Artigo disponibilizado *online*

Revista Ilha Digital

Endereço eletrônico:
http://ilhadigital.florianopolis.ifsc.edu.br/

INSTITUTO FEDERAL
SANTA CATARINA

DESENVOLVIMENTO DE PRODUTO PARA AUTOMAÇÃO RESIDENCIAL COM SISTEMA DROIDLAR

Bruno Rodrigues Silva[1], Clovis Antônio Petry[2]

Resumo: Desde os primeiros anos do século XXI, a automação residencial vem deixando de ser futuro para tornar-se cada vez mais presente. Em 2011, José Roberto Muratori, diretor executivo da Aureside (Associação Brasileira de Automação Residencial), afirmou que o número de projetos em automação residencial cresceu em média 35% nos últimos anos, citando como os principais motivos: a redução do custo de projeto e o perfil das pessoas mais jovens já adeptas às tecnologias que compram imóveis atualmente. Neste contexto está contido o DroidLar, um sistema de automação residencial que permite controlar diferentes equipamentos de uma residência utilizando celulares Android como interface de controle, desenvolvido e aplicado em ambiente acadêmico. Como forma de aproximar o sistema de laboratório ao usuário final, este trabalho apresenta o desenvolvimento dos produtos SAR (servidor de automação residencial) e Controlador, elementos do sistema DroidLar, descrevendo as etapas de definição do gabinete, hardware e firmware dos produtos de modo que sejam implantados em um ambiente real, tornando o sistema mais próximo de uma solução destinada ao usuário final.

Palavras-chave: Automação Residencial. Android. Sistema Embarcado. *Hardware*.

Abstract: *Since the early years of this century, home automation is leaving to be a future to be increasingly present. In 2011, José Roberto Muratori, CEO of Aureside (Brazilian Association of Residential Automation), said the number of projects in home automation grew on average by 35% in recent years, citing as the main reasons: reduction of cost of design and the profile of younger's people and fans of technologies who buy real estate today. In this line is included the DroidLar, a home automation system that lets you control different devices of a residence using Android phones as control interface, developed and applied in academic environment. As a way of approaching the laboratory system to the end users, this work presents the development of the SAR and the Controller of DroidLar system, describing the steps of definition of the enclosure, hardware and firmware of the products so that they are deployed in a real environment, making the system closer to a solution for the end user.*

Keywords: *Home Automation. Android. Embedded System. Hardware.*

[1] Especialista em Desenvolvimento de Produtos Eletrônicos pelo IFSC, *campus* Florianópolis <brsilva.eng@gmail.com>.
[2] Professor do Departamento Acadêmico de Eletrônica (DAELN), *campus* Florianópolis, IFSC <petry@ifsc.edu.br>.

1. INTRODUÇÃO

A automação de uma residência visa, principalmente, proporcionar conforto, segurança e praticidade para as tarefas do dia a dia. Os sistemas de automação residenciais podem ser descentralizados, ou seja, a interface de controle do usuário envia comandos para os controladores de equipamentos residenciais diretamente, ou centralizados, ou seja, todo o controle da automação é realizado por um Servidor de Automação Residencial (SAR).

O sistema DroidLar foi desenvolvido por Euzébio (2011) com a proposta de ser um sistema de automação residencial centralizado, possibilitando o controle de equipamentos domésticos e eletroeletrônicos de uma residência, utilizando como interface de controle o celular com sistema operacional Android. A solução é composta por três elementos: cliente Android, controlador de iluminação e servidor de automação residencial. Em seu trabalho, Euzébio (2011) utilizou *kits* Arduino para prototipagem dos controladores de iluminação

e um computador de propósito geral para atuar como servidor.

Como forma de agregar valor ao sistema, este trabalho está focado na definição e criação dos produtos utilizados como controlador de iluminação e SAR, de modo que sejam aplicados juntamente com o mesmo cliente Android já desenvolvido anteriormente, a fim de implantá-los em um ambiente real, tornando a solução mais próxima do usuário final.

2. METODOLOGIA

Foram consideradas quatro etapas a fim de executar este trabalho, sendo elas: estudo, desenvolvimento, teste e documentação.

Na etapa de estudo, foi realizado um levantamento bibliográfico com o objetivo de acumular informações suficientes para desenvolver os *hardwares* e *firmwares* do controlador de iluminação e do servidor de automação residencial. Este levantamento bibliográfico foi baseado principalmente na documentação de Euzébio (2011), além de livros e artigos sobre as tecnologias envolvidas.

Na etapa de desenvolvimento, foram especificados os gabinetes e desenvolvidas as arquiteturas de *hardware* e *firmware* dos produtos. A arquitetura de *hardware* levou em conta principalmente os requisitos de comunicação entre o SAR, o cliente Android e controlador de iluminação. A arquitetura de *firmware* levou em conta principalmente os protocolos de comunicação entre os dispositivos. Foram criados os diagramas esquemáticos e os *layouts* das placas de circuito impresso (PCI), sendo confeccionados industrialmente a partir da geração dos arquivos Gerber, e os *firmwares* desenvolvidos em C.

Na etapa de teste, os dispositivos desenvolvidos na etapa anterior foram testados em um ambiente controlado simulando uma residência, tendo como objetivo realizar os mesmos testes de funcionalidades, com o cliente Android, realizados por Euzébio (2011), porém em ambiente mais próximo do real.

A etapa de documentação foi realizada ao longo de todo o trabalho, procurando-se registrar a execução das etapas descritas anteriormente para elaboração dos produtos propostos.

3. AUTOMAÇÃO RESIDENCIAL

Pode-se definir automação residencial, ou domótica, como o conjunto de tecnologias que permitem automatizar uma série de tarefas realizadas em um imóvel (PRUDENTE, 2011). Para automatizar uma residência é necessária uma infraestrutura diferente da tradicional. Sabe-se que quanto antes for percebida essa necessidade (ainda na fase de projeto), mais barata tornar-se-á a automação da residência (AURESIDE, 2013).

A Figura 1 apresenta um diagrama da infraestrutura de iluminação de uma residência com um sistema de automação residencial, onde a ação do usuário no interruptor é interpretada por um *hardware* que realiza o controle da iluminação através da sua eletrônica interna. Além da corrente alternada, linhas de neutro e fase da rede elétrica, tem-se ainda a corrente contínua, linhas de terra e tensão contínua fornecida pela fonte de alimentação. A comunicação entre os controladores e o servidor é realizada através da rede de dados (PRUDENTE, 2011).

Para que haja comunicação entre os dispositivos conectados na rede de dados, é necessário que seja estabelecido um protocolo comum entre eles. O protocolo utilizado neste projeto para a rede de dados foi o CAN (*Controller Area Networks*).

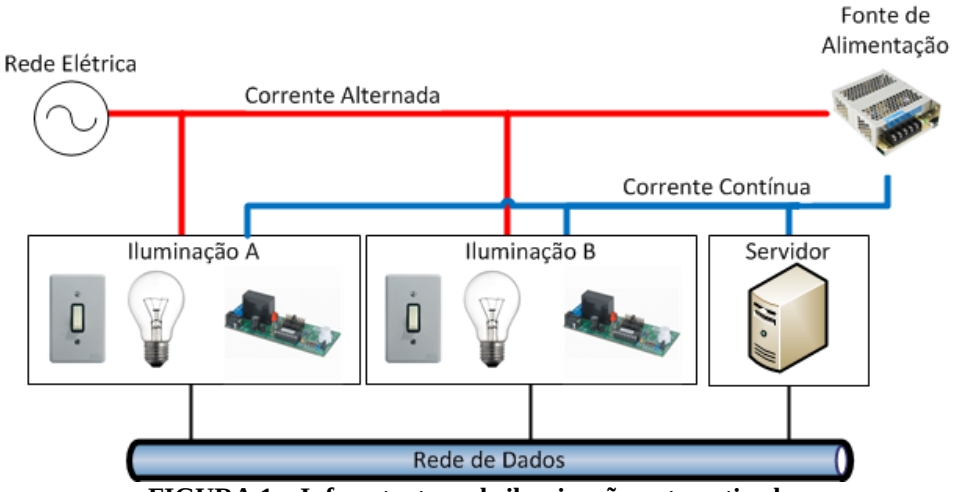


FIGURA 1 – Infraestrutura de iluminação automatizada.

4. PROTOCOLO CAN

O CAN é um protocolo de comunicação síncrono desenvolvido na década de 1980 pela Bosch, destinado principalmente para a indústria automotiva. Atualmente, além da indústria automobilística, o CAN é utilizado em diferentes segmentos, como: automação industrial, automação predial, equipamentos médicos, simuladores de voo, telescópios, entre outros.

A arquitetura CAN especifica apenas as camadas física e de enlace do modelo OSI (BOLZANI, 2004). Guimarães e Saraiva (2002) apontam como principais características da arquitetura CAN:

- Multi-mestre: todos os elementos da rede podem tornar-se mestre ou escravos do barramento;
- *Multicast*: quando um elemento da rede envia uma mensagem, todos os outros recebem;
- CSMA/CD com NDA: antes de enviarem uma mensagem, os elementos da rede verificam se existe alguma mensagem sendo enviada no barramento. Caso dois elementos enviem a mensagem simultaneamente, a de menor prioridade é interrompida para que seja transmitida a de maior prioridade;
- NRZ: cada bit transmitido representa efetivamente um dado;
- A taxa de transmissão de dados pode chegar a 1 Mbps, dependendo do comprimento do barramento;
- Robustez em detecção de falhas;
- Alta imunidade à interferência eletromagnética.

4.1. Barramento CAN

Em geral, o barramento CAN é implantado a partir de um par de fios de sinal, denominados CANH (*CAN High*) e CANL (*CAN Low*). Pode conter ainda um par de fios, VCC e GND, para alimentação dos dispositivos conectados ao barramento (GUIMARÃES; SARAIVA, 2002).

No barramento CAN o bit é interpretado a partir da diferença de potencial entre os fios de sinal, Figura 2. O bit 0 (zero) é denominado como "Dominante", caracterizado pela diferença de tensão entre CANH e CANL, enquanto o bit 1 (um) é denominado como "Recessivo", caracterizado pela variação praticamente nula entre os fios (BOLZANI, 2004).

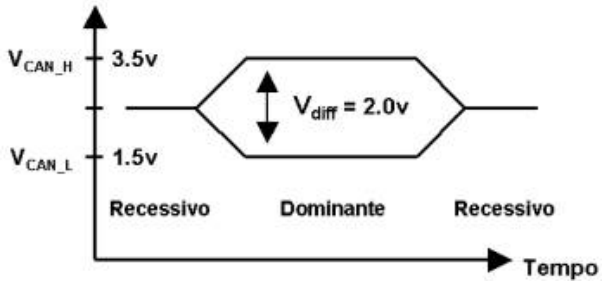


FIGURA 2 – Interpretação dos bits em um barramento CAN.

Devem estar presentes nas extremidades do barramento CAN dois resistores de 120 Ω, um em cada ponta da rede, como mostra a Figura 3. Esse resistor é chamado "Terminador CAN". Eles servem para garantir a propagação dos sinais elétricos pelos fios da rede, ou seja, balancear a rede CAN. O balanceamento da rede gera uma resistência de 60 Ω entre os fios CANH e CANL. Na prática, aplica-se um Terminador CAN no primeiro elemento da rede e outro no último (GUIMARÃES; SARAIVA, 2002).

4.2. *Frame* CAN

O *frame* CAN pode ser de dois tipos: *Standard* ou *Extended*. Na versão *Standard*, o campo do *frame* que identifica o tipo da mensagem a ser transmitida é formado por 11 bits, enquanto que na versão *Extended*, esse parâmetro é formado por 29 bits (11 bits *standard* mais 18 bits *extended*), possibilitando até 5,3 milhões de tipos de mensagens distintas. Em ambas as versões, podem ser transmitidos até 8 bytes de dados por *frame* (GUIMARÃES; SARAIVA, 2002).

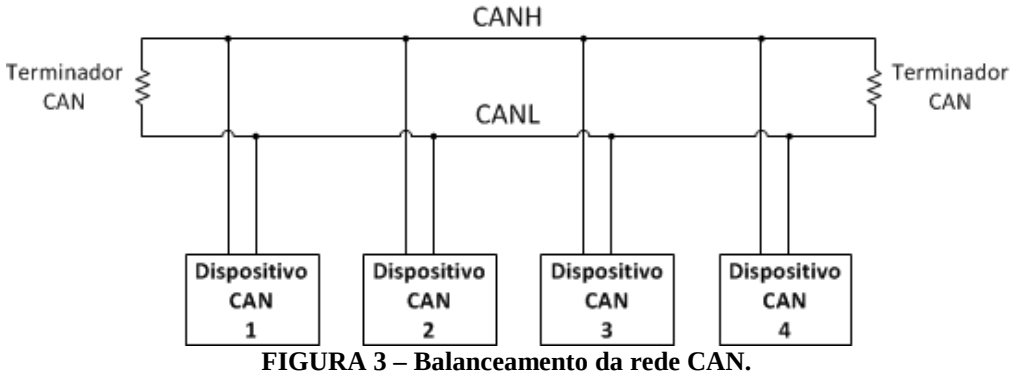


FIGURA 3 – Balanceamento da rede CAN.

4.3. *Hardware* CAN

Em um projeto eletrônico de um dispositivo compatível com o protocolo CAN, deve-se considerar a utilização de um controlador e um transceptor CAN. Alguns microcontroladores já apresentam, em sua arquitetura interna, um módulo de *hardware* controlador CAN, necessitando apenas do transceptor (GUIMARÃES; SARAIVA, 2002).

Neste projeto, devido às características apresentadas, o CAN foi escolhido como o protocolo de comunicação entre o *hardware* do servidor de automação residencial e o *hardware* do controlador de iluminação. Todos eles farão parte de um sistema maior, chamado DroidLar.

5. DROIDLAR

O DroidLar é sistema de automação residencial que permite controlar diferentes equipamentos de uma residência utilizando celulares Android como interface de controle. A arquitetura do sistema é do tipo centralizada, representada na Figura 4, onde o cliente Android estabelece uma conexão IP *via* rede sem fio (WiFi) com o SAR (Servidor de Automação Residencial). Este, por sua vez, se conecta ao Controlador através do barramento da rede de dados (EUZÉBIO, 2011).

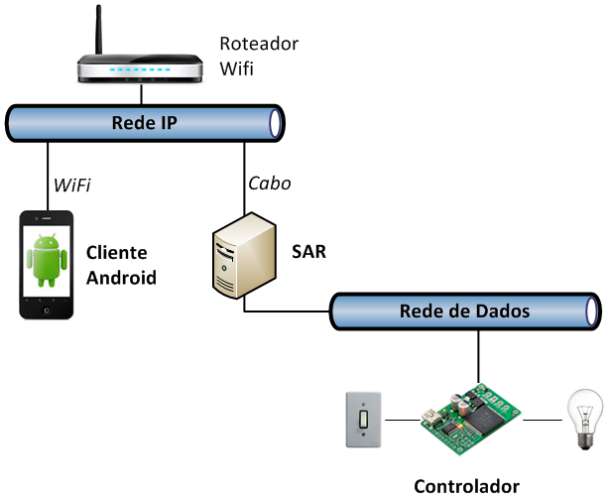


FIGURA 4 – Arquitetura do sistema DroidLar.

A arquitetura prevê apenas um elemento cliente e um servidor, porém podem ser inseridos um ou mais controladores à rede de dados. Cada elemento presente na arquitetura tem sua função bem definida.

5.1. Cliente Android

O cliente Android consiste em uma aplicação, disponível para *smartphones* com sistema operacional Android embarcado. Quando conectado à rede IP, a aplicação estabelece conexão com o servidor e envia comandos *via* protocolo HTTP, com informações de controle para o servidor.

A interface permite abstrair do usuário os outros elementos do sistema, dando-lhe a noção de estar conectado diretamente ao equipamento da residência. O *software* não necessita estar constantemente ligado ao servidor, apenas quando for enviar algum comando (EUZÉBIO, 2011).

5.2. SAR

O SAR deve permanecer constantemente em operação e conectado à rede IP. Sua principal função no sistema é interpretar os comandos do cliente, converte-las para o protocolo da rede de dados e enviá-las para os controladores dos equipamentos da residência. O caminho inverso também é verdade, pois os controladores respondem à solicitação do servidor, que por sua vez repassa-a ao cliente (EUZÉBIO, 2011).

Outra função do servidor é manter-se atualizado quanto aos dispositivos da rede de dados, requisitando regularmente aos controladores as informações sobre o estado dos equipamentos controlados pelo mesmo. Também é função do SAR autenticar o usuário no sistema, onde apenas após esta ação, os comandos de controle do cliente são repassados adequadamente ao controlador específico (EUZÉBIO, 2011).

5.3. Controlador

O Controlador é formado por um circuito eletrônico capaz de controlar a intensidade de até três lâmpadas. As requisições de controle podem ser oriundas do SAR ou por três interruptores presentes no mesmo. Cada interruptor é diretamente relacionado a uma única lâmpada (EUZÉBIO, 2011).

Através dos interruptores, o usuário pode controlar a intensidade da lâmpada em 100% ou 0%. Apenas através das requisições enviadas pelo SAR, a intensidade pode ser controlada em valores intermediários. Qualquer alteração realizada nas lâmpadas é armazenada em memória. Desta forma, sempre que requisitado, o controlador envia ao servidor uma mensagem atualizada da intensidade atual de cada uma das lâmpadas controladas pelo mesmo (EUZÉBIO, 2011).

5.4. Protocolo Cliente Android x SAR

As mensagens trocadas entre cliente Android e SAR apresentam os parâmetros a seguir:

- Tipo: indica a operação a ser realizada;
- Endereço: identificação do controlador presente na rede de dados;
- Conteúdo: dado da operação a ser realizada;
- Opcional: identifica a lâmpada presente no controlador, quando necessário.

A mensagem de autenticação difere-se das demais nos campos Endereço e Conteúdo. Em seu lugar, são enviados os campos Usuário e Senha, contendo o nome do usuário no sistema e sua senha, respectivamente (EUZÉBIO, 2011).

Para toda requisição enviada do cliente, o servidor envia uma resposta, podendo ser uma mensagem de sucesso ou de erro, quanto à operação realizada (EUZÉBIO, 2011).

5.5. Protocolo SAR x Controlador

Sempre que um controlador é conectado à rede de dados, ele envia ao SAR uma mensagem contendo informações sobre o endereço do controlador na rede, o estado e o identificador das lâmpadas que ele controla. Também quando o usuário altera o estado da lâmpada através do interruptor, além de memorizar a informação, o controlador envia uma mensagem ao servidor, mantendo-o sempre atualizado sobre qualquer alteração realizada (EUZÉBIO, 2011).

5.6. DroidLar versão 1.0

Em sua versão atual (1.0), o sistema DroidLar suporta apenas o controle de lâmpadas. Neste projeto, o sistema permanece restrito ao controle de iluminação, pretendendo-se realizar modificações no Controlador e no SAR tornando-os aplicáveis em um ambiente residencial real.

6. VISÃO GERAL DO SISTEMA

A Figura 5 ilustra a visão geral da infraestrutura de rede de uma residência em que o sistema DroidLar está presente. O modelo estabelecido foi gerado a partir do conjunto de informações levantadas, dando início a criação dos blocos SAR e Controlador.

7. GABINETE DO SAR

Para determinar como seria o gabinete utilizado no SAR, foram pesquisados alguns modelos de dispositivos que possuem ao menos um conector de rede IP e um conector de fonte de alimentação externa, a fim de levantar características estéticas e ergonômicas do dispositivo. Dentre os pesquisados, observou-se que os equipamentos têm pouca interação com o usuário, estando restrito à conexão de cabos e sinalização de estados internos através de LEDs. Outra característica marcante foi o formato retangular, comunicando ao usuário que é um dispositivo eletrônico que pode ser instalado em um escritório, sala ou quarto.

Como forma de reduzir o custo do projeto, optou-se por utilizar o gabinete de outro produto, que apresentasse as características identificadas e necessitasse de poucas adaptações. A Figura 6 apresenta a vista frontal do gabinete utilizado para o SAR.

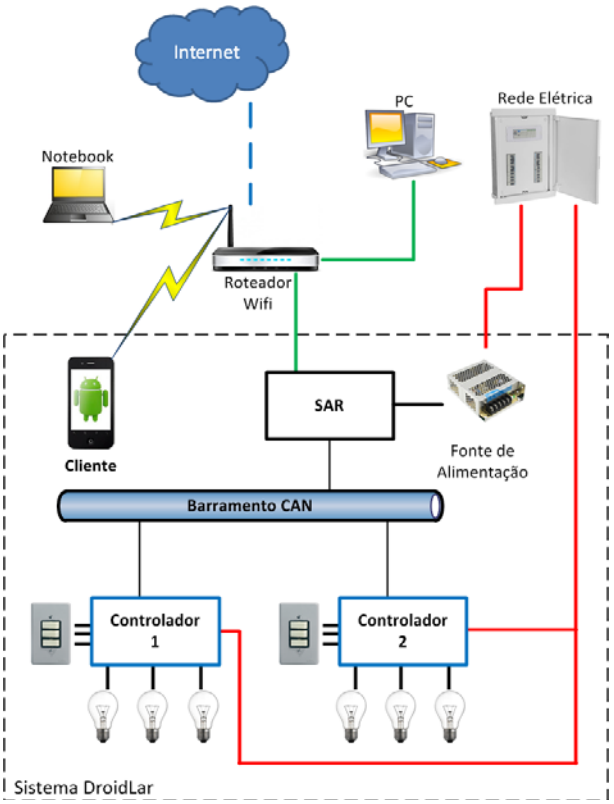


FIGURA 5 – Visão geral do sistema.

FIGURA 6 – Vista frontal do gabinete do SAR.

Na Figura 7 podem-se observar as adaptações realizadas na parte traseira do gabinete para atender os requisitos de conectividade e interação:

- Dois conectores RJ12 para conexão com a rede de dados CAN;
- Um botão para reiniciar o dispositivo (*reset*);
- Um conector RJ12 para gravação ICSP do *firmware*;
- Um conector RJ45 para conexão com a rede IP;
- Um conector DB9 para comunicação serial;
- Um conector Jack J4 para fonte externa.

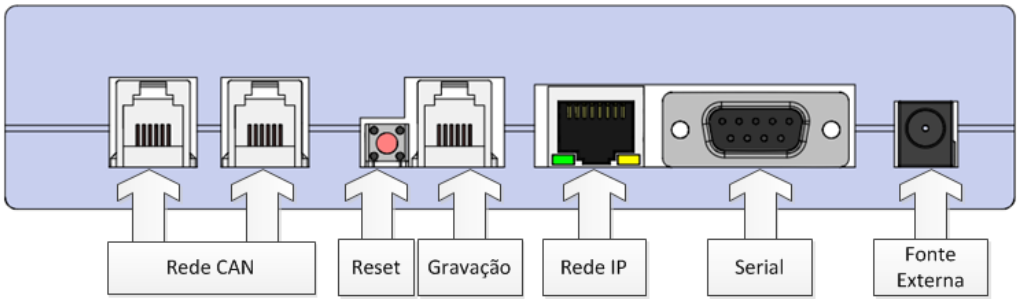


FIGURA 7 – Vista traseira do gabinete do SAR.

A placa de circuito impresso do SAR foi desenvolvida para encaixar-se no gabinete reaproveitado.

8. *HARDWARE* DO SAR

A Figura 8 ilustra a arquitetura de *hardware* do SAR. Sua função é representar os módulos internos necessários para que o servidor embarcado seja empregado no sistema DroidLar, de modo que atenda aos requisitos de conectividade do sistema e de dimensionamento e interação definidos no gabinete.

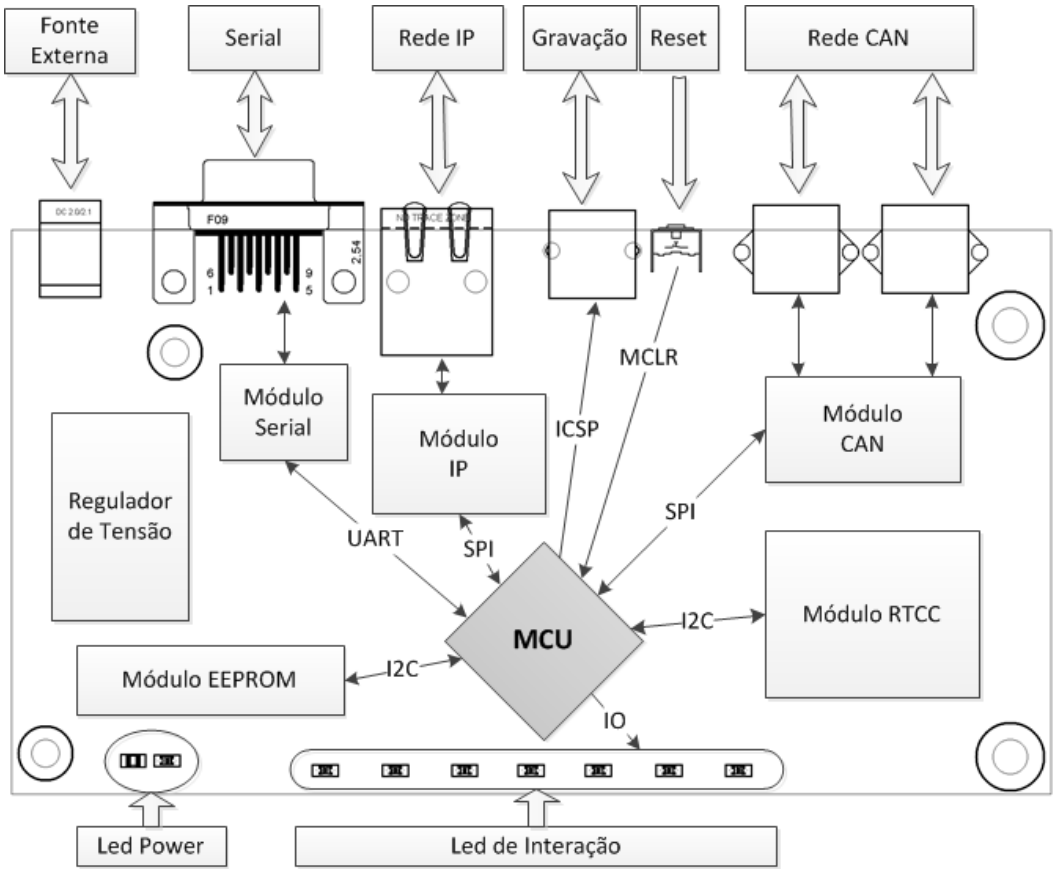


FIGURA 8 – Arquitetura de *hardware* do SAR.

Dentre os blocos contidos na mesma, o regulador de tensão tem a função de regular a alimentação da fonte externa no nível de tensão de operação dos módulos e dos outros elementos do *hardware*. Com exceção deste, os outros blocos são controlados por um microcontrolador (MCU) central.

- Módulo Serial: atua no canal de depuração do *firmware*, possibilitando que o usuário monitore os pacotes recebidos e enviados pelo servidor, tanto pela rede IP quanto pela rede CAN;
- Módulo RTCC: atua como um relógio interno do dispositivo possibilitando registrar os horários dos eventos ocorridos, através do canal de depuração;
- Módulo EEPROM: utilizado para armazenar informações de inicialização do sistema;
- Módulo IP: responsável pela conexão do servidor à rede IP, retirando do *firmware*

toda complexidade envolvida nas camadas física, de enlace e rede do protocolo IP, permitindo-lhe trabalhar diretamente na camada de transporte com os protocolos TCP e UDP, necessitando apenas comandar o módulo na abertura/fechamento de *socket* e envio/recebimento de pacotes;

- Módulo CAN: responsável pela conexão do servidor à rede CAN, tornando-o compatível com o barramento.

Os dois conectores da rede CAN formam o barramento CAN, que pode ser estendido aos demais elementos da rede. Logo, foi necessário estabelecer um padrão de conectividade.

9. CONECTIVIDADE DA REDE CAN

A Figura 9 apresenta o padrão de conectividade estabelecido para a rede CAN deste projeto. Ao todo, são três pares trançados de conexão no RJ12. O par central, pinos 3-4, são os pinos de sinal do barramento, CANH e CANL, respectivamente. O primeiro par, pinos 1-2, correspondem aos pinos de alimentação da fonte externa, VCC e GND respectivamente, conectada ao servidor. O terceiro par, pinos 5-6, são os mesmos do primeiro, servindo como redundância da fonte de alimentação.

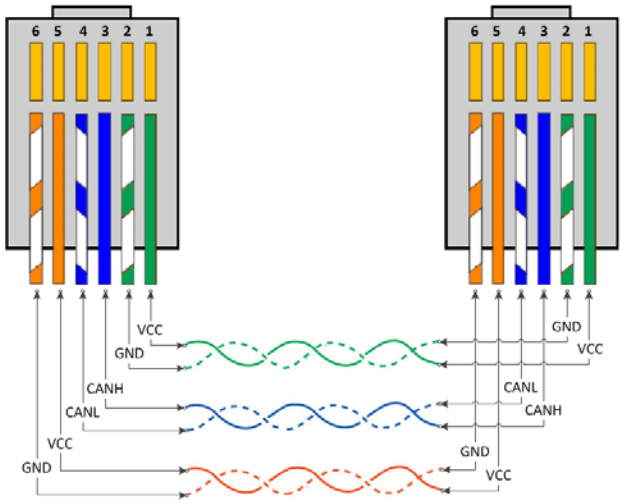


FIGURA 9 – Padrão de conectividade CAN.

O padrão de conectividade CAN deve ser obedecido por todos os elementos da rede, entre eles, o SAR e o Controlador. Caso contrário, o dispositivo pode não conseguir interpretar os *frames* CAN trafegados, ou até causar danos ao mesmo.

Após as definições de conectividade da rede de dados, partiu-se para a definição do gabinete do Controlador, de acordo com o local em que ele será instalado.

10. GABINETE DO CONTROLADOR

O Controlador deve substituir um interruptor tradicional de uma residência, estando este instalado em uma caixa de passagem de 4x2” de embutir em parede de alvenaria. Como referência, foi utilizado um tipo de interruptor modular, permitindo que as estruturas de suporte e espelho sejam aproveitadas, descartando os mecanismos de chaves originais.

A Figura 10 ilustra a visão geral de instalação do Controlador, onde a placa de interface é posicionada na área interna do suporte plástico, conectando-se a segunda placa, que por sua vez, conecta-se a terceira. Com exceção da primeira, as placas ficam embutidas na caixa padrão de 10x15 cm. Sobre o suporte e a primeira placa, é posicionado o espelho do gabinete mantendo a primeira fixa no suporte, expondo apenas os mecanismos presentes na área de interação da mesma.

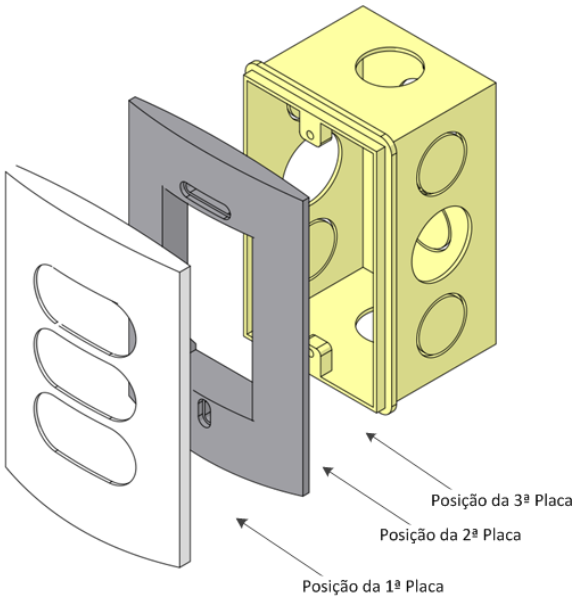


FIGURA 10 – Visão geral de instalação do Controlador.

11. *HARDWARE* DO CONTROLADOR

O Controlador é composto por três placas distintas, denominadas de: Interface (primeira), Controle (segunda) e *Analog* (terceira).

11.1. Placa Interface

A placa Interface tem como principal função permitir a interação do usuário com o dispositivo. Por ela, o usuário poderá ligar e desligar a lâmpada manualmente, através de chaves tácteis, como apresentado na Figura 11. Ao todo, são expostas três chaves de controle, cada uma correspondendo ao acionamento de uma lâmpada de forma independente.

Também é função da placa Interface sustentar a segunda placa conectada por barras de pinos.

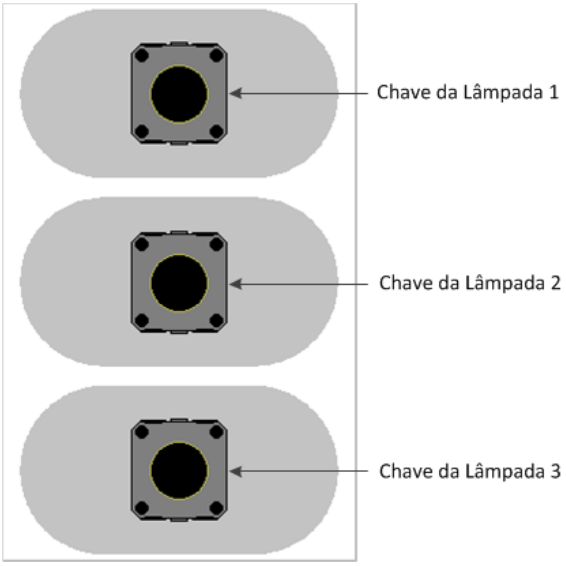


FIGURA 11 – Vista frontal da placa Interface.

11.2. Placa Controle

A segunda placa (Controle) tem a função de realizar o controle digital do acionamento das lâmpadas a partir de comandos provenientes do barramento CAN ou através da interação do usuário com o dispositivo. A Figura 12, apresenta a arquitetura de *hardware* da placa Controle.

Estão presentes na arquitetura da placa Controle alguns blocos que recebem os mesmos nomes e exercem as mesmas funções já citadas no *hardware* do SAR, no item 8, sendo eles: Módulo CAN, Regulador de Tensão, *Reset* e Gravação do *firmware*.

Os pinos contendo os sinais digitais das três chaves, provenientes da placa Interface, estão ligados diretamente ao MCU, identificando se o botão foi pressionado.

A placa Controle também tem a função de sustentar a terceira placa conectada por diferentes barras de pinos. É através deles que são enviados os sinais de controle da lâmpada.

11.3. Placa *Analog*

A Figura 13 apresenta a arquitetura de *hardware* da placa *Analog*. Sua função é realizar a leitura do ponto de inflexão da curva senoidal da tensão alternada e acionar a lâmpada a partir do acionamento digital do microcontrolador.

Os fios da rede elétrica são ligados no conector *Borne In*, onde o neutro é encaminhado para os outros conectores. O fase atua como elemento de entrada para o módulo *Cross* e os módulos *Gate*.

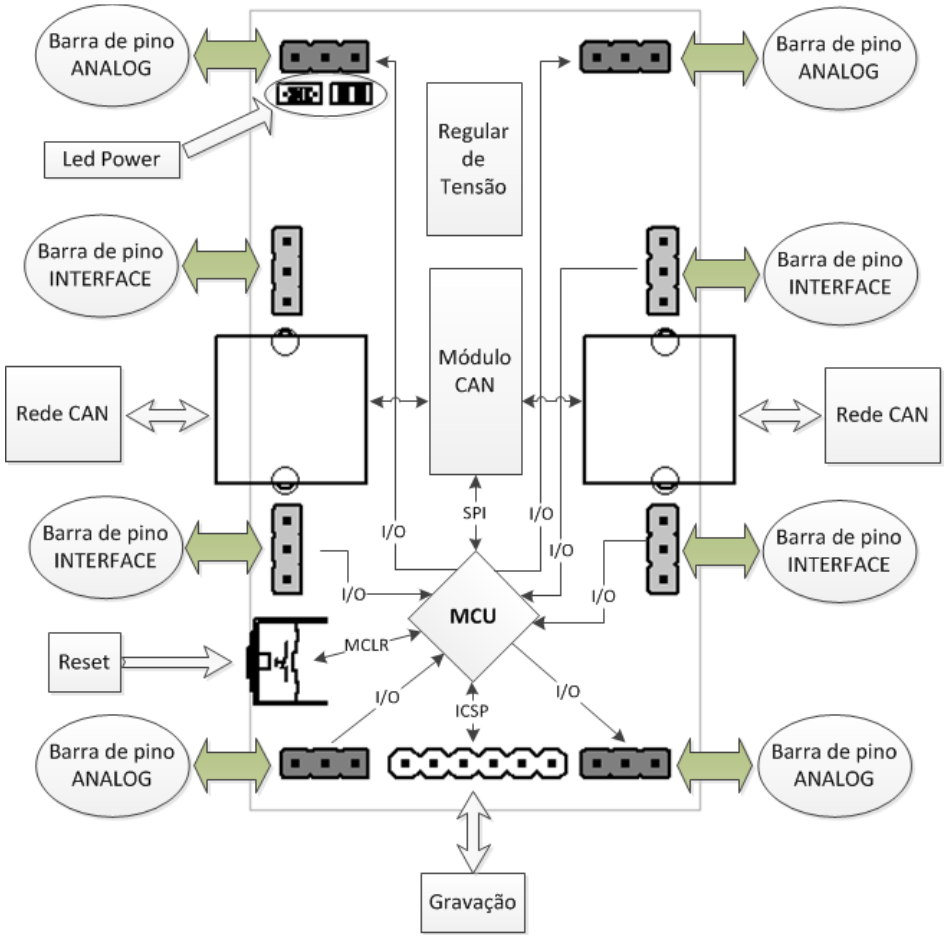


FIGURA 12 – Arquitetura da placa Controle.

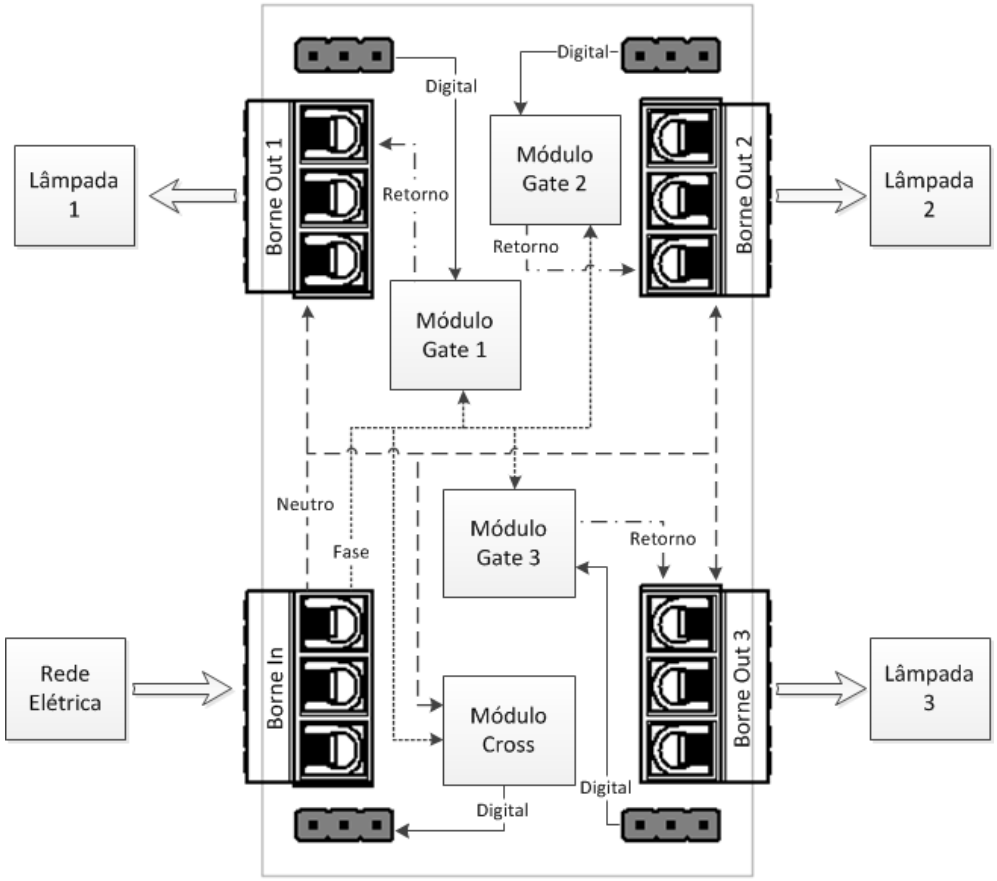


FIGURA 13 – Arquitetura da placa *Analog*.

No módulo *Cross*, enquanto a corrente alternada de entrada for diferente de zero, o sinal digital de saída é 5 V, quando a corrente for é igual (ou próxima) de zero, a saída digital é 0 V, indicando digitalmente o momento em que a tensão alternada está passando por zero. No módulo *Gate* é realizado o disparo do tiristor empregado para conectar a saída (retorno) à entrada (fase).

Após finalizar as definições de *hardware* do Controlador, as placas dos dispositivos foram confeccionadas. Assim que entregues pelo fornecedor, foram montadas dando início ao desenvolvimento do *firmware*.

12. FIRMWARE DO CONTROLADOR

Para o *firmware* do Controlador, ao todo, foram criados quatro *drivers* distintos: *Cross*, *Gate*, Chave e CAN. A aplicação possui até três instâncias dos *drivers Gate* e Chave, referentes ao número de lâmpadas e chaves contidas no *hardware*, respectivamente.

Ao ser inicializada, é atribuído o valor zero para uma variável de controle. Esta variável é utilizada para controlar se uma determinada operação deve ser executada na iteração do laço principal.

Em seguida, são inicializados os *drivers* dos módulos periféricos e habilitadas as interrupções dos *drivers* CAN e *Cross*. As interrupções dos *timers* relacionados aos *driver* dos módulos *Gate* não são habilitadas neste momento. Eles são habilitados apenas no tratamento da interrupção do módulo *Cross*. Desta forma, o disparo é realizado sempre momentos após a passagem da tensão alternada por zero, possibilitando a dimerização da lâmpada de acordo com a intensidade desejada.

A função de tratamento das interrupções apenas seta a variável de controle, indicando no laço principal qual tratamento deve ser realizado.

Ao fim do processo de inicialização, o Controlador envia dois *frames* CAN, um contendo a versão do *firmware* do controlador e outro contendo as informações do estado inicial das lâmpadas para o SAR.

Finalizado o dispositivo Controlador, foi iniciado o desenvolvimento do *firmware* do SAR, integrando ambos dispositivos no sistema DroidLar.

13. *FIRMWARE* DO SAR

O desenvolvimento do *firmware* do SAR foi baseado na lógica de controle de duas máquinas de estados independentes entre si, chamadas de *Device* e *Server*.

A máquina *Device* possui seis estados distintos:

- *IDLE*: estado inicial da máquina *Device*. É realizada a leitura de *frames* do módulo CAN e verifica se é hora de atualizar-se;

- *REQUEST*: trata o *frame* CAN lido;
- *SEND_SAR*: prepara um pacote IP com a resposta do Controlador à solicitação do cliente Android;
- *SEND_DEV*: envia um *frame* CAN;
- *WAITING*: aguarda a resposta do *frame* CAN enviado;
- *NO_ANSWER*: prepara um pacote IP com uma resposta de erro.

O estado da máquina *Device* pode ser alterado para *SEND_DEV* pela ação da máquina *Server*. Isso ocorre quando uma solicitação do cliente Android deve ser repassada ao Controlador. Em contrapartida, o estado da máquina *Server* é alterado pelo *Device* quando este necessita enviar uma resposta do controlador ao cliente, após a preparação do pacote IP no estado *SEND_SAR*.

A máquina *Server* possui 13 estados diferentes:

- *IDLE*: estado inicial da máquina *Device*. É realizada a leitura de pacotes do módulo IP provenientes do cliente Android;
- *REQUEST*: trata o pacote IP lido;
- *AUTHENTICATION*: realiza a validação das informações de usuário e senha de acordo com a lista de permissões pré-definida no *firmware*;
- *LOGGED_ADMIN*: realiza a leitura de pacotes do módulo IP enquanto faz uma contagem de tempo em que o usuário foi autenticado como usuário administrador;
- *LOGGED_NORMAL*: realiza a leitura de pacotes do módulo IP enquanto faz uma contagem de tempo em que o usuário foi autenticado como usuário normal;
- *OPERATION_ADMIN*: faz o tratamento do pacote IP de acordo com as permissões de usuário administrador;
- *OPERATION_NORMAL*: faz o tratamento do pacote IP de acordo com as permissões de usuário normal;
- *DEVICE_LIST*: prepara um pacote IP com a lista de controladores da rede CAN;
- *DEVICE_CHANGE*: prepara um *frame* CAN de acordo com o valor de intensidade para ser enviado ao Controlador;
- *DEVICE_SCAN*: prepara um *frame* CAN requisitando o estado atual dos Controladores;
- *DEVICE_NAME_CHANGE*: atualiza os nomes dados às lâmpadas dos Controladores;
- *RESPONSE_CLIENT*: envia um pacote IP;
- *WAITING*: aguarda a resposta do Controlador ao comando enviado na rede CAN.

O estado da máquina *Server* pode ser alterado para *RESPONSE_CLIENT* pela operação *SEND_SAR* da máquina *Device*. Por outro lado, o *Device* é alterado para o estado *SERV_DEV* quando são realizadas as operações nos estados *DEVICE_CHANGE* e *DEVICE_SCAN* da máquina *Server*.

Com o *firmware* do SAR desenvolvido, os dispositivos puderam ser aplicados e testados em um ambiente controlado.

14. RESULTADOS

Os resultados apresentados a seguir foram obtidos principalmente através do canal de depuração do SAR e da utilização de osciloscópio, para obtenção da forma de onda da corrente alternada aplicada na lâmpada pelo Controlador. Os testes de funcionalidade realizados foram: Acender Lâmpada *via* Controlador, Apagar Lâmpada *via* Controlador e Controle de Intensidade *via* Cliente Android.

14.1. Acender Lâmpada *via* Controlador

O resultado da funcionalidade de acender a lâmpada *via* Controlador é demonstrado na Figura 14. Logo após o usuário pressionar a chave, o Controlador envia um *frame* CAN para atualizar as informações do SAR quanto ao estado da lâmpada. O *frame* de atualização é identificado pelo valor 30h no primeiro byte do campo “data”. Também está demonstrado no comando a alteração da intensidade da lâmpada 0Ah para 100%, indicada pelo quarto byte (FFh) do campo “data”. O *frame* CAN enviado pelo Controlador consiste no comando de atualização, pois não houve alteração no estado da máquina *Server*.

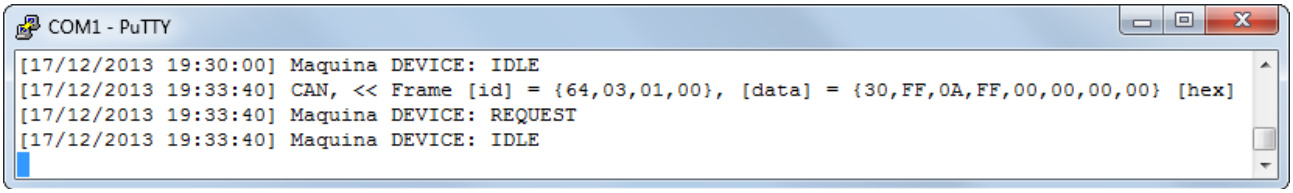


FIGURA 14 – Resultado de ligar lâmpada *via* Controlador.

O resultado da mudança efetiva da intensidade da lâmpada para o valor máximo está representado na Figura 15, onde o canal 2 do osciloscópio representa a corrente alternada aplicada.

Pode-se confirmar, na Figura 16, que a mudança da intensidade da lâmpada para o valor máximo foi gerada a partir do Controlador, onde o canal 1 do osciloscópio representa o *buffer* do módulo CAN do SAR sendo preenchido com o *frame* CAN de atualização enviado pelo Controlador (nível lógico alto quando o *buffer* está cheio), enquanto o canal 2 representa o momento da mudança da forma de onda da corrente alternada.

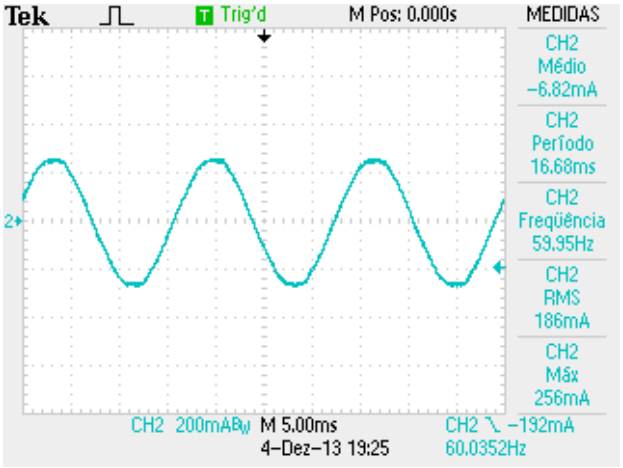


FIGURA 15 – Corrente aplicada na lâmpada em 100%.

O atraso entre o envio do *frame* e a mudança da forma de onda é proposital, pois o Controlador altera o valor da variável de intensidade logo após o envio do comando, porém a mudança é aplicada apenas após a identificação do fim do ciclo da corrente, a partir do *driver Cross*.

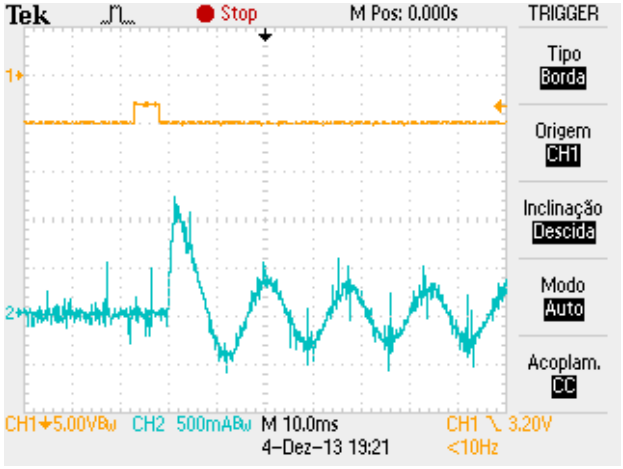


FIGURA 16 – Corrente aplicada na lâmpada ao acendê-la.

14.2. Apagar Lâmpada *via* Controlador

O resultado da funcionalidade de desligar a lâmpada *via* Controlador é demonstrado na Figura 17. Também para esta funcionalidade, logo após usuário pressionar a chave, o Controlador envia um *frame* CAN de atualização da lâmpada para o SAR.

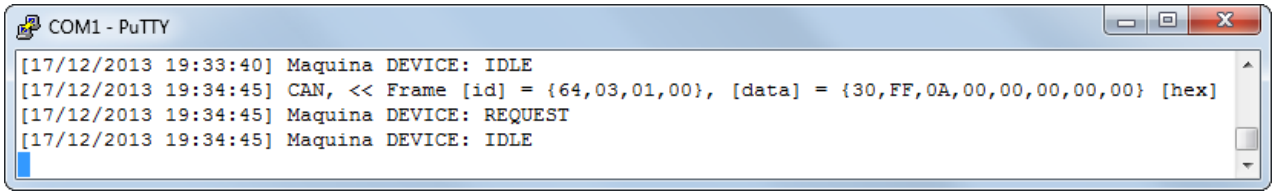


FIGURA 17 – Resultado de desligar a lâmpada *via* Controlador.

Neste caso, o *frame* CAN de atualização apresenta o valor de alteração da intensidade da lâmpada 0Ah em 0% (00h). Também não há alteração no estado da máquina *Server*, confirmando sua veracidade

Pode-se confirmar o resultado da mudança efetiva da intensidade da lâmpada para o valor mínimo na Figura 18, sendo o canal 1 do osciloscópio a representação do *buffer* do módulo do CAN do SAR, enquanto o canal 2 representa o momento da mudança da forma de onda da corrente alternada.

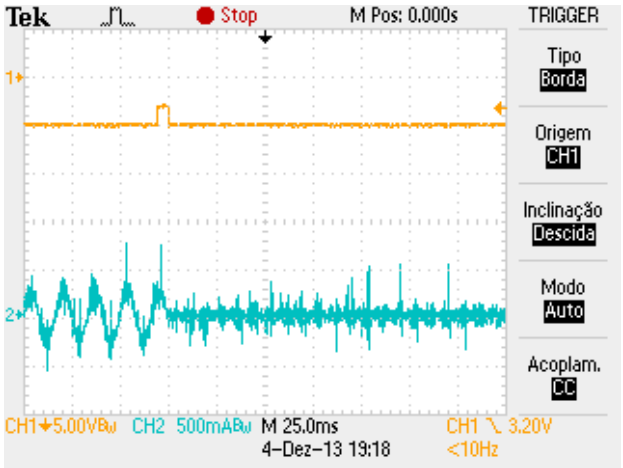


FIGURA 18 – Corrente aplicada na lâmpada ao desligá-la.

14.3. Controle de Intensidade *via* Cliente Android

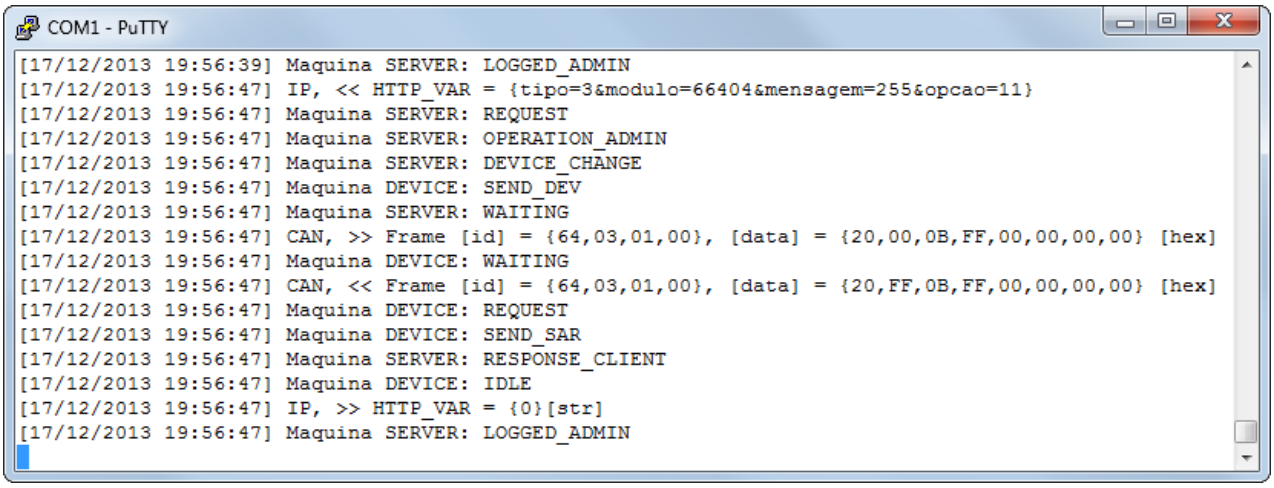
O resultado da funcionalidade de controle de intensidade da lâmpada *via* Cliente Android, é demonstrado na Figura 19(a) e na Figura 19(b). O pacote IP de controle da intensidade, identificado pelo valor 3 da variável "tipo", é enviado ao SAR. Por sua vez, o servidor converte o comando do pacote em um *frame* CAN, repassando-o ao Controlador.

Enquanto o SAR não recebe uma resposta do Controlador, ele mantem as duas máquinas, *Server* e *Device*, em estado *WAITING*. Ao receber a resposta do Controlador, a máquina *Device* é alterada para o estado *REQUEST*, na qual o *frame* CAN é processado. Em seguida, a máquina *Device* entra no estado *SEND_SAR*, preparando um pacote IP de resposta. Neste momento, a máquina *Server* é alterada para *RESPONSE_CLIENT* para que seja enviado o pacote ao Cliente Android.

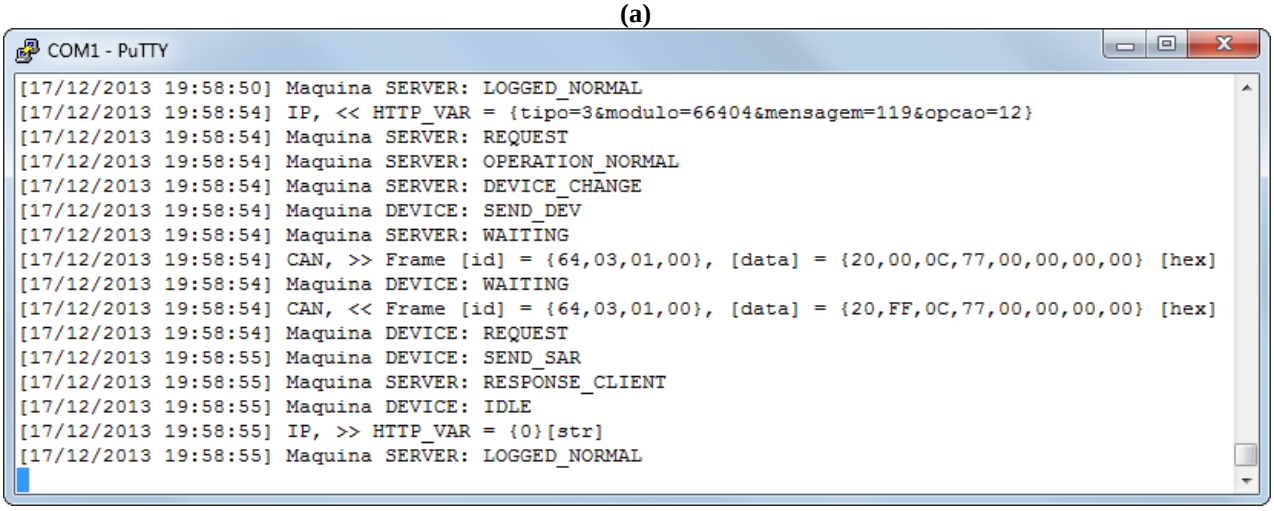
Ao terminar seu trabalho, a máquina *Device* retorna ao estado IDLE, enquanto a máquina *Server*, após o envio do pacote IP, retorna para o estado em que estava antes de receber a requisição.

Pode-se confirmar o resultado do controle de intensidade com o fluxo dos dados transmitidos e com o conteúdo dos pacotes e *frames*. No primeiro pacote IP, a variável "modulo" contém o identificador do Controlador que é utilizado no campo "id" do *frame* CAN (66404 = 10364h ➔ [id] = {64,03,01,00}). Também estão presentes no pacote a variável "mensagem" contendo o valor da intensidade requisitada (255 = FFh) e a variável "opcao" contendo o identificador da lâmpada presente no Controlador (11 = 0Bh). O segundo pacote IP, representa uma resposta de sucesso, indicando que o comando foi realizado pelo Controlador.

Ao comparar os fluxos da Figura 19(a) e da Figura 19(b), pode-se afirmar que o controle de intensidade pode ser realizado tanto por usuário normal quanto por administrador, diferentemente do apresentado na Figura 20. Nesta, o usuário perdeu ou não foi autenticado, sendo no segundo pacote IP uma resposta de erro.



(a)



(b)

FIGURA 19 – Resultado do controle de intensidade em (a) modo admin e (b) modo normal.

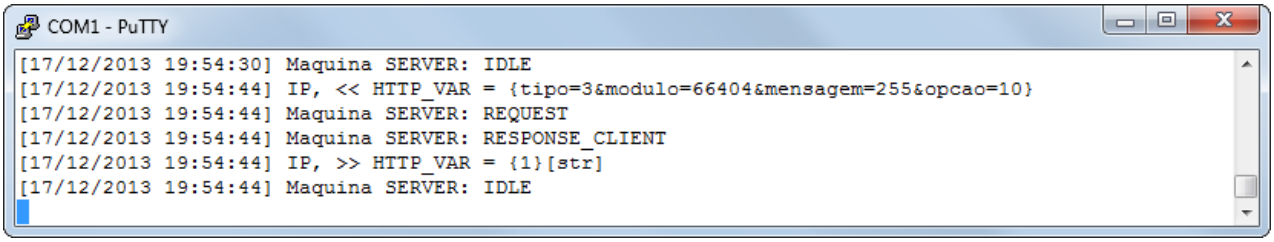
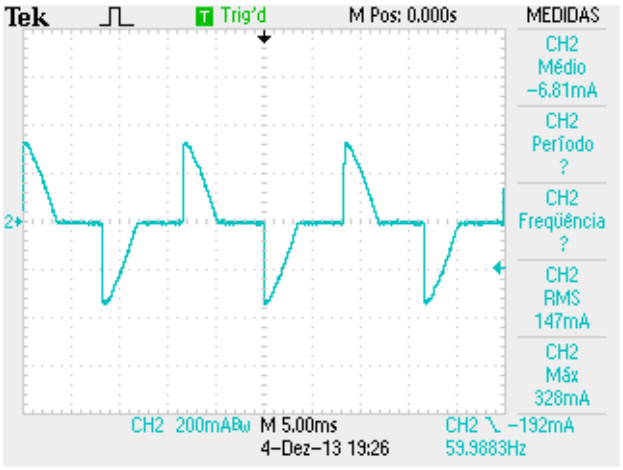
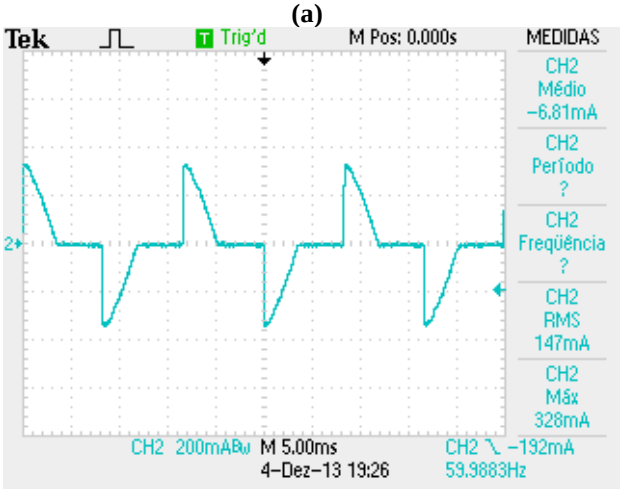


FIGURA 20 – Resultado de erro no controle de intensidade.

Como forma de comprovar a veracidade da execução do controle de intensidade pelo Controlador, a Figura 21 apresenta o resultado da mudança de intensidade da lâmpada para valores intermediários, como por exemplo, enviado pelo usuário normal na Figura 19(b). O canal 2 do osciloscópio representa a corrente alternada aplicada na lâmpada após o comando ser processado pelo Controlador.



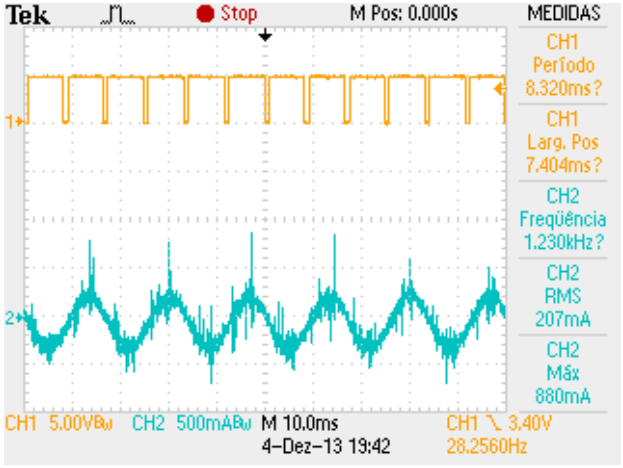
(a)



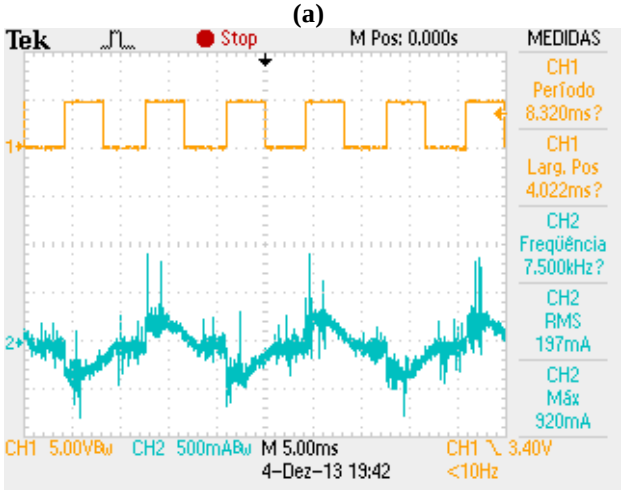
(b)

FIGURA 21 – Formas de onda no controle de intensidade. (FIGURAS IGUAIS)

Complementando as demonstrações da Figura 21, a Figura 22 apresenta como é realizado o controle da intensidade da lâmpada no Controlador, a partir do disparo da corrente alternada pelos *drivers Gate*. O canal 1 do osciloscópio representa o sinal do pino de I/O do microcontrolador ligado à entrada do módulo *Gate*, enquanto que o canal 2 representa o disparo da corrente alternada na lâmpada, a partir da saída do módulo (Retorno).



(a)



(b)

FIGURA 22 – Controle de intensidade no Controlador.

15. CONCLUSÕES

Esse trabalho apresentou e discutiu o projeto e a execução de dois produtos voltados para automação residencial, aplicáveis em um ambiente real, agregando valor a uma solução de código aberto desenvolvido academicamente por terceiros, denominado DroidLar.

Quanto ao desenvolvimento do SAR, os pontos negativos ficaram em torno da opção adotada na definição do gabinete não ter sido uma boa estratégia, pois foi obtida apenas uma amostra do gabinete utilizado no dispositivo, onde em pouco

tempo, mas já com o *hardware* pronto, as hastes de sustentação interna da placa se romperam. O ponto positivo do servidor foi a abordagem empregada para o conector de gravação do MCU mostrando-se ser boa, se não essencial, durante o processo de desenvolvimento do *firmware*, reduzindo significativamente a necessidade de abrir e fechar o gabinete, o que poderia causar ainda mais danos no mesmo.

Quanto ao desenvolvimento do Controlador, as principais considerações estão em torno do *hardware*. Os pontos negativos estão principalmente na utilização de barras de pinos para a sustentação das placas e o posicionamento dos conectores da rede CAN nas laterais do dispositivo, pois no momento em que é realizada a manutenção, e o dispositivo é retirado e colocado na caixa de passagem 4x2", as placas Controle e *Analog* se soltam com facilidade. Os pontos positivos no dispositivo estão nas estratégias aplicadas para o *hardware* e o *firmware*. O *hardware* modular permitiu que as placas fossem modularizadas de acordo com sua funcionalidade. No caso de substituição de uma placa as outras poderiam continuar sendo aproveitadas. Quanto ao *firmware*, o tratamento das interrupções mostrou-se mais eficiente quando aplicado da forma como foi apresentado no trabalho, se comparado ao tratamento da interrupção realizada na própria função de interrupção.

Em geral, os dispositivos comportaram-se bem nos testes realizados. Algumas melhorias podem ser implementadas nos detalhes dos dispositivos, porém as funcionalidades principais do sistema DroidLar já estão satisfatoriamente atendidas.

REFERÊNCIAS

AURESIDE. **Conceitos Básicos**, 2013, Disponível em: < http:// www.aureside.org.br/temastec/ default.asp?file=concbasicos.asp&menu=temas>. Acesso em: 15/11/2013.

BOLZANI, C. A. M. **Residências Inteligentes**. 1ª. ed. São Paulo, SP – Brasil: Livraria da Física, 2004.

EUZÉBIO, M. V. D. M. **DroidLar - Automação residencial através de um celular Android**. IFSC. São José, SC – Brasil, p. 58. 2011.

GUIMARÃES, A. D. A.; SARAIVA, A. M. O **Protocolo CAN: Entendendo e Implementando uma Rede de Comunicação Serial de Dados baseada no Barramento *"Controller Area Network"***. Universidade de São Paulo. São Paulo, SP – Brasil, p. 10. 2002.

PRUDENTE, F**. Automação Predial e Residencial – Uma Introdução**. Rio de Janeiro, RJ – Brasil: LTC, 2011.