

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPTO. DE ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO

SEL0630 - Aplicação de Microprocessadores II

**Controle de ar-condicionado por um
aplicativo de Smartphone.**

Autor: Guilherme José Acra, nº. USP 7150306

Autor: Leonardo Guarnieri de Bastiani, nº. USP 8910434

Orientador: Prof. Dr. Evandro Luis Linhari Rodrigues

São Carlos

2017

Resumo

Este projeto tratou da criação de um sistema que automatiza o funcionamento de um ar-condicionado com IoT. Uma Raspberry Pi ligada a um LED infravermelho foi utilizada para emitir sinais semelhantes a de um controle remoto de ar-condicionado. A implementação do sistema foi por uma aplicação REST e uma interface gráfica desenvolvida para smartphones Android. As aplicações desenvolvidas funcionam e o sistema pode ser usado na prática, com as funcionalidades de ligar e desligar e definir temperatura do ar-condicionado, além de programar uma ação de acordo com o dia da semana e medir a temperatura ambiente por um componente chamado SHT1x com o protocolo de I²C.

Palavras-Chave: raspberry pi, infravermelho, internet das coisas, ar-condicionado, automatização, REST, Android, SHT1x, I²C.

Abstract

This project implements a system that automates the operation of an air conditioner with IoT. A Raspberry Pi connected to an infrared LED was used to emit the same signals as a remote control air conditioner. The implementation of the system was by a REST application and a graphical interface developed for Android smartphones. The developed applications worked and the system can be used in practice, with the functions of turning on and off, setting temperature, scheduling an action according to the day of the week and measuring the ambient temperature by a component called SHT1x with the I²C protocol.

Keywords: raspberry pi, infrared, IoT, air conditioner, automation, REST, Android, SHT1x, I²C.

1. Introdução

1.1 Motivação

Os ares-condicionados são equipamentos que possuem um alto consumo de energia e o racionamento de seu uso traz benefícios para o proprietário e para a comunidade de maneira em geral, além dessa visão, este projeto trata de uma automatização com IoT, ou seja, foi desenvolvida uma aplicação REST que de modo geral são aplicações que possuem uma integração muito fácil e traz uma comodidade para quem o utiliza, permitindo que seu proprietário controle seu ar-condicionado com seu próprio smartphone Android pelo aplicativo ou outro dispositivo com navegador Web.

1.2 Objetivos

Por se tratar de um projeto didático, muitos objetivos não possuem fins práticos, mas foram de grande importância para os desenvolvedores nas trajetórias profissionais. Os objetivos estão listados a seguir:

- Entender como funciona a comunicação entre o controle remoto e o ar-condicionado.
 - Aprender conceitos e estudar a fundo sobre sistemas embarcados.
 - Aprender sobre as aplicações de infravermelho em sistemas embarcados
 - Conseguir executar na prática a integração de um sistema embarcado com outros periféricos.
 - Controlar o ar-condicionado do laboratório de ensino pela internet.
 - Criar um sistema de automação de fácil uso e modularizado.
 - Entender o framework Flask e aplicá-lo.
 - Aprender a linguagem de programação Python e como aplicá-la em um sistema embarcado.
 - Entender o sistema embarcado NodeMCU e como aplicá-lo.
 - Aprender a comunicação I²C e como aplicá-la.
 - Aprender o protocolo RC5 presente nas comunicações de infravermelho.
 - Criar um aplicativo para o sistema Android.
- Todos esses objetivos foram cumpridos no final da disciplina.

1.3 Justificativas

A automação residencial, vem se tornando cada vez mais importante. Também conhecida como Domótica, pode ser definida como o uso de tecnologias e serviços para automatizar a execução de tarefas domésticas corriqueiras. O objetivo de seu uso é o aumento da segurança, comodidade, praticidade e uso racional de energia elétrica [1, 2].

Diversos elementos são usados em uma automação residencial, entre os quais, sensores, controladores, atuadores, barramentos e interfaces. De maneira geral, os sensores são responsáveis por detectar e/ou mensurar grandezas físicas ou eventos; os atuadores ativam e/ou controlam equipamentos automatizados; interfaces são o meio pelo qual o usuário interage com o sistema (visualizando dados e/ou tomando alguma ação); por fim, os controladores interligam os elementos anteriores por meio de barramentos (meio físico onde ocorre o transporte das informações). Os controladores monitoram as informações dos sensores, tomam decisões e enviam comandos aos atuadores. Eles ainda podem enviar dados às interfaces e receber comandos destas [1, 2, 3]. Para haver tal comunicação entre os elementos citados, é necessário a aplicação de protocolos, que nada mais são do que um acordo, regras, padrões entre as partes envolvidas de como se dará a comunicação [1, 4].

O projeto de automação de ar-condicionado via internet realizado neste trabalho foi escolhido por ser bastante representativo para se entender automação residencial, uma vez que inclui diversos dos conceitos anteriormente apresentados. O objetivo é a automatização de uma tarefa extremamente corriqueira (controlar um aparelho de ar-condicionado), entretanto há um potencial muito grande de aumento de comodidade, praticidade e economia de energia elétrica. O projeto inclui o uso de um sensor (de temperatura), de um atuador (LED infravermelho), de uma interface (aplicativo Android), de um controlador (Raspberry Pi), de três protocolos principais (I²C, RC5, HTTP, sendo que o primeiro possui um barramento).

1.4 Organização do trabalho

Inicialmente, será apresentado o **embasamento teórico** do projeto. Nesta parte será falado em detalhes sobre os elementos usados. São eles:

- O atuador, ou seja, o controle infravermelho e seu protocolo RC5, bem como a biblioteca que implementa tal protocolo (LIRC).
- O controlador (Raspberry Pi). Será ainda comentado sobre o controlador NodeMCU, estudado pela dupla, mas que infelizmente não houve tempo hábil para ser usado no projeto.
- O sensor de temperatura SHT1x e seu protocolo I²C, bem como do barramento usado.
- A arquitetura REST e o protocolo HTTP usado para comunicação pela internet.

Em seguida, serão detalhados os **materiais e métodos** usados no desenvolvimento do projeto, como eles se interligam e se complementam para atingir os objetivos esperados.

Em **resultados e discussões**, será analisado o desempenho do sistema. Os objetivos alcançados serão destacados, bem como os que não foram possíveis de se realizar. A adequação com os conceitos de automação residencial também será verificada.

Na **conclusão** será feito um apanhado geral do que foi alcançado no projeto. As tecnologias serão avaliadas no que se refere a aplicabilidade no domínio e facilidade de implementação.

Por fim, em **trabalhos futuros**, há uma avaliação das muitas necessidades de um projeto de automação residencial, que não foi possível alcançar durante o tempo disponível para o desenvolvimento.

2. Embasamento teórico

2.1 O infravermelho e seu protocolo de comunicação

O estudo do protocolo RC-5[10] presente em muitos sistema que possuem infravermelho foi aconselhado pelo professor para ser objeto de estudo, a seguir é apresentado algumas características do protocolo:

- Tipo de modulação: Código de Manchester
- Frequência da portadora: 36 KHz
- Apresenta dois bits de start.
- Um bit Toggle que muda ao soltar e pressionar novamente.
- Cinco bits de endereço.
- Seis bits de comando ou 7 no protocolo RC5-extended.
- O período de cada bit é de 1,778 microsegundos.
- O tempo de duração de um pacote é de 24.892 microsegundos.
- O pacote é repetido a cada 100 microsegundos.

A Fig. 1 apresenta uma ilustração gráfica do protocolo RC5 e nela é possível ver como as características apresentadas anteriormente se enquadram.

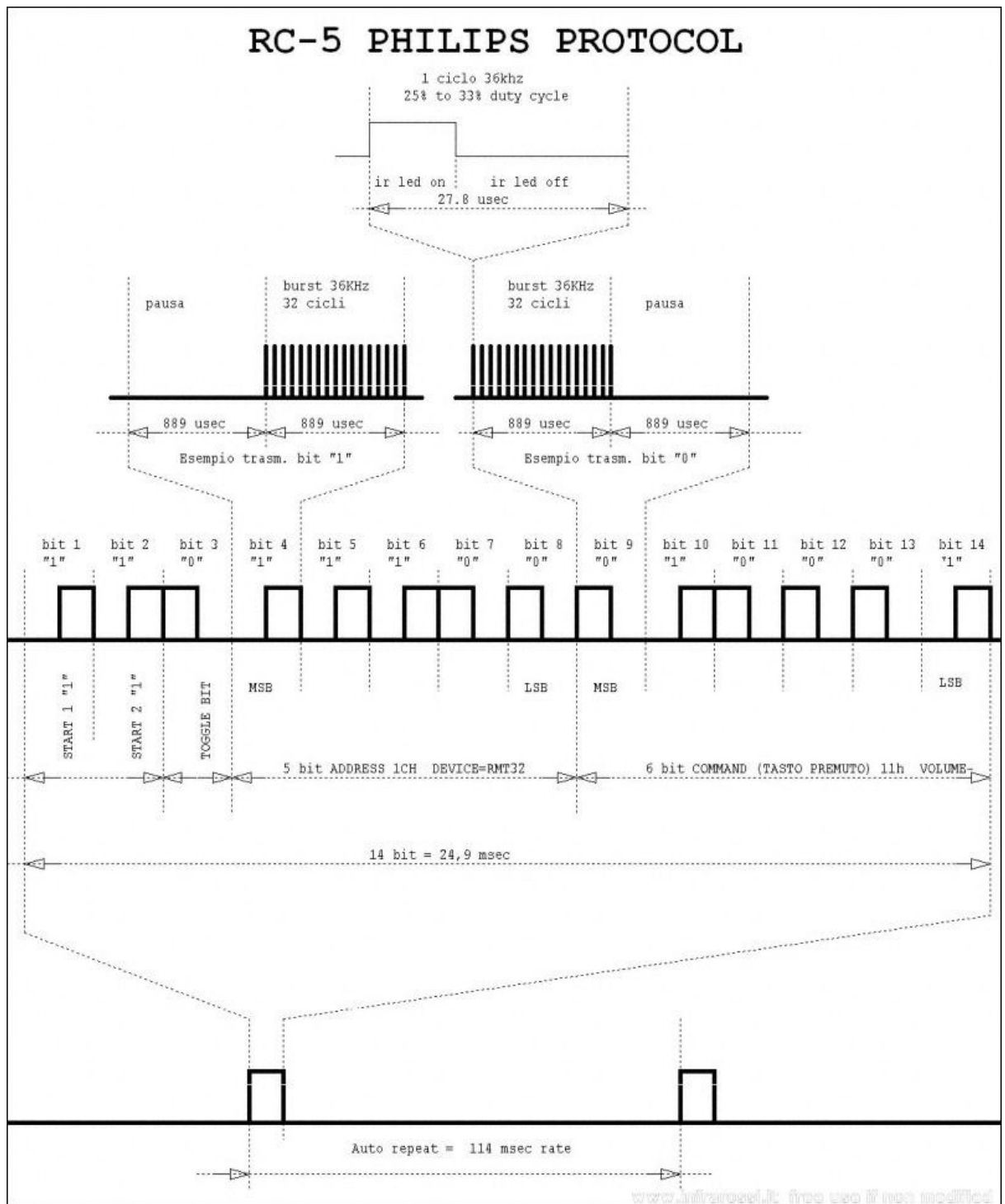


Figura 1. Especificação do protocolo RC-5.

Neste projeto, os conceitos do protocolo RC5 não foram necessários, pois a implementação do envio de sinais pelo LED infravermelho foi feita com o auxílio de uma biblioteca que será apresentada posteriormente, entretanto seu conceito foi estudado e entendido para acrescentar na formação profissional de seus desenvolvedores.

Um ar-condicionado da marca LG foi controlado neste projeto e com o auxílio de um sensor infravermelho e de um osciloscópio, foi possível analisar a forma de onda enviada

pelo controle remoto, as Fig. 2 e 3 apresentam sua forma. É possível ver que o sinal captado pelo sensor infravermelho difere do protocolo RC-5 apresentado anteriormente, podemos concluir que o controle de ar-condicionado da marca LG estudado implementa outro protocolo. Na Fig. 2 é possível notar que há um start bit e na Fig. 3 é possível ver os bits transmitidos pelo controle.

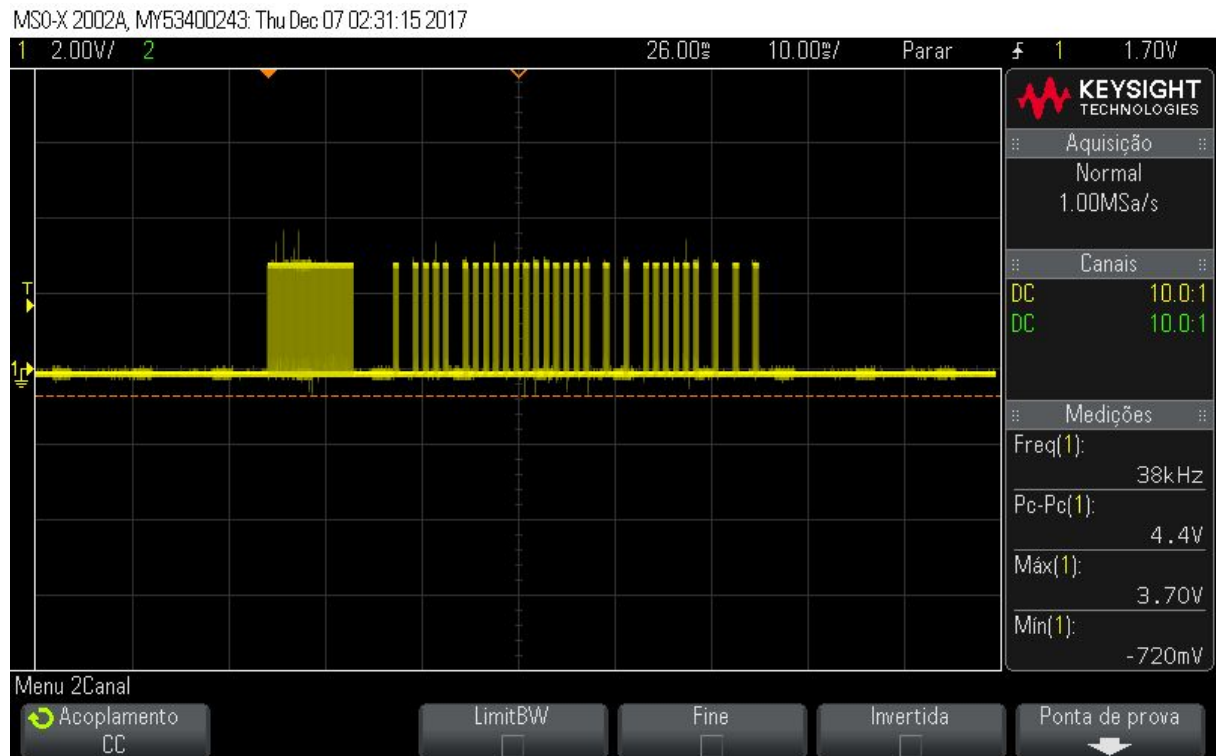


Figura 2. Foto do osciloscópio medindo o sinal de saída de um sensor infravermelho.

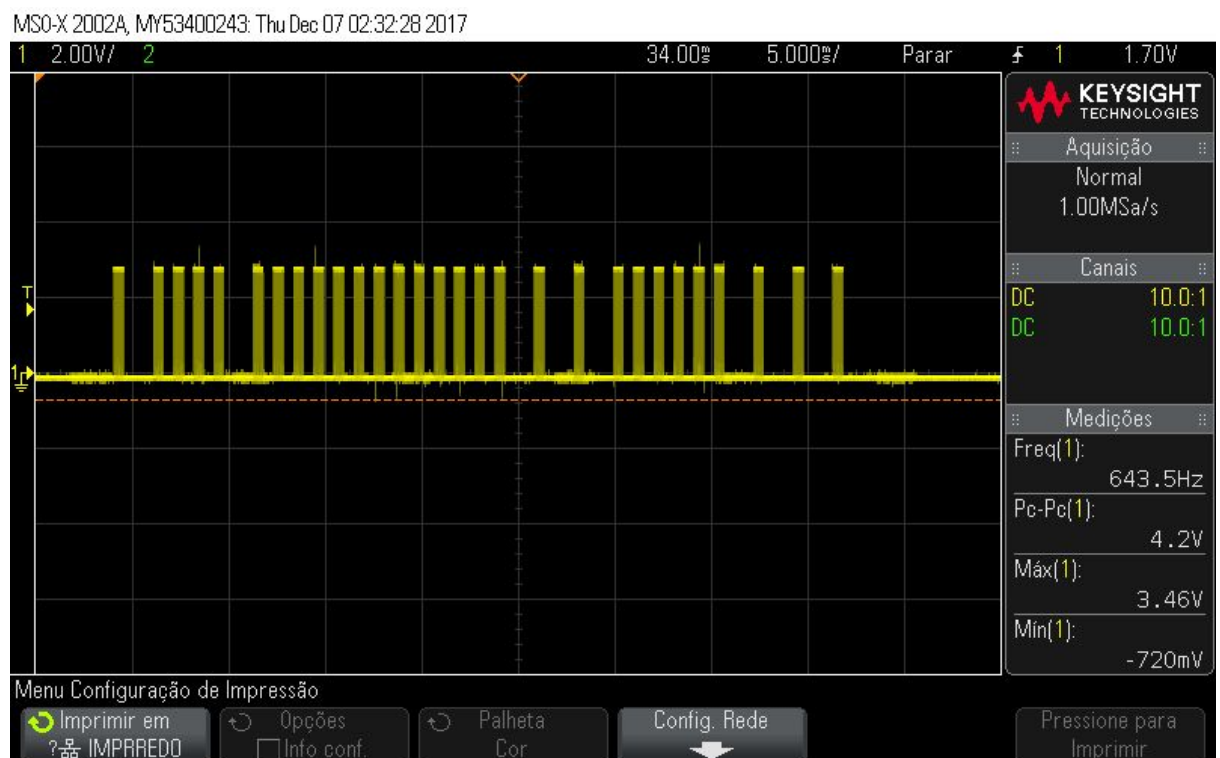


Figura 3. Zoom das informações apresentadas na Fig. 2.

2.2 Raspberry Pi e a biblioteca LIRC

Neste projeto foi utilizada uma Raspberry Pi 2 com o sistema operacional Raspbian, a instalação desse sistema foi seguida de acordo com o passo a passo apresentado no site oficial[11].

A biblioteca LIRC foi utilizada para enviar os sinais infravermelho, sua instalação foi feita da seguinte forma:

- Execute a linha:
 - `$ sudo apt-get install lirc`
- Acrescente as linhas no arquivo `/etc/modules`:
 - `lirc_dev`
 - `lirc_rpi gpio_in_pin=23 gpio_out_pin=22`
- Edite o arquivo `/etc/lirc/hardware.conf` e deixe-o igual as linhas abaixo:
 - `#####`
 - `# /etc/lirc/hardware.conf`
 - `#`
 - `# Arguments which will be used when launching lircd`
 - `LIRCD_ARGS="--uinput"`
 -
 - `# Don't start lircmd even if there seems to be a good config file`
 - `# START_LIRCMD=false`
 -
 - `# Don't start irexec, even if a good config file seems to exist.`
 - `# START_IEXEC=false`
 -
 - `# Try to load appropriate kernel modules`
 - `LOAD_MODULES=true`
 -
 - `# Run "lircd --driver=help" for a list of supported drivers.`
 - `DRIVER="default"`
 - `# usually /dev/lirc0 is the correct setting for systems using udev`
 - `DEVICE="/dev/lirc0"`
 - `MODULES="lirc_rpi"`
 -
 - `# Default configuration files for your hardware if any`
 - `LIRCD_CONF=""`
 - `LIRCMD_CONF=""`
 - `#####`
- Execute as linhas de comando:
 - `$ sudo /etc/init.d/lirc stop`
 - `$ sudo /etc/init.d/lirc start`
- No arquivo `/boot/config.txt`, acrescente a linha:
 - `dtoverlay=lirc-rpi,gpio_in_pin=23,gpio_out_pin=22`

- Encontre o arquivo de configuração desejado em <http://lirc.sourceforge.net/remotes/> e coloque-o em /etc/lirc/lircd.conf
- Execute:
 - \$ sudo apt-get update
 - \$ sudo apt-get upgrade
- Reinicie a Raspberry e então execute:
 - \$ sudo /etc/init.d/lirc start
 - \$ sudo killall lircd
 - \$ sudo lircd -d /dev/lirc0

Por fim, é necessário montar o circuito com o LED infravermelho igual a Fig. 7. A porta GPIO 22 foi utilizada.

2.3 NodeMCU

O NodeMCU foi estudado durante o desenvolvimento desse projeto, ele é uma peça importante em um projeto como esse por se tratar de um componente muito mais barato e com um consumo energético muito menor.

Ele não foi implementado devido ao tempo demandado das outras atividades, porém ele poderia substituir facilmente a Raspberry Pi. Como esse projeto possui fins didáticos a Raspberry Pi foi mantida e o NodeMCU serviu apenas de objeto de estudos, mas em um projeto real de engenharia, ele faria o papel de servidor REST e de emissor de sinais infravermelho.

Por ter um custo reduzido, o NodeMCU seria replicado para os diversos ares-condicionados de ambiente, assim seria aplicado o conceito de mestre e escravo.

2.4 O sensor de temperatura SHT1x e o protocolo I²C

O protocolo I2C (Inter-Integrated Circuit) é um protocolo para comunicação entre circuitos integrados inventado pela Philips Semiconductor (hoje NXP Semiconductor). Ele tem a característica de ser síncrono (possui sinal de clock) e baseado no modelo master-slave. O mestre, que é ao menos um (apesar de normalmente só haver um), coordena a comunicação (gera o sinal de clock e inicia a comunicação). Os escravos, normalmente vários, recebem comandos do mestre e enviam informações [5, 6] .

Os elementos da comunicação são interligados por meio de um barramento I2C, conforme figura 4. Todos os circuitos são unidos por apenas dois fios. O primeiro é o Serial Data (SDA), responsável pela transferência dos dados, sendo bidirecional. O segundo é o Serial Clock (SCL), responsável pela temporização entre os dispositivos. Apenas o mestre escreve no SCL, colocando o sinal de clock. O clock serve para saber quando os bits em SDA devem ser lidos. O fato de haver apenas dois fios no barramento, independente da quantidade de elementos é uma das maiores vantagens do I2C [5, 6].

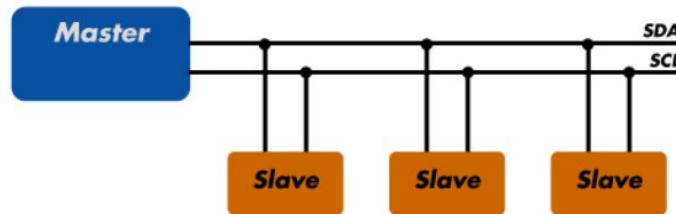


Figura 4. Barramento I2C [5].

Como todos os escravos se encontram em um mesmo barramento, é necessário que eles sejam endereçados, para que o mestre possa especificar com quem quer se comunicar em um determinado instante. Assim, cada dispositivo I2C tem um endereço de 7 bits que o identifica. Último bit (totalizando 8) indica se a operação é de leitura ou escrita (read/write). O mestre envia o endereço (byte) ao início da transmissão para indicar com quem quer se comunicar, e o escravo só responde se for o seu endereço [5].



Figura 5. Endereço I2C [5].

A comunicação em si se dá nas seguintes etapas (conforme figura 5) [7]:

- 1) O mestre inicia a comunicação com um sinal de "start".
- 2) O Mestre envia o endereço (8 bits) do escravo.
- 3) O escravo confirma recebimento com um sinal ACK (nível lógico 0).
- 4) Pode haver uma pausa (para escravo processar").
- 5) Mestre ou escravo (dependendo o último bit do endereço) envia os dados.
- 6) Quem recebe os dados confirma o recebimento, enviando um ACK (nível lógico 0) ou NACK (nível lógico 1).
- 7) Se foi enviado um NACK, o mestre encerra a transmissão com uma sequência de "stop". Caso contrário, mais dados são enviados.

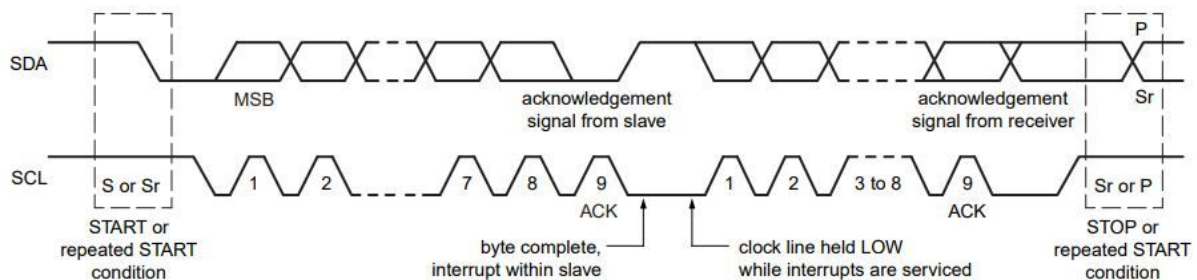


Figura 6. Comunicação I2C ao longo do tempo.

No projeto foi usado o Sensor de humidade e temperatura SHT11. Este circuito usa um protocolo baseado no I2C, apesar de bem parecido. As principais mudanças são [8]:

- O formato da sequência de "start" e inexistência de sequência de "stop".
- Não há endereços propriamente dito. No lugar do endereço o mestre envia o comando que ele deseja que o sensor faça.

2.5 Servidor REST

Atualmente uma variedade de aplicações e serviços estão disponíveis na internet. Uma das maiores dificuldades atualmente é permitir que essas aplicações e serviços possam ser consumidas pela diversidade de dispositivos existentes (computadores pessoais, celulares, tablets, etc), sem que seja necessário refazer toda a aplicação para cada caso. Uma das soluções para esse problema é o uso de REST [9].

O REST (Representational State Transfer) consiste em regras e princípios que permitem a criação de um sistema com interfaces bem definidas, de modo que diversos dispositivos possam usá-la. É uma abstração da arquitetura Web, usando o protocolo HTTP. Esse protocolo é baseado no modelo cliente-servidor, em que o cliente pode fazer requisições ao servidor, que o responde. As duas principais requisições são: GET (o cliente solicita alguma informação) e POST (o cliente envia alguma informação) [9].

3. Material e Métodos

Como controlador, foi utilizada uma Raspberry Pi 2 com o sistema operacional Raspbian.

Para controlar o aparelho de ar condicionado, foi usado um circuito emissor de infravermelho, conforme a figura 7. O resistor ligado aos 5V foi omitido. A entrada do circuito foi ligada a um pino GPIO da Raspberry.

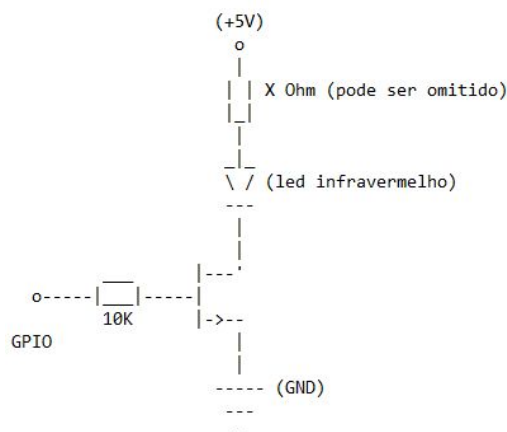


Figura 7. Circuito emissor de infravermelho.

O controle da emissão dos sinais de infravermelho foi feito usando a biblioteca LIRC. Procurando na internet o modelo do ar-condicionado que queríamos controlar, encontramos o arquivo de configuração correto.

Para a medição da temperatura, foi usado o sensor SHT11. Toda a comunicação como o circuito foi realizado por meio de um script Python.

Para disponibilizar o serviço de controle de ar-condicionado na internet, fizemos um servidor REST, usando Python e o framework Flask.

A parte física do sistema pode ser visto na figura 8.

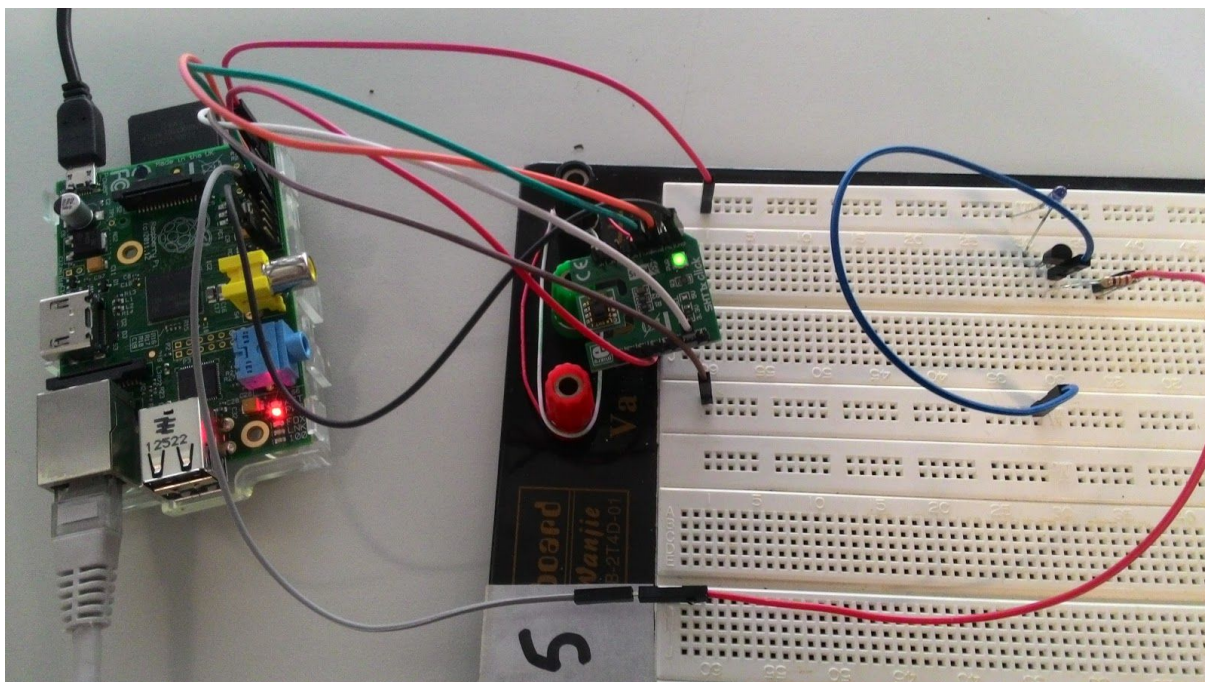


Figura 8. Parte física do sistema.

A interação com o usuário se dá por meio de um aplicativo para dispositivos móveis Android. O aplicativo foi desenvolvido usando o AndroidStudio.

Todos os códigos se encontram no repositório: <https://github.com/gacra/AutoAC>

4. Resultados e discussões

Os objetivos do projeto foram em sua maioria alcançados. Por meio do aplicativo, o usuário teria as seguintes funcionalidades:

- Controlar em “tempo real” do ar-condicionado (ligar/desligar e temperatura).
- Verificar temperatura ambiente.
- Fazer agendamentos (Dia da semana e horário para ligar em uma temperatura ou desligar)

Assim, temos que a automatização da tarefa de controlar um aparelho de ar-condicionado foi alcançada, gerando um possível aumento de comodidade, praticidade e economia de energia elétrica.

O emissor de infravermelho funcionou adequadamente, inclusive a distâncias maiores do que as esperadas. O sensor também funcionou, sendo fundamental para dar ao usuário um feedback (mesmo que não conclusivo) de que o sistema está funcionando.

O aplicativo também ficou de acordo com o esperado, havendo boa (e fácil) interação com o servidor em REST. Infelizmente, não foi possível passar o aplicativo para um dispositivo móvel real, devido a erros que não conseguimos solucionar a tempo. Dessa maneira, a comprovação do funcionamento ocorreu apenas no simulador do AndroidStudio.

5. Conclusão

O projeto idealizado no início da disciplina foi cumprido, no final do projeto foi possível controlar o ar-condicionado por linhas de comandos na Raspberry Pi, controlar o ar-condicionado com uma aplicação REST, controlar o ar-condicionado com uma programação temporizável e controlar o ar-condicionado por um aplicativo de smartphone Android.

A biblioteca lirc acelerou muito no desenvolvimento do projeto, além de ela implementar a saída de dados que controla o funcionamento do LED infra-vermelho, ela possui uma biblioteca com vários modelos de ares-condicionados e isso facilitou o trabalho dos desenvolvedores. Apesar de se ter usado uma biblioteca pronta, tudo o que foi implementado foi estudado e compreendido, pois se trata de um projeto didático.

A Raspberry Pi foi fundamental para o desenvolvimento do projeto, ela foi capaz de emitir sinais para um LED infravermelho por um protocolo conhecido. O NodeMCU que foi apresentado no decorrer do projeto seria uma ótima alternativa para substituir a Raspberry Pi, mas por questões de tempo de implementação não foi incorporado, entretanto, foi estudado e compreendido, caso fosse feito um novo projeto com os mesmos fins certamente seria aplicado, pois o NodeMCU possui um custo e um consumo menor.

5.1 Trabalhos Futuros

O NodeMCU poderia substituir a Raspberry Pi, diminuindo os custos do projeto, mas aumentando a complexidade, pois é um embarcado que não possui sistema operacional e não trabalha com uma linguagem tão simples como o Python.

Este projeto não possui nenhum tipo de autenticação, ou seja, qualquer pessoa conectada a internet poderia enviar comandos para a Raspberry Pi e controlar o ar-condicionado e isso é indesejável. É importante criar um sistema de autenticação e um sistema de criptografia de dados para aumentar ainda mais a segurança.

O projeto possui uma aplicação REST, porém a interface gráfica foi implementada apenas para Android. Seria muito mais interessante se houvesse uma aplicação Web que controlasse o REST e posteriormente outras aplicações Mobile e Desktop. A aplicação Android foi preferencialmente desenvolvida pois, por se tratar de um projeto didático, o desenvolvimento de aplicativos para essa plataforma é de grande curiosidade dos desenvolvedores e foi muito importante para a formação profissional dos mesmos.

Referências

- [1] BOLZANI, C. A. M. Residências Inteligentes. Livraria da Física, 2004.
- [2] ACCARDI A., DODONOV E. Automação Residencial: Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos: Tecnologias, Infraestrutura e Software, 2012.
- [3] CASADOMO. Domótica - Introducción. Agosto 2010. Disponível em: <<http://www.casadomo.com/>>.

- [4] TANENBAUM, A. S. Rede de Computadores. Editora Campus, 2003.
- [5] DOS REIS, V. R. I2C - Protocolo de Comunicação. Dezembro, 2015. Disponível em <<http://www.arduino.br.com/arduino/i2c-protocolo-de-comunicacao/>>
- [6] RENNÓ, D. O protocolo I2C. Disponível em <http://www.geocities.ws/deciorenno/Eletricidade/TD9_I2C.pdf>
- [7] I2C-bus specification and user manual. NXP. Abril, 2014. Disponível em <<https://www.nxp.com/docs/en/user-guide/UM10204.pdf>>
- [8] Datasheet SHT1x (SHT10, SHT11, SHT15) - Humidity and Temperature Sensor IC. Sensirion. Dezembro, 2011. Disponível em <https://cdn.sparkfun.com/datasheets/Sensors/Pressure/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf>
- [9] O que é API? REST e RESTful? Conheça as definições e diferenças. Bencode. Disponível em <<https://bencode.com.br/o-que-e-api-rest-e-restful/>>
- [10] Protocolo RC-5. http://www.pcbheaven.com/userpages/The_Philips_RC5_Protocol/
- [11] Instalação do Raspbian. <https://www.raspberrypi.org/documentation/installation/installing-images/>

Figura 1. <https://en.wikipedia.org/wiki/RC-5>