passa-alta pode ser obtida pela equação:

$$f_c = \frac{1}{2\pi RC} \quad (\text{Equação 3})$$

Caso o filtro for passa-baixa, todas as frequências abaixo da frequência de corte conseguem transpor o filtro. Já no caso do filtro ser passa-alta, de maneira inversa ao passa-baixa, somente as frequências acima da frequência de corte conseguem passar de sua entrada  $V_i$  para sua saída  $V_o$ .

#### **2.2.4 Semicondutores**

O semicondutor é um elemento com propriedades elétricas entre as do condutor e as do isolante. Um semicondutor pode ser dopado para ter um excesso de elétrons livres ou excesso de lacunas, criando-se assim, respectivamente, semicondutores do tipo *n* (negativo) do tipo *p* (positivo) (MALVINO; BATES, 1987).

#### **2.2.5 Transistores**

O transistor pode ser usado para várias funções, como por exemplo, amplificadores ou interruptores de sinal. O transistor tem três regiões dopadas: coletor, base e emissor, podendo ser do tipo *pnp* e *nnp*. A relação de suas correntes é dada por:

$$I_E = I_C + I_B \quad (\text{Equação 4})$$

Onde:

$I_E$  é a corrente do emissor;

$I_C$  é a corrente do coletor;

$I_B$  é a corrente da base.

O transistor tem como principal vantagem o ganho de corrente ( $\beta$ ). Esse ganho é dado pela equação:

$$\beta_{CC} = \frac{I_C}{I_B} \quad (\text{Equação 5})$$

Onde:

$\beta_{CC}$  é a razão da corrente contínua do coletor para a corrente contínua da base;

$I_C$  é a corrente do coletor;

$I_B$  é a corrente de base.

O valor de  $\beta$  é tipicamente de 100 a 300, ou seja, a corrente do emissor é a maior das três correntes e a corrente de base é muito menor (MALVINO; BATES, 1987).

### **2.3 Componentes utilizados**

Para o desenvolvimento dos circuitos foram necessários alguns componentes eletrônicos específicos. Para utilização adequada destes é necessário que se faça um estudo sobre suas características e o conhecimento de seus parâmetros descritos em documentos técnicos.

### 2.3.1 Arduino

Arduino é um protótipo de uma plataforma eletrônica de código aberto, construída para ser flexível, fácil de usar em nível de hardware e software. Ele pode interagir com o ambiente ao receber em suas entradas uma variedade de sensores e atuar em suas saídas em uma variedade de atuadores. Projetos no Arduino podem ser independentes ou podem comunicar com software rodando em um computador (ARDUÍNO..., 2016).

Existem vários modelos de Arduino. Um dos mais populares é o Arduino UNO, que tem como base o microcontrolador ATmega328P e apresenta as seguintes características (Arduino UNO..., 2016):

- Tensão de operação de 5V
- Tensão de entrada recomendada variando de 7-12V
- 14 pinos de entradas digitais sendo 6 com função PWM
- 6 entradas analógicas
- Corrente de entrada/saída por pino de 20 mA
- Memória flash de 32 KB (ATmega328P)
- SRAM de 2 KB (ATmega328P)
- EEPROM de 1 KB (ATmega328P)
- Frequência de clock de 16 MHz
- Comprimento de 68,6 mm e largura de 53,4 mm
- Peso de 25g

O microcontrolador ATmega328P é da família de microcontroladores Atmel AVR de 8 bits, tem 131 instruções e 23 entradas/saídas programáveis (ATMEL..., 2015).

Neste projeto, o Arduino é usado no circuito de transmissão e recepção. No circuito de transmissão ele é o responsável pelo envio da informação codificada. No circuito de recepção, é responsável pela interpretação do sinal recebido e a atuação no hardware (acender ou apagar uma lâmpada, por exemplo). No arduino, quando uma de suas saídas é colocada em nível alto, ou ativada, ela estará numa tensão de 5V. Já quando uma de suas saídas é colocada em nível baixo, ou desativada, ela estará numa tensão de 0V.

### 2.3.2 Timer LM555

O LM555 é um circuito integrado capaz de gerar pulsos com precisão de tempo e oscilações. O tempo entre os pulsos é controlado por um resistor e um capacitor externo, seu diagrama de blocos pode ser observado na Figura 11 (SEMICONDUCTOR..., 2013)

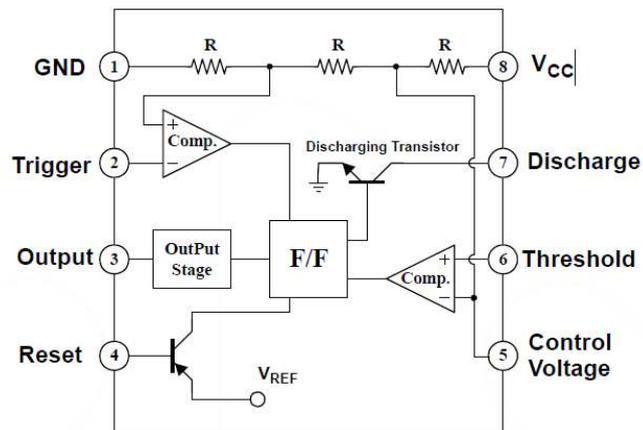


Figura 11 -Diagrama de blocos do LM555.

Fonte: SEMICONDUCTOR..., 2013.

### 2.3.3 Optoacoplador

Um optoacoplador combina numa mesma base, um LED e um fotodetector. Na Figura 12 mostra-se um acoplador formado por um LED e um fotodiodo: o primeiro está a esquerda e o segundo, à direita. A tensão que alimenta o LED força a circulação da corrente através do mesmo, fazendo com que emita uma luz que incide sobre o fotodiodo e cria uma corrente inversa que passa através do resistor  $R_2$ . (ZBRA; MALVINO; MILLER, 2001).

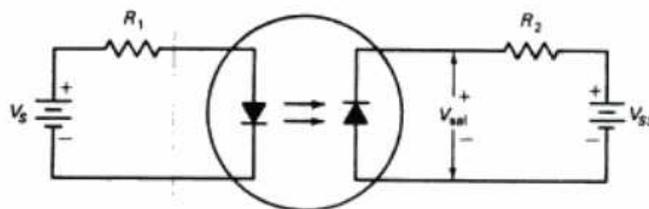


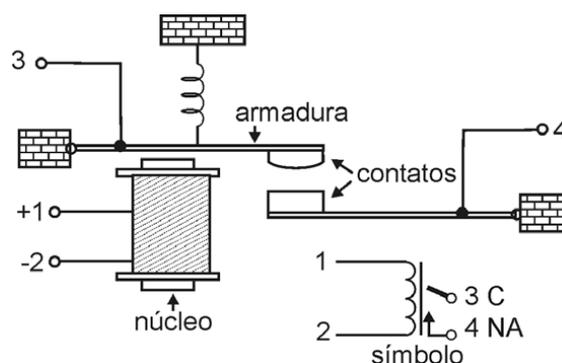
Figura 12 - Circuito optoacoplador

Fonte: ZBRA; MALVINO; MILLER, 2001.

A principal vantagem do optoacoplador é o isolamento elétrico entre o circuito do LED e o circuito do fotodiodo (ZBRA; MALVINO; MILLER, 2001).

### 2.3.4 Relé e triac

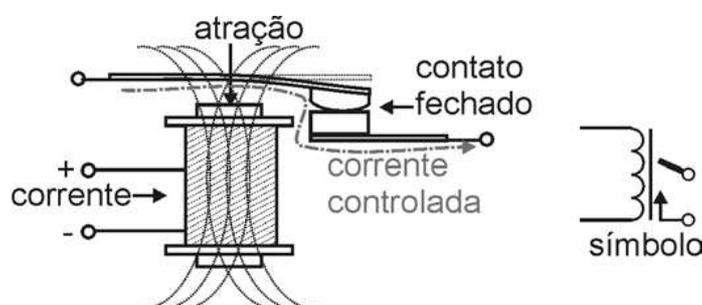
Podemos definir um relé como um dispositivo comutador eletromecânico. A estrutura simplificada de um relé é mostrada na Figura 13 (BRAGA, 2012). Os terminais 1 e 2 são os terminais da bobina, os terminais 3 e 4 correspondem aos contatos.



**Figura 13 - Estrutura simplificada de um relé.**

**Fonte: BRAGA, 2012.**

O relé é basicamente composto por uma bobina, uma armadura móvel e dois contatos. A bobina é um eletroímã quando é submetida a uma corrente elétrica, que assim, cria um campo magnético que atua sobre a armadura que, por sua vez, atua sobre os contatos fazendo com que eles se atraiam. Seu funcionamento, entretanto, faz com que um circuito ligado à bobina nas entradas 1 e 2 atue fazendo chaveamento no circuito ligado às entradas 3 e 4, Figura 14. Estes circuitos não precisam ter nenhum contato físico entre si, podendo trabalhar em níveis de tensão totalmente diferentes.

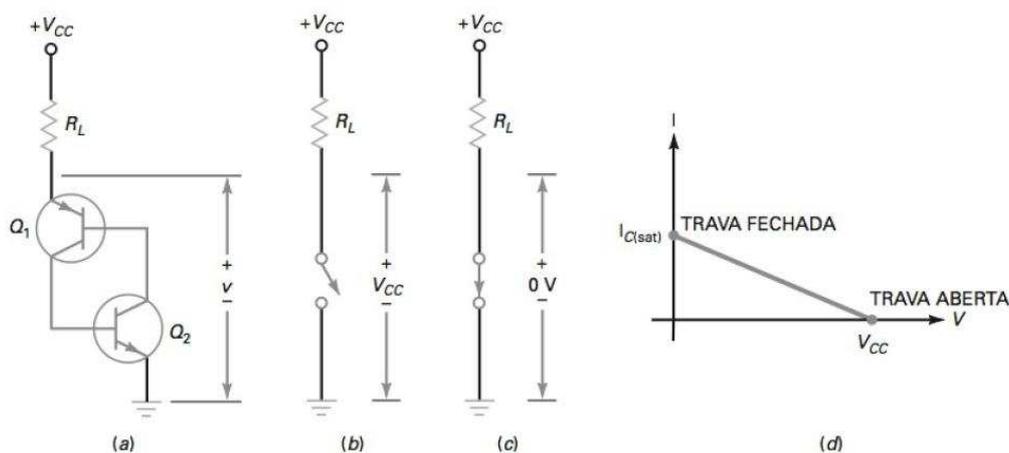


**Figura 14 - Com a atração, os contatos se tocam. Assim, a corrente pode passar pelo circuito controlado.**

**Fonte: BRAGA, 2012.**

Triacs são tiristores bidirecionais. Os tiristores, por sua vez, podem ser explicados como um circuito equivalente, Figura 15a. O transistor  $Q_1$  é um dispositivo *pnp* enquanto o transistor inferior  $Q_2$  é um dispositivo *nnp*. Se a corrente na base de  $Q_2$  aumenta, a corrente no coletor de  $Q_2$  aumenta. Isso aumenta a corrente na base de  $Q_1$  e a corrente no coletor de  $Q_1$ . Isso aumentará ainda mais a corrente na base de  $Q_2$ . A ação de amplificar e realimentar continua até que os dois transistores sejam levados à saturação. Nesse caso, o circuito todo age como uma chave fechada, Figura 15c. Nesse caso, o ponto de operação está no extremo superior da reta de carga  $cc$ , Figura 15d.

Por outro lado, se algo causar a diminuição na corrente de base de  $Q_2$ , a corrente no coletor de  $Q_1$  diminui, causando ação reversa, e o circuito todo age como uma chave aberta, Figura 15b. Nesse caso, o ponto de operação está no extremo inferior da reta de carga  $cc$ , Figura 15c (MALVINO; BATES, 1987).

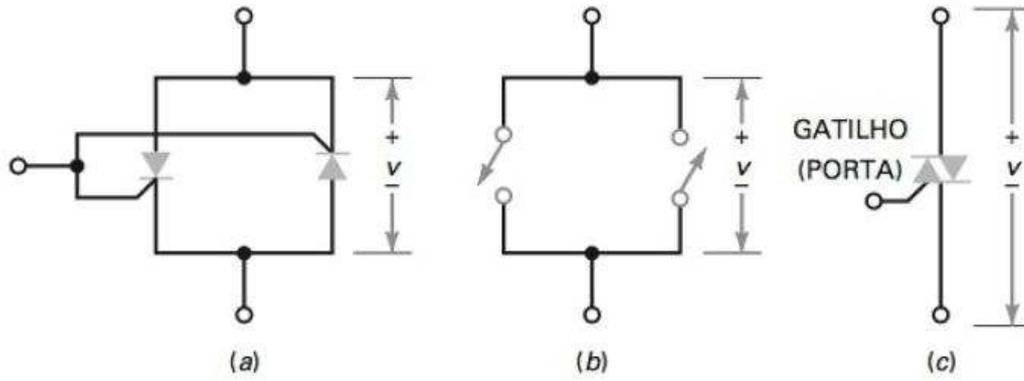


**Figura 15 – Tiristores**

- a) Trava conectada a um resistor de carga com uma fonte de tensão  $V_{CC}$**
- b) Trava aberta**
- c) Trava fechada**
- d) Gráfico mostrando o ponto de operação da corrente e tensão na trava**

**Fonte: MALVINO; BATES, 1987.**

O triac funciona como se fossem duas travas em paralelo, como mostrado na Figura 16.

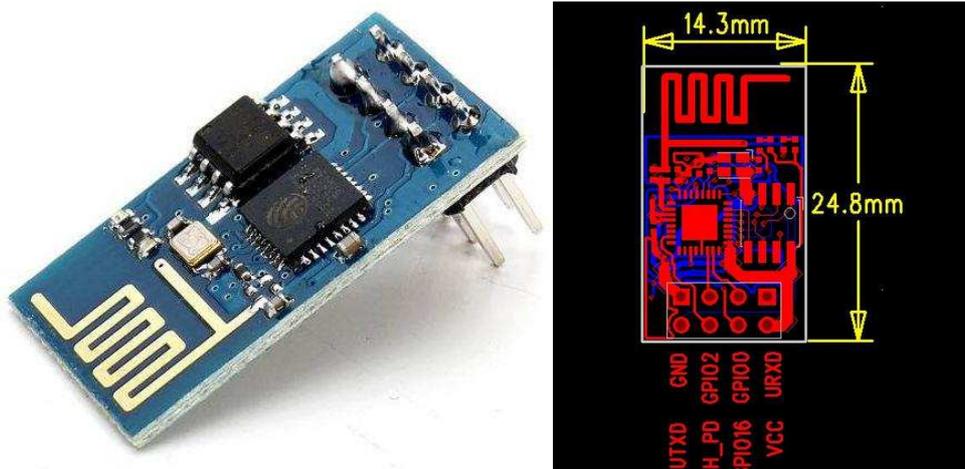


**Figura 16 – Funcionamento do triac**  
 a) Duas travas em paralelo com o gatilho comum  
 b) Duas travas abertas  
 c) Símbolo esquemático para o triac

Fonte: MALVINO; BATES, 1987.

### 2.3.5 ESP8266

O ESP8266, Figura 17, é um módulo WiFi completo de baixo custo, e tem embutido em sua composição o microcontrolador Tensilica L106 de 32-bits, com uma unidade central de processamento de 80 MHz, que pode alcançar um valor de até 160 MHz. Seu consumo máximo típico, dependendo de seus parâmetros, é de 170 mA, num ambiente com temperatura de 25°C.



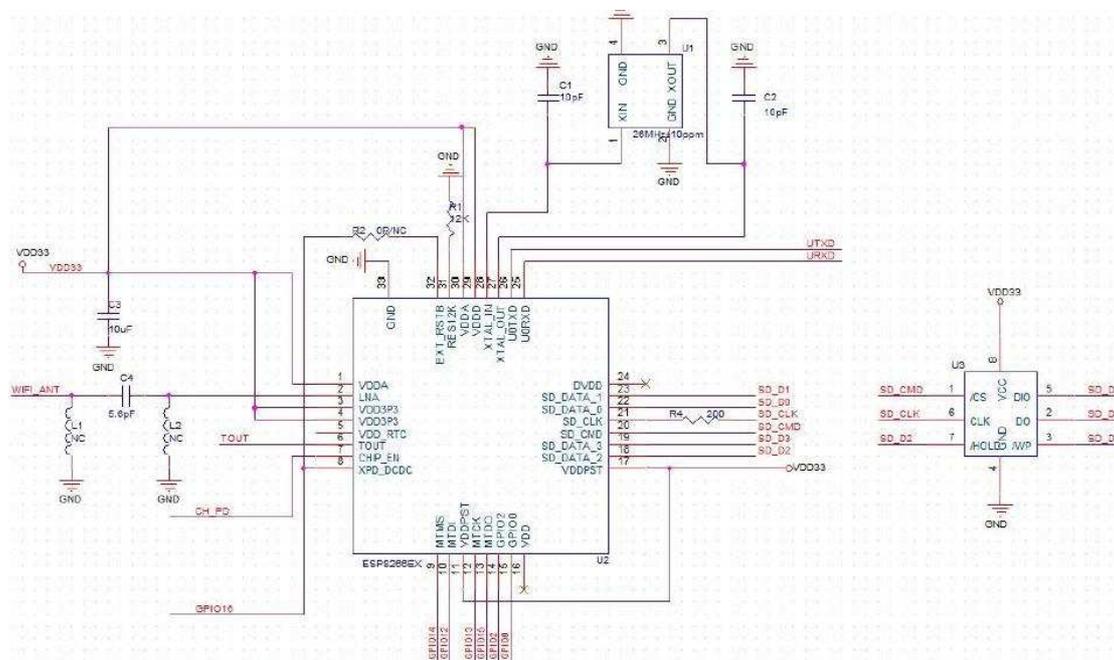
**Figura 17 - ESP8266-01**

Fonte: WIKI, 2015.

Apresenta as seguintes características:

- Protocolo TCP/IP integrado
- Trabalha com uma tensão de alimentação de 3,3V, podendo chegar a no máximo 3,6V.
- WiFi 2.4 GHz, suporta WPA/WPA2 Wi-Fi Protected Access
- PWM e portas de entrada e saída (GPIO)
- Temperatura de operação de -40°C a 125°C

O ESP8266 apresenta várias versões. Além de outras versões extras, seus modelos variam do ESP8266-01 até o ESP8266-14. Suas variações alteram a quantidade de pinos presentes em cada módulo, bem como seus formatos, tamanhos, LEDs na placa e até mesmo a antena (WIKI, 2015). O esquema do ESP866 pode ser observado na Figura 18.



**Figura 18 - Esquema do ESP8266**

**Fonte: SYSTEMS, 2013.**

Uma das maneiras de se comunicar com o ESP8266 é conectando-se a ele através de sua porta serial. Após conectado, para realizar as configurações do ESP8266, utiliza-se AT Commands ou Comandos AT, que é uma linguagem desenvolvida no início do ano de 1980 pela Hayes Company, e é utilizada hoje em dia para comunicação entre computadores e modem (DOSTÁLEK; KABELOVÁ, 2006). Na Tabela 2 apresenta-se alguns comandos AT para o ESP8266.

**Tabela 2 - Alguns exemplos de comandos do ESP8266, suas funções e respostas.**

| <b>Função</b>                                   | <b>Comando AT</b>                        | <b>Resposta</b>  |
|---|--|--|
| Test AT startup                                 | AT                                       | OK   |
| Restart module                                  | AT+RST                                   | OK   |
| List of all available AP's detected by ESP8266ç | AT+CWLAP                                 | +CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<ch>,<freq offset><br>OK<br>ERROR |
| Connect to AP                                   | AT+CWJAP?<br>AT+CWJAP="SSID", "Password" | +CWJAP:<ssid>,<bssid>,<channel>,<rssi><br>OK                       |
| Get IP Address                                  | AT+CIFSR                                 | +CIFSR:<IP address><br>OK<br>ERROR                                 |

Fonte: TEAM, 2015.

### **2.3.6 Buffer não inversor**

Sabemos que ESP8266 trabalha com uma tensão de 3,3V, mas o arduino que se comunica diretamente com ele trabalha com 5V em suas portas quando estas estão em nível alto. Então, se faz necessário converter as tensões de 5V para 3,3V. Para isso utiliza-se o buffer não inversor CD4050BC.

O CD4050 é um buffer não inversor hexadecimal, que trabalha de 3 a 15V de tensão em sua alimentação (SEMICONDUCTOR, 2002). Seu funcionamento consiste em se alimentar o CI, através da porta  $V_{DD}$  na Figura 19, com a tensão desejada em suas saídas (3,3V), e toda vez que sua entrada for acionada por uma tensão que vêm do arduino (5V), a sua saída correspondente apresentará um valor alto de 3,3V. Então, se o arduino aciona a entrada A do CD4050 a saída G subirá seu nível com a tensão convertida igual à que alimenta o CI.

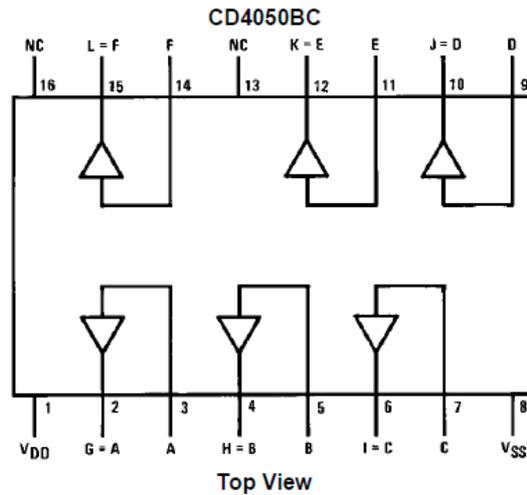


Figura 19 - Buffer não inversor hexadecimal CD4050BC

Fonte: SEMICONDUCTOR..., 2002.

### 2.3.7 Regulador de tensão

O regulador de tensão AMS1117-3.3, Figura 20, é necessário neste projeto pois é necessário garantir uma tensão de 3,3V para alimentar o ESP8266 e o buffer não inversor. Caso se tenha uma fonte de alimentação de 3,3V, não é necessário seu uso. Mas como é mais comum se encontrar fontes de 5, 9 e 12V, ele se torna um componente importante, facilitando a alimentação do circuito.

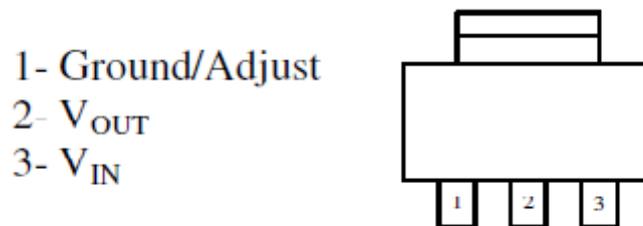


Figura 20 - Vista superior do regulador de tensão AMS1117 SOT-223

Fonte: SYSTEMS..., 2016.

## 2.4 Revisão de procedimentos de programação

### 2.4.1 Operadores de condição/repetição

Operadores de comparação são importantes para estabelecer repetições/condições para a realização de uma parte do código, definida dentro de uma estrutura de repetição/condição. Na Tabela 3 mostra-se esses operadores.

**Tabela 3 - Operadores de comparação.**

| <b>Operador</b> | <b>Definição</b>     |
|-----------------|----------------------|
| $x==y$          | x igual a y          |
| $x!=y$          | x diferente de y     |
| $x<y$           | x menor que y        |
| $x>y$           | x maior que y        |
| $x<=y$          | x menor ou igual a y |
| $x>=y$          | x maior ou igual a y |

Mais de um operador de comparação pode ser utilizado em uma estrutura de repetição/condição. Para isso, tem-se os operadores relacionais (Tabela 4).

**Tabela 4- Operadores de relacionais.**

| <b>Operador</b> | <b>Definição</b>           |
|-----------------|----------------------------|
| &               | E                          |
|                 | OU                         |
| ~               | NÃO                        |
| ^               | OU EXCLUSIVO               |
| <<              | DESLOCA BITS PARA ESQUERDA |
| >>              | DESLOCA BITS PARA DIREITA  |

### 2.4.2 Estrutura if

Tem sua estrutura da seguinte maneira:

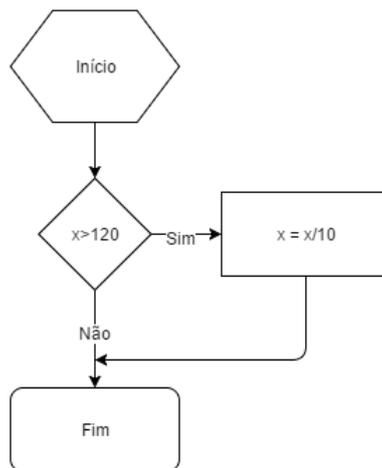
```
if (<condição>){  
    <comandos>  
}
```

Se a condição for satisfeita, os comandos dentro do laço, determinado pelo conteúdo dentro das chaves, são executados. Caso contrário, nenhum comando dentro do laço será executado.

Exemplo:

```
if (x > 120){ // Se a variável chamada x for maior que 120  
    x = x/10; /* A variável x recebe um novo valor que é seu valor antigo  
    dividido por 10. Caso x = 130, seu novo valor será x=13. */  
}
```

Na Figura 21 apresenta-se o fluxograma da execução do código apresentado anteriormente.



**Figura 21 - Fluxograma da função if.**

### 2.4.3 Estrutura if-else

Tem sua estrutura da seguinte maneira:

```

if (<condição>){
    <comandos1>
} else { <comandos2> }

```

Se a condição for satisfeita, os comandos dentro do laço **if** denominados comandos1, são executados. Caso contrário, os comandos dentro do laço **else** denominados comandos2, são executados.

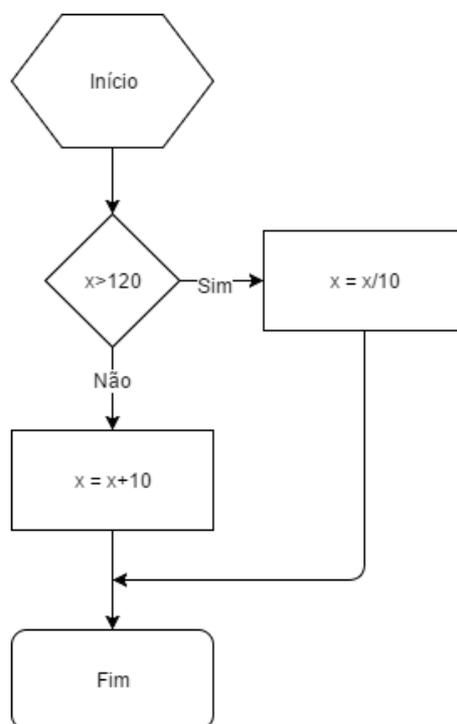
Exemplo:

```

if (x > 120){ // Se a variável chamada x for maior que 120
    x = x/10; /* A variável x recebe um novo valor que é seu valor antigo
    dividido por 10. Caso x = 130, seu novo valor será x=13. */
} else { x = x+10; } /* Se a variável x não satisfizer a condição do if, seu valor
é acrescido de 10. */

```

Na Figura 22 apresenta-se o fluxograma da execução do código apresentado anteriormente.



**Figura 22 - Fluxograma da função if-else.**

#### 2.4.4 Estrutura if-else if-else

Tem sua estrutura da seguinte maneira:

```

if (<condição1>){
    <comandos1>
} else if (<condição2>) {
    <comandos2>
} else if (<condição3>){
    <comandos3>
} else if (<condiçãoN>){
    <comandosN>
} else { <comandosN+1> }

```

Se a condição1 for satisfeita, os comandos dentro do laço, determinado pelo conteúdo dentro das chaves, são executados. Caso contrário, nenhum comando dentro do laço será executado.

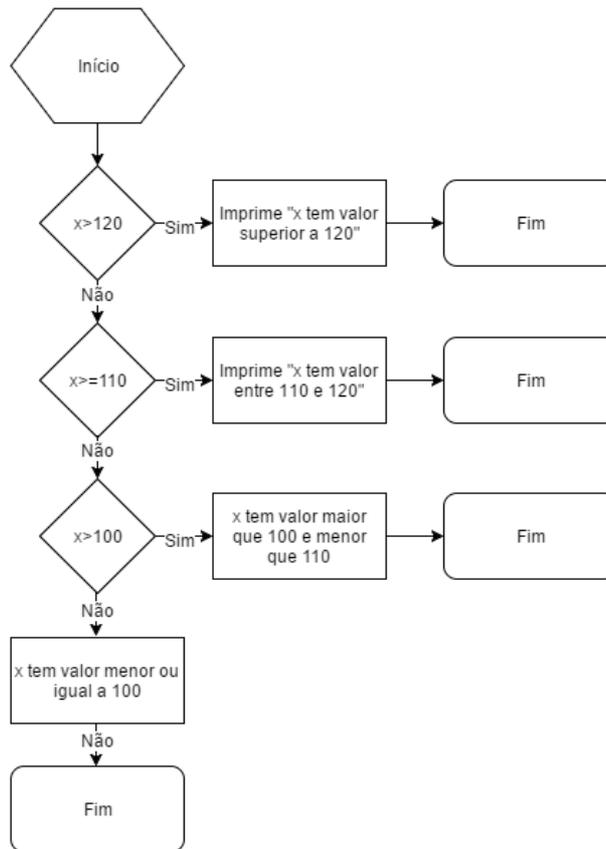
Exemplo:

```

if (x>120){ // Se a variável chamada x for maior que 120
    printf("x tem valor superior a 120"); //A função printf é utilizada na linguagem
    C para imprimir a frase que está entre aspas
} else if (x>=110){ // Se a variável chamada x for maior que 110
    printf("x tem valor entre 110 e 120"); //Imprime a frase que está entre aspas
} else if (x>100){ // Se a variável chamada x for maior que 100
    printf("x tem valor maior que 100 e menor que 110"); //Imprime a frase que está
    entre aspas
} else { printf("x tem valor menor ou igual a 100");} // Se nenhum comando
    acima foi executado, imprime a frase que está entre aspas dentro do laço else

```

Na Figura 23 apresenta-se o fluxograma da execução do código apresentado anteriormente.



**Figura 23 - Fluxograma da função if-elseif-else.**

### 2.4.5 Estrutura while

A estrutura while realiza testes e executa os comandos, até que as condições de testes se tornem falsas. Ela se dá da seguinte maneira:

```

while (<condição for verdadeira>){
    <comandos>
}
  
```

Os comandos são executados infinitamente até que a condição se torne falsa.

Exemplo:

```

int x = 0;
while (x < 5){ // Repete o laço té que o valor da variável não seja menor que 5
    printf("%i",x); //Imprime os valores da variável x (o termo %i é utilizado para
    imprimir variáveis do tipo inteiro)
    x++;
}
  
```

Na Figura 24 apresenta-se o fluxograma da execução do código apresentado anteriormente.

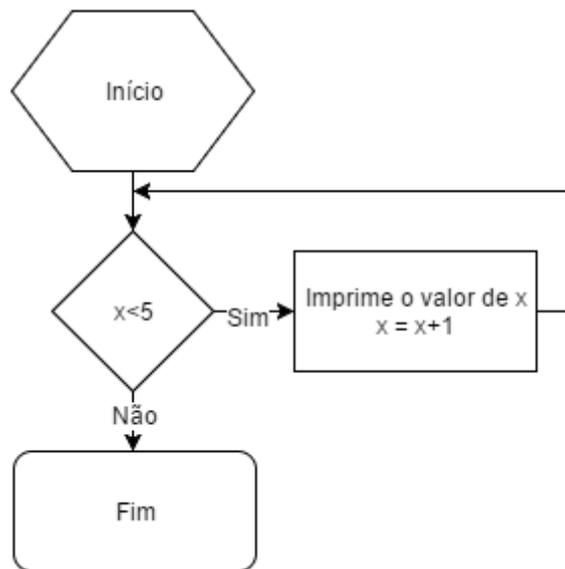


Figura 24 - Fluxograma da função while.

#### 2.4.6 Estrutura for

A estrutura *for* também é uma estrutura capaz de realizar repetição e exige 3 parâmetros: inicialização da variável, condição de parada ou saída da repetição e valor de incremento ou decremento da variável.

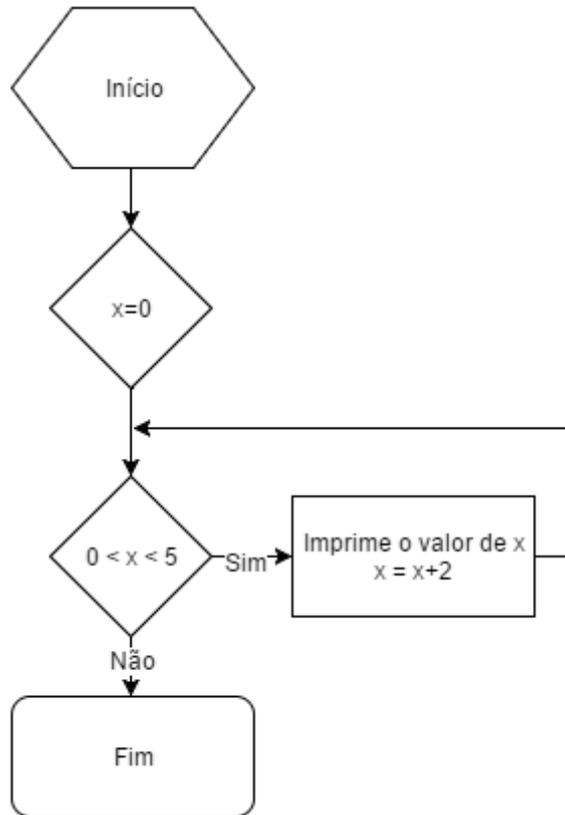
```
for (<inicialização> ; <condição> ; <incremento> ){  
    <comandos>;  
}
```

Assim como no caso da estrutura *while*, os comandos são executados infinitamente até que a condição se torne falsa. Porém no caso do *for*, as variáveis são inicializadas e, incrementadas ou decrementadas, dentro dos parâmetros da estrutura *for*.

Exemplo:

```
for (int x=0; i < 5; x=x+2){ /* x se inicia com 0 e recebe um incremento de 2 no fim de  
    cada repetição da estrutura for */  
    printf("%i",x); //Imprime os valores da variável x  
}
```

Na Figura 25 apresenta-se o fluxograma da execução do código apresentado anteriormente.



**Figura 25 - Fluxograma da função *for*.**

## 2.5 MIT App Inventor

MIT App Inventor é uma introdução para programação e criação de aplicativos que transforma uma linguagem de texto complexa em visual. A interface gráfica simples possibilita que um usuário iniciante consiga criar aplicativos funcionais com facilidade, mesmo com pouco tempo de aprendizado. MIT App Inventor foi criado em 2009 e possibilita a criação online de aplicativos para plataforma Android e conta, hoje em dia, com uma comunidade de aproximadamente 3 milhões de usuários em 195 países (TECHNOLOGY, 2016).

### 3 DESENVOLVIMENTO DO TRABALHO

Após o estudo sobre os protocolos, eletrônica, e outros componentes foi possível começar o desenvolvimento dos circuitos eletrônicos. Como o projeto tem como finalidade a criação de protótipos de baixo custo, os circuitos foram desenvolvidos em placas de fenolite furadas, bastando apenas montagem e solda do circuito.

Considerando alta tensão como a tensão da rede elétrica de uma residência de valor 127 ou 220 Volts, e baixa tensão como a tensão de no máximo 12 Volts, tanto no projeto da aplicação do protocolo X10 quanto no projeto que utiliza ESP8266 para comunicação, para se construir um sistema de acionamento remoto, como o acionamento de uma lâmpada por exemplo, é necessário construir um circuito de acionamento de tensão alta através de baixa tensão. Já que um micro controlador trabalha com tensão de 5 Volts e a tensão rede elétrica residencial é de 127 ou 220 Volts, Uma forma de se fazer isso é utilizando um relé ou triac. O relé é acionado simplesmente fazendo com que exista uma corrente passando pelos terminais da bobina, de valor mínimo dependendo de cada modelo, que faz com que os terminais dos contatos se conectem. A carga então deve ser ligada nos terminais dos contatos e seu acionamento é feito nos terminais da bobina. Já o triac, é preciso que se construa um circuito para seu funcionamento adequado. Nesse projeto, foram utilizados para montagem do circuito o optoacoplador MOC3020, o triac BT136 e dois resistores. Na Figura 26 mostra-se o projeto do circuito acionando uma lâmpada através de um botão de pressão.

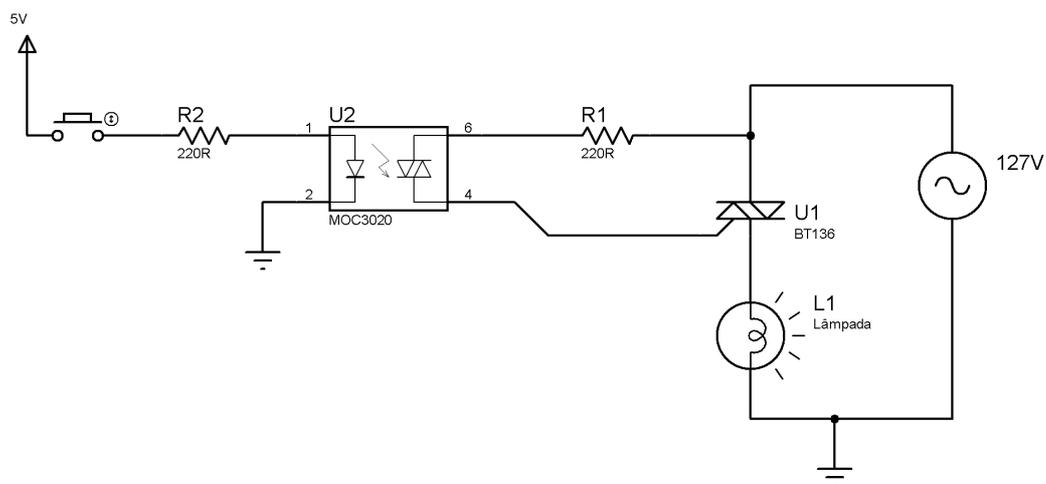


Figura 26 - Circuito de acionamento de uma lâmpada.

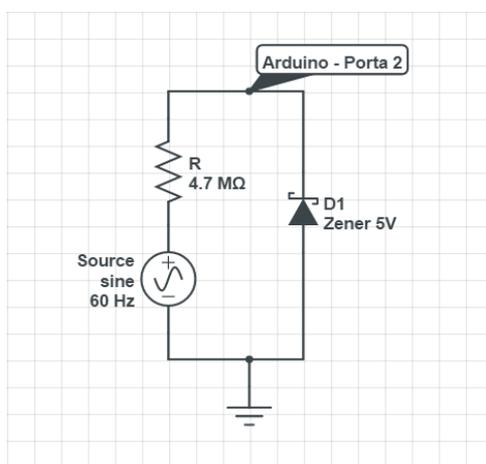
No projeto, o acionamento pelo botão de pressão, que aplica 5V no optoacoplador é simplesmente substituído por uma entrada do arduino, que é ligada e desligada de acordo com a programação implementada.

### 3.1 Circuito para implementação do Protocolo X10

Nessa seção serão descritos o desenvolvimento dos circuitos necessários para implementação do protocolo X10.

#### 3.1.1 Detecção de passagem por zero

Detectar passagem por zero tem como objetivo detectar quando a onda senoidal da rede elétrica realiza a passagem pelo valor 0 V, auxiliando os circuitos de comunicação do protocolo X10 o momento correto de se realizar a comunicação. Para esse fim, o arduino recebe a onda proveniente da corrente alternada instalada em uma residência, Figura 27, e consegue identificar se é ou não o momento ideal de se realizar a comunicação. Como o arduino trabalha na faixa de 0 a 5 V, foi utilizado um diodo zener de 5 V para limitar a tensão na porta de entrada do arduino bem como um resistor de alto valor, 4,7 M $\Omega$ , para limitar a corrente e abaixar tensão.



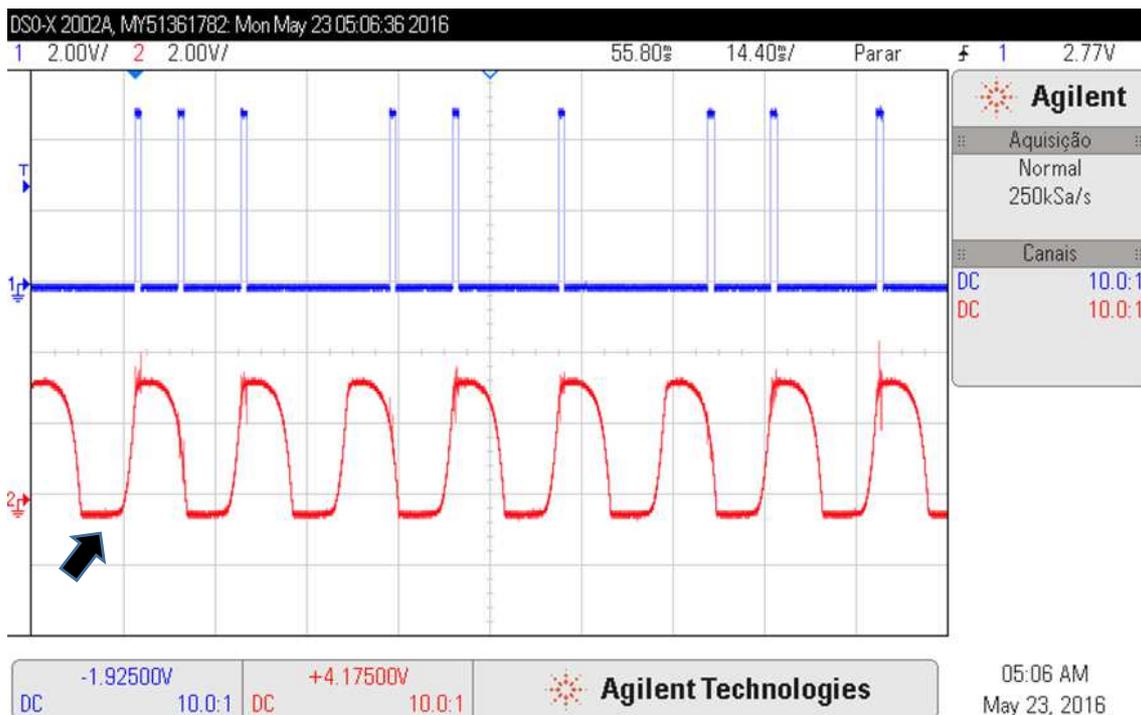
**Figura 27 - Circuito detector de passagem por zero.**

Para a detecção foi utilizada a função de interrupção do arduino. Interrupção consiste em em uma função que é executada no arduino, de acordo com o tipo da interrupção, toda vez que for verificado que um nível de uma porta for alterado. Entenda-se nível baixo como sendo 0 Volts, e nível alto como sendo 5 Volts, as interrupções podem ser dos tipos:

- Low: Ocorre sempre quando a entrada está no nível baixo.

- Change: Acontece quando ocorre uma alteração no nível, tanto de baixo para alto, quanto de alto para baixo.
- Rising: Ocorre somente quando a entrada se altera do nível baixo para alto.
- Falling: Ocorre somente quando a entrada se altera do nível alto para baixo.

Utiliza-se a interrupção do tipo *Change* pois como no protocolo X10, deve-se enviar informações sempre que a onda senoidal estiver no valor de 0 V, precisa-se saber o momento tanto da descida quanto subida da onda. Na Figura 28 pode-se observar a detecção e o acionamento do envio dos bits de comunicação no momento em que acontece a interrupção, que será explicado no próximo tópico. Pode-se perceber que os envios acontecem nos momentos em que ocorrem variações de nível, tanto de baixo para alto quanto de alto para baixo, que representam o momento em que a onda senoidal de 60 Hz da rede elétrica passa pelo valor de 0 V.

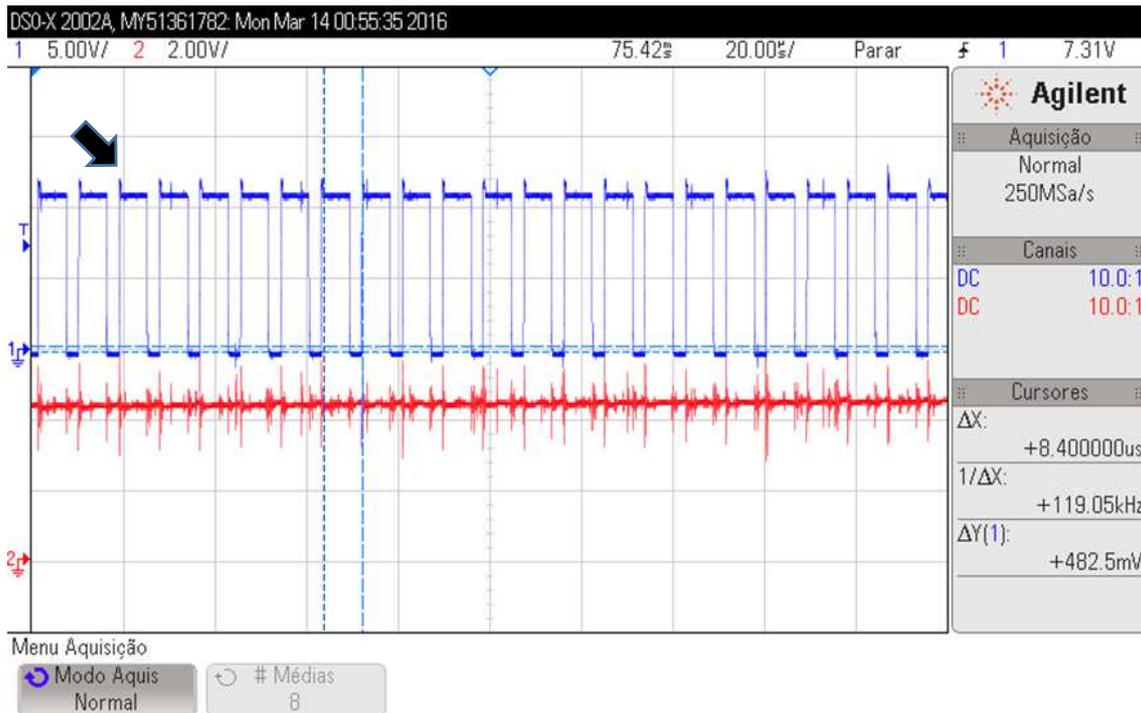


**Figura 28 – Detecção e o acionamento do envio dos bits de comunicação no momento em que acontece a interrupção.**

No canal 2, indicado pela seta, observa-se o circuito detector de passagem por zero. No canal 1, pode-se observar o envio do bit de comunicação.

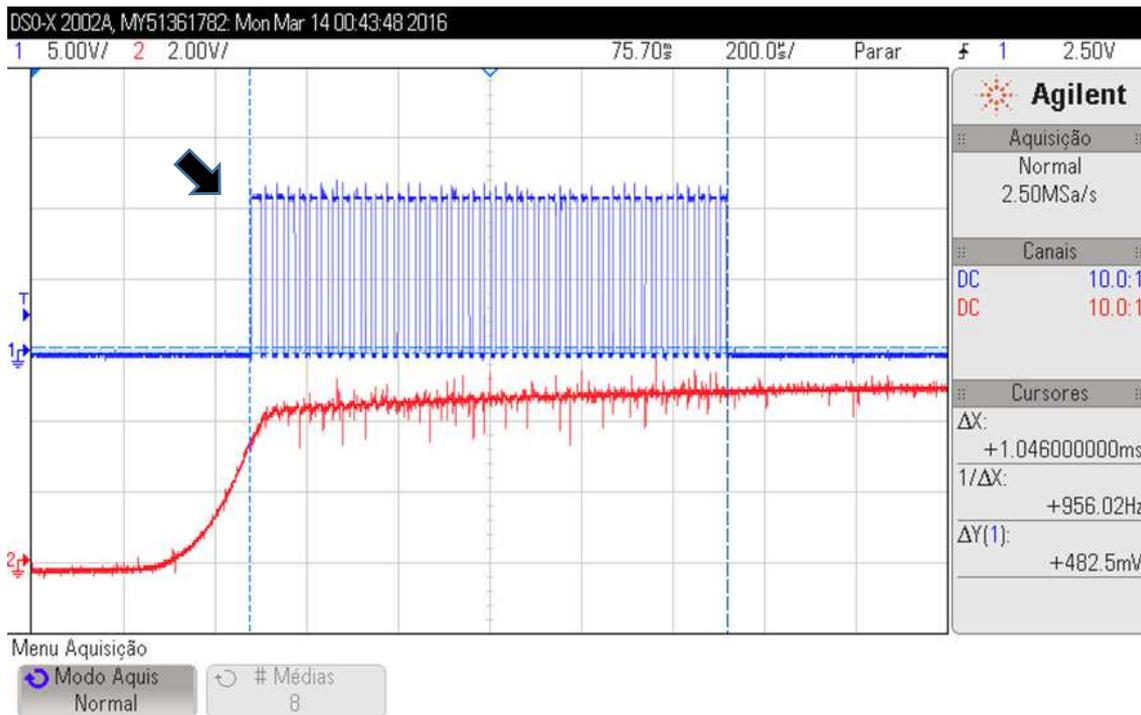
### 3.1.2 Implementação do circuito transmissor

Para que aconteça a comunicação via Protocolo X10, são necessárias oscilações de 120 kHz. Para se conseguir tais oscilações foi utilizado o LM555 operando no modo astável, ou seja, produzindo uma onda quadrada em sua saída. O sinal gerado apresentou um bom resultado, após alguns ajustes, apresentando uma frequência satisfatória de proximadamente 119 kHz, como apresentado na Figura 29.



**Figura 29 – Sinal de alta frequência gerado no LM555 indicado pela seta.**

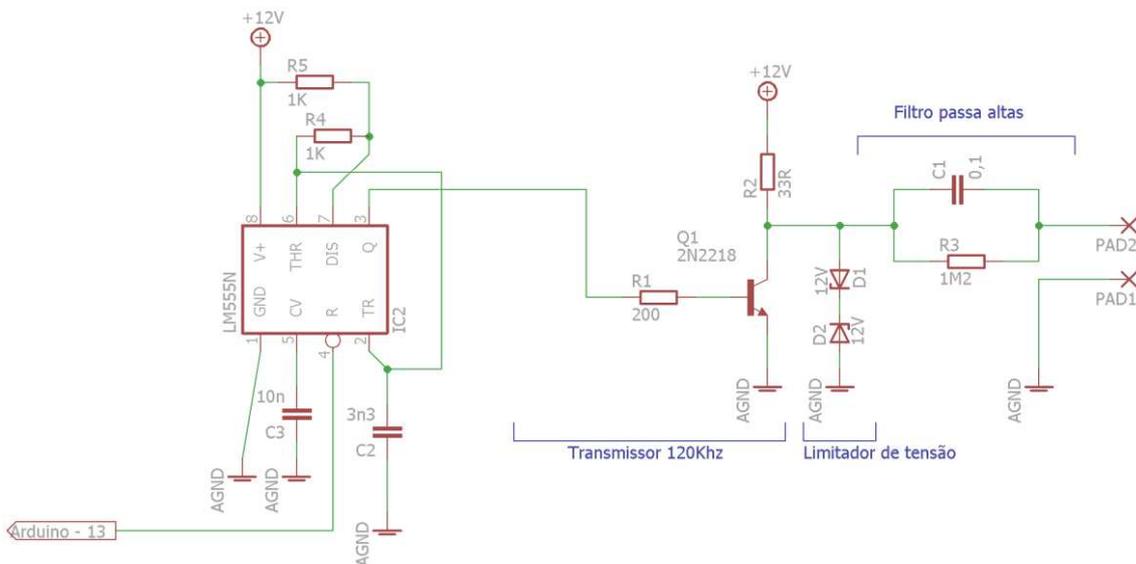
Para envio da comunicação, o arduino, que já consegue perceber o momento certo de enviá-la, aciona o LM555. Assim, caso o bit desejado seja "1", o arduino aciona o LM555, por sua vez, aciona um transistor que "injeta" alta frequência na rede elétrica. Já caso o bit seja "0", o arduino não faz nada pois não é necessário. A geração da frequência foi programada para ter 1 milissegundo de duração (BURROUGHS, 2002). Na Figura 30 pode-se observar o envio do bit "1" no osciloscópio.



**Figura 30 - Envio do bit "1".**

Pode-se observar no envio do bit "1" na Figura 30, indicado pela seta, que sua duração, representada por  $\Delta X$ , é de aproximadamente 1 ms.

O esquema do circuito transmissor é mostrado na Figura 31.



**Figura 31 - Circuito transmissor**

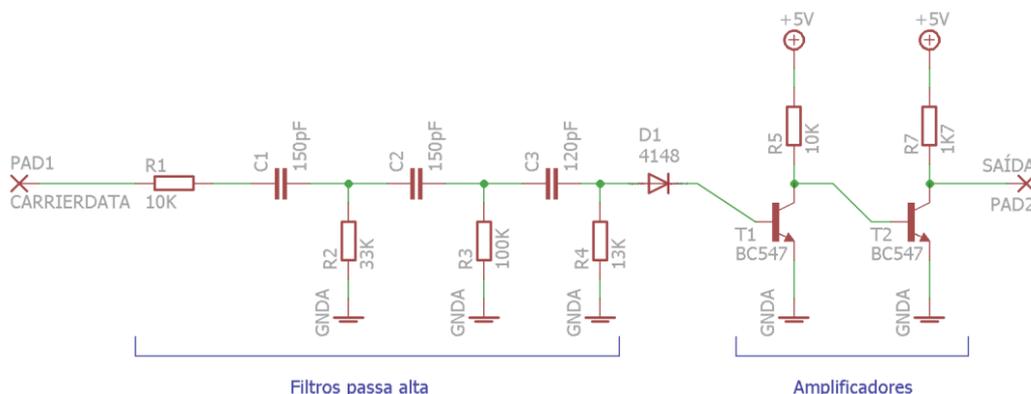
Um filtro passa altas auxilia no desacoplamento da tensão da rede de alimentação do circuito de transmissão dos elementos de baixa potência do circuito transmissor.

### 3.1.3 Implementação do circuito receptor

O circuito receptor, é o responsável por receber toda a informação da rede elétrica e separar a onda senoidal de 60 Hz da alimentação, do sinal de alta frequência de comunicação, injetado na rede pelo circuito oscilador.

Para receber os sinais do protocolo X10, é necessário detectar a presença da frequência de 120 kHz na linha de alimentação de corrente alternada. Como a impedância do capacitor é dada pela Equação 3, um capacitor de 0,1 μF apresenta uma baixa impedância (13 Ω) para uma frequência de 120 kHz, mas uma alta impedância (26,5 Ω) para a frequência de 60 Hz da linha de alimentação (BURROUGHS, 2002).

O circuito é composto por duas etapas: filtro passa alta, seguido por amplificadores, Figura 32.

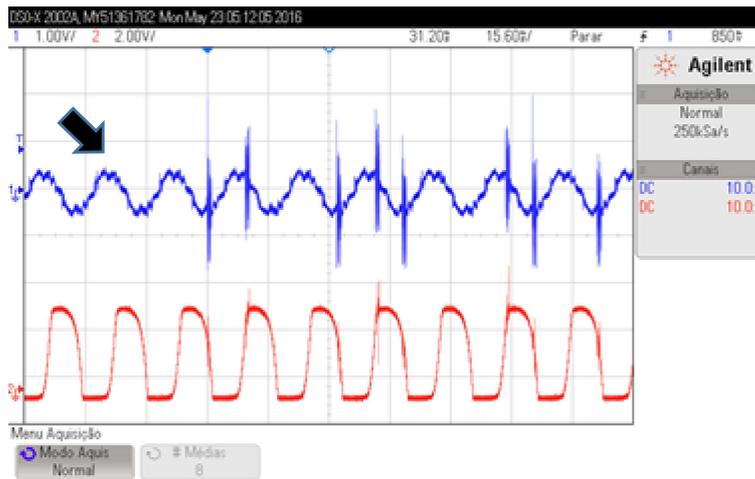


**Figura 32 - Filtro do circuito receptor**

O filtro passa alta permite a passagem somente de altas frequências. Ele é composto por três etapas de filtragem. A primeira etapa é composta por um capacitor (C1) de 150 pF e um resistor (R2) de 33 kΩ. Pela equação Equação 3, tem-se:

$$f_{q\ db} = \frac{1}{2 * \pi * 150\ pF * 33\ k\Omega} \cong 32\ kHz \quad (\text{Equação 6})$$

Isso faz com que se atenua a frequência de 60 Hz (BURROUGHS, 2002) enquanto as frequências acima de 32 kHz conseguem passar para os próximos estágios de amplificação, Figura 33.

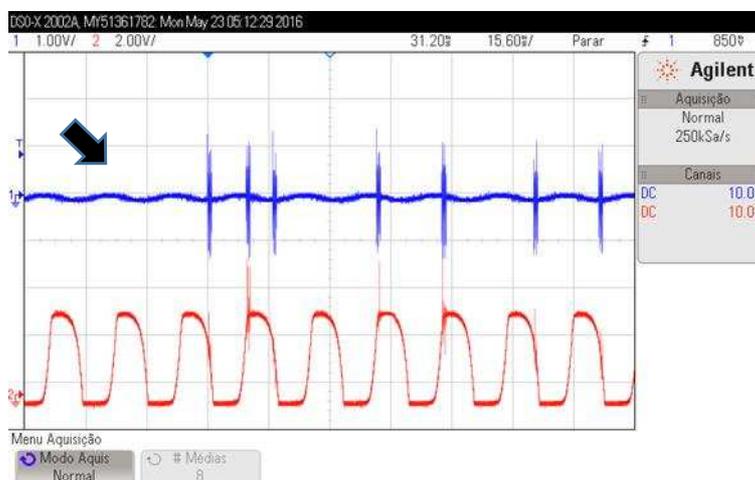


**Figura 33 – Saída do primeiro filtro indicado pela seta.**

Na sequência, tem-se um capacitor (C2) de 150 pF e um resistor (R3) de 100 kΩ. Pela equação Equação 3, tem-se:

$$f_q \text{ db} = \frac{1}{2 * \pi * 150 \text{ pF} * 100 \text{ k}\Omega} \cong 10 \text{ kHz} \quad (\text{Equação 7})$$

A saída do sinal filtrado é exibida na Figura 34.

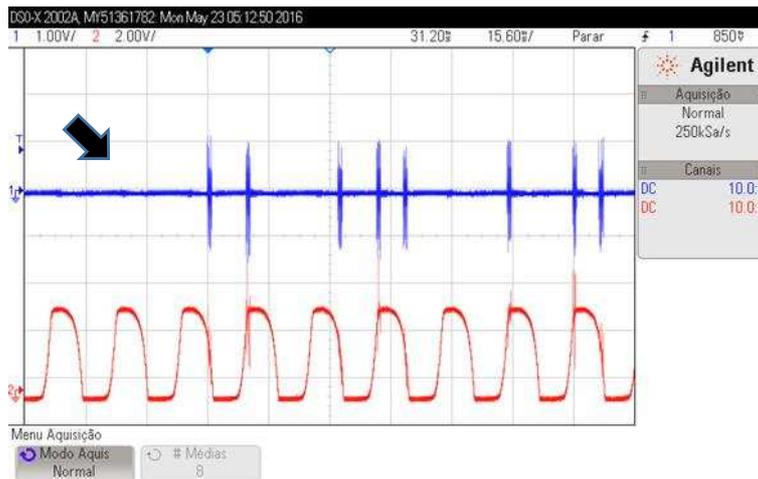


**Figura 34 – Saída do segundo filtro indicado pela seta.**

Atenuando-se as frequências abaixo de 10 kHz. Logo após, tem-se um capacitor (C3) de 120 pF e um resistor (R3) de 13 kΩ. Pela equação Equação 3, tem-se:

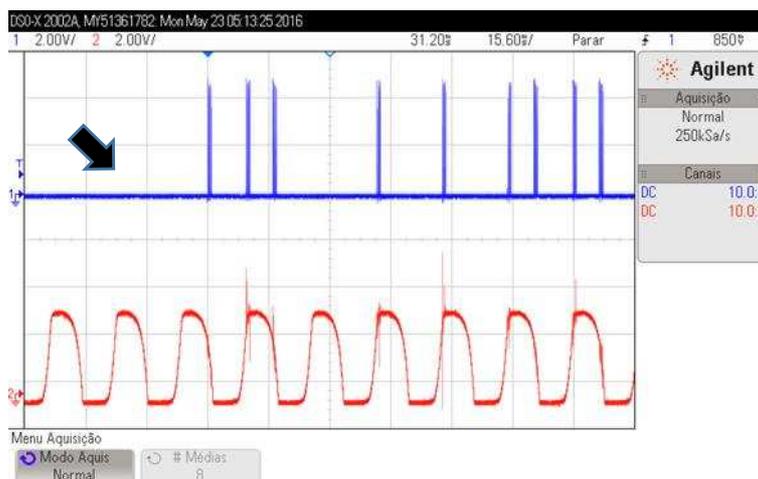
$$f_q \text{ db} = \frac{1}{2 * \pi * 120 \text{ pF} * 13 \text{ k}\Omega} \cong 100 \text{ kHz} \quad (\text{Equação 8})$$

Atenuando-se, agora, as frequências inferiores à 100 kHz, Figura 35.



**Figura 35 – Saída do terceiro filtro indicado pela seta.**

Então, um diodo retifica o sinal, eliminando a parte negativa do sinal, e os amplificadores estão amplificando as frequências passantes, que é exatamente a frequência que deseja-se perceber se está presente ou não na linha de alimentação. A saída é exibida na Figura 36.



**Figura 36 - Saída dos amplificadores indicado pela seta.**

## 3.2 ESP8266

### 3.2.1 Comunicação do ESP8266 com o arduino

Uma das maneiras de se configurar o ESP8266 é realizando uma comunicação serial com o arduino. Para isso deve-se conectar a porta RX de um com a porta TX do outro com o auxílio do buffer não inversor. Após sua conexão deve ser realizada sua configuração. Em seu modo de operação normal, as suas entradas *Reset* e *CH\_PD* devem ser

configuradas em nível alto. No arduíno, para configurar essas saídas, basta configurar da seguinte maneira:

```
#define RST 4 //Atribui o valor de RST como 4, que é a porta do arduíno conectada a
porta RST do ESP8266

#define CH_PD 5 // Atribui o valor de CH_PD como 5, que é a porta do arduíno
conectada a porta CH_PD do ESP8266.

void setup()
{
  pinMode(RST,OUTPUT); //Declara a porta equivalente a RST como saída
  digitalWrite(RST,HIGH); //Coloca a porta RST em nível alto

  pinMode(CH_PD,OUTPUT); //Declara a porta equivalente a CH_PD como saída
  digitalWrite(CH_PD,HIGH); //Coloca a porta CH_PD em nível alto
}
```

### **3.2.2 Configurando ESP8266 para se conectar a um ponto de acesso**

Foi criada uma função que envia os dados pela comunicação serial configurada entre o arduíno e o ESP8266 chamada *enviaESP*. Ela recebe como parâmetros um comando, um tempo de espera e uma variável para monitorar caso aconteça erros na comunicação. Assim configura-se o ESP8266 com os seguintes comandos:

```
enviaESP("AT+RST\r\n",2000,DEBUG); // reseta o ESP8266
enviaESP("AT+CWMODE=1\r\n",1000,DEBUG); // configura como estação
enviaESP("AT+CWJAP=\"NomeWifi\", \"SenhaWifi\"\r\n",1000,DEBUG); //se conecta
a uma rede Wifi
delay(1000); //Aguarda um tempo para que a conexão seja estabelecida com êxito
enviaESP("AT+CIPMUX=1\r\n",1000,DEBUG); // configura para multiplas
conexões. É necessário para configurar o ESP8266 como servidor
enviaESP("AT+CIPSERVER=1,8080\r\n",1000,DEBUG); // Configura o ESP8266
como um servidor na porta 8080
```

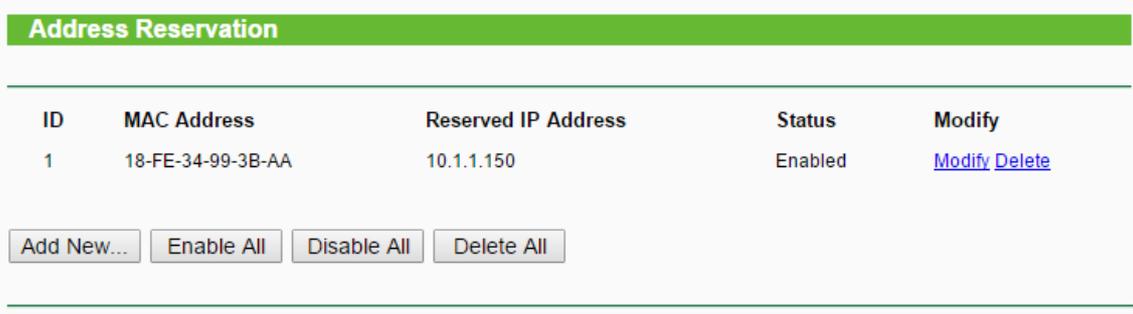
No código acima, *NomeWifi* é o nome da rede wireless que se deseja conectar e *SenhaWifi* é a senha dessa rede. Após a execução da configuração, se não ocorrer nenhum problema, o ESP8266 já estará conectado ao ponto de acesso.

### 3.2.3 Configurando o ESP8266 no roteador com IP fixo

Para realizar a comunicação numa rede wireless é necessário estabelecer a conexão dos dispositivos envolvidos através de uma rede fornecida por um ponto de acesso. Esse ponto de acesso pode ser um roteador, um servidor ou até mesmo pode ser um ESP8266 configurado para tal função. No caso, utilizou-se um roteador, uma ferramenta bastante popular como ponto de acesso de internet encontrado hoje em dia nas residências.

Numa rede, se o roteador estiver configurado com o DHCP (Dynamic Host Configuration Protocol) ativado, todos os dispositivos, quando se conectam com sucesso após passar pelas etapas de segurança, são registrados com um número denominado endereço IP (Internet Protocol). Todos os dispositivos conectados têm seu próprio IP, assim sendo, o ESP8266 ao se conectar a uma rede wifi recebe do roteador um endereço IP, dentre os quais ele está configurado para fornecer. Lembrando que o roteador fornece um número máximo e configurável de endereços IP.

É interessante, nesse caso, que se reserve no roteador um número IP fixo para o ESP8266, pois os comandos são enviados através desse número. Caso ele fique se alterando, teríamos dificuldade em configurar o sistema que realiza o controle remoto do ESP8266. Para isso basta acessar as configurações do roteador e reservar um endereço IP para o endereço MAC do ESP8266, Figura 37.



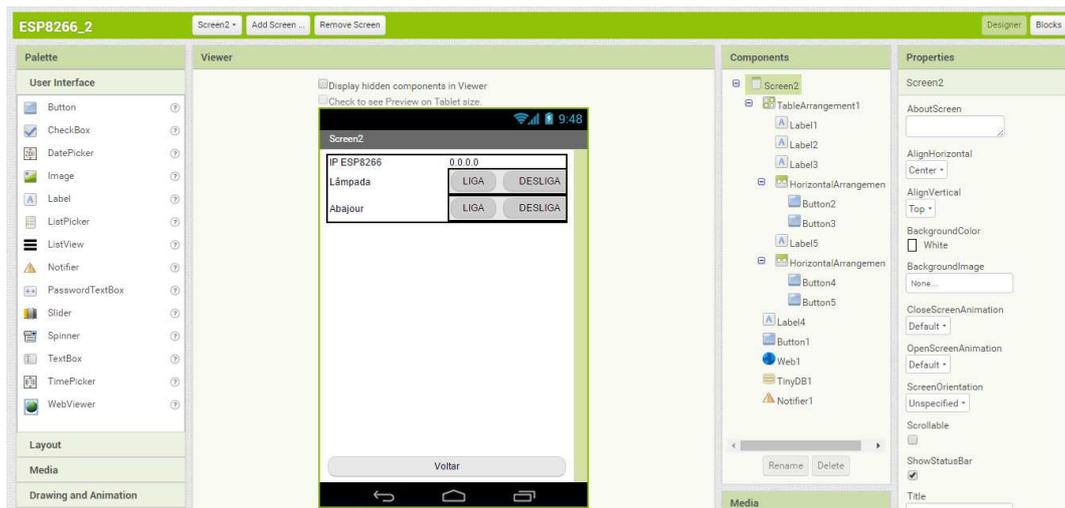
| ID | MAC Address       | Reserved IP Address | Status  | Modify  |
|----|-------------------|---------------------|---------|---|
| 1  | 18-FE-34-99-3B-AA | 10.1.1.150          | Enabled | <a href="#">Modify</a> <a href="#">Delete</a> |

**Figura 37 - Configuração de um endereço de IP para um ESP8266 no roteador TP-Link modelo no TL-WR340G/TL-WR340GD**

### 3.2.4 Programação para Android com MIT App Inventor

A interface do MIT App Inventor é simples, Figura 38, e é dividida basicamente nas opções de Designer e Blocks. A interface de Designer é composta pelas abas Palette, Viewer, Components, Properties. Na aba Palette, temos os elementos que podem ser

adicionados ao seu projeto, na aba Viewer tem-se a parte visual da tela onde se pode arrastar e posicionar os elementos da aba Palette para adicioná-los ao projeto. Na aba Components vê-se os componentes adicionados à tela que se está trabalhando (outras telas podem ser criadas) e, ao selecionar, pode-se ver suas propriedades aparecerem na aba Properties. Essas propriedades podem ser alteradas de acordo com as necessidades do programador.



**Figura 38 - Interface do MIT App Inventor - Designer**

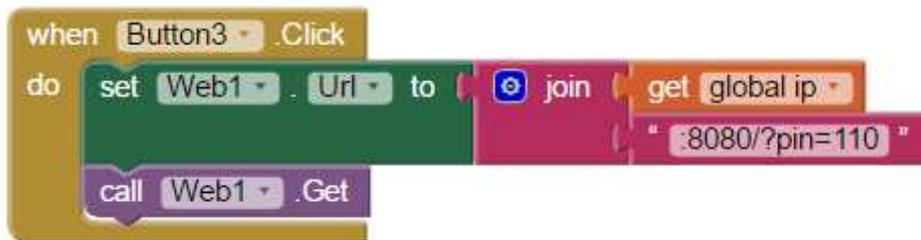
Após inserir e configurar os elementos na opção de Designer, seus elementos já podem ser programados na opção Blocks, Figura 39. A opção Blocks é composta pela aba Blocks e Viewer. Na aba Blocks tem-se acesso a blocos de lógica e blocos dos elementos que colocamos no projeto. Já na aba Viewer pode-se manipular os blocos e construir as lógicas de programação.



**Figura 39 - Interface do MIT App Inventor - Blocks**

Neste caso, o ESP8266 recebe o comando GET de página via rede local. Para isso, conecta-se o ESP8266 na rede local e, sabendo-se seu endereço IP, programa-se o aplicativo Android para lhe enviar informações. Essas informações são códigos dizendo ao ESP8266 o que ele deve fazer.

Para programar o clique de um botão, por exemplo, pega-se o bloco de programação do botão que envia um comando para o arduíno, apresentado na Figura 40. Utiliza-se, para programação, o bloco de comando do botão com o evento ".Click", que é o evento executado assim que o botão for clicado. Dentro da condição, que é quando o botão é clicado, configura-se o aplicativo para enviar a informação via o método GET. A variável "global ip" armazena o endereço IP configurado na tela inicial do aplicativo, que é o endereço do ESP8266 na rede local, assim o aplicativo envia o comando da seguinte maneira, supondo que o endereço do ESP8266 seja 192.168.1.150, por exemplo: `http://192.168.1.150:80/?pin=110`



**Figura 40 - Programação da lógica executada após clicar num botão no aplicativo Android**

A informação enviada é controlada pelos valores "110" enviados para o ESP8266. Nesse caso, os números "11" representam o número correspondente à porta do arduíno na qual está conectada um dispositivo a ser controlado, como um circuito de acionamento de uma lâmpada. O "0", por sua vez, é o comando desligar, que pode ser substituído pelo "1" no botão programado com a função de ligar.

## 4 RESULTADOS E ANÁLISES

### 4.1 Protocolo X10

Para análise de resultados do protocolo X10 foi submetido a testes os circuitos transmissor e receptor. Criou-se uma programação para realizar a comunicação entre eles. No receptor foi adicionado um circuito de acionamento de duas lâmpadas de Led, Figura 41.



**Figura 41 – Montagem dos circuitos de comunicação por Protocolo X10**

O circuito transmissor transmite sequências informando se alguma lâmpada, ou ambas as lâmpadas, deveria ser ligada ou desligada. O circuito receptor recebe essas comunicações e as interpreta. Para isso, foi criada uma função para o envio da comunicação através da rede elétrica. A fluxograma da função é representada no da Figura 42.

Os primeiros resultados não foram satisfatórios, descobriu-se alguns problemas como a alimentação do arduino por fontes chaveadas causavam instabilidades em seu funcionamento e a comunicação não acontecia, pois, sua instabilidade o fazia perder pedaços da comunicação. Outro problema identificado foi que o circuito transmissor precisa de uma referência aterrada, não bastando somente o neutro da rede. Sem o aterramento a referência do circuito varia, apresentando um comportamento não desejável. Um capacitor em sua alimentação amenizou o problema da alimentação irregular e a ligação do circuito transmissor em um aterramento resolveu o problema de flutuação, porém após a correção deste problema, a comunicação realizava apenas o

primeiro comando da sequência enviada, parando após a primeira lâmpada ligar. Descobriu-se posteriormente que a lâmpada de led, que contém também uma fonte chaveada em seu acionamento atrapalhava a comunicação via protocolo X10. A adição de um capacitor na detecção de passagem por 0V, criando um filtro passa baixas amenizou, mas não conseguiu resolver o problema, então, a solução foi a substituição das lâmpadas de led por incandescentes, resolvendo definitivamente o problema e possibilitando que a comunicação acontecesse normalmente.

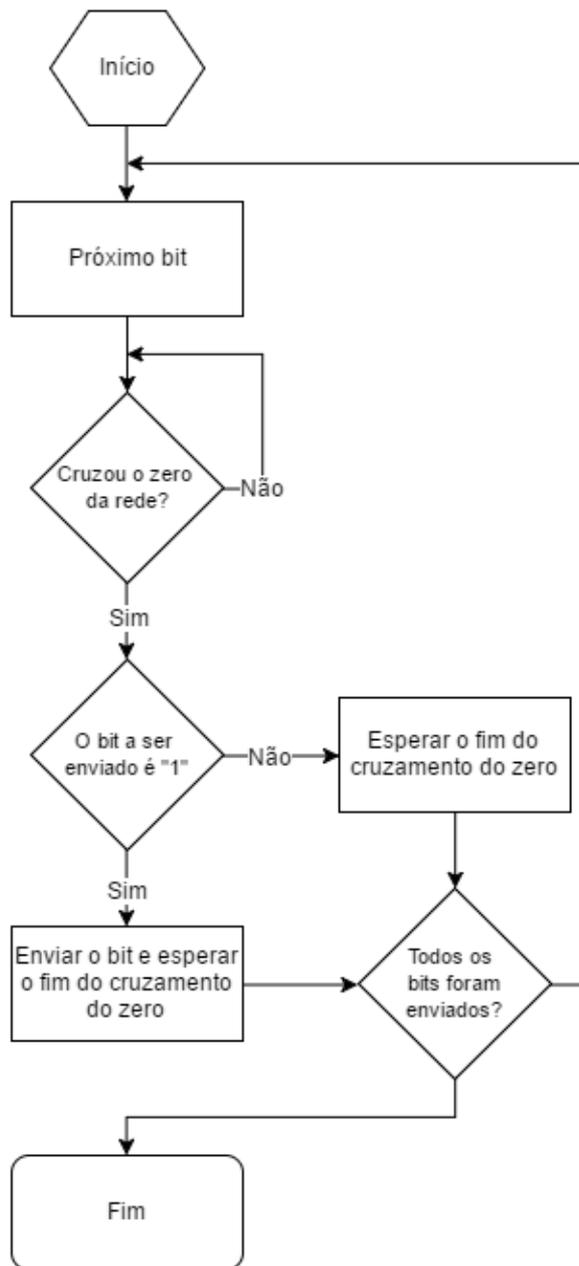
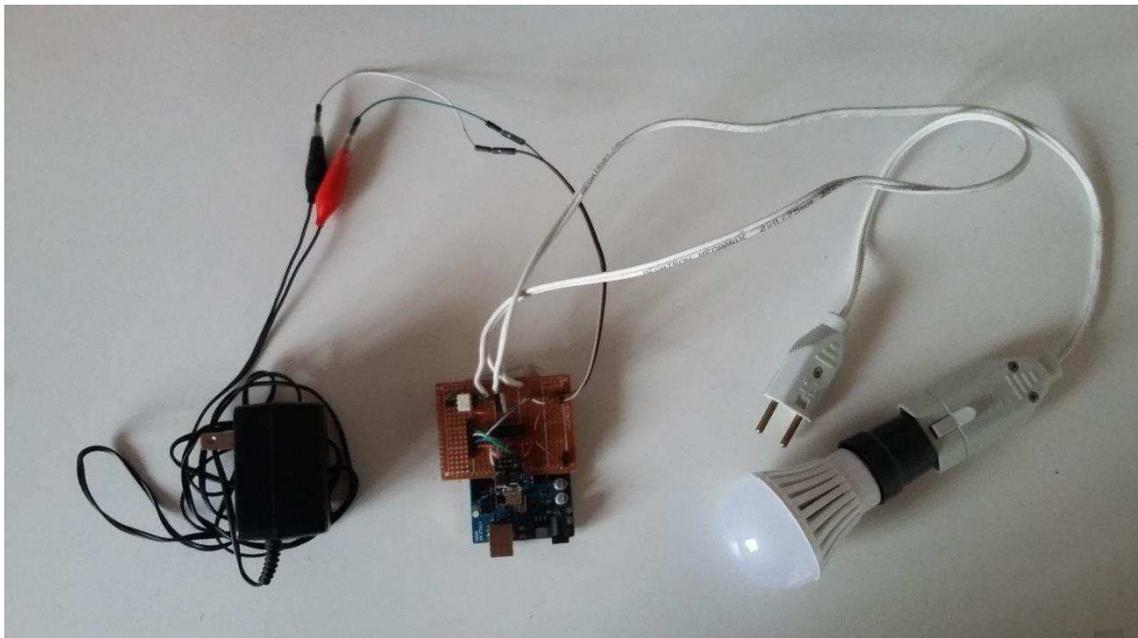


Figura 42 – Fluxograma da função de envio de bits pela rede elétrica.

## 4.2 ESP8266

Para uma análise do comportamento do ESP8266, foi construída uma placa que tem como função conectar o arduíno num ponto de acesso, para que ele possa receber os comandos via wifi, e realizar acionamentos cargas de potência mais alta, Figura 43.



**Figura 43 - Montagem dos circuitos de comunicação com ESP8266**

Ou seja, o arduíno recebeu comandos e acionou, por meio de uma de suas saídas de 5V como programado, uma carga de maior de valor de 110 ou 220V. Um fluxograma completo da lógica programada no arduíno é exibida na Figura 44.

O bloco *Configura ESP8266* é uma série de comandos enviado ao ESP8266 para que ele seja configurado para se conectar à um ponto de acesso e foi tratado no tópico 3.2.2.

É importante ressaltar que, como se utilizou como transporte das mensagens um roteador, o sucesso da comunicação depende do bom funcionamento do mesmo, e o atraso da comunicação depende também do tráfego da rede local. Durante os testes problemas não foram encontrados e o ESP8266 apresentou comportamento estável, respondendo os comandos com precisão.



Figura 44 – Fluxograma da programação do arduino

## 5 CONSIDERAÇÕES FINAIS

O Protocolo X10 mostrou-se funcional para automação residencial, no entanto, apresentou instabilidade devido a interferências externas. Para uma aplicação efetiva, filtros mais robustos podem ser desenvolvidos melhorando a qualidade da comunicação. O legado desse protocolo já foi estabelecido, existindo hoje no mercado produtos que transmitem informações como dados de internet através da rede elétrica.

O ESP8266 apresentou bom funcionamento, e em trabalhos futuros sugere-se o desenvolvimento do circuito trocando-se o Arduíno por ESP8266, fazendo que seu custo, que já é baixo, se reduza ainda mais. Pode ser adicionado ao projeto controle remoto de iluminação em LED dimerizável com auxílio do PWM, sistemas de tomadas inteligentes com acionadores sem contato com auxílio do infravermelho, programações de desligamento automático temporário e até mesmo acionamento via internet com auxílio de protocolos como MQTT (Message Queue Telemetry Transport).

Recomenda-se também, em trabalhos futuros, o uso de protótipos como o NodeMCU, que é um módulo produzido com base no ESP8266, programado em linguagem Lua, e que apresenta algumas facilidades para implementação em um projeto. Outras pesquisas podem ser realizadas em relação à filtros eletrônicos pois, devido ao aumento considerável do uso de componentes que geram ruídos, como o uso de fontes chaveadas como apresentado nesse projeto, as redes de alimentação ficam muito poluídas com harmônicas e subharmônicas. Aparelhos eletrônicos sensíveis sem filtros adequados podem receber essa poluição na fonte de alimentação e apresentar instabilidade no seu funcionamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ARDUÍNO. **What is arduino?**, 2016. Disponível em: <<http://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 05 mai. 2016.
- ARDUÍNO UNO & Genuino UNO. **Arduino UNO**, 2016. Disponível em: <<https://www.arduino.cc/en/main/arduinoBoardUno>>. Acesso em: 05 mai. 2016.
- ATMEL. **ATmega48A/PA/88A/PA/168A/PA/328/P**. Atmel Corporation. San Jose, p. 660. 2015.
- BOYLESTAD, R. L. **Introdução à análise de circuitos**. 12. ed. São Paulo: Pearson Education do Brasil, 2012.
- BOYLESTAD, R.; NASHESKY, L. **Dispositivos Eletrônicos e Teoria de Circuitos**. [S.l.]: Editora LTC, v. 1, 1998.
- BRAGA, N. C. **Relés**. 1. ed. [S.l.]: Editora Newton C., 2012.
- BURROUGHS, J. **X-10 Home Automation Using the PIC16F877A**. [S.l.]: Microchip Technology Inc., 2002.
- CHUNDURU, V.; SUBRAMANIAN, N. **Effects of power lines on performance of home control system**. Power Electronics, Drives and Energy Systems, 2006. PEDES'06.: International Conference on IEEE, 2006.
- CUNCHE, M. I know your MAC Address: Targeted tracking of individual using Wi-Fi. **International Symposium on Research in Grey-Hat Hacking - GreHack**, Grenoble, France, nov. 2013.
- DOSTÁLEK, L.; KABELOVÁ, A. **Understanding TCP/IP: A Clear and Comprehensive Guide to TCP/IP Protocols**. [S.l.]: Packt Publishing Ltd, 2006.
- EUROX10. **O que é X10? EuroX10**, 2015. Disponível em: <<http://www.eurox10.com/Content/X10Information.htm>>. Acesso em: 05 mai. 2015.
- INSTRUMENT, T. **LM555 Timer**. Texas Instrument. Dallas, p. 25. 2015.
- JUNESTRAND, S. Being private and public at home. **Ph.D. dissertation, School of Architecture**, 2004.

LOUREIRO, C. A. H. et al. **Redes de Computadores III: Níveis de Enlace e Físico**. Porto Alegre: Bookman Editora, 2014.

MALVINO, A.; BATES, D. J. **Eletrônica (v. 2)**. [S.l.]: McGraw Hill, 1987.

NETO, A. J. F.; NEVES, F. D.; JUNIOR, P. C. S. **Fundamentos de Circuitos Elétricos**. [S.l.]: AMGH Editora, 2013.

ROSETTO, M. Uso do Protocolo Z39.50 para recuperação de informação em redes eletrônicas. **Scientific Electronic Library Online**, Brasília, v. 26, n. 2, ago. 1997.

SEMICONDUCTOR, F. **CD4050BC Hex Non-Inverting Buffer**. Fairchild Semiconductor. [S.l.], p. 7. 2002.

SEMICONDUCTOR, F. **LM555 Single Timer**. Fairchild Semiconductor Corporation. [S.l.], p. 13. 2013.

SYSTEMS, A. M. **AMS117, 1A LOW DROPOUT VOLTAGE REGULATOR**. Advanced Monolithic Systems, Inc. [S.l.], p. 8. 2016.

SYSTEMS, E. **Espressif Smart Connectivity Platform: ESP8266**. Espressif Systems Inc. [S.l.]. 2013.

TEAM, E. S. I. **ESP8266 AT Instruction Set**. Espressif Systems. [S.l.], p. 59. 2015.

TECHNOLOGY, M. I. O. What is MIT App Inventor? **MIT App Inventor**, 2016. Disponível em: <<http://appinventor.mit.edu/explore/about-us.html>>. Acesso em: 10 mai. 2016.

WIKI, E. C. Modules. **ESP8266**, 2015. Disponível em: <<http://www.esp8266.com/wiki/doku.php?id=esp8266-module-family>>. Acesso em: 05 mai. 2015.

ZBAR, P. B.; MALVINO, A. P.; MILLER, M. A. **Prácticas de electrónica**. 7. ed. [S.l.]: Marcombo, S.A., 2001. 400 p.