INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CAMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE ELETRÔNICA (DAELN)

CURSO SUPERIOR DE TECNOLOGIA EM ELETRÔNICA INDUSTRIAL

RICARDO ADRIANO HORN

ATUADOR SEM FIO PARA APLICAÇÕES EM AUTOMAÇÃO RESIDENCIAL

FLORIANÓPOLIS, DEZEMBRO DE 2018.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CAMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE ELETRÔNICA (DAELN)

CURSO SUPERIOR DE TECNOLOGIA EM ELETRÔNICA INDUSTRIAL

RICARDO ADRIANO HORN

ATUADOR SEM FIO PARA APLICAÇÕES EM AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Eletrônica Industrial

Professor Orientador: Reginaldo Steinbach, Me.

Coorientador: Lucas M. Trichez

FLORIANÓPOLIS, DEZEMBRO DE 2018.

Ficha de identificação da obra elaborada pelo autor.

Horn, Ricardo
ATUADOR SEM FIO PARA APLICAÇÕES EM AUTOMAÇÃO RESIDENCIAL
/ Ricardo Horn; Orientação de Reginaldo Steinbach;
Coorientação de Lucas Trichez. - Florianópolis, SC,
2018. 62 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de
Santa Catarina, Câmpus Florianópolis. CST em Eletrônica
Industrial. Departamento Acadêmico
de Eletrônica.
Inclui Referências.

1. Domótica. 2. IoT. 3. ESP8266. I. Steinbach, Reginaldo.
II. Trichez, Lucas. III. Instituto Federal de
Santa Catarina. Departamento Acadêmico de Eletrônica.
IV. Título.

ATUADOR SEM FIO PARA APLICAÇÕES EM AUTOMAÇÃO RESIDENCIAL

RICARDO ADRIANO HORN

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Eletrônica Industrial e aprovado na sua forma final pela banca examinadora do curso de superior de tecnologia em Eletrônica Industrial do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 28 de dezembro de 2018

Banca Examinadora:

Reginaldo Steinbach, Me

Luis Carlos Martinhago Schlichting, Dr

Renan Augusto Starke, Dr

AGRADECIMENTOS

Na vida não caminhamos sozinhos e não realizamos grandes feitos sem ajuda de pessoas, como diria Henry Ford “(...) Trabalhar juntos é sucesso”. Muitas pessoas participaram do desenvolvimento deste projeto e por isso venho agradecer a estas pessoas que contribuíram diretamente e indiretamente.

Agradeço a família, em especial minha Mãe, Regina Horn que sempre foi meu maior exemplo de dedicação e esforços independentemente do tipo problema ou situação, ao meu Pai, Paulo R. Horn que sempre me estimulou a nunca desistir de uma formação superior e por repassar valores na qual todo pai deveria repassar aos filhos e aos meus irmãos Rafael Horn e Ana Paula Horn, que sempre estiveram presentes em minha vida.

Agradeço ao amor da minha vida Mariana Pinheiro, que abdiquei de momentos juntos para a realização deste trabalho, mas com sua compreensão, seu carinho, seu amor, seu apoio e sua motivação me deram forças para seguir em frente não desistindo nunca. Eu te amo! Agradeço também aos seus pais Adriana e Luciano pelos incentivos e conselhos a não desistir.

Quero agradecer ao meu orientador Reginaldo Steinbach que sem suas orientações, cobranças e dicas este trabalho não seria possível. Meu coorientador, coordenador e amigo Lucas M. Trichez que conseguiu dar um ponto de partida no meu projeto disponibilizando seu tempo e recursos (financeiros, intelectuais e psicológicos) para o desenvolvimento deste trabalho. Ao professor Célio Espindola pela paciência e contribuições metodológicas.

Meus colegas de trabalho que participaram no dia a dia do desenvolvimento, especialmente meu colega e amigo Sérgio Oliveira onde suas dicas, conselhos e ensinamentos não só contribuíram para o desenvolvimento do projeto, mas de minha carreira profissional e pessoal.

Não poderia deixar de mencionar as amizades criadas ao longo do curso (em ordem alfabética) Anderlei Martins, Bryan Douglas Pinheiro, Eric Marques Fortunato, Gero Muller, Júlio L. Piva, Luis G. Bonotto, Vitor P. Mattos. Ao meu colega José V. Bezerra que sempre esteve presente na minha vida e graduação.

Muito obrigado a todos!

“(...) Um dia é preciso parar de sonhar, tirar os planos das gavetas e, de algum modo começar...”

Amyr Klink

RESUMO

Atualmente a demanda por automatizar e controlar tarefas remotamente nas residências cresce a cada ano. Isto graças a evolução tecnológica que temos nos S*martphones*, aplicativos e outras soluções que tornam nossa rotina diária cada vez mais conectada e prática. A inclusão de sistemas para automação em residências hoje, além de facilitar nosso dia a dia, agregam valor ao imóvel. A empresa Legrand, sabendo disso, possui uma linha de produtos chamada *Wi-Connect* na qual visa tornar uma residência mais automatizada. A proposta deste trabalho consiste em desenvolver uma nova versão do produto atual da empresa Legrand, na qual visa automatizar e controlar uma residência remotamente incorporando o módulo ESP8266. Isso fará com que cada dispositivo da residência tenha conexão com a internet dispensando a necessidade de um gateway. Também será desenvolvido uma interface para smartphones para fazer o controle remoto das cargas ou dispositivos.

Palavras-chave: Domótica. IoT. Automação. ESP8266.

ABSTRACT

Currently the demand for automation and remote control of tasks grows every year. This is due to technological evolution in smartphones, apps and other solutions which make our routine more connected and practice. The inclusion of systems to home automation in homes today, besides facilitating our day, add value to the property. The company Legrand, has a product line called Wi-Connect in which aims to make a residence more automated. The proposal of this work is to develop a new version, which aims to automate and control devices remotely, incorporating the module ESP8266, this will make each device of the residence connected the internet without the need of a gateway. An interface will also be developed for smartphones to remotely control the loads or devices.

Key-words: Home Automation. IoT. Automation. ESP8266.

LISTA DE IMAGENS

Figura 1 – Tipos de Automação 18
Figura 2 – Previsão de dispositivos conectados em 2020..... 19
Figura 3 – Acesso à internet por tipo de conexão 22
Figura 4 – *Switch* 2 canais (2x2000W) Legrand e controle de cenários..... 25
Figura 5 – Base controladora (*Gateway*) *Wi-Connect* Legrand..... 25
Figura 6 – *Node*MCU 26
Figura 7 – ESP8266 modelo ESP-01 27
Figura 8 – Proposta de solução..... 28
Figura 9 – Atuador *Switch* 2 canais (2x2000W) Legrand desmontado..... 29
Figura 10 – Placa base atuador *Switch* 2 canais..... 30
Figura 11 – Placa digital atuador *Switch* 2 canais 30
Figura 12 – ESP8266 e Conversor USB/Serial 32
Figura 13 – Fonte 3V3 e Sincronismo da Rede..... 33
Figura 14 – Chaves analógicas e infravermelho 33
Figura 15 – Conexão com a placa base e led's de *status* 34
Figura 16 – *Top* (Vermelho) e *Botton* (Azul) da placa com simulação 3D..... 35
Figura 17 – Simulação 3D de encaixe e case explodido..... 35
Figura 18 – Placas recebidas da OSHPARK 36
Figura 19 – Pinos utilizados na *Flash* externa..... 37
Figura 20 – Correção do erro e montagem mecânica 37
Figura 21 – Diferenças nos protótipos Rev.0 e Rev.1 38
Figura 22 – Placa Rev.1 montada e encaixa na base 39
Figura 23 – Regulador Linear..... 40
Figura 24 – Alteração placa base..... 41
Figura 25 – Arquitetura do protótipo..... 42
Figura 26 – *Sketch* GitHub 43
Figura 27 – Autenticação por *Token* 44
Figura 28 – Botão Virtual, *Blynk Write* e *LED2*..... 45
Figura 29 – Teste prático 46
Figura 30 – Interruptores utilizados 46
Figura 31 – *Log* da leitura das chaves 47
Figura 32 – Mapa de acionamentos 48

Figura 33 – Mapa de acionamentos e atualização de *status* no aplicativo..... 49
Figura 34 – Aplicação ESP-TOUCH..... 50
Figura 35 – Estrutura do pacote de dados enviado pelo ESP-Touch..... 51
Figura 36 – Chamada da função *SmartConfig* 51
Figura 37 – *Feed-Back* Esp-Touch..... 52
Figura 38 – Fluxograma *Firmware* 53
Figura 39 – Cenário de instalação..... 54
Figura 40 – Teste prático 55
Figura 41 – Teste de Ciclagem diagrama 55
Figura 42 – Teste de ciclagem pratico 56
Figura 43 – Teste térmico 57

LISTA DE TABELAS

Tabela 1 – Comparativo produto IHouse com o produto atual Legrand 23
Tabela 2 – Comparativo produto MyWay com o produto atual Legrand 23
Tabela 3 – Comparativo produto FIBARO com o produto atual Legrand 24
Tabela 4 – Valores das chaves no conversor AD..... 47

LISTA DE ABREVIATURAS

IoT – Internet das coisas (*Internet of Things*)

LoRa – Tecnologia de transmissão de dados sem fio

LoRaWAN – Camada lógica da tecnologia Lora

WiFi – Tecnologia de transmissão de dados sem fio

ZigBee – Tecnologia de transmissão de dados sem fio

UART – *Universal Asynchronous Receiver/Transmitter* - Comunicação Serial Assíncrona

SDIO – *Secure Digital Input Output* – Expansão da memória

SPI – *Serial Peripheral Interface* – Protocolo de comunicação com periféricos

I2C – *Inter- Integrated Circuit* – Barramento serial Multimestre

GPIO – *General Purpose Input/Output* – Portas programáveis para entrada e saída de dados de um microcontrolador/microprocessador

ADC – Conversor Analógico-Digital – Pinos de um microcontrolador que realizam uma leitura de um sinal analógico

PWM – *Pulse-Width Modulation* – Modulação da razão cíclica de um sinal

PCI – Placa de Circuito Impresso

SUMÁRIO

1 INTRODUÇÃO 15
1.1 Justificativa 15
1.2 Definição do problema 16
1.3 Objetivo geral 16
1.4 Objetivos Específicos 16

2. REVISÃO LITERÁRIA 17
2.1 Automação residencial (Domótica) 17
2.2 IoT (Internet of Things) 18
2.3 Tecnologias de comunicação sem fio 20
2.4 Produtos usados em domótica atualmente 22
2.5 Produto Legrand 24
2.6 *Node*MCU 26
2.7 ESP8266 27

3 DESENVOLVIMENTO 28
3.1 Desenvolvimento do *Hardware* 29
3.1.1 Definição do circuito 31
3.1.2 Protótipo 1 (Revisão 0) 32
3.1.3 Resultados *Hardware* Rev.0 36
3.1.4 Protótipo 2 (Revisão 1) 38
3.1.5 Resultados Hardware Rev.1 39
3.2 Integração da placa base com a placa digital 39
3.2.1 Resultados Integração 40
3.2.2 Solução 41
3.3 Desenvolvimento do *Software* 42
3.3.1 Arduino 43
3.3.2 Blynk 43
3.3.3 Integração Blynk, Arduino e *Hardware* 43
3.3.4 Autenticação 44
3.3.5 Envio de comandos e botões virtuais 44
3.3.6 Acionamento local e botões 46
3.3.7 *Feed-back* do *status* da tecla no aplicativo 48
3.3.8 Conectando um dispositivo no WiFi com ESPTouch (SmartConfig) 49
3.3.9 Fluxo geral da aplicação 52

4 RESULTADOS 54
4.1 Teste funcional 54
4.2 Teste de ciclagem 55
4.3 Ensaio Térmico 56

5 CONCLUSÃO 58

REFERÊNCIAS 59

ANEXO A Esquemático NodeMCU 61
ANEXO B Lista de Materiais 62

1 INTRODUÇÃO

A demanda por sistemas que tornem as casas “inteligentes” cresce a cada ano. O artigo feito pela Saber Elétrica (2018), aponta que cada vez mais os usuários querem mais conforto em suas residências (desde um simples controle de iluminação a um complexo sistema de cenários com audiovisual e temperatura). O valor investido na automação poderá ser agregado ao preço do imóvel, logo além de conforto, o usuário possui um investimento agregado a sua residência. O controle de cargas de forma remota, seja um conjunto de lâmpadas, sistemas de ar-condicionado dentre outros, é parte indispensável quando se está desenvolvendo um sistema de automação residencial.

A empresa Legrand®, especialista mundial em sistemas elétricos e digitais, possui uma linha de equipamentos que propõe tornar uma residência mais automatizada. Essa linha chama-se *Home Automation,* com a qual conseguimos controlar equipamentos em nossa residência de forma local e ou remotamente, através de tecnologias *wireless*.

1.1 Justificativa

O produto atual da linha de *Home Automation* Legrand automatiza uma residência utilizando a tecnologia sem fio ZigBee, no entanto, só realiza tal tarefa através de uma rede pessoal *PAN,* não podendo controlar cargas remotamente ou ainda controlar cargas via *Smartphone*. Para acionar cargas remotamente ou via celular existe a necessidade de adquirir um *Gateway* conversor de ZigBee para TCP/IP que integra estas funcionalidades ao sistema. Este equipamento agrega um custo maior, que em instalações grandes não chega a ser um problema, mas em instalações pequenas com menos de dez pontos de automação, o *Gateway* acaba sendo o componente mais caro da Automação. Sendo assim o usuário que deseja investir em uma pequena automação para sua residência (menos de dez pontos) não pode controlar remotamente suas cargas.

Portanto pretende-se desenvolver neste trabalho, uma solução que visa conectar os dispositivos a *Internet* de forma que o usuário não precise adquirir um *Gateway* a parte.

1.2 Definição do problema

Apesar de existirem várias soluções que permitem o controle de cargas remotas, algumas delas impedem a integração direta com *smartphones* e outros equipamentos que se comunicam através de redes TCP/IP. Algumas empresas comercializam módulos que implementam um *middleware* de comunicação entre a rede legada e a rede TCP/IP, porém, isso significa um custo que em alguns casos inviabiliza a implementação dos projetos. Com essa proposta algumas soluções de automação podem ser mais viáveis economicamente, pois dispensa a necessidade de um *Gateway* conversor de ZigBee para WiFi, reduzindo o custo da solução para o usuário e permitindo que um dispositivo possa ser controlado diretamente através de celular localmente via WiFi ou remotamente via nuvem.

1.3 Objetivo geral

Desenvolver uma nova versão do produto atual da empresa Legrand, incorporando a tecnologia WiFi, para controlar equipamentos remotamente sem a necessidade de gateway, preservando as funções atuais.

1.4 Objetivos Específicos

- Pesquisar sobre tecnologias sem fio e módulos com tamanho e custos reduzidos.
- Desenvolver um hardware conceito para substituir o produto atual;
- Integrar uma interface (App) para validar o novo produto;
- Implementar recursos de segurança (autenticação);
- Construir um protótipo funcional para validação do conceito.
- Realizar testes (ciclagem e térmico) para acelerar a aparição de possíveis defeitos.

2. REVISÃO LITERÁRIA

Atualmente a automação está cada vez mais presente no cotidiano das pessoas, seja em uma compra *online* ou o controle de uma residência. A solução proposta se enquadra em um conceito de mercado no que atualmente é chamado de *Domótica*, visando atender as necessidades de automação em uma residência.

2.1 Automação residencial (Domótica)

Segundo Muratori e Bó(2017), a definição de Automação Residencial surgiu no Brasil como herança da palavra *Home Automation* utilizado no mercado americano. Isto ocorreu pelo fato de que, no Brasil, os primeiros sistemas voltados para automação eram de fabricantes americanos. Na Europa, o termo mais utilizado é Domótica – junção da palavra latina *Domus* (casa) com Robótica (controle automatizado).

Cada vez mais a indústria está desenvolvendo soluções para o setor residencial, com objetivo de automatizar processos, sistemas e até tarefas complexas existentes em uma residência. O artigo da empresa Comat Releco (2013) mostra que temos hoje um leque de opções para automatização de tarefas que visam atender diversos perfis de usuários, e cada um com um foco de tarefa diferente, são eles: temperatura de um ambiente, controle (on/off) de cargas não utilizadas, monitoramento do consumo de energia elétrica, controle de acesso à residência, segurança contra incêndio, tarefas domésticas (aspirador de pó inteligente, irrigador de gramado, abertura de persianas ou geladeira).

Figura 1 – Tipos de Automação

Fonte: Autoria Própria (2018).

Quando se trata de automação residencial, temos hoje basicamente três tipos de sistemas como mostra a Figura 1:

- *Stand Alone* (Local): Tem como objetivo automatizar uma simples tarefa (Interfone, fotocélula, sensor de presença) e não está conectado à *internet*.
- Integrados: Temos 2 ou mais sistemas conectados para automatizar uma tarefa (entrada de um condóminio).
- Casas conceitos (*Smart Home*): A casa toda estaria conectada a um gerenciador e quanto mais tempo o sistema coleta as informações do perfil do usuário, mais o sistema adapta-se a rotinas e hábitos do usuário, possibilitando um maior conforto para o morador da casa automatizada.

2.2 IoT (Internet of Things)

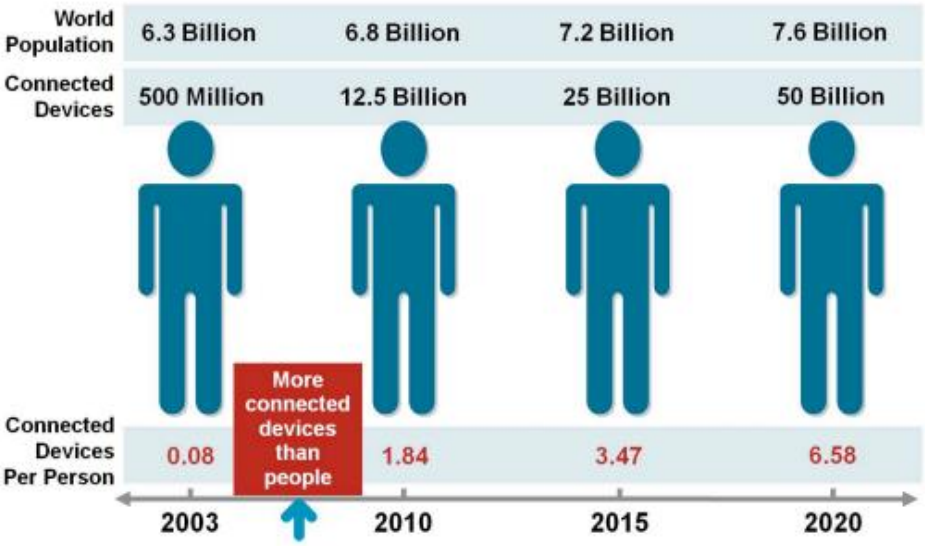
A internet das coisas ou *Internet of Things* está cada vez mais ganhando popularidade devido ao seu alto índice de expectativa e por tornar cada vez mais equipamentos que antigamente funcionavam desconectados da rede em produtos com acesso à internet.

O precursor do IoT chama-se John Romkey que em 1990 criou uma torradeira que era controlada pela internet e apresentou em uma conferência.

Esta torradeira se conectou a um computador com rede TCP/IP, se tornando o primeiro equipamento nascido manual que se conectava à internet na década de 90. (GUMPTION, 2008).

O IoT conecta os aparelhos utilizando a internet com o objetivo de automatizar tarefas, monitorar sistemas, controlar cargas, gerenciar transportes, auxiliar vendas, tornar cidades mais inteligentes (*Smart Cities*) e realizar qualquer outra tarefa que possa ser otimizada ou realizada remotamente. Uma pesquisa recente feita pela Cisco (2011), aponta que até 2020 existirão 50 bilhões de dispositivos conectados, superando em mais de 6 vezes a população mundial (Figura 4).

Figura 2 – Previsão de dispositivos conectados em 2020



Fonte: Cisco IBSG (2011).

Com uma previsão muito otimista de dispositivos conectados à internet, maior será a exposição dos dados de usuários pois cada dispositivo receberá um IP, que pode ser a porta de entrada para *hackers*. Esta vulnerabilidade pode ocorrer por falhas de construção como autenticação fraca, ausência de criptografia, redes inseguras ou usuários irresponsáveis, que não adquirem equipamentos de maior qualidade, não atualizam os equipamentos ou até mesmo não utilizam senhas com uma segurança maior.

Diante desse cenário, a revolução IoT é muito aguardada, porém existem desafios a serem tratados com mais cautela. Deve-se evitar soluções

mau construídas que seriam para facilitar e agilizar nosso dia a dia, mas que acabe expondo o usuário de forma que dados pessoais possam ser utilizados de forma inadequada.

2.3 Tecnologias de comunicação sem fio

De acordo com Rappaport (2009), o primeiro meio de tecnologia para comunicação sem fio veio no século XIX com o advento do rádio e sua capacidade de fornecer contato contínuo entre dois pontos em movimento. Esta tecnologia inicial foi utilizada nas navegações inglesas e foi o início da necessidade de se transmitir dados sem o uso de cabeamento dedicado ao serviço. A tecnologia via rádio evoluiu muito deste então permitindo um maior alcance, mais qualidade e confiabilidade na transmissão de dados, mas isso não impediu que outras formas de comunicação sem fio surgissem como alternativa para outras demandas de curto alcance por exemplo ou de grandes pacotes de dados.

a) BLUETOOTH

Surgiu no final da década de 90 com o propósito de conectar dispositivos sem utilização de fios e com um baixo custo de produção. A tecnologia utiliza uma frequência de rádio de onda curta (2.4 GHz) e pode conectar até 8 dispositivos (utilizando a rede Piconet) com uma distância máxima de 10 metros e velocidade de transmissão de dados de até 2,1 Mbps (Bluetooth 2.0). Por ser de curto alcance ele colabora na segurança dos usuários. Outra forma para aumentar a segurança é a possibilidade de ativar/desativar a função de pareamento automático nos dispositivos até a exigência de um Token para iniciar o pareamento. (CÂMARA, 2012).

b) ZIGBEE

Com poucos anos de existência, o padrão ZigBee chegou no início dos anos 2000 como uma alternativa de conexão de dispositivos dispensando o uso de cabos. Ele opera em frequências de 2,4GHz (padrão global), 915MHz(América) e 868 MHz (Europa). Sua velocidade de transmissão é inferior em relação ao Bluetooth, onde alcança a taxa máxima de 250kbps na frequência

de 2,4GHz. No entanto, seu alcance é maior, podendo chegar de 10 a 100 metros de distância.

Seu conceito foca em baixo consumo de energia, baixo custo, implementação simples, nível de segurança apropriados para a aplicação e sua topologia de implementação permite a criação de redes do tipo *Mesh*.

A transmissão ZigBee é muito utilizada nas propostas de conectar sensores a atuadores, onde o sensor realiza a leitura de um sinal e enviar um comando de ação para um atuador distante, que não necessite a transmissão de um pacote de dados muito grande. (PINHEIRO, 2004).

c) LORA

LoRa (**Lo**ng **Ra**nge – Longo alcance) foi patenteado recentemente pela empresa Semtech®, e seu foco de aplicação é conectar dispositivos de longas distâncias ou de difícil acesso, podendo chegar de 3 a 4 quilômetros de distância em áreas urbanas e 10 ou mais quilômetros em áreas rurais (A patente garante mais de 20km em condições favoráveis). Opera nas frequências de 169 MHz, 433 MHz, 868 MHz (Europa) e 915 MHz (América) e consegue taxas de transmissão de até 22kbps.

O LoRA seria a camada física da rede, e o LoRaWAN é a implementação da rede. O LoRaWAN é implementado com 2 níveis de segurança onde existe segurança do dado a ser transmitido e segurança na transmissão do dado.

Lora está com muitas expectativas de ser a base da transmissão da *Internet das Coisas*, pelos motivos de possuir baixo custo, baixo consumo, sinais propagarem mesmo com barreiras (paredes, morros, árvores), longas distâncias de transmissão e facilidade de implementação. (NUNES, 2017).

d) WiFi

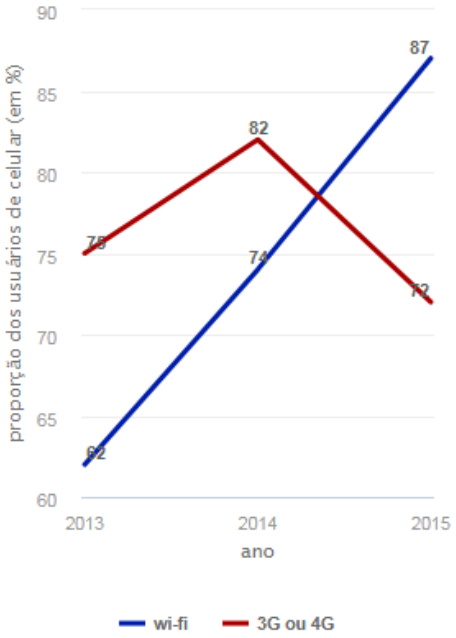
Pertencente ao padrão IEE802.11 o método de transmissão sem fio, WiFi surgiu no final da década de 90, mas que se popularizou nos anos 2000 com a inundação de notebooks e smartphones equipados com esta tecnologia.

As redes WiFi funcionam com ondas de rádio e suas distâncias de transmissão podem variar de 100 a 300 metros dependendo do ambiente (interno ou externo). Ele opera nas frequências de 2,4GHZ e 5,0GHz podendo

alcançar taxas de transferências de até 600Mbps dependendo do padrão utilizado (IEE 802.11n) (LANDIM, 2012).

Atualmente o WiFi é a tecnologia mais utilizada para acesso à internet no Brasil (Figura 5), segundo uma pesquisa feita pelo site Cetic (2015), deixando em segundo lugar a internet móvel 3G e 4G.

Figura 3 – Acesso à internet por tipo de conexão



Fonte: TIC Domicílios (2015).

O padrão WiFi é utilizado para transferências de grandes pacotes de dados, graças a alta velocidade e implementação de diversos níveis de segurança para os usuários que são WEP (Wired Equivalent Privacy), WPA (WiFi Protected Acess) e WPA 2 (WiFi Protected Acess II).

2.4 Produtos usados em domótica atualmente

No mercado atualmente existem empresas nacionais e internacionais, que desenvolvem soluções na área da automação residencial, são eles:

IHouse (The Innovative House), é uma empresa nacional que possui uma linha de produtos para controle de lâmpadas (*dimmer* e cenários),

hidromassagens, acionamento de equipamentos através de Infravermelho, chuveiros inteligentes entre outros. Utilizam a tecnologia ZigBee para acionamento das cargas e é necessário um gateway para fazer acessos remotos. A Tabela 1 mostra um comparativo entre o produto deles com o produto atual da Legrand. (IHOUSE, 2018).

Tabela 1 – Comparativo produto IHouse com o produto atual Legrand

	Módulo de Relé RP212 IHouse	**Switch Small LEGRAND**
Saídas	2x1800W	2x2000W
Tecnologia sem Fio	ZigBee	ZigBee
Acesso Remoto	Não	Não
Acesso local	Sim	Sim
Acesso por Smartphones	Não	Não

Fonte: Autoria própria (2018).

MyWay é uma empresa brasileira que desenvolve soluções residenciais nas áreas de controle de persianas, *dimmers*, som ambiente, painéis de controle e entre outros. Utilizando seus produtos, eles garantem acesso remoto das cargas. A Tabela 2 mostra um comparativo entre o produto deles com o produto atual da Legrand. (WAY, 2016).

Tabela 2 – Comparativo produto MyWay com o produto atual Legrand

	Módulo de Relé MI_RLY_L MyWay	**Switch Small LEGRAND**
Saídas	1x500W	2x2000W
Tecnologia sem Fio	RF	ZigBee
Acesso Remoto	Não	Não
Acesso local	Não	Sim
Acesso por Smartphones	Não	Não

Fonte: Autoria própria (2018).

FIBARO é uma empresa com sede em diversos países, que possui equipamentos de automação residencial com *designer* inovadores. Seus focos são sensores de temperatura, sensor de inundação, sensor de movimento, tomadas inteligentes, *dimmers* e entre outros. Também não possuem controle de cargas remotamente sem o uso de um gateway adicional. A Tabela 3 mostra um comparativo entre o produto deles com o produto atual da Legrand. (FIBARO, 2018).

Tabela 3 – Comparativo produto FIBARO com o produto atual Legrand

	Double Switch FGS-223 FIBARO	**Switch Small LEGRAND**
Saídas	2x1200W	2x2000W
Tecnologia sem Fio	Z-Wave	ZigBee
Acesso Remoto	Não	Não
Acesso local	Sim	Sim
Acesso por Smartphones	Sim	Não

Fonte: Autoria própria (2018).

2.5 Produto Legrand

Hoje a empresa Legrand, possui soluções para a linha de automação residencial sem fio. Possui equipamentos que controlam persianas, lâmpadas (dimmer), atuador para cargas diversas (chave *on/off*) e acionamento de equipamentos via ZigBee ou através de infravermelho. O produto utilizado como base do desenvolvimento foi um atuador *Switch* 2 canais (Figura 2), que por meio da tecnologia ZigBee liga ou desliga um equipamento através de um comando proveniente de um controle remoto ou um interruptor tipo pulsador. (LEGRAND, 2015).

Figura 4 – Switch 2 canais (2x2000W) Legrand e controle de cenários

Fonte: Legrand Wi connect (2018).

O atuador, como mencionado anteriormente, utiliza uma tecnologia ZigBee para realizar o acionamento de cargas, o que impossibilita comandos remotos via tecnologia IP, ou seja, só realiza o controle em uma rede local ZigBee. Possui pontos positivos como dificuldade de invasões *hackers* e ausência de um plano de internet para que a solução funcione. No entanto, se o usuário estiver ausente da residência (em outro país por exemplo) e existir a necessidade de ligar as cargas ou verificar o estado delas, não será possível, pois o atuador não permite conexões com a internet, o que impossibilita realizar esta tarefa. A ausência da tecnologia IP no dispositivo atuador também não permite o controle via Smartphone/App. Para que o dispositivo seja acionado remotamente ou via Smartphone/App, a Legrand disponibiliza um gateway (Figura 3), que pode ser adquirido a parte, cuja função é converter um comando recebido pela internet em um comando ZigBee para o dispositivo conectado.

Figura 5 – Base controladora (*Gateway*) Wi-Connect Legrand

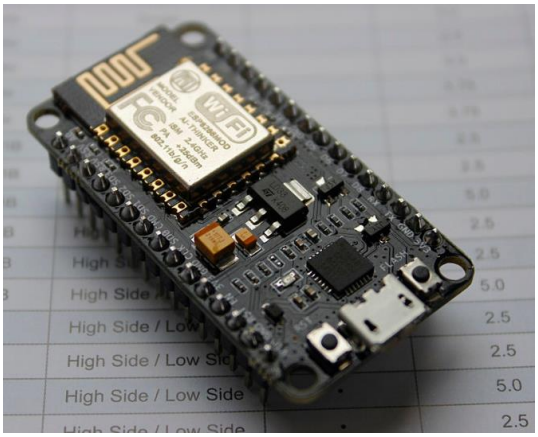
Fonte: Legrand (2018).

A base controladora pode ser adquirida a parte, e exige um investimento maior para automatizar uma residência. Este investimento extra, nem sempre o cliente está disposto a arcar, o que em algumas vezes, penaliza o usuário de ter uma casa automatizada com o conforto do acesso remoto ou até mesmo inviabiliza uma venda, principalmente em pequenas instalações.

2.6 *Node*MCU

Após pesquisar tecnologias para validar o conceito, o modelo que mais se enquadrou com a proposta deste trabalho foi um kit de desenvolvimento chamado *Node*MCU (Figura 6).

Figura 6 – *Node*MCU



Fonte: GitHub (2014).

O Kit *Node*MCU é uma plataforma de desenvolvimento *Open Source*, que embarca o modulo ESP8266 (um microcontrolador que embarca o *hardware* necessário para redes WiFi). O *Node*MCU possui um conversor USB/Serial (CP2102) integrado na qual facilita sua programação. Utiliza a IDE do *Arduino* para programação do *firmware* e disponibiliza os pinos GPIO's nos terminais barra pinos que podem ser facilmente utilizados.

Este kit foi utilizado para realização de alguns testes básicos, que determinou a escolha do módulo ESP8266 para a implementação no produto atual da Legrand.

2.7 ESP8266

O modulo ESP8266 (Figura 7) é produzido pela empresa ESPRESSIF que, segundo o fabricante, criou um microcontrolador com WiFi integrado. Alguns aspectos interessantes deste módulo são o baixo custo (menos de 5 dólares) e dimensões bastante reduzidas.

Figura 7 – ESP8266 modelo ESP-01

Fonte: Open Home automation (2015).

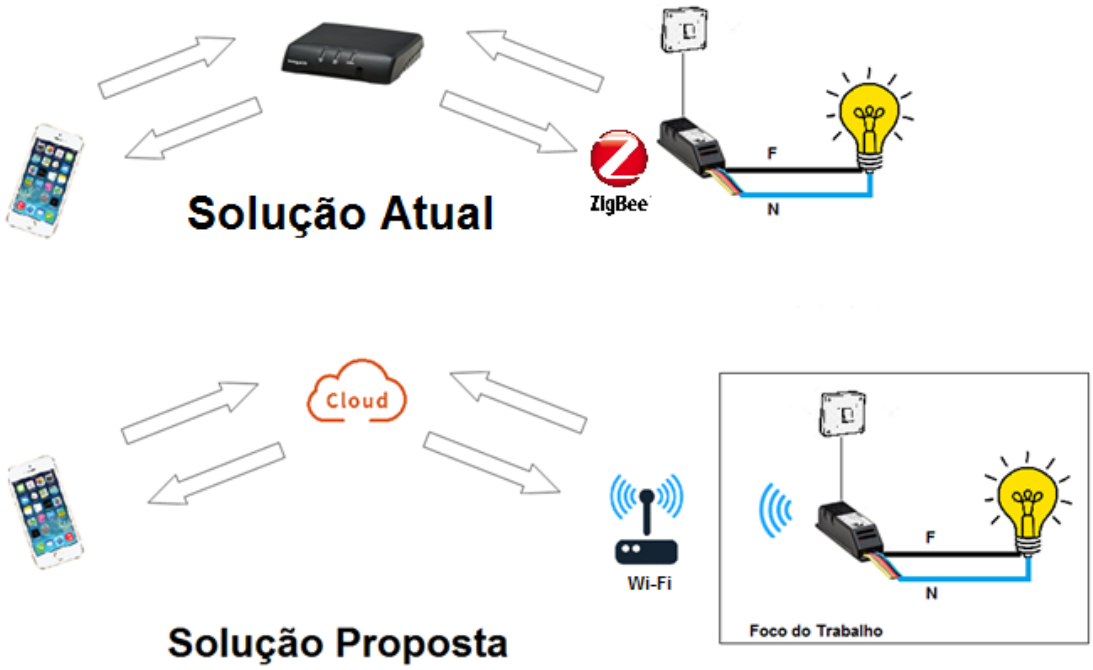
Este módulo está sendo escolhido pela maioria no quesito IoT para validação de conceitos e ele tem motivos para estar no topo da lista dos mais utilizados, pois possui alguns pontos fortes como facilidade de uso e baixo consumo.

Segundo o datasheet do fabricante, o ESP 8266 é constituído de uma CPU que opera em 80 Mhz (com velocidade máxima de 160MHz) de 32 bits. Possui uma tensão de operação entre 2,5V e 3,6V e possui periféricos como UART, SDIO, SPI, I2C, I2S, IR, GPIO, ADC e PWM. Para o WiFi ele possui o protocolo 802.11b/g/n e opera na banda de 2,4GHz. Sua memória RAM é de 32K para instruções, 96K para dados e 64K para boot. Sua memória FLASH externa é de 512K. Sua segurança utiliza WPA e WPA2 e para a rede utiliza os protocolos IPV4, TCP/UDP/HTTP.

3 DESENVOLVIMENTO

A proposta de solução é a conversão da tecnologia do dispositivo atuador, de ZigBee para WiFi, permitindo incorporar a função de atuador e de gateway no mesmo produto (Figura 8). Ou seja, haverá as duas funções, beneficiando o usuário que ao adquirir o atuador, também poderá controlar sua casa em uma rede local e externa via nuvem sem equipamentos adicionais.

Figura 8 – Proposta de solução



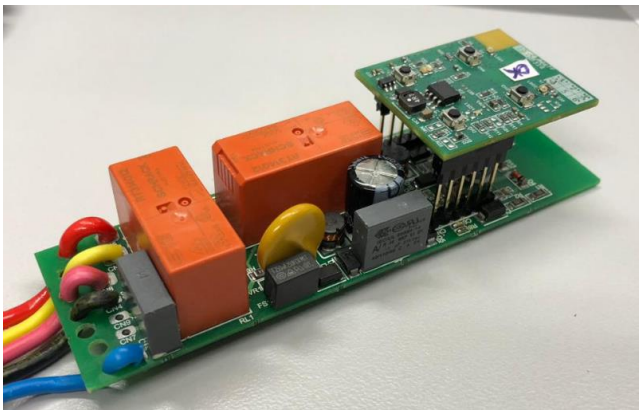
Fonte: Autoria própria (2018).

A proposta inclui também integrar uma interface para *smartphones*, que não existe nos dispositivos nativamente, deixando o usuário dependente do controle de cenários, de páginas *Web* ou de aplicativos de terceiros(pagos), mas mantendo um acionamento remoto em caso de problemas com internet ou ausência de uma smartphone.

3.1 Desenvolvimento do Hardware

Devido a arquitetura dos produtos, a proposta de conversão/atualização do produto pode ser feita apenas com a modificação parcial do sistema digital. Hoje o produto é constituído por duas placas conectadas (Figura 9), uma placa base analógica/potência e uma placa digital.

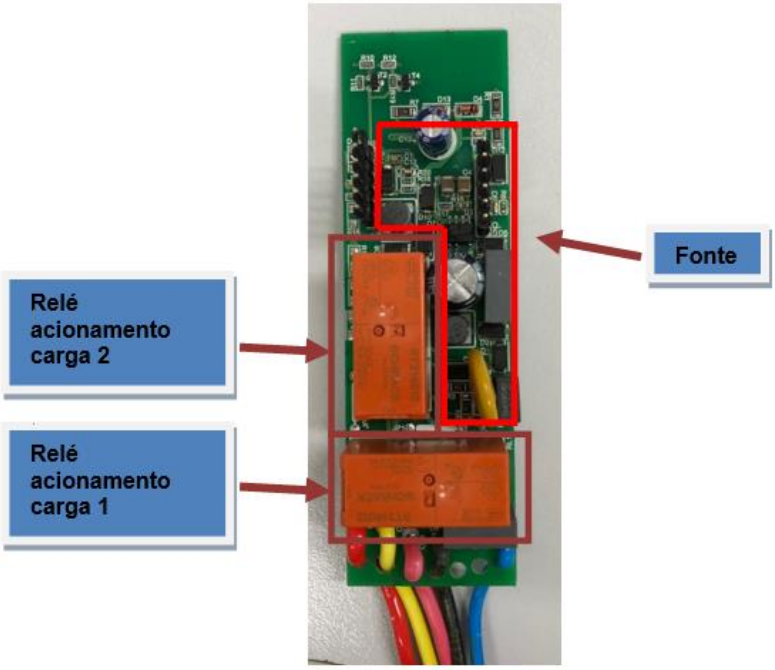
Figura 9 – Atuador *Switch* 2 canais (2x2000W) Legrand desmontado



Fonte: Autoria própria (2018).

A placa base analógica possui duas funções básicas: fonte, transforma a tensão da rede elétrica de 220V/AC para 12V/DC e comportar os relés que acionarão as cargas até 2000W cada (Figura 10).

Figura 10 – Placa base atuador *Switch* 2 canais



Fonte: Autoria própria (2018).

A segunda placa superior (digital) que pode ser vista na Figura 11, é responsável por embarcar o chip EM357 da fabricante *Silicon Labs* que é um microcontrolador que possui a tecnologia de transmissão de dados sem fio ZigBee integrado.

Figura 11 – Placa digital atuador *Switch* 2 canais

Fonte: Autoria própria (2018).

A proposta de modificação do *hardware* será aplicada apenas na placa digital. A placa da fonte não sofrerá alterações pois continuará atendendo

os requisitos e funções para qual foi projetada inicialmente o que permite a conversão do produto com menor impacto nos meios produtivos e nas assistências técnicas.

O desenvolvimento do *hardware* foi baseado em duas referências tecnológicas existentes. A primeira foi a plataforma *Node*MCU, por embarcar o módulo ESP8266 e a segunda foi o próprio produto atual, pois deverá possuir a mesma compatibilidade com a placa base existente que não sofrerá alterações e o mantimento das funções, conexões e outras características existentes no produto atual.

3.1.1 Definição do circuito

A definição do circuito foi determinada de modo que possua os pontos positivos da *Node*MCU e a compatibilidade com a placa base atual. Como a plataforma *Node*MCU é um projeto *open source*, o circuito foi analisado de forma que se pudesse identificar os blocos e utiliza-los de acordo com a proposta.

O kit *Node* MCU (**ANEXO A)** possui basicamente os seguintes blocos:

- Módulo ESP8266 (núcleo).
- Botões de configuração e gravação.
- conversor de USB para Serial.
- Referência do conversor Analógico para Digital.
- Fonte para baixar a tensão de entrada de 5V para 3V3.
- Bloco de IO’s.

Para que seja desenhado a protótipo foram propostas as seguintes modificações:

- Inclusão de quatro push buttons utilizando o conversor Analógico para Digital.
- Inclusão de um led infravermelho para acionamento de cargas com tal tecnologia (televisão, ar-condicionado, aparelho de som).
- Inclusão de quatro leds de status e um led RGB.

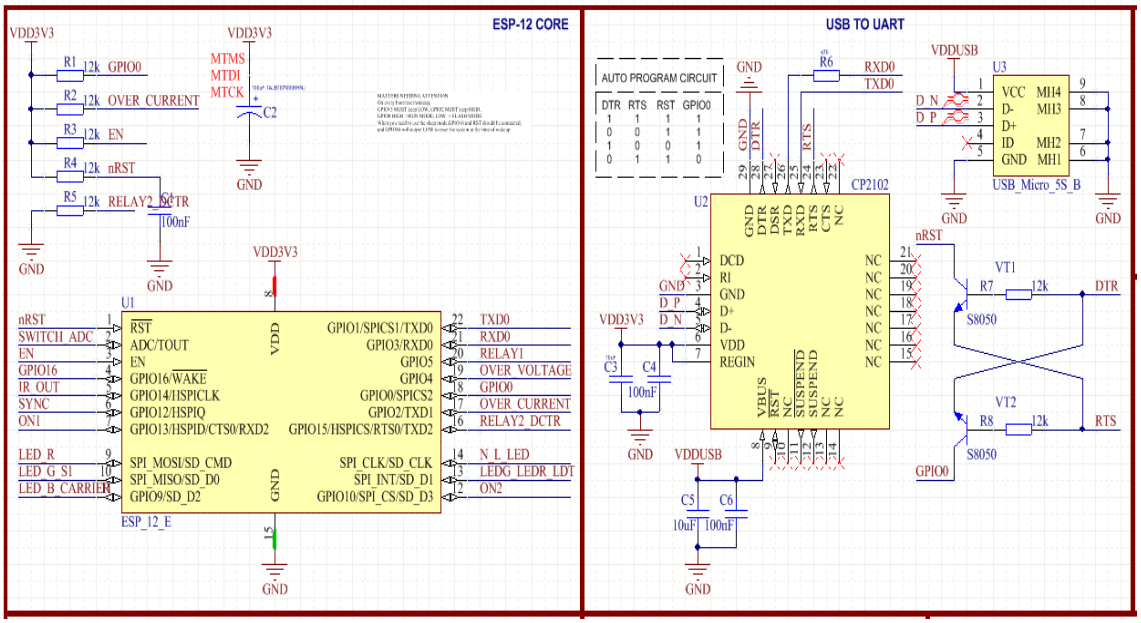
- Substituição do chip conversor USB para Serial para um chip que possuímos no estoque da empresa (CP2102).
- 2 Conectores para interconexão da placa base com a placa digital.
- Substituição do regulador de tensão para um chip que possuímos no estoque da empresa (NCP1117).

Após definidos os requisitos iniciais para atender a proposta de modificação, foi desenhado um diagrama esquemático preliminar, denominado *Protótipo 1 Revisão 0* (Rev.0).

3.1.2 Protótipo 1 (Revisão 0)

O protótipo 1 Rev.0 foi definido com base nos objetivos a serem alcançados, para esta proposta foi desenhado os circuitos que podem ser vistos nas Figuras 12, 13, 14 e 15:

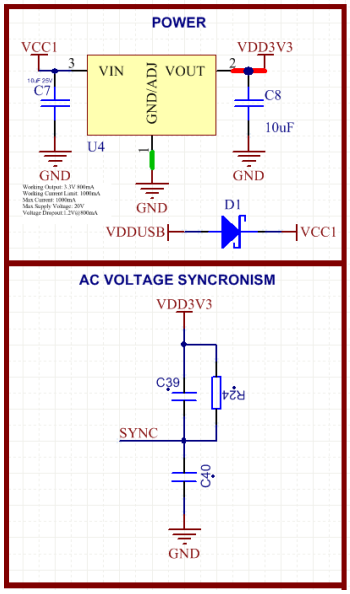
Figura 12 – ESP8266 e Conversor USB/Serial



Fonte: Autoria Própria (2018).

Como visto, foi utilizado o modulo ESP8266 como sugerido pelo Kit NodeMCU. Para a conversão USB/Serial foi utilizado o chip CP2102 na qual a empresa já possui em estoque e fornecedor.

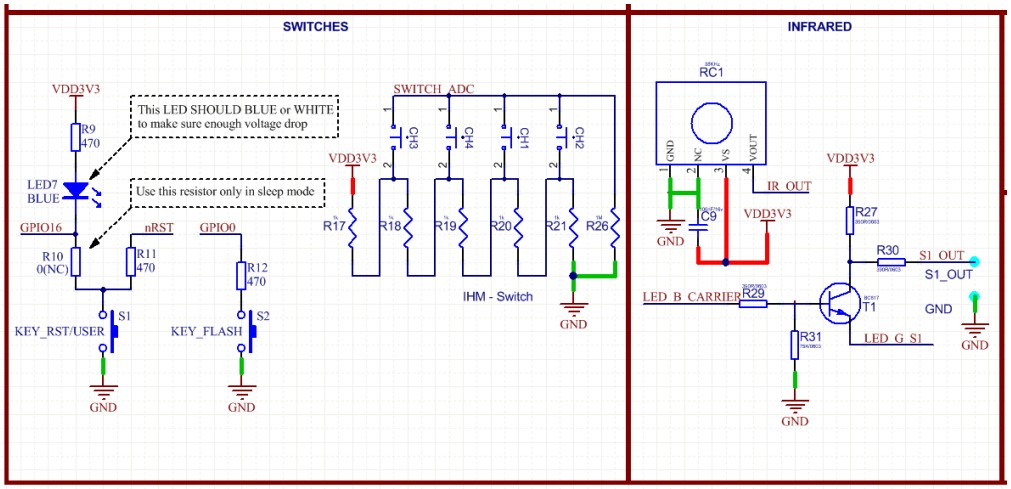
Figura 13 – Fonte 3V3 e Sincronismo da Rede



Fonte: Autoria Própria (2018).

Para a fonte, foi utilizado o regulador linear NCP1117 que tem como função, baixar a tensão fornecida pela base (12VDC) para 3V3DC. O produto da Legrand hoje possui um bloco para sincronismo com a rede, na qual será mantido para testes.

Figura 14 – Chaves analógicas e infravermelho

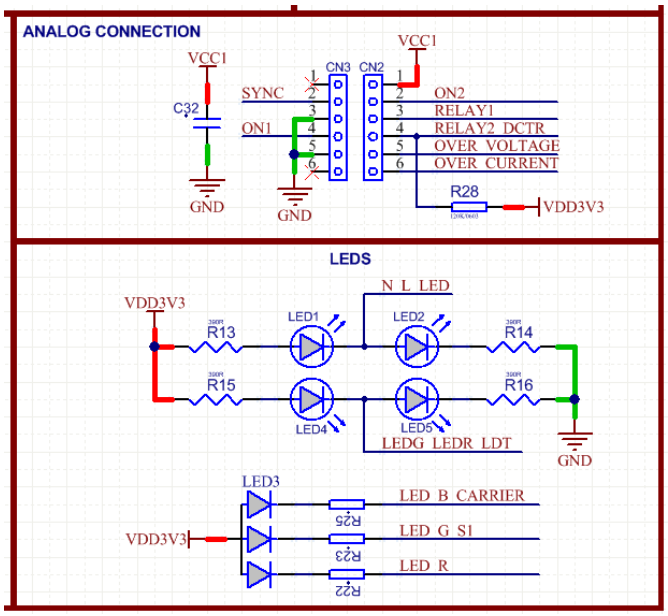


Fonte: Autoria Própria (2018).

Para o circuito das chaves digitais foi feita uma solução a partir do conversor AD pois o modulo ESP não possuía muitos pinos de IO disponíveis,

logo para se conseguir implementar a leitura de quatro chaves foi utilizada esta solução. Também foram mantidos os botões Flash para gravação e Reset para resetar o módulo durante a fase de desenvolvimento. O circuito de Infravermelho foi adicionado para implementar alguma futura solução para controle de dispositivos ativados por infravermelho.

Figura 15 – Conexão com a placa base e *led's* de *status*

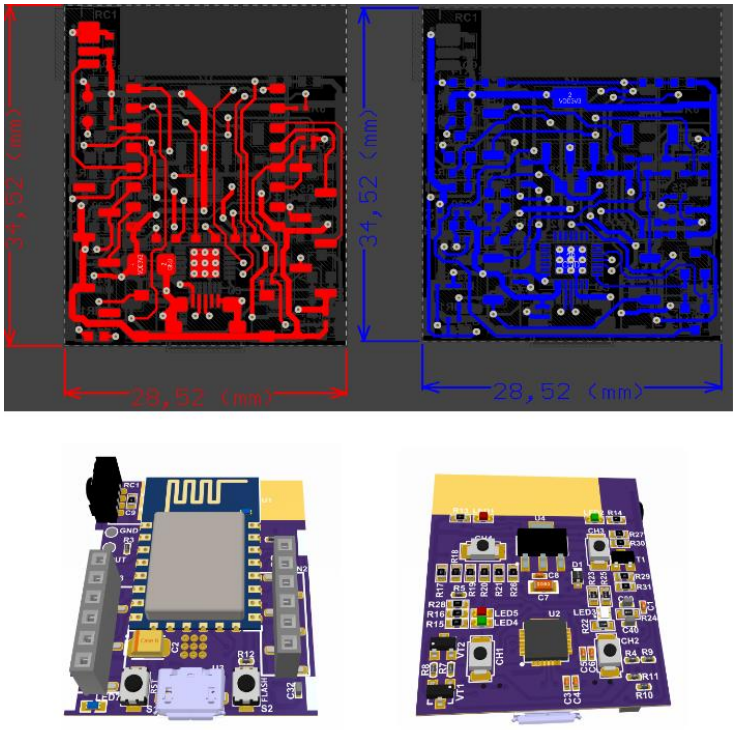


Fonte: Autoria Própria (2018).

Para a conexão com a placa base foram adicionados dois conectores (CN3 e CN2) que recebem a alimentação, e enviam comandos para acionamento dos relés da placa digital para a placa base. Também foram adicionados *led's* para sinalização de carga ativa e um *Led* RGB com o intuito de poder gerar mais cores a fim de sinalizar erros ou até mesmo indicar alguma função ativa.

O PCB foi desenhado (Figura 16) respeitando as dimensões e as mesmas conexões da placa base. O protótipo foi feito com dimensões aproximadas de 28,52mm de comprimento com 34,52mm de largura, em uma PCI dupla face (trilhas na parte de cima e trilhas na parte de baixo.

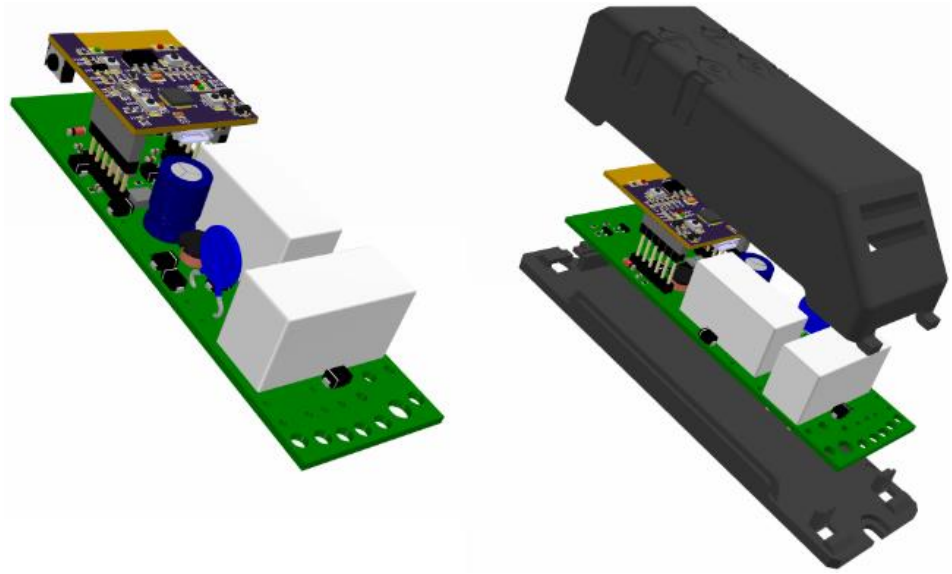
Figura 16 – *Top* (Vermelho) e *Botton* (Azul) da placa com simulação 3D



Fonte: Autoria Própria (2018).

Foi realizado uma simulação 3D de encaixe (Figura 17) da placa digital com a placa base, a fim de validar e evitar qualquer conflito mecânico possível. Garante-se assim que o protótipo fabricado encaixe na placa base com precisão milimétrica.

Figura 17 – Simulação 3D de encaixe e case explodido

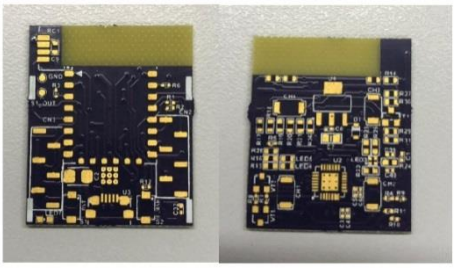


Fonte: Autoria Própria (2018).

Após realizadas as simulações 3D e seus devidos ajustes, todos os conectores de encaixe foram travados no arquivo, para que caso seja necessário realizar alguma alteração futura na placa, não venha a perder a localização e inviabilizando o encaixe na base conforme simulado.

O protótipo foi fabricado nos Estados Unidos em uma empresa chamada OSHPARK (Figura 18) com o intuito de desenvolver um novo fornecedor com um custo mais adequado para prototipagem de placas em baixa escala. O resultado da prototipagem é satisfatório para protótipos, no entanto com um prazo de entrega um pouco longo (2 meses aproximadamente).

Figura 18 – Placas recebidas da OSHPARK



Fonte: Autoria Própria (2018).

3.1.3 Resultados Hardware Rev.0

Após a montagem do protótipo, deu-se início a primeira integração com um *firmware* de teste simples para piscar *led's* e ler chaves. Nestes testes foi encontrado um problema para gravação do módulo ESP8266. O CI utilizado na conversão USB para Serial era identificado pelo computador, no entanto o *firmware* não estava sendo descarregado para o microcontrolador.

Com base neste defeito o circuito e o datasheet do microcontrolador foram analisados e encontrado um problema na utilização dos pinos da comunicação SPI do microcontrolador. Para controlar um *led* RGB acabaram utilizados pinos que são dedicados para memória *flash* externa com interface SPI, conforme mostra o datasheet do componente na Figura 19.

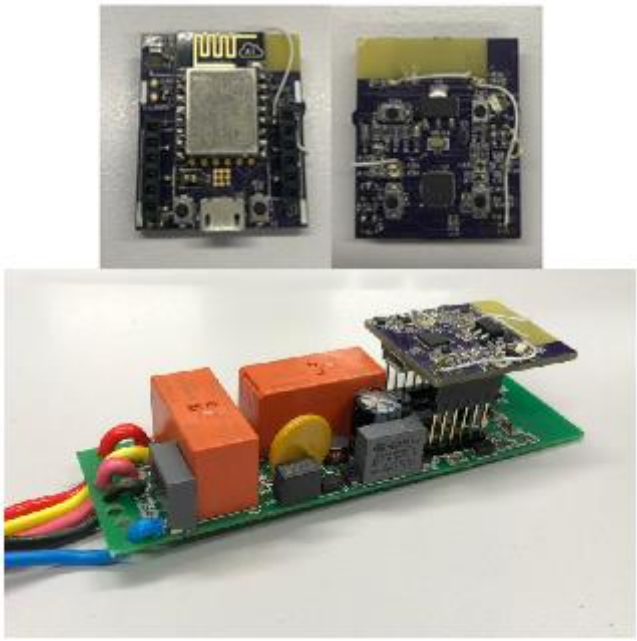
Figura 19 – Pinos utilizados na *Flash* externa

NO	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion result.Input voltage range 0-1v,scope:0-1024
3	EN	Chip enable pin. Active high
4	GPIO16	GPIO16; can be used to wake up the chipset from deep sleep mode
5	GPIO14	GPIO14; HSPI_CLK
6	GPIO12	GPIO12; HSPI_MISO
7	GPIO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input
11	IO9	GPIO9
12	IO10	GBIO10
13	MOSI	Main output slave input
14	SCLK	Clock

Fonte: AI THINKER (2015).

Desta forma, após a montagem, as trilhas que ligavam o *led* RGB foram alteradas por fora da placa (Figura 20), utilizando outros pinos de IO, permitindo assim a gravação de um *firmware* de teste para a validação do protótipo Rev.0.

Figura 20 – Correção do erro e montagem mecânica

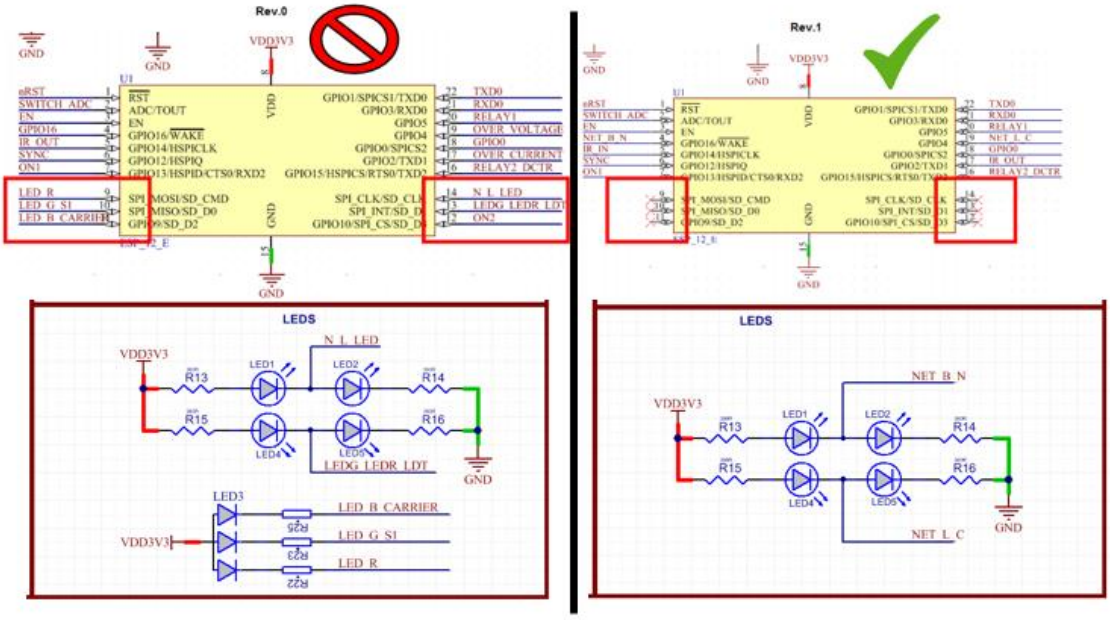


Fonte: Autoria Própria (2018).

3.1.4 Protótipo 2 (Revisão 1)

Ao montar o protótipo da revisão 0 foram identificados alguns erros de ligações, na qual impossibilitavam a gravação do firmware. Tendo isto em vista, foi gerado uma segunda revisão da placa, na qual visava corrigir esta falha (Figura 21). Desta forma foi gerada a revisão 2 (Rev.1) do protótipo para que pudesse ter uma placa funcional sem a necessidade de jumpers, o que poderia acabar gerando defeitos intermitentes devido à instabilidade das conexões.

Figura 21 – Diferenças nos protótipos Rev.0 e Rev.1

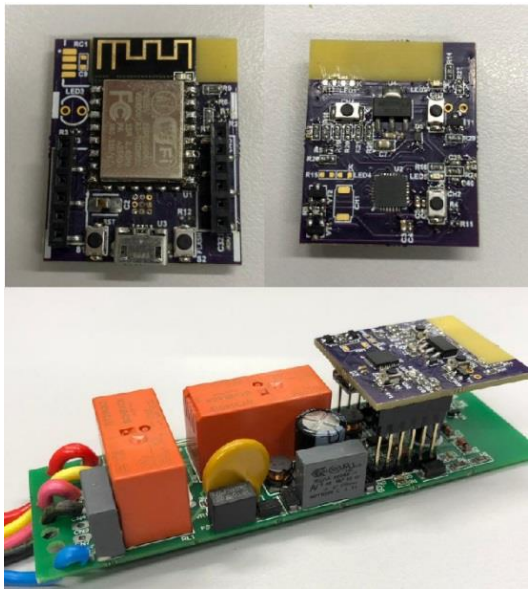


Fonte: Autoria Própria (2018).

Atualizou-se os erros de conexão da Rev.0 para melhorar o suporte a gravação criando a Rev.1, novamente fabricado pela OSHPARK.

Partindo diretamente para a montagem da PCI, não foram encontrados problemas nesta etapa e por ter realizados modificações apenas nas trilhas da PCI o protótipo encaixou perfeitamente na base como na revisão anterior (Figura 22). Após a montagem, foi gravado um *firmware* de teste sem maiores problemas, validando o protótipo e permitindo avançar para a etapa de desenvolvimento do *firmware*.

Figura 22 – Placa Rev.1 montada e encaixa na base



Fonte: Autoria Própria (2018).

3.1.5 Resultados Hardware Rev.1

Após a correção dos problemas da revisão 0, a placa Rev.1 funcionou perfeitamente, onde realizou a gravação diversas vezes de firmwares de testes aleatórios, o que permitiu dar início aos primeiros testes com uma rede WiFi.

3.2 Integração da placa base com a placa digital

Antes de iniciar o desenvolvimento do software, foi realizado uma breve tentativa de integração com a base para verificar os seguintes aspectos:

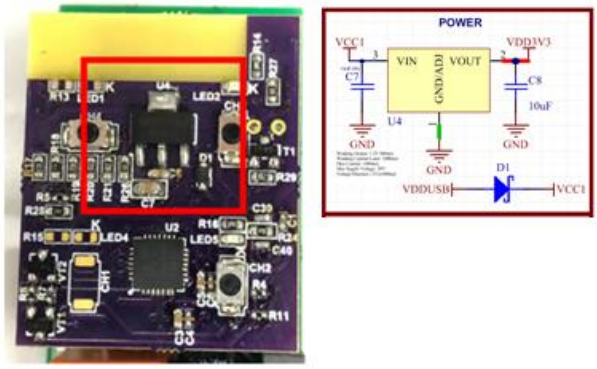
- Verificar o comportamento da placa digital sendo alimentada pela base
- Verificar o acionamento dos relés para validação do hardware
- Verificar o acionamento local através das teclas Push-Button
- Testar a integração 24h para verificar a existência de comportamentos anormais

3.2.1 Resultados Integração

Até o momento, a placa digital estava sendo alimentada e testada pelo conector USB, então foi realizado a integração da base com a placa digital e energizado na rede elétrica (220VAC) para verificar o comportamento da fonte. Foi desenvolvido e gravado um firmware que acionava os relés individualmente em um intervalo de dois segundos para validar o hardware que será utilizado posteriormente. Ao energizar a placa base, tudo funcionou como o previsto, no entanto, após aproximadamente cinco minutos de teste, a placa digital desligou e não realizou mais os acionamentos dos relés. O produto foi desligado e ligado novamente onde o mesmo voltou a operar normalmente. No entanto, após pouco tempo ligado (aproximadamente um minuto) voltou a desligar. Neste momento constatou-se um aquecimento anormal na placa digital.

Ao rastrear a origem do aquecimento incomum, o calor estava sendo produzido pelo regulador (Figura 23) linear NCP1117 (U4), que é responsável por regular a tensão da fonte da base de 12DC para 3V3DC. Neste momento foi verificado que a queda de tensão e a potência dissipada eram muito grandes, o que fazia produzir um calor excessivo na placa, e como ele foi montado sem um dissipador de calor, sua proteção térmica acabava atuando, desligando o chip e impedindo da placa digital de queimar.

Figura 23 – Regulador Linear

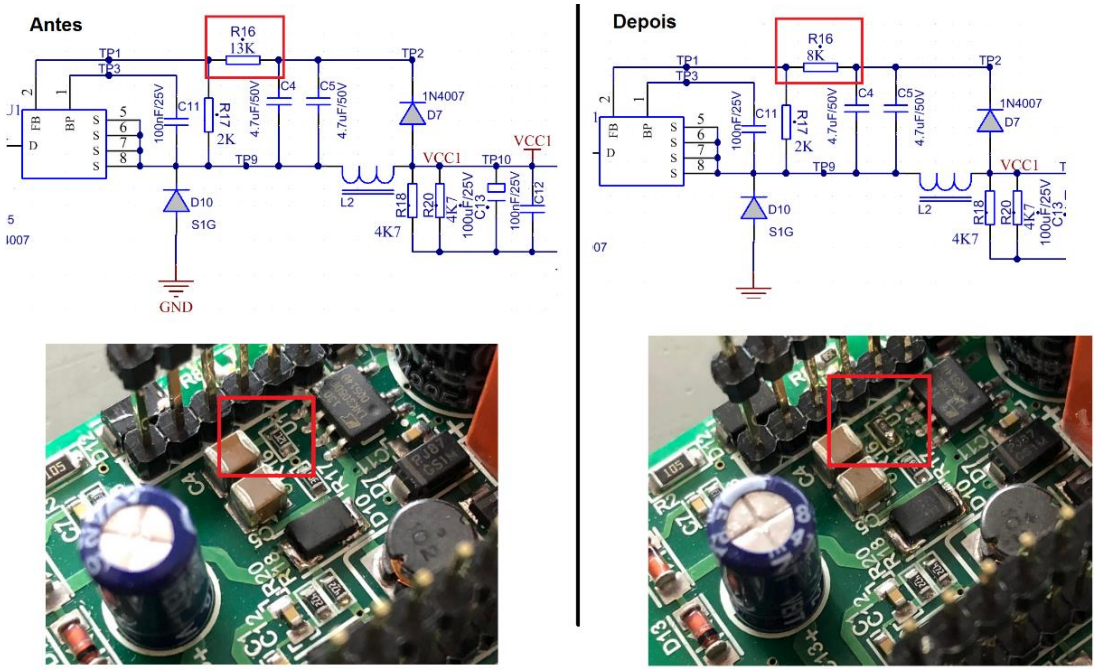


Fonte: Autoria Própria (2018).

3.2.2 Solução

Para que não existisse uma terceira revisão da placa digital, a solução mais simples e rápida encontrada foi reduzir a tensão de saída da placa base, para que o regulador linear não seja submetido a uma tensão de 12V, mas sim entre 8V e 9V. Alterou-se a realimentação da fonte para que regule a tensão da rede para aproximadamente 8VDC.

Figura 24 – Alteração placa base



Fonte: Autoria Própria (2018).

Com esta alteração, o regulador linear da placa digital passou a operar em uma tensão de entrada de 8VDC, diminuindo sua potência dissipada. Com esta tensão os relés da placa base continuam operando, no entanto, com tensão inferior a especificada pelo fabricante e com velocidade de comutação um pouco menor.

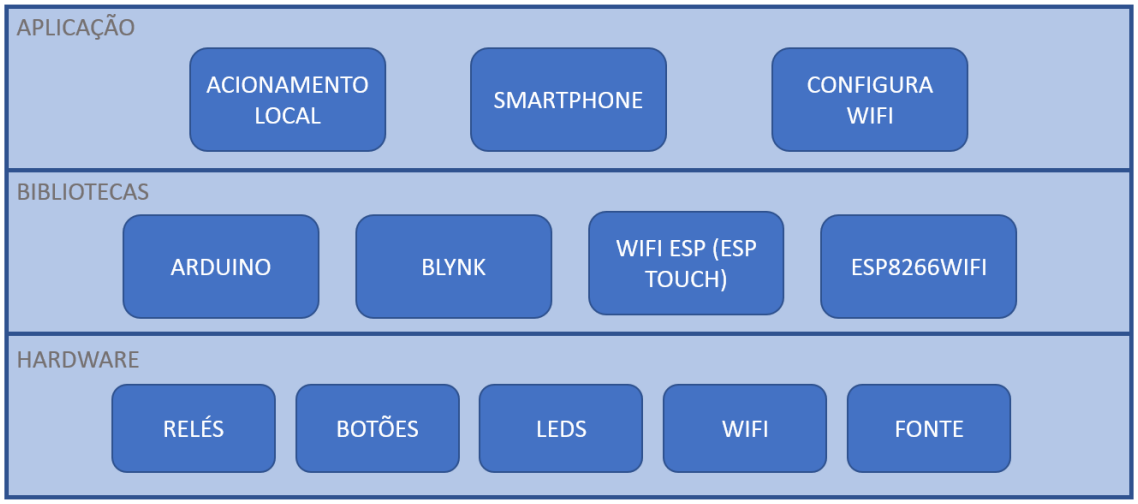
O *firmware* de teste ficou rodando durante duas horas (aproximadamente) sem problemas de aquecimento acionando ininterruptamente os dois relés das cargas.

3.3 Desenvolvimento do Software

A proposta de conversão e atualização do atual produto da Legrand, contempla uma modificação no layout da placa digital, para embarcar o modulo WiFi ESP8266, no entanto para validação dos conceitos foi necessário implementar uma interface para permitir ao usuário controlar as cargas local ou remotamente com o uso de um smartphone conectado à internet.

Para realizar a validação do conceito, foi utilizado o aplicativo *Blynk* para com interface gráfica e de rede, e o aplicativo *ESP-Touch* para conectar/comissionar o modulo ESP a uma rede WiFi, a IDE do Arduino para programação com a biblioteca *WifiManager* para gerenciamento das redes WiFi. O diagrama da figura 25 ilustra a arquitetura dos componentes do firmware do produto.

Figura 25 – Arquitetura do protótipo



Fonte: Autoria Própria (2018).

3.3.1 Arduino

Para o desenvolvimento do firmware, foi escolhido a plataforma e IDE do Arduino para desenvolvimento de um protótipo rapidamente. Também por utilizar a linguagem de programação C que foi base do curso do autor.

O Arduino possui uma biblioteca de terceiros chamada *WifiManager* que possui funções para gerenciamento das redes, onde foi utilizado para fazer as primeiras conexões do módulo ESP em rede WiFi conhecida.

3.3.2 Blynk

Para a criação da interface com smartphones foi utilizado o aplicativo Blynk que é um dos mais utilizados no desenvolvimento de IoT, Blynk (2018). Com ele é possível montar uma interface no celular, mesmo sem conhecimentos de programação e enviar comandos para um dispositivo que esteja conectado com o aplicativo. Também possui muita documentação de operação e pode ser adquirida uma versão paga para fins de desenvolvimento profissional. Todo o desenvolvimento da solução atual foi realizado na versão gratuita.

3.3.3 Integração Blynk, Arduino e Hardware

A integração inicial foi realizada através de um *sketch* (Figura 26) exemplo disponível no Github, Vshymanskyy (2017), onde ele tem como objetivo conectar o módulo ESP8266 no WiFi e integrar com o Blynk. Basicamente precisamos fornecer de forma fixa (sem interface) um SSID e uma senha de uma rede WiFi e executar a função *Blynk.run* para iniciar a sincronia.

Figura 26 – *Sketch GitHub*

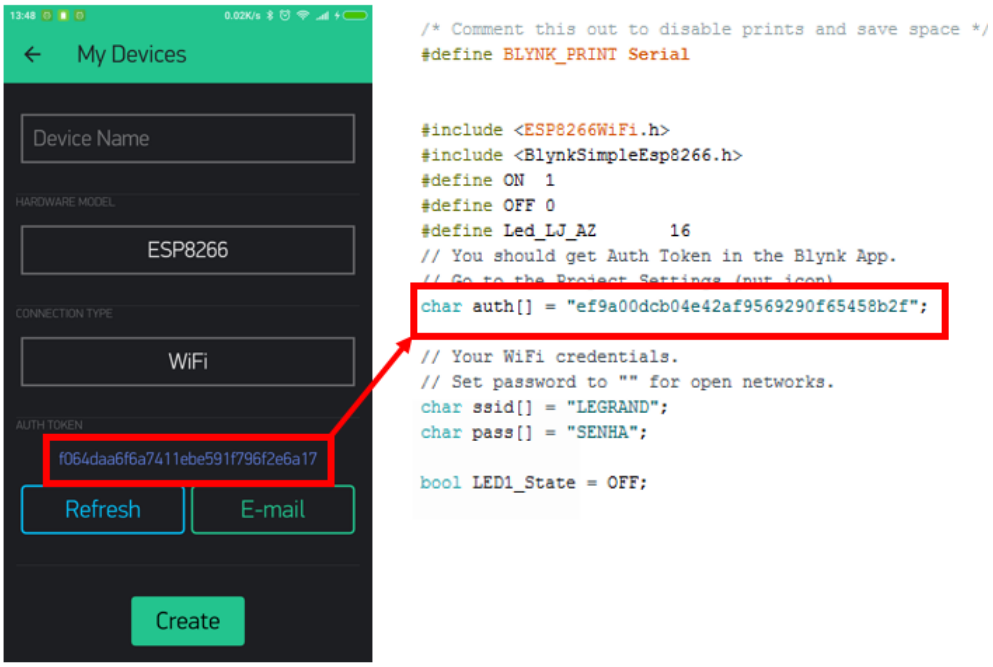
```
39  // You should get Auth Token in the Blynk App.
40  // Go to the Project Settings (nut icon).
41  char auth[] = "YourAuthToken";
42
43  // Your WiFi credentials.
44  // Set password to "" for open networks.
45  char ssid[] = "YourNetworkName";
46  char pass[] = "YourPassword";
47
```

Fonte: GitHub (2017).

3.3.4 Autenticação

Outra informação que precisamos inserir no firmware do dispositivo seria uma autenticação por *Token* que o próprio aplicativo Blynk gera como forma de segurança de informação. Ao criar um projeto novo no Blynk, ele automaticamente cria um Token e envia por e-mail para que seja adicionado no código fonte. Devemos preencher o campo do *Token* (Figura 27) conforme código gerado pelo Blynk no aplicativo.

Figura 27 – Autenticação por *Token*



Fonte: Autoria Própria (2018).

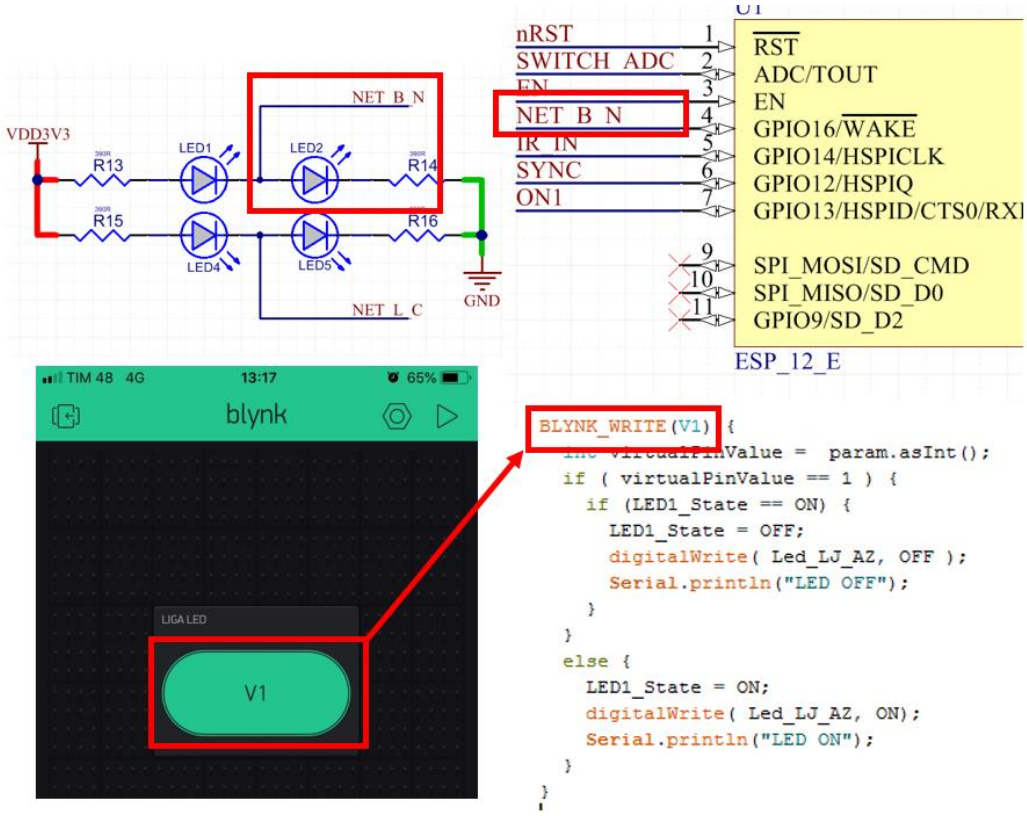
3.3.5 Envio de comandos e botões virtuais

Após conseguir conectar o Blynk ao dispositivo com *firmware* de exemplo, foi necessário validar o envio de comandos. Para isto foi utilizado uma função disponível na documentação do Blynk, Blynk(2018) que envia dados do aplicativo para o hardware do dispositivo, mais especificamente os I/O. Para isto foi utilizada função *Blynk_Write(V1) da biblioteca Blynk* que permite receber um comando enviado/escrito de um botão virtual do aplicativo Blynk. Este comando

é um valor inteiro (0 ou 1) na qual podemos implementar rotinas após a leitura do valor.

Neste caso foi implementada na função Brynk_Write o controle de um *LED* via GPIO. Ou seja, se recebe do aplicativo o valor 1, o *LED* 2 que está conectado no GPI016 do ESP8266 é ligado. Quando recebe o valor 0 o *LED* 2 é desligado, conforme ilustrado na Figura 28.

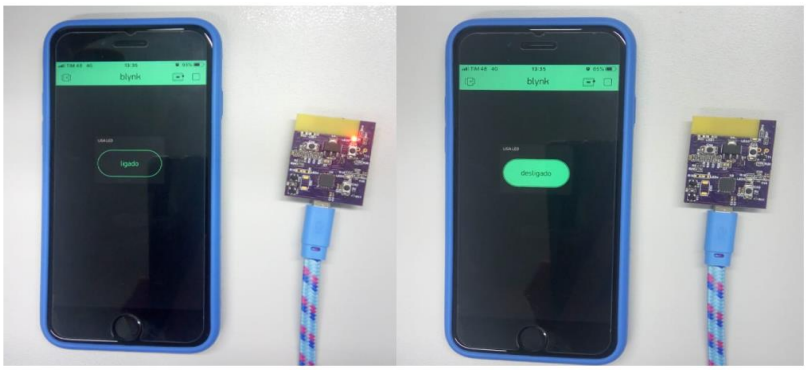
Figura 28 – Botão Virtual, *Blynk Write* e LED2



Fonte: Autoria Própria (2018).

Ao testar o envio de comandos, tanto o hardware quanto o aplicativo funcionaram da forma prevista, conforme a Figura 29, validando a função que será utilizada posteriormente.

Figura 29 – Teste prático



Fonte: Autoria Própria (2018).

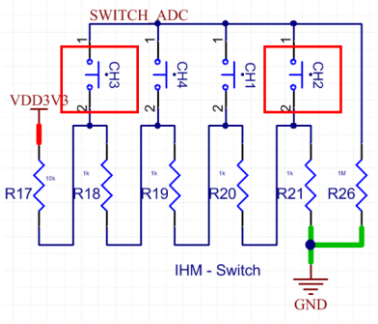
Para adequar o aplicativo a placa base que possui dois relés de acionamentos, foi necessário adicionar mais uma função *write*. O dispositivo e a placa base na qual será comandada, possui dois relés, necessitando de dois botões virtuais.

3.3.6 Acionamento local e botões

A próxima etapa no desenvolvimento foi adicionar o acionamento através dos botões locais (interruptor. Para acionamento local existe no circuito da placa digital dois botões que serão dedicados a esta tarefa. Estes botões serão lidos constantemente e caso exista um acionamento local eles também deverão comandar a carga.

Os interruptores usados serão as chaves 2 e 3 como mostra a figura 30 (CH2 e CH3). As chaves estão conectas a um circuito divisor de tensão que serão lidas pelo conversor AD do microcontrolador.

Figura 30 – Interruptores utilizados



Fonte: Autoria Própria (2018).

Ao utilizar o conversor AD para a leitura das chaves, foi necessário montar uma tabela de conversão entre tensão lida para cada chave. Logo, com a utilização da função do Arduino *AnalogRead(A0)* foi possível mapear os valores que representam cada chave como mostra a Tabela 4.

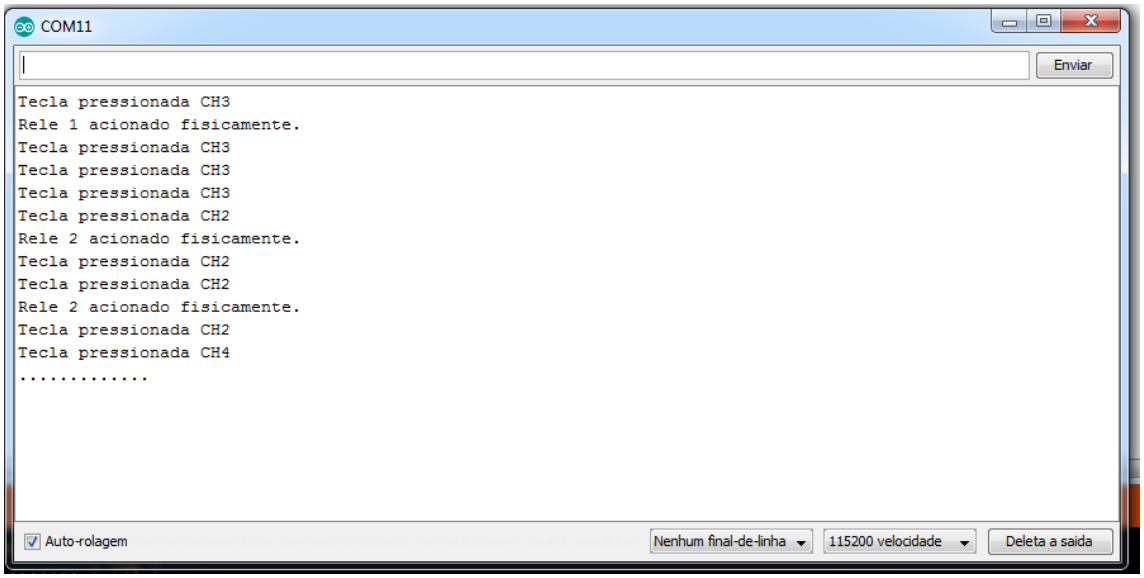
Tabela 4 – Valores das chaves no conversor AD

CHAVE LIDA	VALOR APROXIMADO
CH1	477
CH2	255
CH3	937
CH4	753

Fonte: Autoria Própria (2018).

Depois de descobrir os valores que o conversor AD gera para cada chave, foi necessário implementar uma rotina de leitura das chaves para saber qual chave está sendo acionada e poder acionar os respectivos relés/cargas. A rotina então, compara a leitura com os valores mapeados, identifica a tecla pressionada e gera um *log* (Figura 31) através da serial para a verificação do hardware.

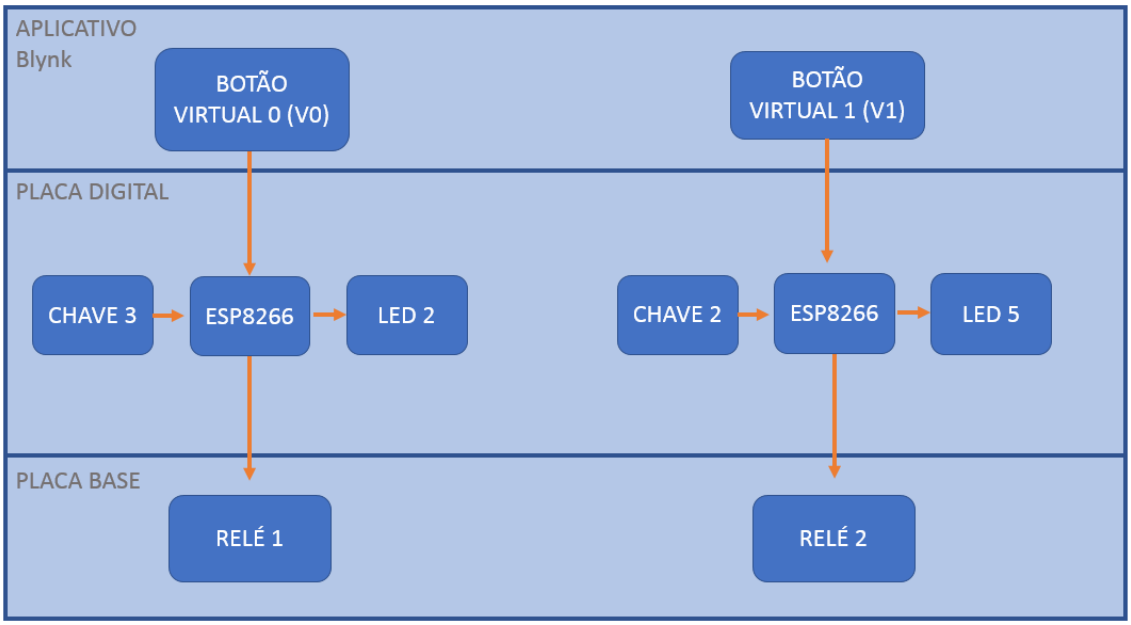
Figura 31 – *Log* da leitura das chaves



Fonte: Autoria Própria (2018).

Foi associado os botões aos *LEDS* de *status*. Ficou definido que a chave 2 (CH2) acionará o relé 2 da placa base e o *LED* de *status* 5 (LED5). A chave 3 (CH3) acionará o relé 1 da placa base e o *LED* de *status* 2 (LED2). Este mapeamento de funcionamento está ilustrado na figura 32.

Figura 32 – Mapa de acionamentos



Fonte: Autoria Própria (2018).

3.3.7 *Feed-back* do *status* da tecla no aplicativo

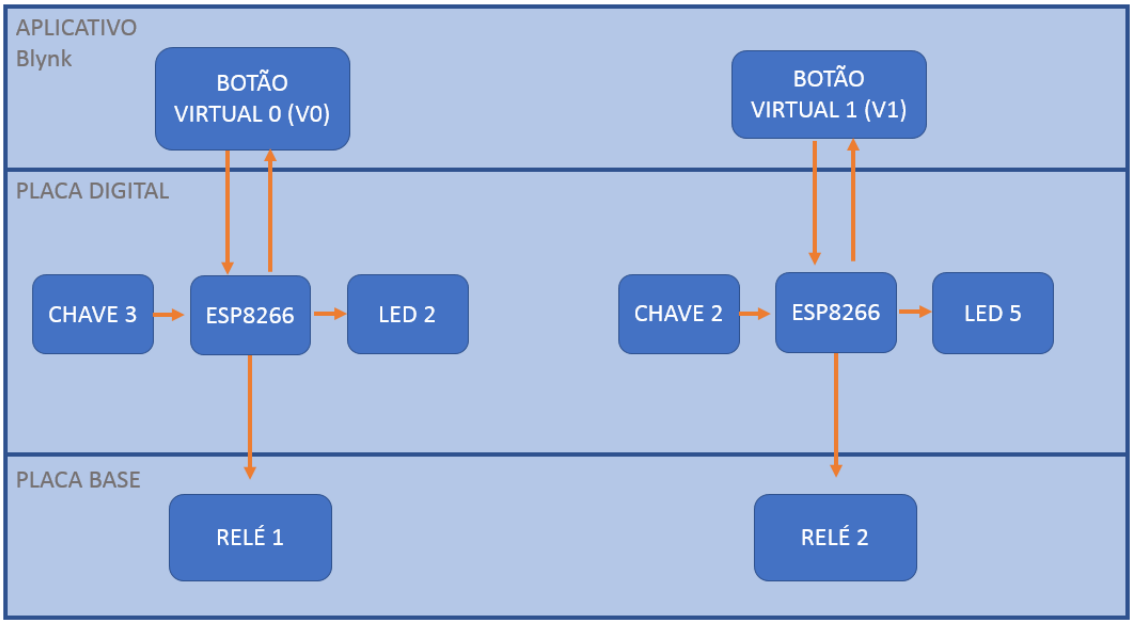
Ao implementar a rotina de leitura dos botões locais e acionamento dos relés da placa base, foi detectado a necessidade de atualizar o status da tecla pressionada também no aplicativo, comportamento chamado de *feedback* de carga. Sem o *feedback,* quando era realizado um acionamento local (na placa digital) o aplicativo não atualizava o status do seu botão virtual, sendo necessário em algumas situações pressionar a tecla duas vezes para inverter o estado da carga. Para resolver este problema foi necessário implementar uma rotina onde uma tecla acionada localmente, além de acionar a carga associada a tecla, também enviasse o acionamento para o aplicativo.

Ao analisar a documentação do aplicativo Blynk, há uma função (*Blynk.virtualWrite*) que envia comandos para o aplicativo, podendo desta forma

atualizar o aplicativo com as informações do que está acontecendo no *hardware*. Foi modificada a rotina da leitura da tecla, acrescentando uma variável de *status* para o envio da realimentação. A modificação funcionou e, ao realizar um acionamento local, o botão do aplicativo também alterava o status.

Para representar melhor estes comportamentos, foi necessário atualizar o mapa dos acionamentos conforme a Figura 33.

Figura 33 – Mapa de acionamentos e atualização de *status* no aplicativo



Fonte: Autoria Própria (2018).

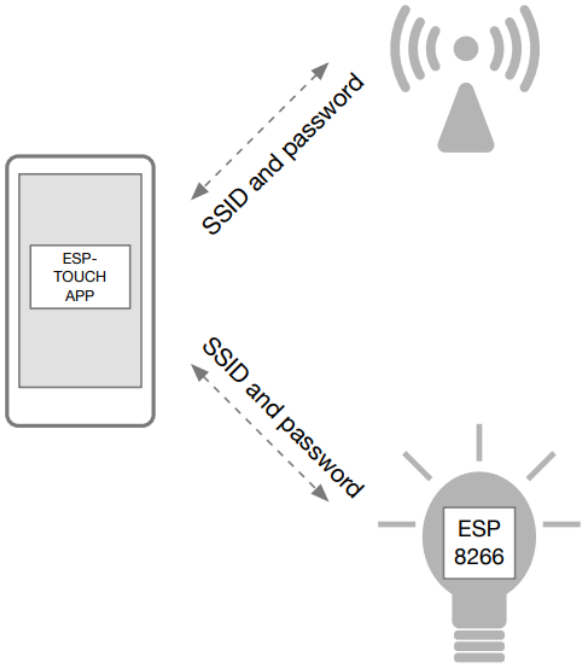
3.3.8 Conectando um dispositivo no WiFi com ESPTouch (SmartConfig)

Nos primeiros *firmwares* de teste, a configuração do WiFi era inserida no código fonte, fazendo com que para trocar de WiFi seja necessário modificar o *firmware*, compilar e gravar o dispositivo. O aplicativo Blynk não disponibiliza uma interface para conectar o dispositivo em uma rede WiFi, processo conhecido também por comissionamento, por isso, foi necessário encontrar uma solução para conectar o ESP8266 a uma rede WiFi de forma simples para o usuário final

Durante as pesquisas de desenvolvimento foram encontradas duas formas de conectar o ESP8266 a rede WiFi, uma através da biblioteca WiManager e outra através de um método que a própria fabricante do ESP8266 (Espressif) disponibiliza chamando de *ESP Touch*. Neste método, um celular

conectado a uma rede WiFi se conecta a um ESP8266 que está sem rede configurada, e envia as informações (SSID e senha) para o ESP se conectar na mesma rede WiFi que o celular se encontra, conforme ilustrado no diagrama da figura 34.

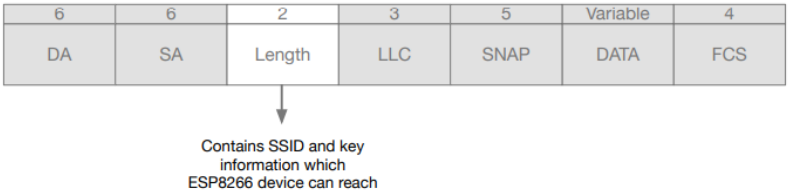
Figura 34 – Aplicação ESP-TOUCH



Fonte: Espressif (2016).

O método utiliza a função disponível na biblioteca do ESP8266 para Arduino chamada *SmartConfig* que coloca o ESP8266 no modo *STATION*, um tipo de modo onde o ESP8266 fica monitorando o tráfego de todas as redes WiFi ao seu alcance. Enquanto isso a aplicação ESP-Touch que roda em um smartphone fica enviando pacotes (Figura 35) via UDP que contém a SSID e senha da rede WiFi na qual o celular está conectado. Desta forma, quando o ESP8266 que está monitorando os tráfegos das redes encontra este pacote ele se auto configura e entra na rede WiFi com os dados encontrados.

Figura 35 – Estrutura do pacote de dados enviado pelo ESP-Touch



Fonte: Espressif (2016).

Este método de conexão segundo o blog Automalabs (2017), é uma forma prática de conectar o dispositivo, mas não segura pois ele (ESP-Touch) codifica as informações (SSID e senha) em um campo de transmissão do WiFi que qualquer um pode ver o tamanho do pacote, e este tamanho corresponde a uma letra basicamente. Desta forma qualquer outro dispositivo compatível com o SmartConfig pode ver este trafego e recuperar as credenciais de acesso.

Como a proposta é uma validação de conceito, foi implementado esta forma de conexão devido a sua praticidade e facilidade de conexão. A estrutura para funcionamento desta aplicação consiste basicamente em chamar a função SmartConfig para que o dispositivo ESP8266 entre em *Mode Station*, então foi associado uma tecla (CH4) para ficar dedicada a esta tarefa conforme ilustra a Figura 36.

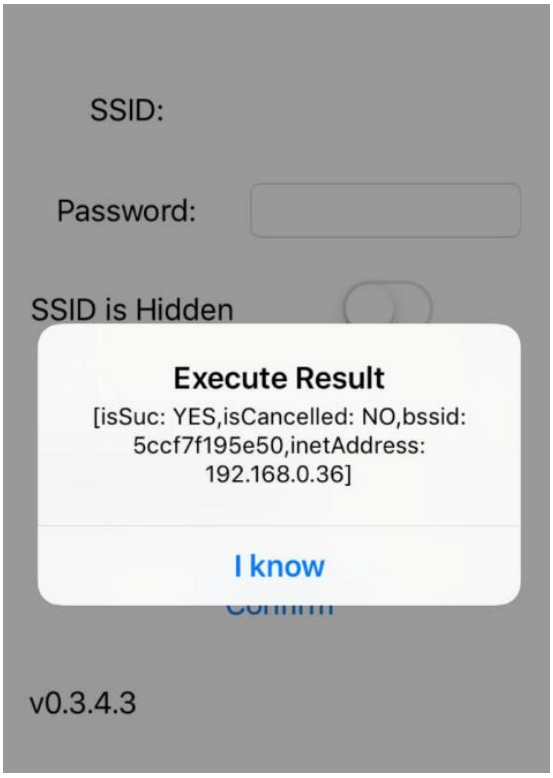
Figura 36 – Chamada da função *SmartConfig*

```
//main loop
void loop() {
  tecla_lida = le_teclado();
  if (tecla_lida == TECLA_NET) {
    smart_config();
  }
```

Fonte: Autoria Própria (2018).

Quando pressionada a tecla CH3, o ESP entra em modo *Station* e fica aguardando o SSID e senha proveniente do aplicativo ESP-Touch. Quando as informações são passadas para o dispositivo, o aplicativo retorna uma mensagem (Figura 37) de sucesso para o usuário saber que a conexão deu certo.

Figura 37 – *Feed-Back* Esp-Touch

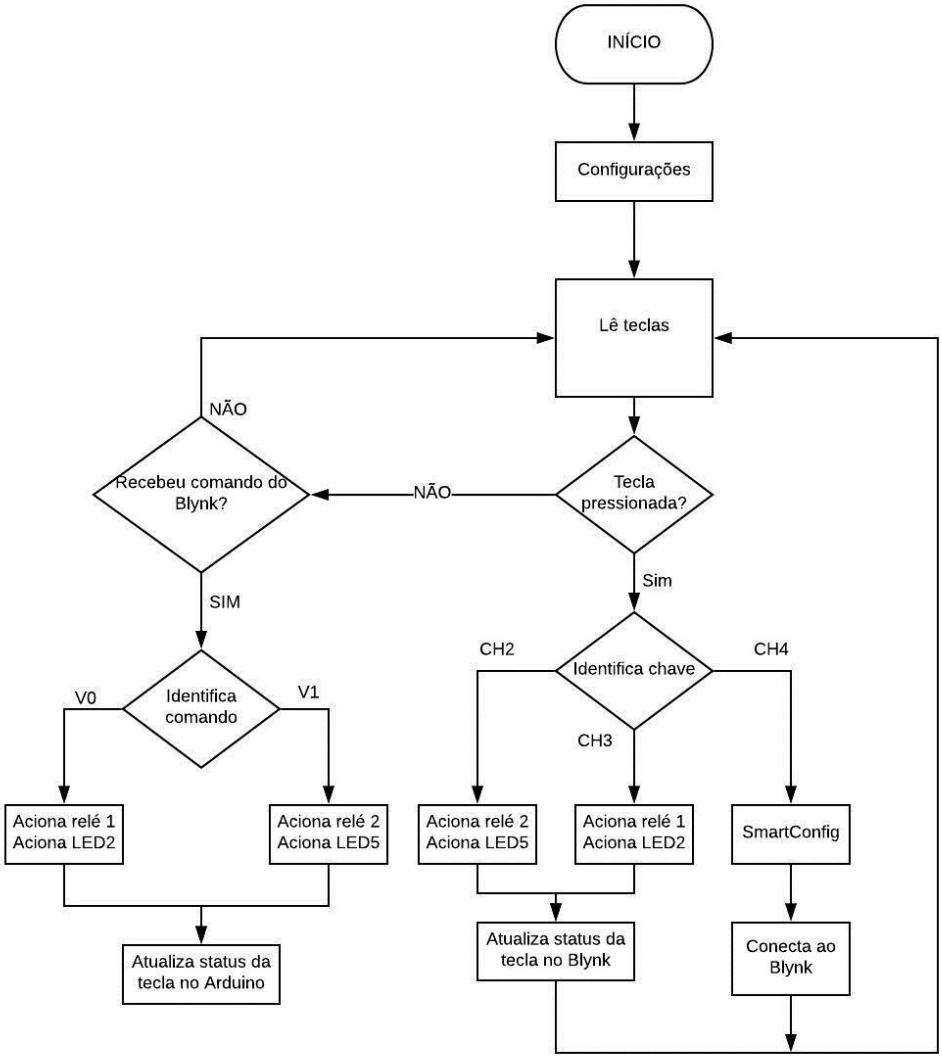


Fonte: Autoria Própria (2018).

3.3.9 Fluxo geral da aplicação

Ao finalizar a última etapa (conectar o ESP8266), o sistema ficou pronto para validação do conceito. Desta forma a aplicação completa pode ser vista em um fluxograma (Figura 38) que ajuda a entender a sequência de operação. Apesar de ser uma aplicação simples, é o suficiente para validar o conceito proposto pelo trabalho

Figura 38 – Fluxograma *Firmware*



Fonte: Autoria Própria (2018).

Basicamente ao ligar o dispositivo a rede elétrica, ele faz as suas inicializações e fica em um loop aonde ele monitora se houve alguma tecla pressionada e se recebeu algum comando do aplicativo *Blynk*. Caso receba um comando proveniente do aplicativo (carga acionada através do *smartphone*) ele identifica o comando, liga ou desliga a carga respectiva e atualiza o status da carga para o *smartphone*. Feito isso ele retorna para o loop de tecla ou comando. O mesmo ocorre para algum evento do hardware, caso identifique um botão pressionado (carga acionada localmente) ele identifica o botão pressionado, liga ou desliga a carga e atualiza o *status* no aplicativo. Após a função realizada, ele retorna para o laço de tecla ou comando.

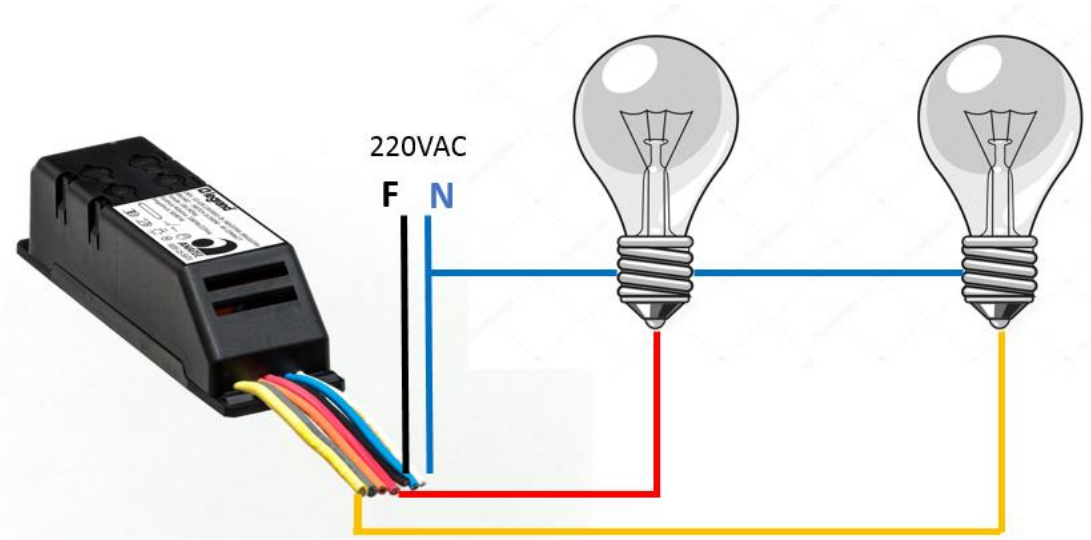
4 RESULTADOS

Alguns testes foram realizados para análise do comportamento da solução. Além destes, outros testes foram realizados para testar a robustez do equipamento com recursos e ensaios internos na empresa. Ao final dos ensaios, o protótipo final foi finalizado e montado com os componentes do **Anexo B**.

4.1 Teste funcional

A montagem para a verificação do funcionamento é descrita na figura 39 onde sugere duas lâmpadas como carga, lembrando que poderiam ser qualquer carga de até 2000Watts por saída.

Figura 39 – Cenário de instalação



Fonte: Autoria Própria (2018).

Com o cenário montado (Figura 40), a proposta realizou o acionamento das lâmpadas normalmente, tanto pelo aplicativo quanto localmente sem indícios de defeitos. O produto foi montado com case transparente para ficar melhor a visualização das placas e a montagem mecânica.

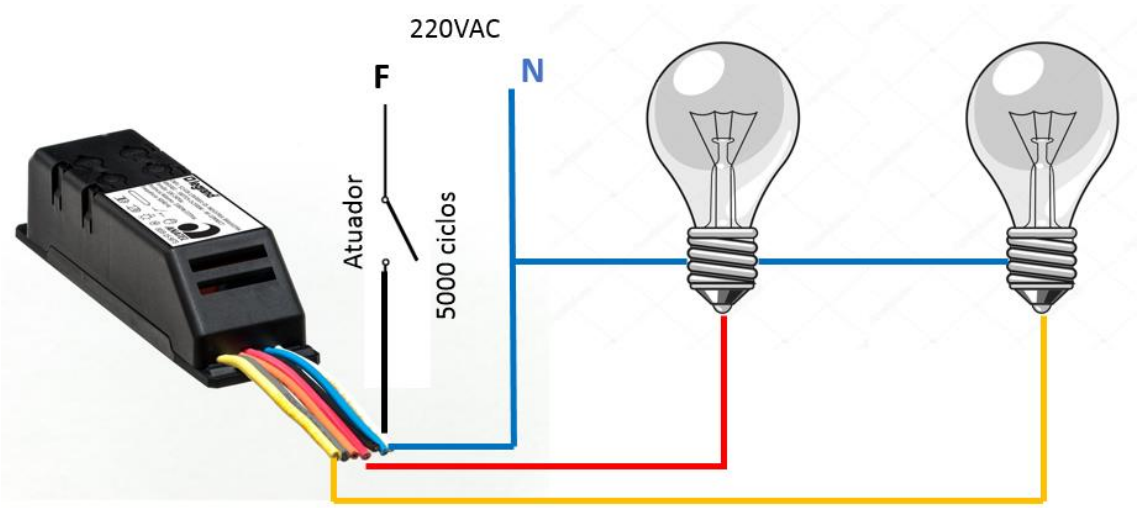
Figura 40 – Teste prático

Fonte: Autoria Própria (2018).

4.2 Teste de ciclagem

Após os testes funcionais, foi realizado um ensaio de ciclagem, que consiste em desligar a energia elétrica e ligar novamente em um intervalo de 1,5 segundos durante 5000 ciclos para verificar a resistência a quedas de energia. O teste consiste em inserir um atuador (chave) em série com o condutor fase que energiza o produto para simular quedas de energia como ilustra a Figura 41.

Figura 41 – Teste de Ciclagem diagrama



Fonte: Autoria Própria (2018).

O atuador utilizado foi um Arduino Nano conectado a um relé (Figura 42), onde foi feito um firmware para acionar o relé 5000 vezes em um intervalo de 1,5 segundos. Como resultado, não foi detectado nenhum problema durante estes ensaios.

Figura 42 – Teste de ciclagem pratico

Fonte: Autoria Própria (2018).

4.3 Ensaio Térmico

O ensaio térmico visa analisar a resistência do produto todo em situações que a temperatura ambiente é mais elevada, forçando os componentes a trabalharem em condições mais severas. Isto faz com que acelere os surgimentos de problemas, podendo então verificar quais circuitos são mais sensíveis ou que estão mal dimensionados.

Para isto, o produto foi colocado em operação durante 24 horas em uma estufa (Figura 43) e a temperatura foi ajustada para 50º Celsius. Durante as 24 horas de operação o produto não apresentou falhas nem perda de funções

Figura 43 – Teste térmico

Fonte: Autoria Própria (2018).

5 CONCLUSÃO

A proposta deste trabalho consistiu em desenvolver uma nova versão do produto atual (*Switch* 2 canais) da empresa Legrand, incorporando a tecnologia de transmissão sem fio WiFi, sem a necessidade de um *gateway*.

A solução contou com duas revisões de placa para embarcar o módulo ESP8266. A primeira revisão alguns pinos dedicados a uma memória externa do ESP8266 foram usados, o que implicou em problemas de gravação do *firmware*. Na segunda revisão da placa, foi detectado um problema de aquecimento no regulador linear responsável por regular a tensão da fonte para alimentar o ESP8266.

No desenvolvimento do software, muitos dos problemas encontrados foram na aplicação e integração com o aplicativo Blynk, como por exemplo o *feed-back* de *status* do *hardware* para o aplicativo e o comissionamento do módulo ESP8266 na rede WiFi que pode existir brechas de segurança.

Alguns testes foram realizados na solução, o primeiro teste foi de ciclagem para verificar se a partida da fonte poderia acabar queimando a solução e o segundo testes foi térmico para acelerar possíveis defeitos. Após os ensaios nenhum problema foi encontrado.

Com o intuito de aprimorar a solução para trabalhos futuros, alguns pontos poderiam ser melhorados, que são eles:

- Substituir o regulador linear por um conversor buck.
- Pesquisar formas seguras de comissionamento do modulo ESP8266.
- Implementar notificação de cargas esquecidas ligadas quando o usuário se ausenta da residência.
- Utilizar um SDK para substituir a IDE do Arduino.

Apesar de alguns problemas encontrados durante o desenvolvimento da solução, os objetivos específicos foram alcançados comprovando que é possível modificar o produto atual da Legrand, sem grandes impactos nas linhas de produção e assistência técnica.

REFERÊNCIAS

AUTOMALABS. **ESP8266:** Usar ESP Touch / SmartConfig é seguro?. 2017. Disponível em: <https://www.automalabs.com.br/esp8266-usar-esp-touch-smart-config-e-seguro/>. Acesso em: 27 out. 2018.

BLYNK. **Send data from app to hardware.** 2018. Disponível em: <https://docs.blynk.cc/#blynk-main-operations-send-data-from-app-to-hardware>. Acesso em: 15 out. 2018.

CÂMARA, Marlon. **Bluetooth:** O que é e como funciona. 2012. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/01/bluetooth-o-que-e-e-como-funciona.html>. Acesso em: 01 out. 2018.

COMAT RELECO (São Paulo). **Domótica e Automação Residencial – Residências Inteligentes.** 2013. Disponível em: <https://www.comatreleco.com.br/domotica-automacao-residencial/>. Acesso em: 27 ago. 2018.

ELÉTRICA, Saber. **Automação Residencial e Casa Inteligente:** Bem vindo ao futuro. 2018. Disponível em: <https://www.sabereletrica.com.br/casa-inteligente/>. Acesso em: 30 out. 2018.

ESPRESSIF. **ESP Touch User Guide.** 2016. Disponível em: <https://www.espressif.com/sites/default/files/30b-esp-touch_user_guide_en_v1.1_20160412_0.pdf>. Acesso em: 25 out. 2018.

FIBARO. **Home Automation:** Smart Home. 2018. Disponível em: <https://www.fibaro.com/us/>. Acesso em: 21 ago. 2018.

GUMPTION. **What Would Your Toaster Say to Your TV?**. 2008. Disponível em: <http://gumption.typepad.com/blog/2008/05/what-would-your.html>. Acesso em: 27 ago. 2018.

IHOUSE. **Pioneirismo em automação residencial.** 2018. Disponível em: <http://www.ihouse.com.br/ihouse.php>. Acesso em: 21 ago. 2018.

LANDIM, Wikerson. **O que é Wi-Fi.** 2012. Disponível em: <https://www.tecmundo.com.br/wi-fi/197-o-que-e-wi-fi-.htm>. Acesso em: 05 out. 2018.

LEGRAND. **Wi Connect.** 2015. Disponível em: <http://www.legrand.com.br/automacao/wiconnect//>. Acesso em: 21 ago. 2018.

MURATORI, José Roberto; BÓ, Paulo Henrique dal. **Automação Residencial:** Conceitos e Aplicações. 2. ed. Belo Horizonte - Mg: Educere, 2017. 200 p.

NUNES, Bruno. **Introdução a Lora, NB-IoT e SigFox.** 2017. Disponível em: <https://www.embarcados.com.br/lora-nb-iot-e-sigfox/>. Acesso em: 05 out. 2018.

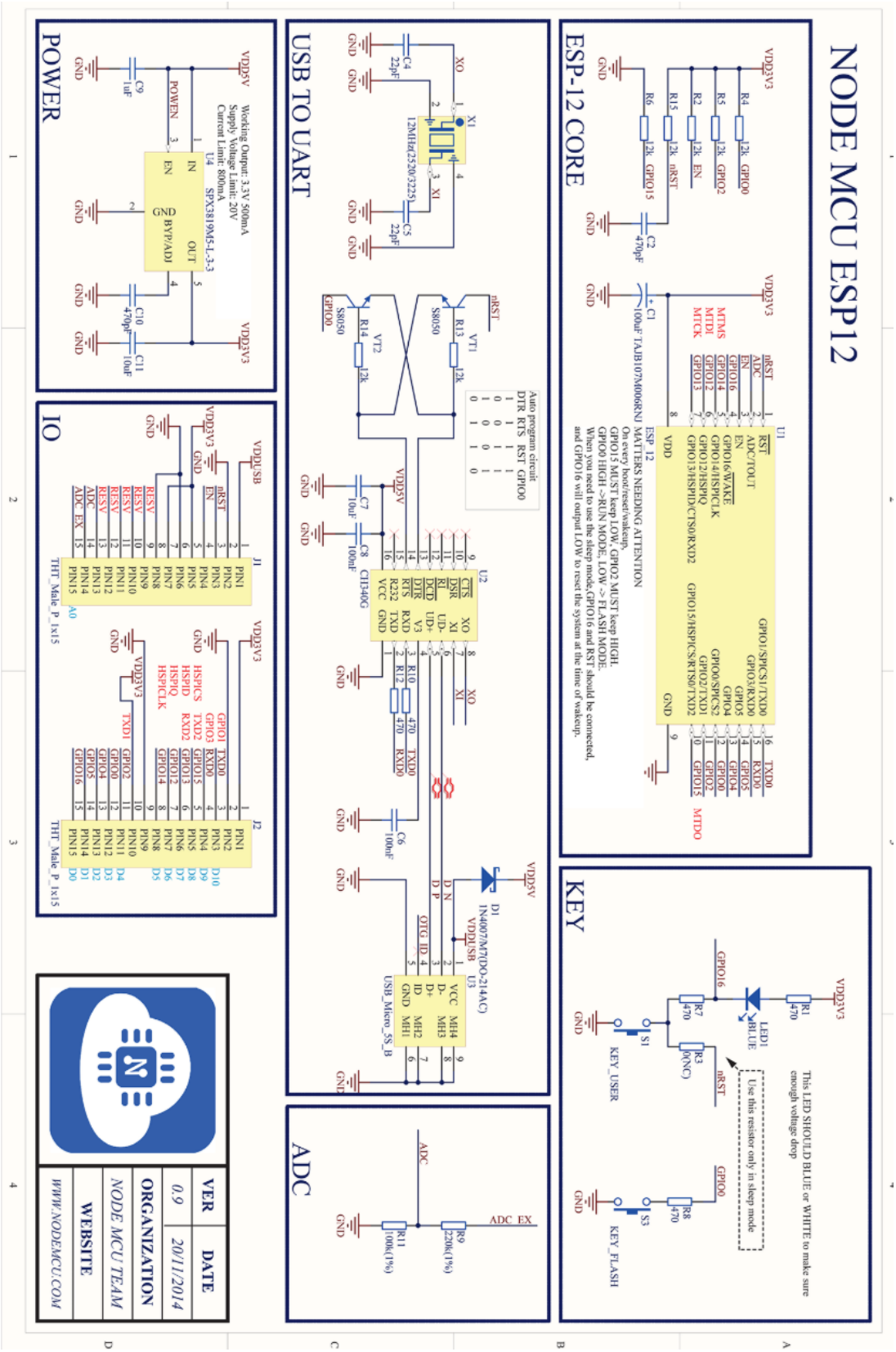
PINHEIRO, José Mauricio Santos. **As redes com ZigBee.** 2004. Disponível em: <https://www.projetoderedes.com.br/artigos/artigo_zigbee.php>. Acesso em: 05 out. 2018.

RAPPAPORT, Theodore S. **Comunicações sem fio:** Princípios e práticas. 2. ed. São Paulo: Pearson, 2009.

VSHYMANSKYY. **ESP8266 Standalone.** 2017. Disponível em: <https://github.com/blynkkk/blynk-library/blob/master/examples/Boards_WiFi/ESP8266_Standalone/ESP8266_Standalone.ino>. Acesso em: 25 out. 2018.

WAY, My. **Automação de sistemas.** 2016. Disponível em: <https://sistemamyway.com.br/>. Acesso em: 21 ago. 2018.

ANEXO A Esquemático NodeMCU



ANEXO B Lista de Materiais

legrand®

LISTA DE MATERIAIS

DESCRIÇÃO	QUANTIDADE	POSIÇÃO
CAPACITOR SMD 100nF/16V 0402	4 PC	C1, C4, C6,C32
CAPACITOR TANTALO 100uF/16V 3,5mm X 2,8m	1 PC	C2
CAPACITOR SMD 10uF/16V 0402	3 PC	C3, C5, C8
CAPACITOR SMD 10uF/25V 0805	1 PC	C7
CAP.CER.SMD 100nF +/-10% 16V 0603 X7R GP C	1 PC	C9
CAPACITOR SMD 1nF/50V 00603	2 PC	C39, C40
CHAVE TACTIL SMD	4 PC	CH1, CH2, CH3, CH4
CONECTOR MINI MODUL SMD 180 Graus 1x6	2 PC	CN2, CN3
DIODO SMD BAT54 0603	1 PC	D1
LED LARANJA SMD 0603	1 PC	LED1
LED VERDE SMD 0603	2 PC	LED2, LED4
LED VERMELHO PTH 3mm	1 PC	LED3
LED VERMELHO SMD 0603	1 PC	LED5
RESISTOR SMD 12K 0402 +/-5% 1/10W	6 PC	R1, R3, R4, R5, R7, R8
RESISTOR SMD 1K 0402 +/-5% 1/10W	1 PC	R2
RESISTOR SMD 470R 0402 +/-5% 1/10W	3 PC	R6, R11, R12
RESISTOR SMD 0R 1/10W 5% 0603((1608)	2 PC	R9, R10
RESISTOR SMD 390R 1/10W 5% 06003 (1608)	4 PC	R13, R14, R15, R16
RESISTOR SMD 10K 1/10W 5% 06033(1608)	1 PC	R17
RESISTOR SMD 1K 1/10W 5% 0603((1608)	5 PC	R18, R19, R20, R21, R29
RESISTOR SMD 47K 0603 +/-5% 1/10W	1 PC	R24
RESISTOR SMD 1M 1/10W 5% 0603(1608)	1 PC	R26
RESISTOR SMD 390R 1% 0,1W 50V 0603 CINTA	1 PC	R27
RESISTOR SMD 120K 1% 0,1W 50V 0603 CINTA	1 PC	R28
TRANSMISOR INFRAVERMELHO 38Khz TSOP62	1 PC	RC1
CHAVE TACTIL SMD	2 PC	S1, S2
TRANS. SMD NPN BC847 SOT23 CINTADO	1 PC	T1
ESP-12 Wi-Fi Module by AI-Thinker	1 PC	U1
CONVERSOR USB/SERIAL CP2102	1 PC	U2
CONECTOR MICRO USB FEMEA SMD	1 PC	U3
REGULADOR LINEAR NCP1117	1 PC	U4
TRANS. SMD NPN BC847 SOT23 CINTADO	2 PC	VT1, VT2

Total:	60 PC