

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ANDERSON LAZZARI

**DESENVOLVIMENTO DE UM PROTÓTIPO DE SISTEMA CIBERFÍSICO PARA
ACIONAMENTO DE CONDICIONADOR DE AR EM AMBIENTE RESIDENCIAL**

Porto Alegre

2014

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE SISTEMA CIBERFÍSICO PARA
ACIONAMENTO DE CONDICIONADOR DE AR EM AMBIENTE RESIDENCIAL**

Projeto de diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Prof. Dr. Renato Ventura Bayan Henriques

Porto Alegre

2014

DEDICATÓRIA

Dedico este trabalho à minha família, em especial aos meus pais, por todo suporte e apoio dados que foram fundamentais durante todo o curso de graduação.

AGRADECIMENTOS

Agradeço ao professor e orientador Dr. Renato Henriques e ao idealizador do projeto, Dr. Marcelo Pias, pela confiança concedida na realização desse trabalho.

Agradeço à professora Dra. Silvia Silva Botelho da Costa Botelho pela oportunidade de obtenção da bolsa de iniciação científica do CNPq.

Agradeço ao CNPq pela bolsa de iniciação científica concedida para o desenvolvimento desse estudo.

Agradeço ao colega de projeto e amigo Eduardo Maciel pelas constantes motivação e troca de experiência dispendidas ao longo deste trabalho.

Agradeço aos colegas de projeto Fernanda Mota, Jonas Casarin e Robledo Castro pelas discussões e pelo apoio.

Agradeço ao Eng. Maicon Ramos, da AES Sul, pelos depoimentos técnicos concedidos como suporte para este trabalho.

RESUMO

O presente trabalho se insere no contexto de novas tecnologias de controle residencial, de conforto térmico e de eficiência energética, dentro de um projeto de integração mais amplo, o projeto SapiEns. O objetivo deste trabalho é o desenvolvimento de uma unidade de acionamento para um aparelho de ar condicionado e de uma plataforma de comunicação remota entre uma unidade central de comando e a unidade de acionamento. O projeto teve como condição de contorno permitir uma integração com serviços de acesso remoto de dados em servidor web e com banco de dados MongoDB, de modo a que o acionamento do condicionador de ar pudesse ser feito via *Smartphone* ou outro modo de acesso à internet. Como unidade central de comando, foi utilizado um Raspberry PI modelo A; como unidade de acionamento, foi utilizada uma plataforma Arduino UNO; como meio de comunicação, foi empregado o módulo nRF24L01(+); como condicionador de ar principal foi utilizado um Electrolux Split Hi-Wall 9000 BTU, além de se terem efetuado testes comparativos em um Komeco Portátil Ambient ABP 09QC 9000 BTU. O kit também contou com uma lógica de ajuste de temperatura de referência, baseada na leitura de temperatura de um sensor LM35, integrado ao módulo de acionamento.

Palavras-chave: Domótica, sistema ciberfísico, eficiência energética, conforto térmico, RF, infravermelho, condicionador de ar, Arduino, Raspberry PI.

ABSTRACT

This work is inserted on the context of new technologies on residential control, thermal comfort and energy efficiency, all inside a more global integration project, the SapiEns project. The goal of this work is to develop a drive system unity for an air conditioner and a communication platform between a command central unity and the drive system unity. This project had to work in harmony with services of remote data access in a webserver and with data base MongoDB, so that the activation of the air conditioner could be made by Smartphone or by any other method with Internet access. For the command central unity, it was used the Raspberry PI model A; for the drive system unity, it was used an Arduino UNO platform; for the communication between them it was used a nRF24L01(+) module; as primary air conditioner it was used an Electrolux Split Hi-Wall 9000 BTU, whereas comparative tests were made on a Komeco Portátil Ambient ABP 09QC 9000 BTU. This kit also had a logic system for adjusting the setpoint, based on the reading of the temperature from a LM35 sensor, which is integrated to the drive system unity.

Keywords: Domotics, cyber-physical system, energy efficiency, thermal comfort, RF, infrared, air conditioner, Arduino, Raspberry PI.

LISTA DE FIGURAS

Figura 1. Espectro eletromagnético.	23
Figura 2. Diagrama de integração entre os diversos dispositivos presentes em uma residência típica.	24
Figura 3. Roteador D-Link DI-604A Express 4 portas.	27
Figura 4. Aspecto externo da tomada sem fio.	28
Figura 5. Painel frontal do monitor.	29
Figura 6. Comunicação do monitor com o sistema principal.	29
Figura 7. Raspberry PI modelo A.	30
Figura 8. Plataforma de prototipagem eletrônica Arduino UNO.	31
Figura 9. IDE Arduino.	32
Figura 10. Sensor de temperatura LM35.	33
Figura 11. Vista e pinagem do TSOP38328.	34
Figura 12. Diagrama de blocos do TSOP38238.	34
Figura 13. Max Power IR LED Kit.	35
Figura 14. Circuito esquemático do Max Power IR LED Kit.	35
Figura 15. Imagem ilustrativa do módulo RF nRF24L01(+).	36
Figura 16. Sistema binário para o protocolo NEC para controle remoto de televisor.	39
Figura 17. Padrão de envio de mensagem para o protocolo NEC.	39
Figura 18. Controle remoto universal MXT.	41
Figura 19. Arquitetura do projeto SapiEns.	43
Figura 20. Dois modelos de condicionador de ar estudados: (a) Komeco Portátil Ambient ABP 09QC 9000 BTU, (b) Electrolux Split Hi-Wall 9000 BTU	44
Figura 21. Arquitetura de operação do módulo de comunicação e do módulo central.	45
Figura 22. Interface gráfica do programa AnalysIR.	48
Figura 23. Código hexadecimal dos A/C Electrolux (parte superior) e Komeco (parte inferior), para o comando Ligado, frio, ventilação automática, 25°C, swing desligado.	51
Figura 24. Placas de circuito impresso para (a) Raspberry PI e (b) Arduino.	56
Figura 25. Fluxograma de operação para o Raspberry Pi.	64
Figura 26. Fluxograma de operação para o Arduino.	64

LISTA DE TABELAS

Tabela 1. Opções de taxa de transmissão de dados para a comunicação RF.	37
Tabela 2. Modos de atenuação do sinal RF.	37
Tabela 3. Parâmetros de acionamento para o condicionador de ar.	46
Tabela 4. Lógica de ajuste do condicionador de ar.	47
Tabela 5. Códigos hexadecimais Electrolux para acionamento em ventilação automática com swing desligado.	52
Tabela 6. Códigos hexadecimais Komeco para acionamento em ventilação automática com swing desligado.	52
Tabela 7. Cronograma de etapas desenvolvidas ao longo do projeto e da implementação do protótipo	60

LISTA DE ABREVIATURAS E SIGLAS

CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CPS	Sistemas Ciberfísicos (<i>Cyber-Physical Systems</i>)
Abradee	Associação das Empresas Distribuidoras de Energia Elétrica
EPE	Empresa de Pesquisa Energética
Cieam	Centro das Indústrias do Estado do Amazonas
ABRAVA	Associação Brasileira de Refrigeração, Ar Condicionado, Ventilação e Aquecimento
API	Interface de Programação de Aplicações (<i>Application Programming Interface</i>)
IR	Infravermelho (<i>Infrared</i>)
RF	Radiofrequência
LED	Diodo Emissor de Luz (<i>Light Emitting Diode</i>)
IDE	Ambiente Integrado para Desenvolvimento de Software (<i>Integrated Development Environment</i>)
A/C	Condicionador de Ar com acionamento via controle remoto (<i>Air Conditioner</i>)
AGC	Controle Automático de Ganho (<i>Automatic Gain Control</i>)
OEM	Onda eletromagnética
LIRC	Controle Remoto Infravermelho Linux (<i>Linux Infrared Remote Control</i>)
USB	Barramento Série Universal (<i>Universal Serial Bus</i>)
SSH	<i>Secure Shell</i>
TCP/IP	Protocolo de Controle de Transmissão/Protocolo de Internet (<i>Transmission Control Protocol/Internet Protocol</i>)

SUMÁRIO

1	INTRODUÇÃO.....	15
2	CONTEXTO E MOTIVAÇÃO DO PROJETO	18
2.1	Sistema ciberfísico.....	18
2.2	O projeto SapiEns e o consumo de energia	19
2.3	Conforto térmico.....	21
3	FUNDAMENTOS TEÓRICOS.....	22
4	DESCRIÇÃO DA ARQUITETURA.....	24
4.1	O Sistema Residencial SapiEns	24
4.1.1	<i>Visão global do sistema</i>	24
4.1.2	<i>Roteador</i>	26
4.1.3	<i>Tomada sem fio</i>	27
4.1.4	<i>Módulo de monitoramento</i>	28
4.1.5	<i>Unidade central</i>	30
4.1.6	<i>Módulo de comunicação</i>	31
4.1.7	<i>O módulo RF</i>	36
4.2	Controle remoto infravermelho	38
4.2.1	<i>Teoria geral de funcionamento de controles remotos</i>	38
4.2.2	<i>O controle remoto universal e os condicionadores de ar</i>	40
5	DESCRIÇÃO DO PROJETO DESENVOLVIDO	42
5.1	Cronograma e etapas adotadas.....	42
5.2	Arquitetura geral do protótipo para o Sistema SapiEns.....	43
5.3	Arquitetura de operação dos módulos central e de acionamento.....	44
5.4	A ferramenta de análise de protocolos IR AnalysIR	47
5.5	Testes	49
5.5.1	<i>Teste de funcionalidade</i>	49
5.5.2	<i>Teste de integração</i>	50
6	VALIDAÇÃO EXPERIMENTAL	51
6.1	Protocolos de acionamento IR	51
6.2	Teste de funcionalidade	54

6.3	Teste de integração	55
6.4	Placas de circuito impresso.....	56
7	CONCLUSÕES.....	57
8	PROPOSTA PARA TRABALHOS FUTUROS	58
	REFERÊNCIAS BIBLIOGRÁFICAS	59
	ANEXO A. CRONOGRAMA DESENVOLVIDO.....	60
	ANEXO B. FLUXOGRAMA DE OPERAÇÃO PARA RASPBERRY PI E ARDUINO	64

1 INTRODUÇÃO

Os avanços recentes na tecnologia permitem que se viva em ambientes cada vez mais confortáveis e automatizados. Graças à disseminação dos sistemas embarcados, dos processos de controle e dos sistemas de comunicação, tornou-se possível a automatização de tarefas e essa é uma tendência global, que se estende naturalmente aos ambientes residenciais. Segundo (BOLZANI, 2007), a automação, tanto residencial quanto predial, demonstra que a integração de dispositivos eletroeletrônicos e eletromecânicos aumenta consideravelmente os benefícios, comparativamente aos sistemas isolados, de eficiência limitada. Segundo o mesmo autor, ela é também uma aliada na redução do consumo de recursos como água e energia elétrica, além de trazer maior conforto e segurança aos usuários. Tipicamente, costumam-se monitorar a automatizar processos de controle de temperatura ambiente, de iluminação e de som por meio de uma central de comando, que pode estar conectada a um computador ou à internet. Nesses casos, é comum que se usem sensores de presença, de luminosidade, de temperatura e câmeras; como atuadores, encontram-se motores para o acionamento de portas e janelas, lâmpadas, aquecedores, alto-falantes e sistema de alarme.

Ao mesmo tempo, contudo, as exigências com relação ao conforto aumentam constantemente, além do fato de cada usuário apresentar comportamentos e exigências peculiares. Numa dimensão mais ampla do que a simples automação residencial, fala-se em domótica, ciência de caráter multidisciplinar (Arquitetura, Engenharia, Ciência da Computação, Medicina, Sociologia e Psicologia) que visa a estudar as necessidades do usuário frente às possibilidades oferecidas pelo mundo digital e suas interações com a residência automatizada (BOLZANI, 2007). Assim, não somente a automatização de tarefas é contemplada, mas também o desenvolvimento de sistemas inteligentes (*smart*), capazes de se adaptar às diferentes necessidades do cliente.

Enfim, a necessidade de que sistemas desse tipo possam estar presentes em todo lugar, possam ser utilizados por qualquer pessoa para realizar qualquer tarefa, possam tomar decisões e se comunicar de maneira autônoma com quaisquer outros dispositivos e possam se conectar diretamente com o mundo físico motivam o uso de outro conceito, que são os chamados Sistemas Ciberfísicos (*Cyber-physical Systems - CPS*). Um CPS é a integração entre dispositivos eletrônicos computacionais e processos físicos, isto é, a intersecção, não a união, entre o que é físico e o que é virtual (CHASE, 2013).

Outra questão de interesse de um tal sistema concerne a eficiência energética. Usar a energia de maneira eficiente é um dos desafios das inovações tecnológicas do século XXI.

Trata-se não apenas de uma questão de tecnologia, mas também de conscientização por parte do consumidor. No Brasil, o setor residencial representa a segunda maior parcela do consumo total de energia elétrica, perdendo apenas para o setor industrial (ANEEL, 2011). A análise dos dados de consumo energético do país apontam para uma conscientização para o racionamento de energia por volta dos anos 2000. Contudo, a partir de 2003, o consumo residencial cresceu monotonamente, o que, para muitos analistas, é consequência do abandono gradual, pela população em geral, das práticas de consumo eficiente de racionamento, adotadas durante o racionamento (ANEEL, 2011). A fim de que o usuário doméstico mude de postura e possa controlar de maneira eficiente o seu consumo de energia, é necessário o fácil acesso às informações detalhadas relativas ao seu consumo, fato que é diferente da realidade, como comprova um estudo realizado pela IBM (IBM Institute for Business Value, 2007).

Com o intuito de atender a essas exigências e de mudar esse cenário, o projeto Sapi-Ens, da empresa Globosense, idealiza uma solução voltada especificamente aos consumidores domésticos brasileiros. O projeto em si prevê a possibilidade de gerenciar equipamentos à distância, via internet e via *app smartphone*, podendo-se ter fácil acesso ao consumo energético de cada equipamento em tempo real. O projeto também objetiva a proposta de soluções de controle inteligente de acionamento de recursos, como o estudo de acionamento do condicionador de ar na manutenção do conforto térmico e seu impacto no consumo energético, o uso de câmeras e/ou de sensores de presença para o desligamento de equipamentos como televisores e lâmpadas em ambientes nos quais não há pessoas presentes, o uso de câmeras para detecção de situações de perigo devido a equipamentos esquecidos ligados, entre outros. A proposta é levar ao cliente um produto que lhe forneça maior conforto, segurança e eficiência energética. Assim, será possível que se executem tarefas de maneira autônoma, que o cliente entenda melhor o valor da sua fatura de consumo de energia elétrica e que se reduzam desperdícios de energia. O sistema descrito deverá ter baixo custo, ter interface amigável e ser de fácil instalação a fim de que ele se torne um produto acessível aos consumidores brasileiros de todas as classes.

Com essa motivação, o presente trabalho ataca a questão do projeto e da implementação do *hardware/software*, considerando a abordagem de CPS, com foco no desenvolvimento de um dispositivo de acionamento de um condicionador de ar, visto que esse equipamento é o responsável pela maior parcela na fatura de energia elétrica residencial. Propõe-se, assim, a elaboração de uma arquitetura modular para o sistema, devendo este contemplar um módulo dedicado ao acionamento do condicionador de ar e à comunicação com um módulo, denominado central, de maneira a contemplar as especificações de projeto supracitadas. Dessa manei-

ra, esse dispositivo acionador deverá contar com uma lógica de tomada de decisões que permita o ajuste automático dos parâmetros A/C de modo a manter uma temperatura considerada agradável por um consumidor particular que o utilize.

Este trabalho está dividido de tal maneira que se entendam os conceitos envolvidos, os passos adotados e os resultados obtidos. Assim, no Capítulo 2, serão apresentadas as principais motivações que justificam o objetivo do projeto. No Capítulo 3, são apresentados os fundamentos teóricos referentes às ondas infravermelho. No Capítulo 4, é descrita a arquitetura do sistema bem como as características básicas de seus equipamentos constituintes. No Capítulo 5, encontra-se a metodologia de trabalho adotada e dá-se um enfoque técnico mais detalhado ao modo como o protótipo foi desenvolvido e integrado, de modo a atender as especificações de projeto. No Capítulo **Erro! Fonte de referência não encontrada.**6 são discutidos os resultados obtidos. No Capítulo 7, encontram-se os principais elementos de conclusão do projeto. No Capítulo 8, sugerem-se eixos de desenvolvimento para trabalhos futuros.

2 CONTEXTO E MOTIVAÇÃO DO PROJETO

2.1 Sistema ciberfísico

Um CPS pode ser entendido como uma extensão do conceito de sistemas embarcados. Assim como um sistema embarcado, um CPS é caracterizado por ser um sistema microprocessado, no qual o computador é completamente dedicado ao(s) dispositivo(s) que controla. A principal diferença entre esses dois conceitos, conforme (WAN-A, 2010) consiste na característica de que um CPS é projetado para conectar dispositivos físicos de modo a ser criada uma rede de interação. A maneira mais típica de construção de um CPS consiste em se embarcarem sensores e atuadores em dispositivos eletrônicos. Assim, a informação proveniente do mundo físico é coletada pelos sensores e é transmitida a um Sistema de Decisão (também chamado controlador) ou ao próprio usuário. Uma vez recebida a informação dos sensores pelo Sistema de Decisão, este compara os dados aos valores desejados, por meio de uma sequência de processos de controle, e aciona os atuadores correspondentes às tarefas necessárias. Visto que as redes *wireless* de sensores e de atuadores são a ponte entre os mundos físico e cibernético, elas são de extrema importância para o bom funcionamento do sistema.

Como descrito em (LEE, 2010), um modelo de CPS deve comportar o mundo físico, sensores e atuadores, plataforma de *hardware*, *software*, rede (*network*) e sistema de controle. Além disso, estão inclusos os conceitos de *timing*, de concorrência e de dinâmica.

Campos de pesquisa envolvendo CPS em engenharia vêm sendo consideravelmente explorados devido às questões de convergência de sensores, de controle, de computação e de comunicação. As aplicações de CPS's apresentadas em (WAN-B, 2010) englobam sistemas e dispositivos médicos, assistência à autonomia, controle de tráfego, sistemas automotivos avançados, processos de controle, instrumentação, infraestrutura crítica (como energia e água), robótica distribuída, sistemas de armamentos, manufaturas, controle e comando distribuídos, estruturas inteligentes, biosistemas e sistemas de comunicação.

A utilização dos conceitos de CPS no presente trabalho visa ao controle de temperatura de um ambiente residencial, independentemente das mudanças dinâmicas de temperatura do ambiente externo. A motivação para esse ponto de partida será explorada na seção sobre conforto térmico.

2.2 O projeto SapiEns e o consumo de energia

O projeto SapiEns é um projeto bastante amplo, em fase de pesquisa, que visa ao desenvolvimento de um produto de elevado valor agregado para o sistema energético brasileiro. Em seu aspecto amplo, o projeto tem como objetivo o estudo e o desenvolvimento de metodologias e tecnologias capazes de incluir o consumo como elemento inteligente em *Smart Grids* sustentáveis e eficientes de forma a contribuir para o desenvolvimento científico e tecnológico e para a inovação do Setor Elétrico Brasileiro.

De maneira mais específica, propõe-se que se desenvolvam:

- Uma taxonomia de métodos para automação do consumo residencial de energia, levando em conta aspectos psicológicos associados à motivação do indivíduo para a conscientização de consumo sustentável
- Um estudo de caso de forma a avaliar os perfis de consumo individual em residências, de modo a se tornar possível a elaboração de gatilhos motivacionais de posturas de consumo consciente;
- Um protótipo *Smart Trigger*, a ser integrado em domicílios presentes em *Smart Grids* (30 residências no Brasil), capaz de motivar o usuário a realizar um consumo consciente de energia elétrica, além de monitorar o consumo e atuação remotos em tempo real via *web apps*, SMS e *smartphones*.
- Um protótipo SapiEns, o qual adiciona a possibilidade de adotar diferentes níveis de automação do consumo, de forma a permitir a realização de operações e controles autônomos (sem a interferência explícita do usuário);
- Um estudo de caso de forma a testar e validar a metodologia e o protótipo SapiEns como quarta etapa do processo de fornecimento de energia em um escopo Smart Grid.

A execução da proposta engloba uma equipe multidisciplinar em rede com o Grupo de Automação e Robótica Inteligente da FURG, o Programa de Engenharia Elétrica representado pelo Grupo de Controle, Automação e Robótica (GCAR) da UFRGS, a Empresa do Setor Elétrico de Distribuição CEEE-D e a empresa Globosense, que se localiza no espaço Tecnopuc, em Viamão. O CNPq é o órgão que auxilia financeiramente a pesquisa por meio da distribuição de bolsas às equipes de pesquisa.

O foco da equipe da UFRGS está na questão da definição dos dispositivos a serem utilizados nas residências dos futuros clientes, bem como o método de comunicação entre tais dispositivos. O foco deste trabalho, em particular, foi o desenvolvimento da arquitetura básica

para um módulo de comunicação autônomo entre o sistema SapiEns e um módulo de acionamento para um aparelho de condicionador de ar. O projeto do módulo de acionamento deveria contemplar algoritmos de ajuste da temperatura de referência e de modo de operação do condicionador de ar baseado na leitura de temperatura do ambiente e nas preferências ajustáveis do usuário do equipamento. À equipe da FURG, reservou-se a tarefa de implementar métodos em *software* para a extração do consumo elétrico dos equipamentos a partir de tomadas inteligentes, que foram disponibilizadas pela empresa Globosense e cujas características principais serão detalhadas mais adiante neste trabalho. A execução desta tarefa passaria pelo uso de ferramentas de banco de dados e de servidor *web*.

O motivo de escolha dos A/C como ponto de partida para a parte do acionamento deve-se ao fato de que eles representam, hoje, grande parte do consumo de energia elétrica das residências brasileiras. Embora não haja estudos que quantifiquem precisamente o percentual médio gasto em energia elétrica com condicionamento de ar em residências, devido ao fato de isso ser fortemente dependente com a localização geográfica da residência e com o perfil de cada usuário, a análise de sistemas de potência sugere que o aparelho de ar condicionado é responsável por parcela significativa do consumo elétrico residencial nos últimos anos. Em depoimento concedido especialmente para a realização deste projeto, o Eng. Maicon Ramos, engenheiro de operação de sistema da AES Sul, conta que um dos indícios de que os condicionadores de ar representam parcela significativa do consumo é o fato de que a carga elétrica total é fortemente dependente da temperatura. Segundo ele, em épocas muito quentes, a demanda de energia elétrica durante a madrugada fica praticamente constante, sugerindo uma tendência de uso do aparelho durante este período do dia. Outro indício é o deslocamento gradual, nos últimos anos, do pico de consumo para a faixa das 14:00 às 15:00, o que costumava acontecer em um horário mais próximo do final da tarde.

Além disso, segundo o presidente da Abradee, Nelson Leite, o ar condicionado é o novo chuveiro elétrico, como declarado em (PORTAL PCH, 2014). Segundo dados da (EPE, 2013), o consumo residencial brasileiro cresceu 11,4% para o mês de janeiro de 2013, registrando o maior crescimento da classe desde 2005. Segundo a mesma fonte, o consumo residencial de energia elétrica é muito sensível à temperatura. Calor intenso e permanência de temperaturas elevadas por vários dias refletem-se necessariamente na expansão do consumo e, não raro, em maior pressão sobre a rede elétrica de distribuição. Segundo o Cieam, somente no último trimestre de 2012 haviam sido produzidos cerca de 1 milhão de condicionadores, volume cerca de 200% maior do que o produzido em igual período de 2011.

Segundo dados de (EPE, 2014), em 2013 o aumento do consumo anual de eletricidade no país foi de 3,5%, somando 463,7 mil gigawatts-hora (GWh), puxado novamente pelo consumo das residências, especialmente na região nordeste, em que a taxa de crescimento atingiu 11,5%. O órgão aponta, novamente, uma grande parcela de influência dos aparelhos de ar condicionado para o crescimento. De acordo com a ABRAVA, cerca de 15% das residências do país possuem, hoje, equipamentos de condicionamento do ar, com perspectiva positiva para ampliação desta parcela em bom ritmo nos próximos anos, segundo a mesma empresa.

Em 2013, o professor Edmilson Moutinho dos Santos, presidente do Programa de Pós-Graduação em Energia da Universidade de São Paulo, afirmou: “as redes de energia nos principais centros consumidores estão saturadas”.

Portanto, o desenvolvimento de um módulo de comunicação com o condicionador de ar, assim como de algoritmos de decisão que evitem o desperdício de energia, tem uma motivação econômica justificada.

2.3 Conforto térmico

A necessidade em se controlar o conforto térmico dentro de casa é um tema largamente reconhecido e estudado mundialmente. O conforto térmico é a condição mental que expressa satisfação com o ambiente térmico em que se está inserido (ASHRAE, 2010). O conforto térmico é subjetivo. Porém, de maneira geral, ele depende de parâmetros não controláveis como sexo, idade e características fisiológicas de cada pessoa e isolamento das paredes do ambiente. Ele também depende de parâmetros controláveis como temperatura, umidade e velocidade do ar.

O protótipo tratado por esse trabalho destina-se ao ajuste automático dos parâmetros temperatura de referência e modo de operação do condicionador de ar. Visto que, uma vez definida a temperatura de referência do condicionador de ar pelo usuário, o controle de temperatura do ambiente é feito pelo próprio aparelho de ar condicionado, a proposta aqui não é modificar esse controle.

A topologia de CPS deve assegurar a possibilidade de que outros parâmetros sejam incluídos na malha de controle em projetos futuros (expansibilidade), de sorte a refinar o modelo e a garantir uma maior eficiência no controle de conforto térmico. Como exemplos desses parâmetros, pode-se citar a umidade relativa do ar, o horário do dia, o estado de outros aparelhos no cômodo e o histórico de utilização do aparelho de ar condicionado.

3 FUNDAMENTOS TEÓRICOS

Este capítulo apresenta uma revisão de conceitos pertinentes envolvendo ondas eletromagnéticas e, mais especificamente, ondas infravermelho. Esses conceitos são uma base para o melhor entendimento da operação de controles remotos infravermelho utilizados em eletrodomésticos convencionais.

O princípio de funcionamento do controle remoto está baseado nas ondas eletromagnéticas. Heinrich Hertz mostrou a existências das OEM ao utilizar um aparelho que produzia faíscas elétricas e um receptor colocado a alguns metros de distância, em que era possível observar a chegada das OEM emitidas, com comportamento similar ao da luz visível. Em 1895, Guglielmo Marconi, Nobel de física de 1909 e inventor do rádio, realizou experimentos com essas ondas, emitindo impulsos que constituíam as letras da palavra SOS do código Morse. O experimento de Marconi foi um marco importante para impulsionar o desenvolvimento da tecnologia da transmissão de informação por meio de OEM.

O controle remoto pode utilizar uma de duas tecnologias: RF (radiofrequência) e IR (infravermelho). Neste trabalho, será enfatizada a segunda tecnologia, por ela ser o objeto de estudo do projeto e por ela representar a maior parte das aplicações comerciais e residenciais.

Ao se pressionar um botão em um controle remoto, um circuito eletrônico aciona um ou mais LEDs IR, os quais enviam um sinal na faixa do infravermelho ao dispositivo a ser acionado.

A luz infravermelha é uma categoria específica de OEM. Conforme (HALLIDAY & RESNICK, 2007), uma OEM é a interação entre campos elétrico e magnético, variantes no tempo, perpendiculares entre si e perpendiculares à direção de propagação da onda, com conseqüente transporte de energia. A propagação de OEM independe da existência de meio material e a velocidade de propagação é dada pela Equação (1):

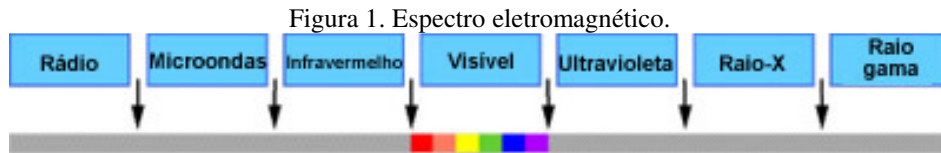
$$v = \lambda \cdot f \quad (1)$$

em que v é a velocidade [m/s], λ é o comprimento de onda [m] e f é a frequência [Hz]. Para propagação no vácuo, a velocidade v é denotada por c e vale $3 \cdot 10^8 m/s$. A energia transportada pela OEM é proporcional à sua frequência, conforme a Equação (2):

$$E = h \cdot f \quad (2)$$

em que E é a energia [J], h é a constante de Planck [$6,6 \cdot 10^{-34}$ J · s] e f é a frequência [Hz].

As OEM são comumente subdivididas em classes de ondas, conforme sua faixa de frequência e o que elas representam em termos práticos. O espectro eletromagnético é mostrado na Figura 1.



Fonte: <http://crcontroleremoto.blogspot.com.br/2012/06/como-funciona-seu-controleremoto.html>.

Assim, a luz infravermelha é um tipo particular de OEM. Tipicamente referida por calor, trata-se de OEM que interagem com os objetos do mundo físico, sendo absorvidas quase integralmente, como se os objetos fossem corpos negros. Dessa forma, esses objetos sofrem conseqüente aumento de temperatura. A faixa de frequências para o infravermelho é de 3×10^{12} Hz a $4,3 \times 10^{14}$ Hz, ou, em termos de comprimento de onda, de $0,7 \mu\text{m}$ a $1000 \mu\text{m}$. Nessa faixa de frequências, portanto, o IR é invisível ao olho humano. Contudo eles podem ser visualizados em câmeras digitais com dispositivo de carga acoplada, ou sensores CCD.

4 DESCRIÇÃO DA ARQUITETURA

Neste capítulo, é apresentado o contexto técnico do trabalho, em que se contemplam os subsistemas existentes dentro do protótipo residencial SapiEns e o funcionamento de aparelhos de controle remoto comerciais.

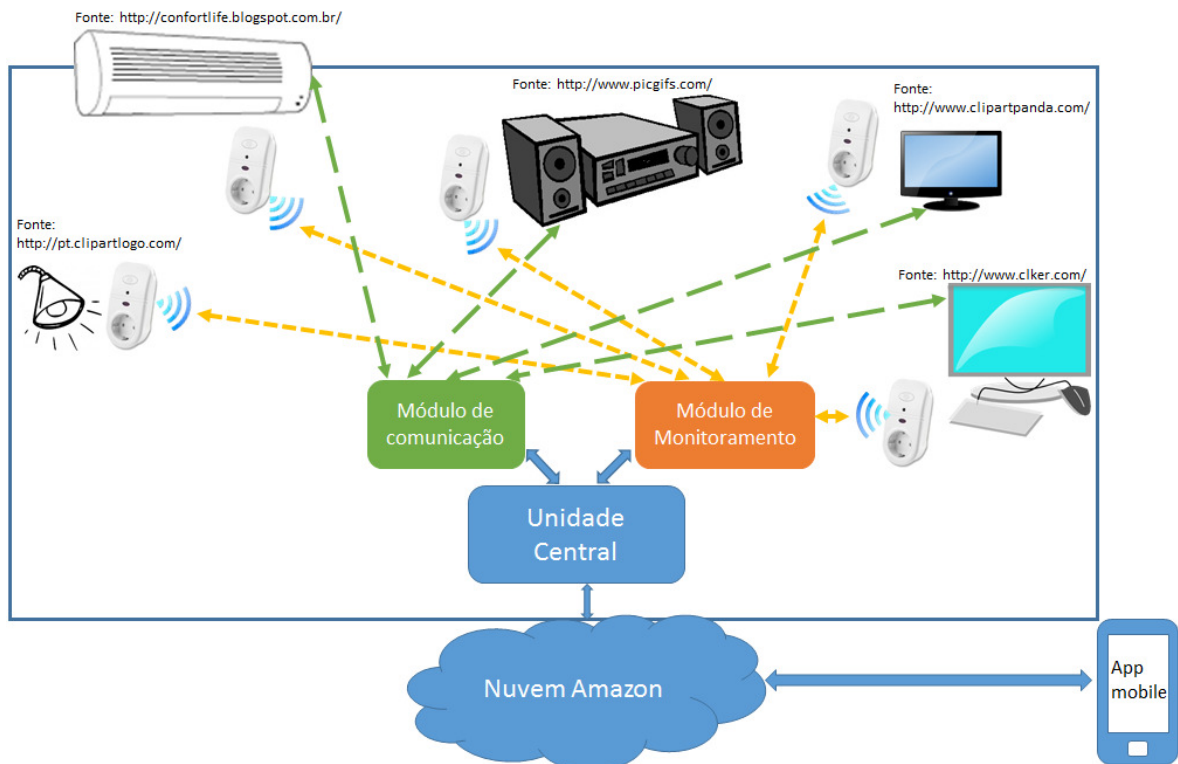
4.1 O Sistema Residencial SapiEns

Nesta seção, descreve-se o funcionamento dos principais recursos do projeto SapiEns e como eles se relacionam entre si e com equipamentos da residência do cliente.

4.1.1 Visão global do sistema

A Figura 2 descreve os principais componentes envolvidos em termos de arquitetura em uma residência típica em conjunto com os sistemas projetados de modo a atender as especificações de projeto.

Figura 2. Diagrama de integração entre os diversos dispositivos presentes em uma residência típica.



Do diagrama da Figura 2 identificam-se os dispositivos já presentes na residência do cliente, que compreendem basicamente dois grupos de aparelhos: os puramente elétricos, como lâmpadas convencionais, aquecedores de água e refrigeradores, os quais podem ser desligados por simples interrupção da passagem de corrente elétrica, e os eletrônicos, como televisores, condicionadores de ar, computadores e aparelhos de som, os quais requerem, em geral, algum procedimento para seu desligamento seguro, como pressionar um botão de um controle remoto, por exemplo. As especificações de projeto exigem que o consumo instantâneo de cada grupo de equipamentos seja monitorado além de que o estado de cada equipamento seja comandado à distância. Desse modo, faz-se necessário o desenvolvimento de um módulo dedicado à comunicação com o segundo grupo de equipamentos, de modo a garantir o desligamento seguro dos mesmos. Os detalhes desse módulo serão discutidos em seções subsequentes deste trabalho.

Outro elemento empregado nesse sistema é a tomada elétrica sem fio, da própria empresa Globosense. Esse dispositivo permite que se monitorem o acionamento e o consumo energético de um equipamento da residência do cliente. As informações são trocadas via protocolo *wireless Zigbee* com o módulo de monitoramento. Em termos de produto final, deverá haver uma tomada dessas para cada equipamento consumidor de energia elétrica na casa do cliente. A razão pela qual esse dispositivo deve ser usado é a condição de projeto que exige que o sistema seja de fácil instalação por parte do usuário. Assim, com ela, dispensa-se qualquer alteração na instalação elétrica na residência do cliente. Maiores detalhes da tomada sem fio são apresentados na seção 4.1.3.

Quanto ao módulo de monitoramento, trata-se de um *watchdog*, fornecido pela empresa Globosense, o qual permite gerenciar até 14 tomadas sem fio. Esse dispositivo possibilita a troca de informações com um PC por meio de um cabo RJ45-USB e de uma API apropriada. Mais detalhes sobre o funcionamento desse dispositivo são apresentados na seção 4.1.4.

O módulo de comunicação ilustrado na Figura 2 é um dos focos deste trabalho e terá, portanto, um maior detalhamento em seções subsequentes deste. Em linhas gerais, a existência desse módulo visa a permitir o desligamento de equipamentos eletrônicos que não podem simplesmente ter sua alimentação elétrica cortada de forma abrupta. Além do acionamento e do desligamento, objetiva-se, com o módulo, ajustar parâmetros do equipamento, como se faz com o uso de um controle remoto. Assim, o módulo se localiza próximo ao equipamento a que se destina, devendo, portanto, se comunicar com o sistema principal sem o uso de cabos ou fios. No caso do presente trabalho, o equipamento a ser comandado é o A/C. O núcleo desse módulo é uma plataforma de prototipagem do tipo Arduino.

O sistema principal é composto de um computador Raspberry PI, com um sistema operacional ArchLinux, uma plataforma gratuita, recente, estável e bem documentada. Inicialmente havia sido escolhida uma plataforma Linux para o minicomputador. Contudo, essa plataforma não suporta o serviço de banco de dados MongoDB, necessário para salvar o histórico de consumo elétrico dos equipamentos monitorados pelas tomadas inteligentes. Assim, usando-se a plataforma ArchLinux, usa-se um *webserver* que permite o gerenciamento e a comunicação com todos os dispositivos dentro da residência. Ademais, esse servidor se comunica com um servidor Linux configurado no serviço Amazon, o que possibilita o acesso remoto via *app smartphone* ou via navegador de Internet.

Nas seções numeradas de 4.1.2 a 4.1.7, serão apresentados com maior grau de detalhe os principais dispositivos de *hardware* envolvidos na elaboração do projeto. Serão descritos seus modos de operação e o papel que desempenham dentro do projeto SapiEns como um todo.

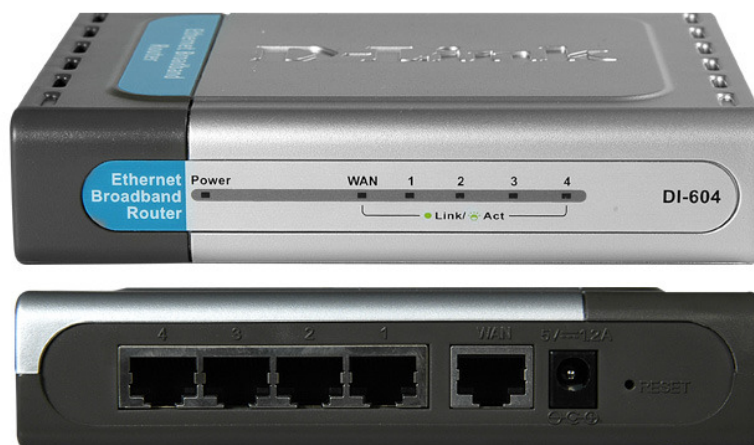
4.1.2 Roteador

O roteador é um aparelho eletrônico que encaminha pacotes de dados entre redes de computadores. Um roteador é conectado a duas ou mais linhas de redes de comunicação diferentes. Quando um pacote de dados chega em uma das linhas, o roteador lê a informação de endereço no pacote para determinar seu destino final. A partir da sua tabela de encaminhamento, ele direciona o pacote para a rede mais apta a recebê-lo. Assim, os roteadores respondem pelo tráfego de dados na Internet. Os roteadores mais conhecidos são aqueles encontrados em residências e em escritórios, que simplesmente entregam dados como páginas *web*, *email* e vídeos ao computador a partir da internet.

Além disso, o roteador permite a criação de redes locais de dispositivos. O roteador define os números IP aos dispositivos conectados às suas portas, de modo que seja possível acessá-los a partir de outros dispositivos conectados à mesma rede. Essa funcionalidade é o que permite o acesso ao módulo central a partir da interface *frontend*, por parte do usuário.

O roteador adotado no protótipo é o modelo DI-604A Express, 4 portas, da D-Link, cujo aspecto físico está apresentado na Figura 3.

Figura 3. Roteador D-Link DI-604A Express 4 portas.



Fonte: <http://shopmidia.com.br/produto/Roteador-D-LINK-DI%252d604%252dA-Express-4-Portas.html>.

O modelo adotado para o projeto permite a criação de uma rede com até 4 dispositivos via cabo, além do compartilhamento de internet entre eles. O acoplamento de dispositivos às portas do roteador é feito por cabos Ethernet, ou cabo RJ5.

4.1.3 Tomada sem fio

A tomada sem fio da Globosense pode ser entendida com um dispositivo de tomada dotado de um módulo de comunicação *Zigbee*, de relés eletromecânicos comandáveis a partir da recepção de comandos específicos do protocolo citado e de um medidor de consumo elétrico. Essa tomada deve ser inserida entre a rede elétrica e um equipamento residencial cujo consumo e cujo acionamento se deseje monitorar. Dessa maneira, a comunicação *wireless* com o módulo de monitoramento permite que se ligue/desligue o equipamento à distância, além de se conhecer o seu consumo a todo instante. Por meio do comando apropriado, a tomada permitirá/cortará a condução de corrente através dos terminais do equipamento ao qual está associada, por meio do chaveamento eletromecânico. O consumo de energia elétrica é medido e enviado em tempo real para o módulo de monitoramento.

A Figura 4, a seguir, mostra o aspecto da referida tomada.

Figura 4. Aspecto externo da tomada sem fio.



Se a condução estiver acionada, o LED vermelho frontal estará aceso, sinalizando o contato entre os terminais da tomada e os da rede. O botão, também na parte frontal, permite que se corte/habilite a condução manualmente, caso seja necessário.

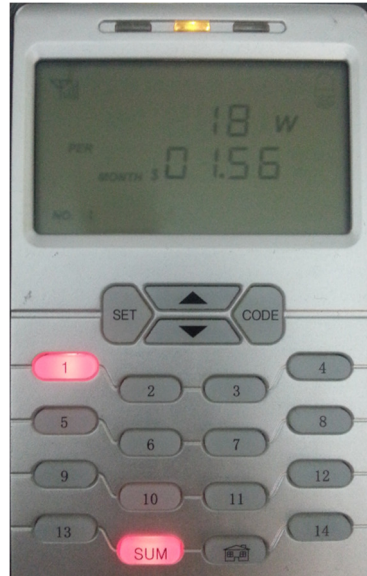
Devido ao atual fornecedor da empresa Globosense, o padrão de tomada utilizado é o europeu, o que requer o uso de adaptadores para que sejam possíveis os testes no Brasil. A empresa Globosense trabalha atualmente para que as tomadas do seu produto sejam produzidas no padrão brasileiro, o que dispensaria o uso de adaptadores.

4.1.4 *Módulo de monitoramento*

O módulo de monitoramento, que pode ser chamado de monitor de energia ou simplesmente de monitor, é um dispositivo dotado de um *display* digital e de um teclado na parte frontal, e de um suporte plástico e de uma entrada RJ45 fêmea na parte traseira. O monitor de energia permite o pareamento com as tomadas descritas na seção 4.1.3 de modo a comandá-las à distância (ligar ou desligar) e a obter o consumo em watts dos equipamentos a elas acoplados. A forma de comunicação entre o monitor e as tomadas é o protocolo Zigbee, cujas vantagens são a baixa potência de operação, a baixa taxa de transmissão de dados e o baixo custo de implantação. Comparativamente às redes Wi-Fi e Bluetooth, o alcance Zigbee é inferior (cerca de 100m). Contudo, essa rede é projetada para que seus dispositivos operem em modo de repetidor, ou seja, a comunicação entre duas unidades pode ser repetida sucessivamente pelas outras unidades existentes na rede até se atingir o destino final.

O monitor descrito pode se comunicar com até 14 tomadas de forma independente. O aspecto visual do painel frontal é mostrado na Figura 5.

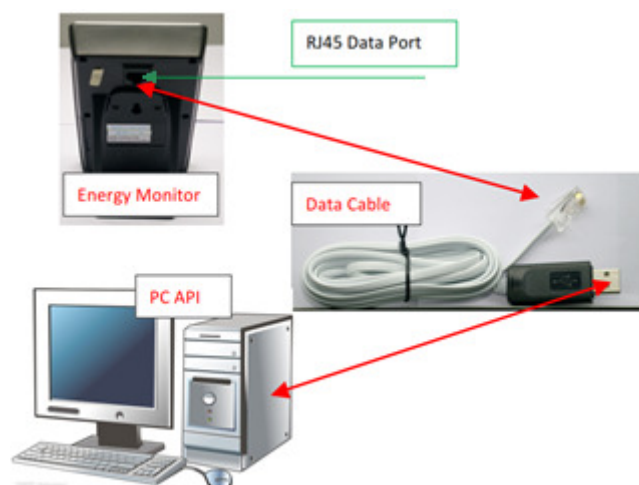
Figura 5. Painel frontal do monitor.



No painel frontal, podem ser visualizadas informações como os estados dos plugues pareados, assim como seus consumos. Uma média por hora também é calculada e apresentada no display.

Além de exibir informações no display, o monitor também oferece a opção de enviar informações para um sistema computadorizado por meio de uma *API* e de um cabo RJ45 – USB. O diagrama apresentado na Figura 6 mostra a comunicação entre o monitor e o sistema principal.

Figura 6. Comunicação do monitor com o sistema principal.



4.1.5 Unidade central

A unidade central de processamento, a qual é destinada ao gerenciamento de todos os dispositivos relativos à residência do cliente, é um sistema computadorizado do tipo Raspberry PI. Esse componente foi comparado a similares, como o chamado Beagle Bone e foi escolhido devido ao interesse da empresa Globosense em se usar um dispositivo com boa capacidade de processamento para vídeo, visto que o sistema, futuramente, fará uso de câmeras. Assim, apesar de outros dispositivos possuírem melhor processamento global e disporem de mais pinos de entrada/saída, optou-se pelo Raspberry Pi como solução para o projeto. Além disso, a empresa Globosense, devido a sua origem inglesa, tem facilidades para a aquisição desse componente. A Figura 7 mostra o aspecto físico desse dispositivo.

Figura 7. Raspberry PI modelo A.



Fonte: Adaptado de <http://www.rs-online.com/designspark/electronics/blog/raspberry-pi-model-a>.

O computador de pequenas dimensões comporta um sistema operacional ArchLinux, configurado em um cartão de memória SD, o qual também funciona como disco rígido. Assim como em um computador, o Raspberry Pi tem capacidade para gerenciar periféricos como monitor, teclado, mouse, *pen-drive*, entre outros. Para o desenvolvimento das aplicações, criou-se uma rede com roteador e fez-se o acesso ao dispositivo via SSH, a partir de um terminal Linux Ubuntu 14.04.

O Raspberry PI comporta serviços de *webserver* (Node.js) e de banco de dados (MongoDb), cuja configuração foi realizada pela equipe da FURG. O Raspberry Pi também é responsável pelo gerenciamento de módulo RF para comunicação com o módulo de comunicação, descrito nas próximas seções. Além disso, a unidade central tem acesso e controla as informações referentes ao monitor de energia via porta serial, a qual recebe o cabo de comunicação do monitor.

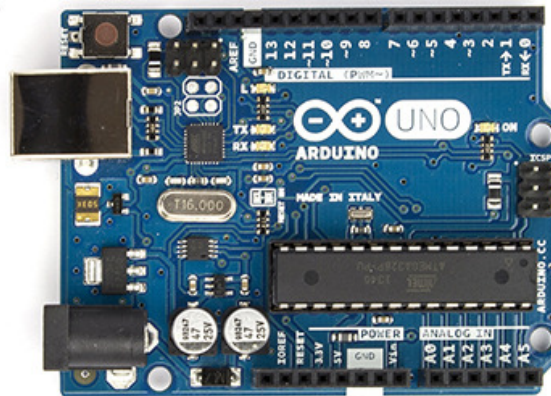
4.1.6 Módulo de comunicação

Nesta seção, serão apresentados os principais recursos de *hardware* e de *software* que compõem o kit de comunicação com o sistema central de gerenciamento.

4.1.6.1 A plataforma de prototipagem eletrônica Arduino

O elemento fundamental do módulo de comunicação é, conforme já mencionando na seção 4.1.1, uma plataforma de prototipagem eletrônica Arduino UNO. A Figura 8 mostra o aspecto visual do dispositivo.

Figura 8. Plataforma de prototipagem eletrônica Arduino UNO.



Fonte: <http://arduino.cc/en/Main/arduinoBoardUno>

A escolha desse dispositivo deve-se ao fato de ele ter ampla documentação, de estar em constante desenvolvimento, de possibilitar uma programação alto nível (C++), de ser livre tanto em *hardware* quanto em *software* e de fazer parte de uma linha com outros modelos, podendo-se alterar, assim, o *hardware* a todo momento sem grandes complicações.

Esta plataforma comporta um microcontrolador ATMEL ATMEGA328 com suporte de entradas/saídas embarcado e uma linguagem de programação padrão, baseada em C/C++. O microcontrolador ATMEGA328 é de 8 bits RISC, com 32 kB de memória ISP *flash* com recurso de leitura durante escrita, 1 kB de memória EEPROM, 2 kB de memória SRAM e 32 registradores de uso geral. Para a programação, compilação e transferência do código ao microcontrolador, usa-se uma IDE própria do Arduino, de caráter livre. Esta IDE, multiplataforma, possibilita que se desenvolvam códigos em um computador pessoal e, via porta USB,

transfira-se o código binário para o microcontrolador. A IDE conta, ainda com uma janela gráfica na qual podem ser impressas mensagens e lidos comandos de teclado, a fim de que sejam feitos testes e simulações.

Além das vantagens de a plataforma ser livre, compacta, relativamente barata, bem difundida e documentada mundialmente, existe a possibilidade de se reproduzir a arquitetura da placa em uma placa dedicada, de tamanho menor, voltada à aplicação de projeto com que se trabalha. Assim, uma vez implementados e validados os códigos na placa Arduino UNO, pode-se transferir o microcontrolador para a placa projetada, sendo esse um dos motivos principais que levaram à escolha dessa ferramenta de trabalho no projeto.

A Figura 9 mostra a interface da IDE Arduino utilizada para o desenvolvimento do programa do microcontrolador.

Figura 9. IDE Arduino.

```
codeKitEnergy_ino | Arduino 1:1.0.5+dfsg2-2
File Edit Sketch Tools Help
codeKitEnergy_ino printf.h
#include <ac_intelligence_kit.h>
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include "printf.h"

RF24 radio(0,10);
// Radio pipe addresses for the 2 nodes to communicate.
const uint64_t pipes[2] = {
    0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };

// Temperature sensor parameters
const int PIN_TEMPERATURE_SENSOR = A0;
char TEMPERATUREROOMTOSEND[7];
const int REFRESH_RATE = 2000; // Tempo de atualização da leitura em
const float CELSIUS_BASE = 0.4888; // Base de conversão para Celsius
int INITIAL_DELAY = 1000;

// Parametros do Led infra vermelho
char ACIR_mode;
int IRSENDERLED = 3;

1 Arduino Uno on /dev/ttyACM0
```

4.1.6.2 O sensor de temperatura LM35

O sensor de temperatura LM35, fabricado pela National Semiconductor, é um sensor de baixo custo, de comportamento linear, calibrado diretamente para a escala °C, com sensibi-

lidade de $10\text{mV}/^\circ\text{C}$ e precisão de $0,5^\circ\text{C}$. Sua faixa de operação é de -55°C a $+150^\circ\text{C}$. Este sensor drena uma corrente inferior a $60\ \mu\text{A}$ e pode ser alimentado entre $4\ \text{V}$ e $30\ \text{V}$, que é uma faixa compatível com as tensões de trabalho do Arduino.

Esse sensor é incorporado diretamente aos terminais de entrada da plataforma de prototipagem. A programação pela IDE permite que se obtenha diretamente a leitura da temperatura em $^\circ\text{C}$ a partir do sensor.

A Figura 10 mostra o sensor LM35.

Figura 10. Sensor de temperatura LM35.



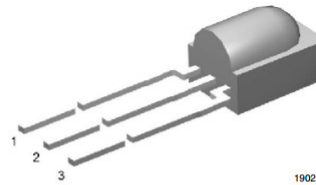
Fonte: <http://www.rustamaji.net/arduino/sensor-suhu-lm35>

O protótipo inicial utiliza apenas um sensor LM35. Contudo, discute-se a possibilidade de se utilizar a informação de 2 ou mais sensores de temperatura a serem distribuídos pelo ambiente do cliente, de modo a se obter uma leitura de temperatura média mais precisa do local. Além disso, com o refinamento e a evolução da malha de controle de temperatura, pode-se discutir a possibilidade de se empregar um sensor de temperatura e de umidade, devendo-se alterar a programação convenientemente.

4.1.6.3 O receptor IR

O componente receptor IR utilizado para efetuar a aquisição de dados foi o TSOP38238, que é um demodulador de sinais IR. Este componente é compatível com os dispositivos de controle remoto comerciais mais encontrados no mercado, a saber aparelhos de televisão, aparelhos para assinatura de canais fechados de televisores e condicionadores de ar. A Figura 11 mostra o aspecto externo e a pinagem do TSOP38238.

Figura 11. Vista e pinagem do TSOP38328.



MECHANICAL DATA

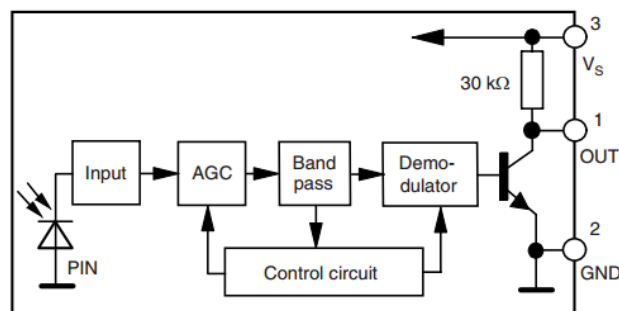
Pinning:

1 = OUT, 2 = GND, 3 = V_S

Fonte: Adaptado de SEMICONDUCTORS, Vishay. IR Receiver Modules for Remote Control Systems. 2009.

O TSOP38328 realiza demodulação para sinais modulados em fase (PM) com frequência portadora de 38kHz. Internamente ao componente, são feitas operações de ajuste controle de ganho (AGC), filtragem passa-baixa e demodulação, como mostrado na Figura 12.

Figura 12. Diagrama de blocos do TSOP38328.



Fonte: Adaptado de SEMICONDUCTORS, Vishay. IR Receiver Modules for Remote Control Systems. 2009.

A tensão de alimentação do TOP38328 deve estar entre 2,5V e 5,5V e a corrente de funcionamento é de 3mA, ao passo que a tensão de saída vai de 0,3V ao valor da tensão de alimentação acrescido de 0,3V. A corrente de saída é de 5mA. Além disso, o dispositivo conta com circuitos de imunização contra luz ambiente e contra variações espúrias da tensão de alimentação.

A partir da análise das especificações do componente, aliada ao baixo custo e ao baixo tamanho, concluiu-se que ele representaria uma ferramenta compatível com os objetivos do projeto. Cabe salientar que esse componente não faz parte do protótipo SapiEns em si, mas é uma ferramenta pela qual passa a aquisição de sinais infravermelho emitidos pelos controles remotos para posterior reprodução com o circuito LED IR emissor, apresentado na seção 4.1.6.4.

4.1.6.4 O circuito de LED IR emissor

O circuito de emissão IR consiste em um circuito transistorizado cuja função é polarizar o LED IR de modo a que este emita o sinal IR na frequência de 38 kHz, com tempos de subida e de descida suficientemente pequenos para o correto acionamento do condicionador de ar. O circuito é vendido pela Sparkfun com o nome de Max Power IR LED Kit. A Figura 13 mostra o aspecto visual desse kit.

Figura 13. Max Power IR LED Kit.

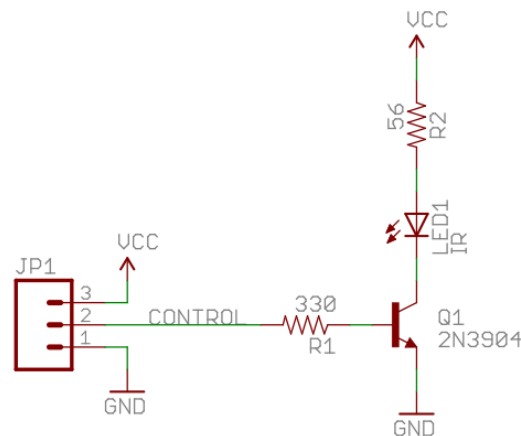


Fonte: <https://www.sparkfun.com/products/10732>.

Em sua operação, os três terminais do módulo IR são diretamente acoplados aos pinos de alimentação, gnd e saída pwm do Arduino, sendo o controle de emissão feito por software.

A Figura 14 mostra o diagrama esquemático correspondente.

Figura 14. Circuito esquemático do Max Power IR LED Kit.

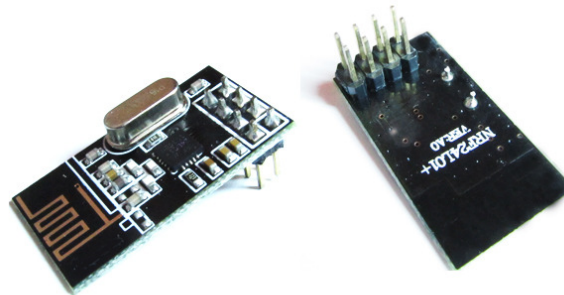


Fonte: <http://dlmh9ip6v2uc.cloudfront.net/datasheets/Kits/PowerIR-v11.pdf>.

4.1.7 O módulo RF

A comunicação entre o Raspberry PI e o Arduino é feita por meio do módulo nRF24L01(+) 2.4 GHz Wireless Transceiver, da empresa Nordic Semiconductor. O uso deste componente para o projeto deve-se ao fato de ele ter licença livre, ter custo acessível, ter pequenas dimensões (15mm x 29mm), não ser direcional e ser compatível tanto em hardware quanto em software com ambos os sistemas microprocessados utilizados no projeto. O fato de o componente não ser direcional permite que os módulos principal e de comunicação possam se encontrar em peças diferentes da residência de um cliente, mesmo com portas fechadas. A Figura 15 mostra o aspecto visual do módulo RF.

Figura 15. Imagem ilustrativa do módulo RF nRF24L01(+).



Fonte: http://www.electfreaks.com/wiki/index.php?title=2.4G_Wireless_nRF24L01p.

Neste trabalho, a comunicação realizada por esse componente será referida simplesmente por RF, por questões de simplicidade. A integração do módulo é feita pela conexão dos seus 8 pinos aos pinos de entrada e saída de uso geral do Arduino e do Raspberry PI, observando-se o nível de tensão utilizado de 3,3V. A biblioteca utilizada para a programação desse componente é a biblioteca RF24.h, desenvolvida para as linguagens C e C++.

Existem quatro modos de operação para cada módulo: baixo consumo, responsivo, transmissor e receptor. Eles podem ser configurados, respectivamente, utilizando-se os métodos `powerDown()`, `powerUp()`, `openReadingPipe()` e `openWritingPipe()`.

Os módulos RF podem operar em uma de três taxas de transmissão de dados: 250kbps, 1Mbps e 2Mbps, bastando-se chamar o método `setDataRate()` com uma das opções indicadas na Tabela 1.

Nome da opção	Taxa de transmissão
RF24_250KBPS	250 kbps (apenas para módulos +)
RF24_1MBPS	1 Mbps
RF24_2MBPS	2 Mbps

Tabela 1. Opções de taxa de transmissão de dados para a comunicação RF.

Outra opção configurável diz respeito à atenuação do sinal enviado, que está relacionado diretamente ao consumo e à distância máxima de comunicação. Utilizando-se o método `setPALevel()`, pode-se escolher uma dentre quatro opções possíveis, conforme mostrado na Tabela 2.

Nome da opção	Atenuação [dBm]
RF24_PA_MIN	-18
RF24_PA_LOW	-12
RF24_PA_MED	-6
RF24_PA_HIGH	0

Tabela 2. Modos de atenuação do sinal RF.

Para que seja possível a comunicação entre dois módulos, devem-se definir *pipes* de comunicação, de modo a se definir qual deles será o emissor e qual será o receptor. O receptor recebe dados de até seis emissores, isto é, existem seis *pipes* distintas dedicadas a essa finalidade. Contudo, no caso do projeto, há apenas um emissor e um receptor.

Quando o emissor emite um sinal, uma *flag* acusa o sucesso ou a falha no envio, permitindo o tratamento de erros. Caso um pacote não seja percebido pelo receptor, o emissor tenta enviá-lo automaticamente por um número de até quinze vezes, antes de que a transmissão seja dada como falha – este número é escolhido pelo método `setRetries()`. Caso o receptor esteja disponível, então o último pacote será enviado para o *buffer*. Além disso, após a recepção de um dado pelo receptor, é possível que os papéis de emissor e transmissor sejam momentaneamente invertidos sem que a *pipe* de comunicação seja alterada. Assim, pode-se operar em modo envio, recepção, resposta.

Cada módulo conta com 3 *buffers* do tipo FIFO, cada um com um tamanho máximo de 32 bytes. Ao se transmitir uma informação com tamanho inferior ao máximo, os primeiros

bytes do *buffer* são preenchidos e os demais bytes recebem o valor zero. Pode-se configurar um tamanho máximo de *buffer* inferior a 32 bytes a fim de se aumentar o desempenho.

Segundo informações do fabricante, a distância máxima de comunicação pode chegar a aproximadamente 100 m, a 250 kbps. A corrente de consumo varia entre 11,3 mA e 13,5 mA e a alimentação deve ficar entre 1,9 V e 3,6 V. É garantido o correto funcionamento do dispositivo na faixa de temperatura de -40°C a 85°C.

4.2 Controle remoto infravermelho

Nesta seção serão apresentados alguns conceitos básicos referentes aos controles remotos via sinais infravermelho e o controle remoto universal utilizado no projeto.

4.2.1 Teoria geral de funcionamento de controles remotos

O conceito dos controles remotos surgiu durante a Primeira Guerra Mundial, com os alemães, que os utilizavam para dirigir navios à distância e os fazer colidir com navios aliados. Já durante a Segunda Guerra Mundial, a tecnologia foi aprimorada e passou a se utilizar o controle remoto para fazer os primeiros acionamentos de bombas. Anos após a grande guerra, começou-se a utilizar a tecnologia de controle remoto para o acionamento de televisores, como alternativa aos botões existentes. Hoje em dia, essa tecnologia está presente não só em televisores, mas também em aparelhos de som, *home theaters*, A/C, entre outros.

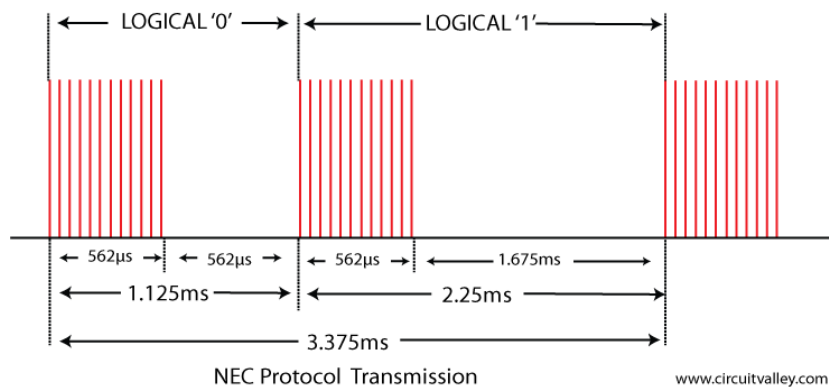
Para utilizar a radiação IR para a comunicação, os circuitos de controles remotos fazem com que o LED IR ligue/desligue em um padrão específico. Diz-se que o LED IR pisca nesse padrão. Assim, por meio da associação de um sistema de códigos (bits 0 e 1) a esse piscar, torna-se possível a comunicação. Dessa forma, quando um LED IR emissor transmite um sinal pelo ar, um receptor IR contido no equipamento a ser acionado recebe o sinal e o transforma apropriadamente em sinais elétricos, que são, em seguida, tratados como dados binários. Para que esse sistema de comunicação seja possível, tanto emissor quanto receptor devem estar minimamente sincronizados (ALMEIDA CAVALCANTE, 2013).

A fim de tornar a comunicação imune a interferências provenientes da luz do sol e de fontes de luz IR, como lâmpadas, velas, sistemas de aquecimento e o próprio corpo humano, o LED não é ligado/desligado estaticamente. O sinal emitido pelo LED é modulado em largura de pulso com uma frequência de modulação de 38 kHz (em alguns aparelhos, também 36 kHz

e 40 kHz), como explicado em (BERGMANS, 2001) e em (SHIRRIFF, 2009). O período de tempo durante o qual o LED está emitindo uma onda quadrada com a frequência de modulação é chamado de marca; aquele durante o qual o LED não está emitindo, espaço.

Com a definição de marcas e de espaços, pode-se codificar e transmitir informação em forma de bits (0 e 1). A Figura 16 mostra um exemplo desse padrão, adotado pelo protocolo NEC, muito utilizado em controles remotos de televisores.

Figura 16. Sistema binário para o protocolo NEC para controle remoto de televisão.



Fonte: <http://www.circuitvalley.com/2013/09/nec-protocol-ir-infrared-remote-control.html>.

Da Figura 16, pode-se entender como os bits são transmitidos e interpretados pelo receptor. Um bit zero corresponde a uma marca de 562 μ s seguida de um espaço de 562 μ s, ao passo que um bit 1 corresponde a uma marca de 562 μ s seguida de um espaço de 1.675 ms.

Outros protocolos para televisores amplamente documentados são: ITT, JVC, Mitsubishi, Nokia, Sharp, Sony, Philips RC-5, Philips RC-6, Philips RC-MM, Philips RECS80, RCA e X-Sat. A diferença básica entre cada protocolo diz respeito às definições dos bits 0 e 1 e a parâmetros adicionais de emissão, a saber, frequência de modulação, sinais de início de comunicação, sinal de cabeçalho, número de envios consecutivos e quantidade de bits por mensagem.

A Figura 17 mostra o envio completo de uma mensagem para o protocolo NEC.

Figura 17. Padrão de envio de mensagem para o protocolo NEC.



Fonte: (BERGMANS, 2001)

A mensagem, transmitida no formato mostrado pela Figura 17 para o protocolo NEC, é recebida e decodificada pelo receptor. É comum que, além da mensagem binária a ser transmitida, sejam enviados sinais de referência, como a marca e o espaço iniciais longos, que servem para iniciar a comunicação.

Embora protocolos de controle remoto para televisores estejam amplamente documentados, o mesmo não ocorre para condicionadores de ar, devido ao seu surgimento mais recente. Em termos de realização de projetos, os protocolos típicos para televisão já se apresentam sob forma de bibliotecas de programação, como a biblioteca IRremote, de Ken Shriff e a base de dados LIRC, ferramentas gratuitas que auxiliam na realização de projetos envolvendo protocolos IR. Contudo, protocolos para condicionadores de ar ainda não estão integrados a tais ferramentas. A base de dados LIRC apresenta apenas um modelo de condicionador de ar (hokkaido), enquanto a biblioteca IRremote trata os protocolos de A/C como formato bruto, ou sem padrão. No entanto, constata-se que a frequência típica de modulação para condicionadores de ar é de 38kHz e que a quantidade de informação transmitida é maior do que para protocolos de televisão, como pode ser constatado em (SHIRRIFF, 2009) e em (SILVA, 2012). Como será visto na seção 5.1, utilizou-se uma ferramenta computacional para Arduino denominada AnalysIR. Essa ferramenta é dedicada à análise de sinais IR e suas características serão explanadas na seção 5.4. Segundo o suporte desse *software*, contatado por e-mail, protocolos de A/C podem ultrapassar 256 bits de informação (contra 32 bits do protocolo NEC), muito mais do que protocolos de TV convencionais.

É comum se encontrarem projetos em que os códigos brutos (tempos de marca e de espaço) são armazenados e depois retransmitido sem se analisar em muito detalhe o funcionamento do protocolo. Contudo, como será mostrado na seção 5.4, é necessária uma mínima compreensão do protocolo a fim de que as informações transmitidas pelos protocolos possam ser compactadas e armazenadas em um sistema com memória limitada, como o Arduino, principalmente se a inclusão futura de outros modelos de protocolos é prevista. Segundo o suporte do AnalysIR, o número de protocolos de acionamento de A/C em todo o mundo é superior a 1000, existindo cerca de 20 tipos diferentes de protocolos para cada fabricante.

4.2.2 *O controle remoto universal e os condicionadores de ar*

Dado o objetivo do projeto em se construir um sistema funcional e passível de incorporar novos protocolos de controles remotos para condicionador de ar, fez-se uso de um con-

trole universal para a reprodução de mais de um protocolo desses equipamentos. O controle remoto universal, da empresa MXT, é mostrado na Figura 18.

Figura 18. Controle remoto universal MXT.



Esse controle universal conta com 1028 variantes de protocolos, que podem ser escolhidas pressionando-se a tecla SET. O produto, importado da China, não contém manual técnico referente aos tipos de protocolos suportados.

Tipicamente, o usuário de um controle remoto desse tipo deve varrer os protocolos possíveis até que o equipamento seja acionado.

5 DESCRIÇÃO DO PROJETO DESENVOLVIDO

Neste capítulo serão detalhadas as características de funcionamento do protótipo, assim como as etapas adotadas durante o seu desenvolvimento de modo a que se alcançassem os objetivos traçados nos prazos solicitados. Na seção 5.1, descreve-se o cronograma executado durante o projeto. Na seção 5.2, apresenta-se a topologia do protótipo residencial integrado. A seção 5.3 foca os aspectos de desenvolvimento dos módulos de acionamento e da sua comunicação com o módulo central, escopo do trabalho. Na seção 5.4, aborda-se o uso do *software* AnalysIR como ferramenta de trabalho para decodificação dos protocolos IF. A seção 5.5 descreve ensaios realizados durante e após o desenvolvimento do sistema.

5.1 Cronograma e etapas adotadas

O Anexo A contém o cronograma seguido durante o desenvolvimento do protótipo. No Anexo A, também se encontra uma lista mais detalhada acerca de cada fase do projeto.

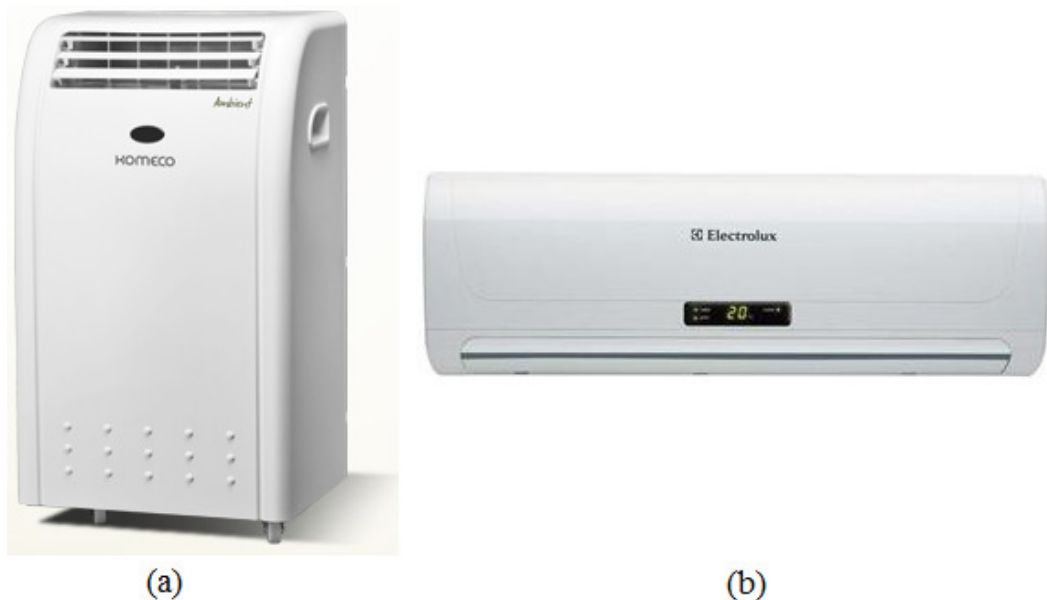
De modo a se chegar a uma arquitetura de *hardware* e de *software* compatível com os objetivos do projeto, inicialmente realizou-se uma revisão bibliográfica sobre as soluções disponíveis aplicáveis ao projeto. Em uma segunda etapa, durante um *workshop* realizado no dia 7 de julho de 2014, a equipe da UFRGS e a equipe da FURG no Parque Tecnológico Tecnopuc se encontraram e foram definidas as metas a serem alcançadas, o protótipo a ser desenvolvido, o *hardware* e o *software* adotados e a divisão das tarefas entre equipes. Definiu-se que haveriam reuniões semanais via Skype entre os integrantes do projeto e que a primeira versão do protótipo deveria ser entregue em três meses a partir da data daquele evento. Para fins de organização e de documentação, utilizou-se uma plataforma de gerenciamento de tarefas *online* chamada Pivotal Tracker.

Os componentes escolhidos para a implementação do protótipo, que são aqueles apresentados no Capítulo 4, foram escolhidos por questões de custo de aquisição, de documentação, de facilidade de aquisição no Brasil, de capacidade de processamento (em havendo), de expansibilidade e de tamanho físico. Também foi adquirida a ferramenta AnalysIR, a qual pode ser adquirida mediante uma doação de valor livre e auxilia na análise de sinais IR sem a necessidade de um osciloscópio ou outro equipamento similar. Maiores detalhes sobre essa ferramenta são fornecidos na seção 5.4.

Uma rotina de comunicação com o monitor de energia é responsável por extrair o consumo e o estado das tomadas inteligentes, assim como por comandá-las à distância. Uma segunda rotina que roda em paralelo no Raspberry PI é responsável pela comunicação via RF com o módulo de acionamento do condicionador de ar. Maiores detalhes dessa comunicação e do funcionamento do módulo de acionamento são reservados à seção 5.3, por se tratarem do foco de implementação do presente trabalho.

Para fins de estudo de caso, elegeram-se dois modelos de condicionador de ar: um Electrolux Split Hi-Wall 9000 BTU e um Komeco Portátil Ambient ABP 09QC 9000 BTU, cujas imagens estão mostradas na Figura 20.

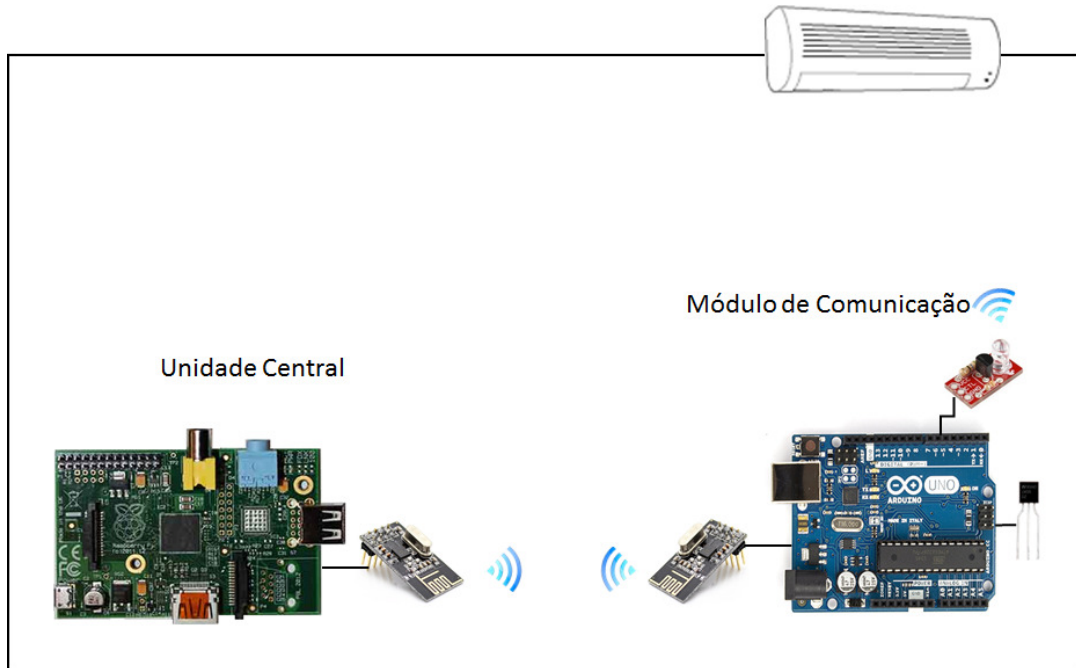
Figura 20. Dois modelos de condicionador de ar estudados: (a) Komeco Portátil Ambient ABP 09QC 9000 BTU, (b) Electrolux Split Hi-Wall 9000 BTU



5.3 Arquitetura de operação dos módulos central e de acionamento

As tarefas desenvolvidas e contempladas mais diretamente neste trabalho são, como já mencionado, o desenvolvimento, tanto em *hardware* quanto em *software*, do funcionamento do módulo de acionamento do condicionador de ar, do controle de temperatura, e da rotina de comunicação entre esse módulo e o módulo central. A arquitetura de comunicação entre os componentes de *hardware* envolvidos estão esquematizados na Figura 21.

Figura 21. Arquitetura de operação do módulo de comunicação e do módulo central.



A sequência lógica de operações executadas está mostrada no Anexo B e começa pela rotina de gerenciamento contida no Raspberry PI, desenvolvida em linguagem C/C++. Inicialmente, lê-se um arquivo de texto, nomeado *dbACcontroller.txt*, o qual contém as preferências do usuário. Esse arquivo define o modo de operação do módulo de comunicação do A/C (modo inteligente ou modo manual), assim como a condição de operação para o A/C em caso de ter-se escolhido o modo manual – ligado/desligado, frio/quente/seco, nível de ventilação e temperatura. As preferências do usuário são interpretadas por uma interface gráfica feita em JQuery e convertidas nesse arquivo de texto por uma rotina desenvolvida pela equipe da FURG. Vale mencionar que o uso de arquivos de texto como meio de ligação entre as rotinas referentes ao aparelho de ar condicionado e aquelas referentes ao armazenamento do consumo elétrico com armazenamento em banco de dados foi a solução encontrada para a impossibilidade de se realizar o acesso ao banco de dados do tipo MongoDB via linguagem C/C++.

A cada ciclo, então, as preferências do usuário são lidas e, em seguida, em havendo variação, essas informações são passadas para o Arduino via RF. Em não havendo variação, o Raspberry PI envia requisições de leitura da situação do condicionador de ar bem como da leitura de temperatura pelo sensor. Aguardam-se 5s para que a resposta correspondente seja retornada.

Toda a comunicação via RF ocorre em modo envio de informação com requisição de resposta. Desse modo, no ciclo em que é enviado um comando de alteração de modo de ope-

ração do condicionador de ar, o Raspberry também recebe uma resposta por parte do Arduino. Nesse formato de comunicação, o tempo de ciclo daquele fica condicionado ao tempo de ciclo deste, que foi ajustado com atrasos para adquirir o valor de aproximadamente 1s. Além disso, com o uso de um tempo de aguardo, no caso de 5s, evita-se que o sistema pare caso ocorra alguma perda de comunicação eventual. Na verdade, caso isso ocorra, a consequência maior para o sistema é o fato de que as variáveis que dependem da resposta do Arduino não são atualizadas naquele ciclo de programação.

O fluxo de tomada de decisão para o Arduino se encontra no Anexo B. Inicialmente, lê-se a temperatura indicada pelo sensor LM35. Em seguida, verifica-se se há alguma mensagem proveniente do Raspberry Pi. Caso haja mensagem, ela é interpretada e classificada conforme um dos 4 casos: operação em modo manual, operação em modo automático, requisição de temperatura ou requisição do status atual do A/C. Em seguida, uma resposta é enviada ao Raspberry Pi e são realizadas as alterações necessárias conforme o tipo de mensagem interpretada. A seguir, serão detalhados os modos de operação manual e automático.

Caso a mensagem de modo manual seja transmitida para o Arduino, então este simplesmente acionará o condicionador de ar com as preferências desejadas. No arquivo de texto *dbACcontroller.txt*, constam as definições dos parâmetros de acionamento do condicionador de ar dados pela Tabela 3.

Parâmetro	Quantidade de níveis	Níveis	Formato para comunicação
Alimentação	2	Ligado/desligado	0, 1
Modo de funcionamento	2	Quente/Frio	1, 2
Ventilação	4	Automático, alto, médio, baixo	0, 1, 2, 3
Temperatura de referência	15	De 16 a 30	De 16 a 30
<i>Swing</i>	2	Ligado/desligado	0, 1

Tabela 3. Parâmetros de acionamento para o condicionador de ar.

A última coluna da Tabela 3 mostra o formato adotado para o envio de dados via RF. Esse é o formato em que as informações são apresentadas no arquivo de texto *dbACcontroller.txt* e são trocadas entre Arduino e Raspberry PI. A escolha desse padrão simplifica a troca

de informações pelos dois sistemas microprocessados e foi acordada como padrão a ser seguido pelos outros programadores ao traduzir as opções escolhidas pelo usuário no *frontend*.

Já no caso de se ter escolhido o modo inteligente, então o Arduino deve executar sua rotina de controle de temperatura do ambiente, com base na leitura do sensor de temperatura conectado aos seus terminais. O Arduino também recebe uma informação da faixa de temperatura de conforto dentro da qual o A/C deve ficar desligado. Essa informação também se encontra no documento de texto *dbACcontroller.txt*, sob forma de dois parâmetros: temperatura central de conforto e tolerância. A rotina de ajuste de parâmetros do condicionador de ar para o protótipo apresenta o funcionamento dado pela Tabela 4, na qual a temperatura proveniente do sensor LM35, a temperatura central de conforto e a tolerância são denotadas, respectivamente, por T_a , T_C e ΔT_C .

Situação	Ação
$T_a < T_C - \Delta T_C$	Ligado; Quente; Alto; T_C ; Desligado
$T_C - \Delta T_C < T_a < T_C + \Delta T_C$	Desligado
$T_a > T_C + \Delta T_C$	Ligado; Frio; Alto; T_C ; Desligado

Tabela 4. Lógica de ajuste do condicionador de ar.

Então, conforme explicado, a rotina Arduino provê a rotina Raspberry PI com informações relativas à situação do condicionador de ar e à leitura de temperatura do ambiente. Uma vez recebidas pelo Raspberry PI, essas informações são armazenadas nos arquivos de texto denominados *dbACCurentStatus.txt* e *dbRoomTemperature.txt*, de modo a serem tratadas pelas outras rotinas existentes no projeto. Caso não seja retornado um valor no tempo estipulado (5s), então deixa-se de esperar e retoma-se a sequência de operações sem essa informação. Nesse caso, os documentos de texto mencionados não sofrem alteração durante aquele ciclo. Isso evita que a rotina Raspberry fique bloqueada enquanto aguarda resposta da rotina Arduino. No próximo ciclo, caso não haja variação dos parâmetros pelo usuário, torna-se a requerer as informações de situação do A/C e de temperatura do ambiente.

5.4 A ferramenta de análise de protocolos IR AnalysIR

Conforme dito na Seção 5.1, usou-se a ferramenta AnalysIR para a visualização e a interpretação dos sinais IR, devido à escassez de bibliotecas funcionais que compreendessem os principais protocolos de comunicação de controles remotos de A/C comerciais. A ferramenta

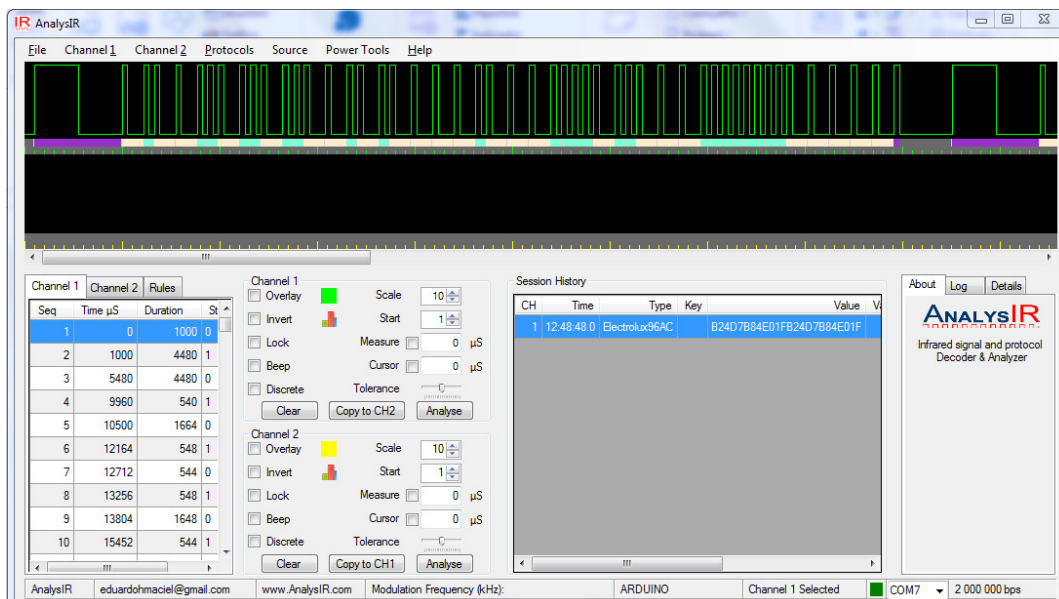
foi adquirida por meio de uma doação livre de US\$ 10,00. Nesta seção, será dada uma visão global deste programa, ficando a análise dos resultados reservada para o Capítulo 6.

O programa AnalysIR foi desenvolvido em Dublin, Irlanda, e é atualmente mantido por uma equipe de três programadores. A proposta do produto é facilitar a análise, a decodificação e a engenharia reversa de protocolos de controles remotos, tanto por leigos quanto por profissionais, tudo isso a um baixo custo.

Inicialmente desenvolvido para a plataforma de prototipagem Arduino, o produto é adquirido por meio de doações, as quais se revertem em integração com outras plataformas microprocessadas, como o Raspberry PI. Além de dever adquirir uma licença, é necessário que o usuário possua uma plataforma de prototipagem compatível devidamente configurada – no caso do projeto, um Arduino e um computador com a IDE Arduino – e um demodulador IR como o que foi apresentado na seção 4.1.6.3. Visto que o custo é definido praticamente pelo custo da plataforma de prototipagem, o uso do *software* AnalysIR possibilita que se estudem sinais IR sem a necessidade de equipamentos como analisadores lógicos ou osciloscópios, tipicamente de valor significativamente mais elevados.

Da obtenção da licença do produto, o fabricante fornece um algoritmo a ser programado no microprocessador do Arduino via IDE Arduino. Em se efetuando a operação com sucesso, deve-se, em seguida fechar a IDE e abrir o *software* AnalysIR. Estando o pino de sinal (pino 1) do sensor TSOP conectado ao pino 3 do Arduino, basta pressionar-se um botão do controle remoto a ser decodificado para que o sinal seja capturado e exibido em uma interface gráfica como a que está mostrada na Figura 22.

Figura 22. Interface gráfica do programa AnalysIR.



A interface gráfica mostra o sinal decodificado pelo sensor infravermelho e, em se tratando de um protocolo conhecido, exibe o protocolo utilizado, bem como a palavra que está sendo transmitida. A existência de dois canais de visualização permite a comparação entre dois sinais. Nessa janela, constam também todas as durações, em microssegundos, da forma de onda pulsada decodificada. Essa informação é exibida em forma de tabela na parte inferior esquerda da janela. Dentre ou demais recursos disponíveis, estão: copiar a imagem dos canais para a área de transferência do Windows, salvar os dados obtidos, visualizar o histórico e habilitar/desabilitar os protocolos a serem incluídos na análise.

Além disso, pode-se, também, copiar para a área de transferência do Windows em formato de texto: o nome do protocolo e a palavra em sistema hexadecimal e em sistema binário que se está transmitindo.

O foco do programa sendo, originalmente, os protocolos de controles remotos de televisores, existem poucos protocolos de A/C suportados. O protocolo exibido na Figura 22, por exemplo, apesar de ser o protocolo do A/C de base utilizado no presente projeto, não era originalmente suportado pelo *software* e foi, portanto, necessário incluí-lo na base de dados. A adição de um novo protocolo pode ser feita pelo próprio usuário, bastando-se alterar um arquivo de nome *AnalysIR.ini* devidamente, de modo a se incluírem as características de cabeçalho, de marca e de espaço, de número de bits da palavra.

Visto que a base de dados do programa reside na contribuição de usuários do programa, e que há suporte a vários tipos de equipamentos possíveis, é aconselhável que se desabilitem protocolos de controles remotos de televisão enquanto se trabalha com A/C e vice-versa, a fim de que não ocorram falsas identificações de protocolos.

5.5 Testes

A fim de se validar o trabalho realizado, foram realizados testes de verificação das funcionalidades implementadas. Os testes realizados foram divididos em duas categorias: teste de funcionalidade e teste de integração.

5.5.1 *Teste de funcionalidade*

Os testes foram conduzidos em dois ambientes diferentes: um apartamento com 50 m², onde estava instalado o A/C Electrolux, e um apartamento de 80 m², onde estava lo-

calizado o A/C Komeco. Em ambos os ambientes, o processador ATMEGA 328 foi programado usando-se a IDE Arduino via cabo USB e, em seguida, o kit foi desconectado do computador, passando a ser alimentado por uma fonte contínua 12V. Em cada ambiente, fizeram-se basicamente dois testes de acionamento: testes em modo manual, esperando-se verificar que as aparelhos fossem acionados no modo desejado, e um teste em modo automático, esperando-se que cada aparelho regulasse a temperatura do ambiente durante um período de 20h. O teste inteligente foi conduzido com uma taxa de transmissão de 250 kbps com atenuação de 0dBm e foi adotada a temperatura de conforto de 25°C com uma tolerância de 2°C para cima ou para baixo.

5.5.2 *Teste de integração*

A integração da programação desenvolvida durante o período de desenvolvimento do projeto foi concluída na semana do dia 3 de novembro de 2014, conforme apontado no cronograma da Tabela 7 do Anexo A. O objetivo da integração foi reunir, em um mesmo sistema, os códigos desenvolvidos pelas equipes da UFRGS e da FURG, visando-se a atender aos objetivos estipulados no primeiro *Workshop SapiEns* e durante as reuniões semanais realizadas desde então.

No funcionamento do Raspberry PI, foi implementado processamento paralelo a fim de que fossem executadas as rotinas servidor web, de interface gráfica, de aquisição de consumo elétrico e de comunicação com o módulo de acionamento do A/C.

As rotinas de servidor web são responsáveis por gerenciar a rede de comunicação, em conjunto o roteador D-Link, o qual gerencia os endereços IP. O acesso ao servidor web via navegador de internet permite a interação do usuário com o sistema por meio da interface gráfica implementada.

As rotinas de aquisição de consumo operam em conjunto com o serviço de banco de dados, MongoDB, realizando leitura e a inclusão no histórico os valores de consumo de cada tomada inteligente, uma vez a cada 1 ms. Após 1000 registros consecutivos no banco de dados, esses valores são substituídos pela sua média aritmética para se salvar espaço em disco.

A integração foi feita em Porto Alegre, no mesmo ambiente onde foi realizado o teste com o A/C da marca Komeco que foi descrito na seção 5.5.1. Os ajustes que levaram ao melhor desempenho do sistema e os resultados decorrentes disso se encontram na seção 6.3.

6 VALIDAÇÃO EXPERIMENTAL

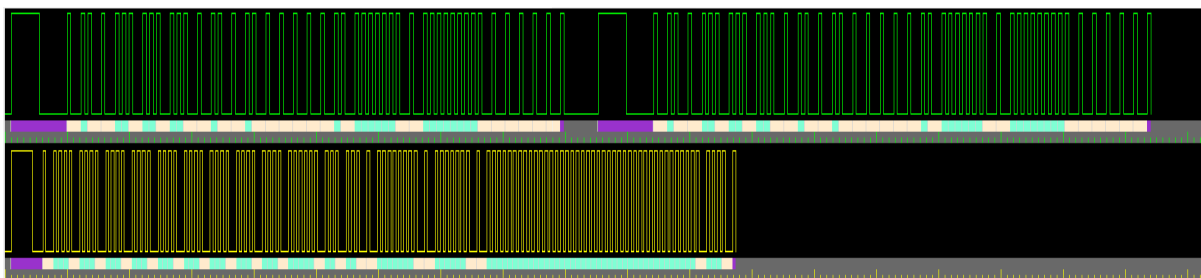
Este capítulo é dedicado à apresentação e à discussão dos resultados obtidos a partir da metodologia desenvolvida neste trabalho.

6.1 Protocolos de acionamento IR

Conforme mencionado na seção 5.4, usou-se a ferramenta AnalysIR para realizar a aquisição dos sinais emitidos pelo controle remoto universal para cada aparelho de ar condicionado do estudo de caso. O objetivo foi interpretar a forma de envio de códigos IR pelo controle aos A/C e, assim, desenvolver um meio de armazenar os códigos hexadecimais de maneira compacta e uma função que reproduza o mesmo método de envio capaz de acionar o A/C. Uma primeira constatação observada foi que mais de um protocolo do controle remoto universal podem acionar um mesmo condicionador de ar, o que sugere que foram incluídas variações de um mesmo protocolo a fim de garantir o correto acionamento do equipamento para algum deles.

Como mencionado anteriormente, foram considerados dois tipos de protocolo de envio, um Electrolux e um Komeco. Apesar de para os dois protocolos enviarem-se sinais modulados em 38 kHz, como explicado na seção 4.2.1, o tamanho e as informações contidas nos códigos hexadecimais, bem como a forma de envio dos mesmos ao A/C são específicas a cada protocolo. A Figura 23 mostra a forma de onda das marcas e dos espaços obtidos com o software AnalysIR para os dois modelos de A/C estudados, para o acionamento em modo resfriamento, ventilação automática, 25 °C e swing desligado.

Figura 23. Código hexadecimal dos A/C Electrolux (parte superior) e Komeco (parte inferior), para o comando Ligado, frio, ventilação automática, 25°C, swing desligado.



Na Figura 23, as marcas correspondem aos momentos em que a onda se encontra em nível alto, e os espaços ocorrem quando a onda se encontra em nível baixo. Pode-se constatar

que o sinal IR enviado pelo controle remoto Electrolux tem uma extensão temporal maior que a do Komeco. O tempo total de envio do primeiro é de 184 ms; o do segundo, 117 ms. No entanto, para o segundo protocolo, o número de marcas e espaços é superior ao do primeiro. Há 100 pares marca-espaço para o protocolo Electrolux contra 114 para o Komeco.

A Tabela 5 mostra os códigos hexadecimais obtidos com a ferramenta AnalysisIR para modos frio e quente, ventilação automática, temperaturas de referência de 25, 26 e 27 °C e swing desligado para o protocolo do A/C Electrolux.

Electrolux		
	Frio	Quente
25°C	B24DBF40C03FB24DBF40C03F	B24D9F60CC33B24D9F60CC33
26°C	B24DBF40D02FB24DBF40D02F	B24D9F60DC23B24D9F60DC23
27°C	B24DBF40906FB24DBF40906F	B24D9F609C63B24D9F609C63

Tabela 5. Códigos hexadecimais Electrolux para acionamento em ventilação automática com swing desligado.

A Tabela 6 mostra os códigos hexadecimais obtidos com a ferramenta AnalysisIR para modos frio e quente, ventilação automática, temperaturas de referência de 25, 26 e 27 °C e swing desligado para o A/C Komeco.

Komeco		
	Frio	Quente
25°C	888888888824C060600000000011	8888888888248060400000000041
26°C	888888888824C0A06000000000E1	88888888882480A0400000000081
27°C	888888888824C060600000000011	8888888888248020400000000001

Tabela 6. Códigos hexadecimais Komeco para acionamento em ventilação automática com swing desligado.

No protocolo Electrolux, são enviados 24 caracteres hexadecimais, ou 96 bits de informação. Pode-se perceber da análise dos hexadecimais que todos eles são a composição de dois hexadecimais de 12 caracteres iguais, os quais são enviados duas vezes (pode-se visualizar na forma de onda da Figura 23). Assim, o método de envio do protocolo Electrolux é da forma H-C-H-C, em que H denota o início da comunicação, dado por duas sequências espaço/marca dadas pelos tempos 1000µs/4488µs, 4488µs/544µs, e C denota a sequência de bits de informação. Para este protocolo, os bits 0 e 1 são codificados pelas sequências de espa-

ço/marca $548\mu\text{s}/548\mu\text{s}$ e $1660\mu\text{s}/548\mu\text{s}$, respectivamente. Juntando-se os 4 tempos de cabeçalho aos $12\text{caracteres} \cdot 4\text{bits/caractere} \cdot 2\text{tempos/bit}$ tempos, obtém-se 100 tempos de marca e de espaço. Multiplicando-se esse valor por 2 (sinal enviado duas vezes), obtém-se o número total de tempos armazenados pelo *buffer* do programa.

Além da repetição do envio, notou-se que os 4 primeiros caracteres (B24D) permanecem os mesmos para todos os casos de acionamento do A/C. Assim, comprimindo-se a informação pela metade, e transformando-se esses 4 caracteres em um cabeçalho, pode-se dizer que a informação está contida em 8 caracteres, em vez de 24. Essas duas operações foram efetuadas, de modo a reduzir em 66,67% o tamanho de armazenamento no Arduino. Além disso, constata-se que um mesmo código hexadecimal é enviado quando se deseja desligar o A/C e um outro quando se deseja mudar o modo do swing. Isso reduz o espaço de armazenamento para essas tarefas. No entanto, para se acionar o swing, deve-se inicialmente, acionar o ar com os demais parâmetros desejados (modo, ventilação e temperatura) e, em seguida, enviar o sinal IR correspondente ao swing (para o usuário, isso representa ouvir 2 bipes sonoros).

No protocolo Komeco, são enviados 28 caracteres hexadecimais, ou 112 bits de informação. Nesse protocolo, o sinal é enviado apenas uma vez, diferentemente ao do Electrolux. Assim como no caso anterior, há duas sequências marca/espaço que marcam o início de comunicação, dadas pelos tempos $1000\mu\text{s}/3432\mu\text{s}$ e $1660\mu\text{s}/440\mu\text{s}$. Após esses tempos, o código hexadecimal é enviado. Os bits 0 e 1 são gerados, respectivamente, pelas sequências marca/espaço $400\mu\text{s}/448\mu\text{s}$ e $1240\mu\text{s}/440\mu\text{s}$. Somando-se os 4 intervalos de tempo com os $28\text{caracteres} \cdot 4\text{bits/caractere} \cdot 2\text{tempos/bit}$ intervalos de tempo, obtém-se o valor de 228, que é a quantidade total de tempos apresentada pelo *buffer* apontada pelo programa.

Dos 28 caracteres, os primeiros 12 são constantes para todos os modos de acionamento, os 5 seguintes representam informação, os 9 que sucedem são outra sequência que permanece constante e os últimos 2 fazem parte da informação. Assim, transformando-se os 21 caracteres em constantes na programação, comprimem-se o espaço de armazenamento da informação em 75,00%. Cabe dizer que, diferentemente do protocolo Electrolux, existe um código hexadecimal de swing para cada combinação dos demais parâmetros de acionamento do A/C, isso significa que ele pode ser ligado ou trocar de estado diretamente com a função swing ligada ou não. Contudo, isso requer o dobro de espaço em memória para o armazenamento. Foi constatado, também, que existe um hexadecimal para se desligar o aparelho para cada combinação de parâmetros, inclusive o swing. Embora isso representaria mais uma duplicação de espaço em memória, testes constataram que para uma mesma combinação de modo de operação com opção de swing, o mesmo código hexadecimal desliga o A/C, indepen-

dentemente dos demais parâmetros. Assim, registraram-se 4 códigos hexadecimais (2 modos de acionamento considerados e 2 modos de swing). Uma última observação diz respeito ao fato de que o penúltimo caractere da primeira parte constante passa de 2 para 0 quando se trata de desligar o A/C.

6.2 Teste de funcionalidade

Esta seção traz as principais observações e conclusões decorrentes do teste de funcionalidade, descrito na seção 5.5.1.

Para ambas as marcas de aparelho consideradas, foi possível realizar o correto acionamento em qualquer combinação de parâmetros – modo (quente/frio), ventilação (automático, mínimo, médio, máximo), temperatura (de 17°C a 30°C) e *swing* (ligado/desligado). Deve-se comentar foram capturados códigos hexadecimais para o A/C Komeco a partir do controle remoto universal da Figura 18 referentes ao nível de ventilação automático, porém este modo não é suportado pelo aparelho. Nesse modo, o aparelho não era acionado.

A posição do led IR influencia no correto acionamento do A/C. Posicionando-se o led de modo a apontar para o receptor do A/C, pode-se realizar o acionamento com uma distância de até 5 metros. Quanto mais distante o led IR do A/C, mais sensível à orientação se torna o sistema. Testando-se o acionamento apenas com o led IR (sem o circuito transistorizado da Figura 14), o A/C não foi acionado.

Na IDE Arduino 1.0.5+dfsg2-2, o compilador faz com que os tempos de subida e de descida do sinal enviado ao led IR sejam de $8\mu\text{s}$, ao passo que na versão 1.0.6 esses tempos passaram para $5\mu\text{s}$. Logo foi necessário corrigir os tempos de atraso após uma subida ou uma descida de sinal IR, de modo que o período da onda quadrada seja de $26\mu\text{s}$ (frequência de 38kHz).

Em modo manual, alterando-se entre os três níveis de taxa de transferência de dados do RF dados pela Tabela 1, não foi notada diferença perceptível no desempenho da comunicação. Não foram notadas perdas de comunicação em nenhum dos casos. No caso do ambiente em que estava instalado o condicionador de ar da marca Komeco, este se encontrava em um cômodo diferente daquele em que se localizava o módulo principal. Mesmo com as portas fechadas, a comunicação foi efetuada sem problemas aparentes.

Alternando-se entre os modos de atenuação apresentados na Tabela 2, não houve perdas de comunicação para o acionamento em modo manual.

No teste em modo inteligente, em ambos os ambientes testados, avaliou-se que a influência da posição do sensor de temperatura influencia os resultados obtidos. Caso o sensor esteja posicionado na região do jato de ar do condicionador de ar, o controle de temperatura do ambiente fica prejudicado. Quando o A/C é acionado em modo de resfriamento a 25°C, a temperatura do ar expelido pelo aparelho fica entre 21°C e 22°C, conforme indicado pelo sensor. Passados os 5 minutos de tempo de regulação térmica, o sistema volta a ler a temperatura do ambiente e interpreta que o ambiente necessita ser aquecido. Assim, o A/C é acionado em modo aquecimento, e o efeito contrário ocorre. Já deixando-se o sensor fora da linha de ação do A/C, esse fenômeno não ocorre. Quando o sensor não está posicionado no caminho tomado pelo jato de ar, a regulação de temperatura do ambiente é atingida em cerca de 20 min. Assim, para que a manutenção da temperatura do ambiente seja efetiva, deve-se posicionar o sensor de modo a que sua leitura não seja influenciada diretamente pelo jato de ar do A/C.

Constatou-se que a existência de “mau contato” entre os terminais do sensor de temperatura LM35 e os do Arduino, causado pelo uso de fios e de conexão, provoca leituras de temperatura elevadas, podendo ser superiores a 100°C.

6.3 Teste de integração

Como descrito na seção 5.5.2, houve uma fase de integração de hardware e de software, envolvendo os códigos implementados pelas equipes da UFRGS e da FURG. O protótipo integrado reuniu as seguintes funcionalidades:

- Interface visual contendo tela de identificação com senha;
- Interface visual que traz informações da temperatura do ambiente monitorado, temperatura da cidade local e as opções de controle dos plugues e do A/C;
- Análise e histórico do consumo, salvo em banco de dados;
- Modos manual e inteligente de controle do conforto térmico.

A integração se mostrou funcional, porém com o surgimento inicial de falhas de operação e de lentidão do sistema. Efetuaram-se, assim, alguns ajustes para melhorar o desempenho do sistema. Os próximos parágrafos dispõem os pontos principais observados.

A taxa de dados da comunicação RF que melhor se adequou aos testes foi a de 250 kbps. Neste modo, o número de perdas de comunicação foi menor. O nível de sinal RF que apresentou melhor desempenho foi o nível HIGH (0dB de atenuação), vide Tabela 2. A dis-

tância a partir da qual a comunicação RF começa a falhar é de cerca de 10m, levando-se em conta as portas fechadas e as paredes.

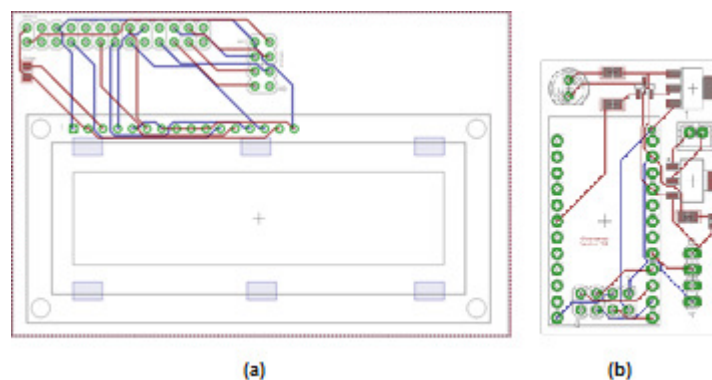
Em operação contínua, notou-se que o Raspberry PI perde a comunicação com o Arduino cerca de 1 vez a cada 20 min. Como o primeiro é programado para continuar normalmente caso ele não receba nenhuma resposta durante 5s, o sistema logo recupera o sincronismo e volta a operar normalmente. Em outras situações mais críticas, o Raspberry PI fica sobrecarregado, fazendo com que os acionamentos ao A/C e às tomadas sofra um atraso de dezenas de segundos. Também ocorre de o sistema se desconectar da rede, tornando-se necessária a reconexão.

Buscando-se melhorar os tempos de identificação na rede e de execução das tarefas, trocou-se o cartão SD inicialmente utilizado (apesar de não identificado, estima-se que seja de classe 4) por um cartão de classe 10. O desempenho geral do sistema melhorou. Atribui-se essa melhora à maior capacidade de leitura/escrita no cartão.

6.4 Placas de circuito impresso

Como dito na seção 4.1.6.1, o interesse do uso da placa Arduino é a possibilidade de reprodução do *hardware* em placas dedicadas. Com este fim, a equipe da UFRGS desenvolveu, em ambiente Eagle, uma placa para o módulo de acionamento e outra para ser acoplada ao módulo central, esta última recebendo o acréscimo de um display o qual exibe um número de identificação ao usuário. A Figura 24 mostra o leiaute proposto para as referidas placas.

Figura 24. Placas de circuito impresso para (a) Raspberry PI e (b) Arduino.



As dimensões físicas das placas ficaram em 8,7 cm x 5,6 cm para a placa a ser acoplada ao Raspberry PI e em 4,4 cm x 2,6 cm para a placa que substitui o Arduino.

7 CONCLUSÕES

Oferecer às residências brasileiras um meio simples de se controlarem as despesas com energia elétrica é possível e apresenta interesse econômico tanto ao mercado consumidor de energia quanto às concessionárias de energia. A evolução dos sistemas microprocessados permite que se implementem soluções integráveis, escaláveis e dedicadas às aplicações de automação residencial.

Com este trabalho, foi possível desenvolver um módulo cujas funcionalidades corresponderam aos objetivos traçados para o protótipo e cujo funcionamento se integrou aos demais blocos do sistema. O entendimento das especificações do projeto e do funcionamento global do sistema durante todas as etapas de desenvolvimento foi fundamental para que a implementação do apresentasse o funcionamento adequado e o retrabalho fosse mínimo.

O uso da ferramenta AnalysIR e de seu suporte permitiu uma melhor compreensão tanto da forma de envio dos protocolos IR aos aparelhos de ar condicionado quanto do mercado desses equipamentos. Para o correto envio dos códigos hexadecimais ao A/C, é importante que a onda modulada em 38 kHz tenha precisão de μs . Do contrário, o aparelho não reconhecerá o sinal transmitido. A compactação dos códigos hexadecimais é um passo necessário a ser tomado para que um maior número de marcas de A/C possam ser incorporados em uma mesma quantidade de memória. Nesse sentido, a gestão de memória e o conhecimento dos recursos do *hardware* podem evitar sua troca apressada, o que poderia causar um aumento do custo do projeto.

Notou-se uma acentuada diferença entre fabricantes no que diz respeito ao modo de envio dos códigos hexadecimais e na própria gestão desses códigos pelo aparelho de ar condicionado, como apontado na questão do swing, do desligamento e dos níveis de ventilação.

A comunicação RF entre Raspberry Pi e Arduino, sob forma de requisição de informação com envio de resposta, permitiu o sincronismo entre as duas plataformas e, ao mesmo tempo, a independência de operação de cada uma caso a outra unidade se encontre desligada.

Adicionalmente, este trabalho serviu de base para a implementação das ideias do projeto residencial SapiEns e poderá auxiliar na implementação de trabalhos futuros desse ou de outros projetos relacionados.

8 PROPOSTA PARA TRABALHOS FUTUROS

Nesta seção, são sugeridos próximos passos a serem tomados de modo a melhorar o desempenho do protótipo residencial SapiEns, acrescentar-lhe funcionalidades e aproximá-lo de um produto que possa ser adquirido pelos consumidores.

No quesito conforto térmico, deve-se pesquisar sobre a influência da temperatura externa, da época do ano e da umidade relativa na percepção do conforto térmico pelo ser humano. Além disso, deveriam ser estudados métodos de aprendizado que possibilite ao sistema aprender quais as principais preferências do usuário. Tudo isso seria incorporado ao modo inteligente de controle do condicionador de ar, de sorte a tornar esse modo realmente inteligente, a ponto de tomar decisões pelo próprio usuário. Paralelamente, deveriam ser incorporados códigos hexadecimais referentes a outros modelos e marcas de condicionadores de ar. A estruturação das classes e das bibliotecas dentro da programação do módulo de acionamento permite que as informações de cada novo modelo de A/C adicionado fiquem em uma biblioteca própria, sem que o código principal do Arduino sofra alterações. Contudo, todos esses acréscimos ao módulo de acionamento requerem um estudo de soluções em aumento de processamento e de memória. Caso contrário, a placa de prototipagem Arduino UNO logo chegará à sua saturação.

Em se contemplando a influência da umidade na percepção de conforto térmico, deve-se optar por um sensor de temperatura e umidade em substituição ao sensor LM35. Sensores combinados, contudo, tendem a ter um custo um pouco mais elevados e são limitados pela frequência máxima de leitura de temperatura e umidade (por exemplo, existem sensores em que a leitura só pode ser feita a cada 2s no mínimo). Deve ser discutida, também, a ideia de fusão de sensores dentro de um mesmo ambiente, o que aumentaria as possibilidades de controle de temperatura, uma vez que a distribuição do calor no ambiente poderia ser avaliada. Também seria possível identificar se algum dos sensores está mal posicionado, devendo ser realocado.

Outro passo a ser tomado é a avaliação do consumo do módulo de acionamento, a fim de que seja abordada a questão da alimentação por baterias ou pilhas. Tratando-se de um produto residencial, devem-se escolher pilhas cujo padrão seja facilmente encontrado pelo cliente e que possam manter o sistema em funcionamento por um período de tempo a ser discutido com a equipe. Um ponto de partida seria uma durabilidade de 1 ano.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA CAVALCANTE, Marisa et al. Controle Remoto: Princípio De Funcionamento (parte 1 de 2). **Caderno Brasileiro de Ensino de Física**, v. 30, n. 3, 2013.

BERGMANS, San. **SB-Projects**. 2001. Disponível em: <<http://www.sbprojects.com/knowledge/ir/index.php>>. Acesso em: 01 nov. 2014.

BOLZANI, Caio. Desmistificando a domótica. **Escola Politécnica da Universidade de São Paulo**, 2007.

CHASE, Otavio. Sistemas Ciberfísicos.

DO BRASIL, Atlas de Energia Elétrica. Energia no Brasil e no Mundo. **Aneel, Informativo**, 2011.

EMPRESA, DE PESQUISA ENERGÉTICA. Resenha Mensal do Mercado de Energia Elétrica: Dezembro. **Rio de Janeiro**, 2013.

EMPRESA, DE PESQUISA ENERGÉTICA. Resenha Mensal do Mercado de Energia Elétrica: Dezembro. **Rio de Janeiro**, 2014.

GLOBOSENSE. USB Spec for Globosense Energy Kit. **Cambridge, CB2 UF**. 2014.

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. Fundamentos de Física: Óptica e Física Moderna, vol. 4. **Editora: LTC**, 2007.

PORTAL PCH. **Ar condicionado, o novo vilão**. 2014. Disponível em: <<http://www.portalpch.com.br/index.php/noticias-e-opniao/noticias-pch-s/2437-22-01-2014-ar-condicionado-o-novo-vilao#.VASqEfl5OuM>>. Acesso em: 01 set. 2014.

SILVA, Sillas. **Clonando Qualquer Controle Remoto**. 2012. Disponível em: <<https://arduinolivres.wordpress.com/2012/07/31/clonando-qualquer-controle-remoto/>>. Acesso em: 13 jul. 2014.

SHIRRIFF, Ken. **A Multi-Protocol Infrared Remote Library for the Arduino**. 2009. Disponível em: <<http://www.righo.com/2009/08/multi-protocol-infrared-remote-library.html>>. Acesso em: 20 jun. 2014.

VALOCCHI, Michael et al. Plugging in the consumer: Innovating utility business models for the future. **IBM Institute for Business Value**, v. 114, 2007.

WAN, Kaiyu et al. Composition challenges and approaches for cyber physical systems. In: **Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on**. IEEE, 2010. p. 1-7. A

WAN, Kaiyu; MAN, K. L.; HUGHES, D. Specification, analyzing challenges and approaches for cyber-physical systems (CPS). **Engineering Letters**, v. 18, n. 3, p. 308, 2010. B

STANDARD, ASHRAE. 55, Thermal environmental conditions for human occupancy. **American Society of Heating, Refrigerating and Air conditioning Engineers**, 2010.

LEE, Edward A. CPS foundations. In: **Proceedings of the 47th Design Automation Conference**. ACM, 2010. p. 737-742.

ANEXO A. CRONOGRAMA DESENVOLVIDO

Tabela 7. Cronograma de etapas desenvolvidas ao longo do projeto e da implementação do protótipo

PERÍODO	TAREFAS
1º/06/2014	Implementação da bolsa CNPq.
1º/06/2014 - 28/07/2014	Pesquisa e revisão bibliográfica. Estudo de arquitetura e componentes básicos.
28/07/2014 - 07/07/2014	Primeiro <i>Workshop</i> SapiEns.
07/07/2014 - 16/07/2014	Programação do sensor de temperatura.
16/07/2014 - 23/07/2014	Sistema de controle básico de temperatura (simulação).
23/07/2014 - 30/07/2014	Pesquisa e desenvolvimento de sistema de acionamento para condicionador de ar com LED IR em Arduino.
31/07/2014 - 03/09/2014	Revisão bibliográfica sobre protocolos infravermelho dedicados a condicionadores de ar. Utilização do software AnalysIR. Organização do código em classes de modo a suportar o acréscimo de outros modelos de condicionador de ar.
04/09/2014 - 09/09/2014	Adaptação do código Arduino para suportar um segundo modelo de condicionador de ar. Testes de funcionamento do protótipo de controle de temperatura do ambiente.
10/09/2014 - 30/09/2014	Configuração de biblioteca RF para Arduino e para Raspberry PI. Definição e implementação da lógica de leitura e de envio de informações do Raspberry PI para o Arduino.
30/09/2014 - 07/10/2014	Definição da lógica de operação de cada dispositivo para a troca de informações e funcionamento autônomo.
08/10/2014 - 17/10/2014	Procura por soluções de gestão de memória em Arduino. Realização de testes. Atualização da IDE para 1.0.6. Correção de erros devido à atualização. Uso da função PROGMEM.
18/10/2014 - 25/10/2014	Documentação. Implementação da funcionalidade de escolha de faixa de temperatura de conforto no modo inteligente.
26/10/2014 - 06/11/2014	Integração dos códigos da equipe da UFRGS e da FURG. Realização de testes.
07/11/2014 - 08/11/2014	Preparação de apresentação para <i>Workshop</i> .
10/11/2014	Segundo <i>Workshop</i> SapiEns. Apresentação do projeto no auditório da FURG como parte da semana acadêmica desta universidade.

A seguir, consta uma lista com os detalhes das etapas desenvolvidas durante o desenvolvimento do protótipo residencial mencionado no trabalho.

- Pesquisa acerca de componentes existentes no mercado que poderiam a ser uma solução economicamente viável para o projeto: Em reunião com a equipe da UFRGS, montou-se uma apresentação na qual foram reunidas as principais soluções em tecnologia disponíveis para o projeto. Como critério de seleção para os possíveis dispositivos a serem utilizados, foram levados em conta: preço do dispositivo, capacidade de processamento (em havendo), documentação disponível, expansibilidade, facilidade de encomenda e tamanho físico. Na mesma apresentação, foram levantadas ideias a respeito do entendimento da equipe sobre os objetivos do projeto.
- Primeiro *workshop* do projeto Sapiens: reunindo a equipe do projeto, a equipe da UFRGS e a equipe da FURG no Parque Tecnológico Tecnopuc, o evento ocorreu no dia 7 de julho de 2014. Neste dia, as equipes foram apresentadas umas às outras e foram apresentadas as ideias pensadas e discutidas separadamente previamente à data do evento. Por meio da apresentação e da discussão dos temas propostos por cada equipe, definiram-se as metas a serem atingidas em um primeiro momento de modo a se obter um primeiro protótipo do sistema para a realização de ensaios. Nesse dia, também, definiram-se quais os componentes seriam empregados, quais as linguagens de programação seriam utilizadas e quais os enfoques a serem adotados por cada equipe. Assim, deu-se um prazo de 3 meses para a apresentação do primeiro protótipo SapiEns. Estipulou-se também, que fossem feitas reuniões semanais via Skype entre as equipes de desenvolvimento, a fim de se trocarem informações e ideias.
- Aquisição da placa Arduino e do sensor de temperatura LM35: programou-se a placa de prototipagem para a leitura da temperatura a cada 1s de tempo. Além da captura da temperatura, objetivou-se a ambientação com o ambiente e a estrutura de programação.
- Desenvolvimento de uma lógica básica de controle de temperatura: a partir da informação de temperatura fornecida pelo sensor, desenvolveu-se lógica básica de controle de temperatura, tendo como parâmetros um intervalo de conforto. O objetivo foi deixar disponível uma lógica de controle monovariável disponível para testes futuros e aprimoramentos.
- Primeiro estudo do acionamento do aparelho de ar condicionado: O emissor IR deveria ser responsável pela captura do sinal emitido pelo controle remoto, de modo a que este sinal fosse caracterizado e armazenado no Arduino. Já o circuito transistorizado do LED IR emissor deveria reproduzir o sinal armazenado e acionar, assim, o aparelho de ar condi-

onado. Por meio da leitura de normas e de fóruns dedicados a essa questão, algumas bibliotecas próprias à manipulação de sinais IR foram testadas, como a biblioteca de Shirriff e a de Itamaro, as quais propunham a captura da ação do controle remoto durante o momento em que uma tecla fosse pressionada. Embora nenhuma tenha se mostrado funcional para um condicionador de ar da marca Electrolux, a leitura de vários códigos permitiu uma melhor compreensão de como os sinais IR deveriam ser enviados e interpretados e de como isso deveria ser adaptado para aparelhos de ar condicionado, já que estas bibliotecas funcionam melhor para aparelhos de TV. Assim, a reunião de várias ideias permitiu que se chegasse a um código funcional para o acionamento do aparelho.

- Estudo sobre protocolos de acionamento infravermelho: Consultaram-se normas de protocolos IR e fóruns dedicados ao acionamento de dispositivos por IR, na busca de um repertório de códigos hexadecimais e formas de envio para os modelos de condicionadores de ar mais comuns, para alguma linguagem de programação como o c++, porém não se encontrou tal recurso. Assim, adquiriu-se a ferramenta AnalysIR de análise de sinais IR concebida para Arduino, para facilitar o estudo e o tratamento de sinais IR. Com base nas informações provenientes desta ferramenta, desenvolveram-se as rotinas de acionamento para os condicionadores de ar Eletrolux e Komeco. Pensando-se na questão de, futuramente, se incorporarem novos modelos de protocolos de condicionadores de ar, foi necessário reestruturar o código em bibliotecas de modo que cada novo modelo incorporado na base de dados não comprometesse o tamanho do código principal.
- Configuração do módulo de comunicação RF entre Arduino e Raspberry PI: inicialmente, configurou-se a biblioteca RF24 em cada componente do módulo até se acertarem todos os parâmetros de modo a estabelecer a comunicação básica. Em seguida, passou-se à elaboração de testes mais sofisticados de comunicação.
- Elaboração de rotinas de troca e de tratamento de informações entre Arduino e Raspberry PI: pensando-se na leitura das informações provenientes do banco de dados e do servidor *web*, desenvolveram-se rotinas de operação para cada um dos sistemas microprocessados. O objetivo foi tornar cada sistema autônomo e funcional, mesmo quando o outro é não fornece dados via RF. Assim, informações referentes ao modo de operação do condicionador de ar (manual ou inteligente) escolhido pelo usuário, à temperatura do ambiente proveniente do sensor de temperatura e à temperatura da cidade foram transmitidas e tratadas conforme necessário.

- Adaptação do código arduino para comportar os códigos hexadecimais dos protocolos IR: após se averiguarem erros de funcionamento conforme se registravam novas palavras hexadecimais à memória do arduino, concluiu-se que o problema ocorria devido à falta de memória RAM (total 2kb). Utilizando-se a biblioteca MemoryFree.h, foi possível confirmar essa hipótese e o método de armazenamento dos códigos hexadecimais teve de ser alterado. Esses foram passados para a memória de programa (32kb) com o auxílio do comando PROGMEM. A atualização da IDE para 1.0.6 foi necessária para se realizar esta tarefa.
- Inclusão da funcionalidade de escolha de faixa de temperatura de conforto: Foi acrescentada a funcionalidade segundo a qual o usuário define a faixa de temperatura de conforto e esta é transmitida ao Arduino via RF.
- Realização de testes e integração de códigos: foram realizados testes de funcionalidade com o código implementado, como alteração da taxa de transmissão e da potência do sinal, aumento do tempo de espera por resposta no Raspberry PI. Também foi realizada a integração dos códigos desenvolvidos pela equipe da UFRGS e da FURG, chegando-se a um sistema dotado de interface com o usuário, com tela de cadastro e de identificação, com monitoramento da temperatura do ambiente e da cidade do cliente, com o consumo das tomadas inteligentes e com o controle do condicionador de ar.
- Apresentação do protótipo na semana acadêmica da FURG: As equipes da UFRGS, da FURG, e da empresa Globosense se reuniram no campus da FURG, em Rio Grande, para apresentar o protótipo de demonstração durante o evento da semana acadêmica. Aproveitou-se para discutir o cumprimento dos objetivos definidos no primeiro *workshop* e para definir as próximas etapas do projeto.

ANEXO B. FLUXOGRAMA DE OPERAÇÃO PARA RASPBERRY PI E ARDUINO

Figura 25. Fluxograma de operação para o Raspberry Pi.

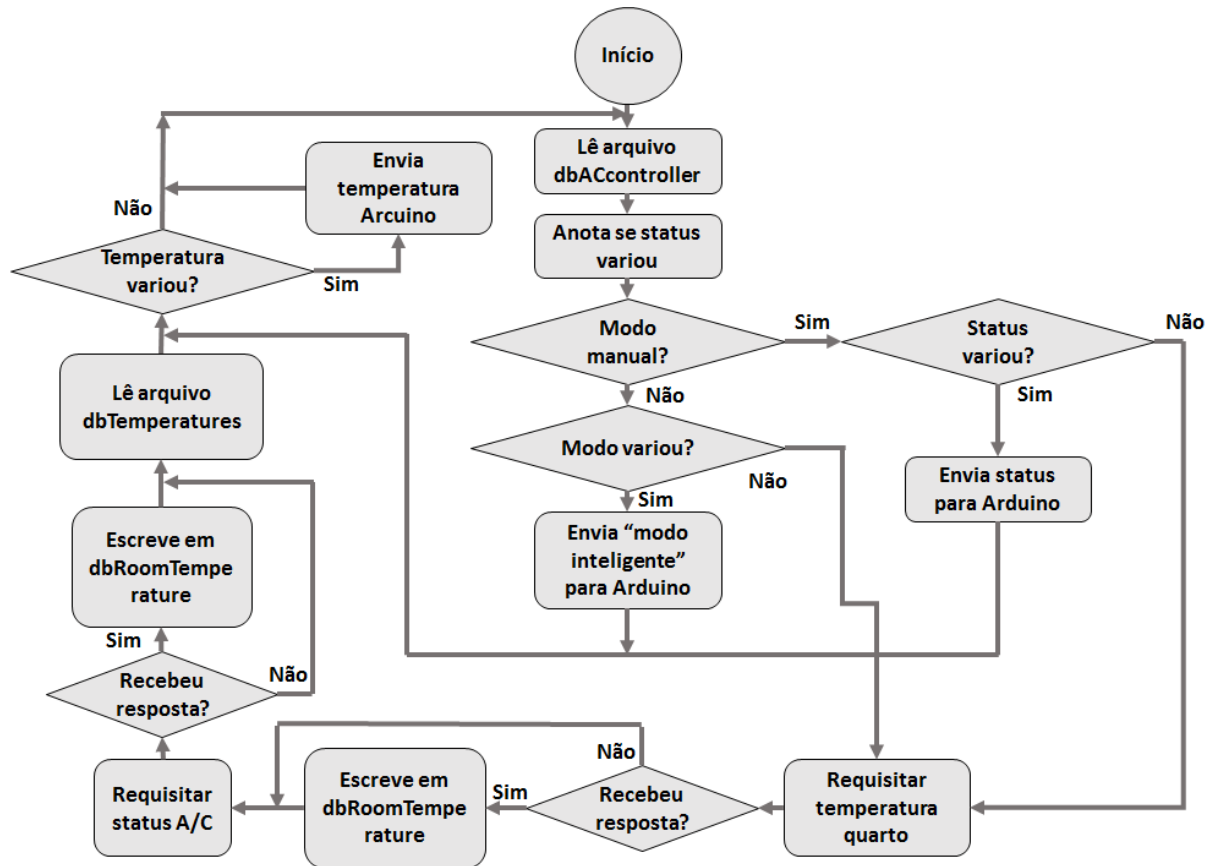


Figura 26. Fluxograma de operação para o Arduino.

