
Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

ACIONAMENTO AUTOMÁTICO DE DISPOSITIVO DE SEGURANÇA.

Iury Rodrigues Prates

MSc. André Chastel Lima (orientador)

Dourados – MS
2021

ACIONAMENTO AUTOMÁTICO DE DISPOSITIVO DE SEGURANÇA

IURY RODRIGUES PRATES

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Iury Rodrigues Prates e aprovada pela Banca Examinadora como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 10 de Novembro de 2021.

Prof. MSc. André Chastel Lima(orientador)

P925a Prates, Iury Rodrigues

Acionamento automático de dispositivo de segurança / Iury Rodrigues Prates. - Dourados, MS: UEMS, 2021.
56p;

Monografia (Graduação) - Ciência da Computação -
Universidade Estadual de Mato Grosso do Sul, 2021.
Orientador: Prof. MSc. André Chastel Lima.

1. Internet das coisas 2. Raspberry Pi 3B. Automação.
Lima, André Chastel | Título.

CDD: 23. ed. - 004.678

ACIONAMENTO AUTOMÁTICO DE DISPOSITIVO DE SEGURANÇA.

Iury Rodrigues Prates

Novembro de 2021

Banca Examinadora:

Prof. MSc. André Chastel Lima (Orientador)
Área de Computação – UEMS

Prof^a. Dra. Glaucia Gabriel Sass
Área de Computação – UEMS

Prof. Dr. Rubens Barbosa Filho
Área de Computação – UEMS

Dedico este trabalho a todos que me apoiaram em toda minha jornada acadêmica.

AGRADECIMENTOS

Agradeço a todos os meus familiares, em especial a meus pais que me deram todo suporte e apoio para sair da minha cidade natal e concluir minha graduação em Dourados.

Agradeço meu professor e orientador MSc. André Chastel que me auxiliou com esse trabalho e sempre esteve pronto para tirar minhas dúvidas e aconselhar.

Agradeço aos amigos que estiveram comigo e em especial Ronaldo Tafarel que me ajudou nas dificuldades das disciplinas e minha irmã Brenda Camila que sempre me deu forças e apoio moral nos momentos de dificuldades.

RESUMO

Um campo de grande abrangência dentro da área da computação é a Internet das Coisas, do inglês, *Internet of Things* (IoT). A IoT traz a integração de variados dispositivos e diversas aplicações que visam facilitar o nosso dia a dia. Sendo assim, um campo de atuação que tem crescido substancialmente é o de automação residencial. A automação residencial busca desenvolver aplicações com intuito de otimizar pequenas funções dentro de uma residência, que podem ser automatizadas, fazendo uso de Sistemas Embarcados (SE's). Sistemas embarcados são sistemas embutidos em microprocessadores que executam tarefas específicas. Neste trabalho é proposto uma aplicação que pode ser utilizada a fim de automatizar diferentes tarefas dentro de uma residência utilizando os conceitos de IoT e fazendo uso de um SE dentro de um Raspberry Pi.

Palavras-chave: Internet das coisas, Sistemas embarcados, Raspberry Pi.

ABSTRACT

A broad field within the computing field is the Internet of Things (IoT). The IoT brings the integration of various devices and various applications that aim to facilitate our daily lives. Therefore, a field of activity that has grown substantially is that of home automation. Home automation seeks to develop applications with the aim of optimizing small functions within a home, which can be automated, making use of embedded systems (SE's). Embedded systems are systems embedded in microprocessors that perform specific tasks. This work proposes an application that can be used to automate different tasks within a residence using the concepts of Internet of Things and making use of an SE within a Raspberry Pi.

Key words: Internet of things, Embedded systems, Raspberry Pi.

SUMÁRIO

1. INTRODUÇÃO.....	20
1.1. OBJETIVO.....	21
2. REFERENCIAL TEÓRICO.....	22
3. INTERNET DAS COISAS.....	24
3.1. INTRODUÇÃO À INTERNET DAS COISAS.....	24
3.2. CONCEITOS DE INTERNET DAS COISAS.....	24
4. SISTEMAS EMBARCADOS.....	26
4.1. CONCEITOS.....	26
4.2. HISTÓRIA.....	26
4.3. CARACTERÍSTICAS.....	28
4.4. ESTRUTURA.....	29
4.4.1. HARDWARE.....	30
5. RASPBERRY PI.....	33
5.1. HISTÓRIA.....	33
5.2. RASPBERRY PI 3 MODEL B.....	33
5.3. PROCESSADOR.....	36
5.4. CONECTIVIDADE.....	37
5.5. ARMAZENAMENTO.....	37
6. SISTEMA OPERACIONAL.....	39
6.1. DISTRIBUIÇÕES.....	39
6.2. RASPBERRY PI OS.....	39
6.3. INSTALAÇÃO.....	40
7. METODOLOGIA.....	41
7.1. DESENVOLVIMENTO.....	41
8. CONCLUSÃO.....	49
9. REFERÊNCIAS BIBLIOGRÁFICAS.....	50
APÊNDICE A - <i>Script</i> do arquivo cliente.py.....	54
APÊNDICE A - <i>Script</i> do arquivo servidor.py.....	56

LISTA DE FIGURAS

Figura 1 - Raspberry Pi 3B.....	34
Figura 2 - Raspberry Pi 3B componentes.....	35
Figura 3 - Conexão <i>socket</i> lado cliente.....	42
Figura 4 - Conexão <i>socket</i> lado servidor.....	43
Figura 4 – Configuração dos pinos GPIO.....	44
Figura 6 - Módulo relé de 2 canais.....	45
Figura 7 - Placa de prototipagem e cabos <i>jumper</i>	45
Figura 8 - Saídas do módulo relé.....	46
Figura 9 - <i>Leds</i> de funcionamento do módulo relé	47
Figura 10 - Circuito utilizado.....	47
Figura 11 - Circuito do teste prático.....	48

LISTA DE SIGLAS

GPIO - *General Purpose Input/Output*

IoT - *Internet of Things*

SE - Sistema Embarcado

SO - Sistema Operacional

1. INTRODUÇÃO

Com a consolidação da internet das coisas, cada vez mais o fluxo de dados que trafega na internet se torna maior, dados esses de diferentes formatos que podem ser provenientes de vários dispositivos diferentes. O fenômeno da internet das coisas transformou o modo com que construímos aplicações. Cada vez mais novos tipos de aparelhos são integrados a internet.

Segundo alguns dados estimados por pesquisas em alta tecnologia, mais de 90% dos microprocessadores fabricados mundialmente são destinados a máquinas que usualmente não são chamadas de computadores. Dentre alguns destes dispositivos estão aparelhos celulares, eletrodomésticos, automóveis, aparelhos de DVD e PALM´s. O que diferencia este conjunto de dispositivos de um computador “convencional” (PC – *Desktop, Notebook*), conhecido por todos, é o seu projeto baseado em um conjunto dedicado e especialista constituído por Hardware, Software e Periféricos – um Sistema Embarcado. (CHASE, 2007)

A internet das coisas tem como objetivo transformar a vida das pessoas, tornando suas rotinas mais simples e mais eficazes. Com a comunicação entre os equipamentos e objetos, será possível antecipar situações ou necessidades. Assim, o principal objetivo e funcionalidade da internet das coisas é, portanto, conectar objetos rotineiros à internet, dando a eles uma funcionalidade mais ampla, com mais inteligência e maior capacidade de integração à outras fontes de informação. (SCHULTZ, 2020)

Sistemas embarcados possuem grande destaque e abrangência dentro do campo de internet das coisas, visto que a maioria desses sistemas atendem os seus conceitos.

Os sistemas embarcados estão presentes desde sistemas de aplicações simples, como os temporizadores de semáforos, até sistemas mais robustos como controles de navegação aérea. Com o crescente número de especialistas e entusiastas de sistemas embarcados, se fazia necessário a criação de um componente que fosse lúdico e pudesse fornecer a teoria e prática para o desenvolvimento de um sistema embarcado. Sendo assim em 2006, o Engenheiro Eben Upton e seu grupo de pesquisa criaram as primeiras placas do Raspberry Pi,

que tinha por finalidade o aprendizado para crianças, jovens e entusiastas da área de tecnologia, estas placas eram concebidas como microcomputadores, por possuir a maioria das interfaces de um computador pessoal, e deveria ser de fácil aquisição aos usuários. (HEIN, 2013)

1.1 OBJETIVO

O objetivo deste trabalho é primeiramente fazer um estudo sobre os conceitos de internet das coisas, e também sobre a placa Raspberry Pi. Ao final apresentar resultados de testes feitos com uma pequena aplicação que atenda e faça uso dos conceitos e ferramentas que foram estudadas. Essa aplicação fará a integração de um sistema embarcado que será executado dentro de um Raspberry Pi para poder automatizar determinada tarefa doméstica. Para implementar essa automatização será utilizado os pinos GPIO(*General Purpose Input/Output*) que é basicamente um conjunto de pinos responsável por fazer a comunicação de entrada e saída de sinais digitais do Raspberry Pi. O sistema embarcado deverá ser acionado através de um comando recebido a partir de outro dispositivo após ser estabelecida uma conexão remota utilizando *socket*.

2. REFERENCIAL TEÓRICO

Neste tópico são mencionados alguns dos trabalhos que serviram de base para criar este projeto.

Diante de um cenário de rápido avanço tecnológico os sistemas embarcados estão cada vez mais introduzidos no nosso dia a dia, isso abriu um imenso campo de pesquisa para desenvolvimento de soluções que buscam suprir as mais diversas necessidades.

Cruz e Lisboa (2014) propõem um modelo de sistema de controle automatizado para casas inteligentes. Esse sistema automatiza e gerencia tarefas dentro de uma residência, tais como, acionamento de luzes, controle de temperatura dos ambientes, controle de estados de portas e janelas entre outros. Essa proposta utiliza equipamentos que possuem acesso a rede *wi-fi*, conectando-os todos a uma mesma rede onde essa terá uma aplicação que faz o gerenciamento de todos os equipamentos. A aplicação proposta pode ser acessada em página web através de qualquer navegador de um computador ou dispositivo *android*. As ferramentas utilizadas nesse trabalho foram a placa de prototipagem Arduíno e o Raspberry Pi. Este modelo é voltado para a automação residencial com intuito de atender as necessidades de uma *Smart Home*(Casa inteligente), que são casas que possuem dispositivos que podem ser controlados à distância via internet ou comandos de voz.

Em Silva et al. (2018) também é apresentado um projeto de automatização residencial utilizando o Raspberry Pi e um sistema que centraliza as informações. Nessa proposta o Raspberry Pi foi utilizado como servidor web. Esse servidor funciona como um armazenador e distribuidor de dados, desta forma as informações são compartilhadas. As páginas da internet trocam dados com o servidor, onde ocorrem solicitações e recebimentos de informações entre página web e servidor e esses dados são transferidos pela internet através de um protocolo HTTP.

Existem vários projetos com a finalidade de se gerenciar dispositivos domésticos, quase todos apresentando o mesmo tipo de proposta. Em Lusembo (2017) é proposto um protótipo de sistema residencial utilizando o Raspberry Pi com um sistema que não fosse baseado em Linux, esse sistema faz o controle e monitoramento de algumas funções dentro da residência.

Já em Fiorin e Antonello (2017) é desenvolvido um sistema de automação utilizando o Raspberry Pi e um Arduíno onde o gerenciamento das ações a serem executadas é feita através de uma aplicação para dispositivo *android*.

3. INTERNET DAS COISAS

Neste capítulo são apresentados alguns conceitos de internet das coisas. No tópico 3.1 é feita uma breve introdução ao assunto e em 3.2 vemos de onde vem o surgimento do termo e as suas definições.

3.1 INTRODUÇÃO À INTERNET DAS COISAS

A internet das coisas, do inglês (*Internet of Things*(IoT)) emergiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. De fato, a IoT tem recebido bastante atenção tanto da academia quanto da indústria, devido ao seu potencial de uso nas mais diversas áreas das atividades humanas. (SANTOS et al, 2016)

3.2 CONCEITOS DE INTERNET DAS COISAS

A internet das coisas, em poucas palavras, nada mais é que uma extensão da Internet atual, que proporciona aos objetos que usamos no nosso dia a dia (quaisquer que sejam), que possuem capacidade computacional e de comunicação, se conectarem à Internet. A conexão com a rede mundial de computadores viabilizará, primeiro, controlar remotamente os objetos e, segundo, permitir que os próprios objetos sejam acessados como provedores de serviços. Estas novas habilidades, dos objetos comuns, geram um grande número de oportunidades tanto no âmbito acadêmico quanto no industrial. (SANTOS et al, 2016)

A IoT tem alterado aos poucos o conceito de redes de computadores, neste sentido, é possível notar a evolução do conceito ao longo do tempo como mostrado a seguir. Tanenbaum (2002) define rede de computadores como um conjunto de computadores autônomos que são interconectados por uma única tecnologia. Essa tecnologia pode ser de diferentes tipos. Peterson and Davie (2011) afirmam que a principal característica das redes de computadores é a sua generalidade, isto é, elas são construídas sobre dispositivos de propósito geral e não são otimizadas para fins específicos tais como as redes de telefonia e TV.

Já Kurose and Ross (2012) argumentam que o termo “Redes de Computadores” já começa a soar um tanto envelhecido devido à grande quantidade de equipamentos e tecnologias não tradicionais que são usadas na Internet.

A internet das coisas refere-se à integração de objetos físicos e virtuais em redes conectadas à Internet, permitindo que os objetos colem, troquem e armazenem dados que serão processados e analisados, gerando informações e serviços em grande escala. São muitas as possibilidades de objetos conectados: automóveis, *smartphones*, eletrodomésticos, artigos de vestuário, fechaduras, entre outros aparelhos. (ALMEIDA, 2015)

4. SISTEMAS EMBARCADOS

Neste capítulo são apresentadas as definições de um SE. Primeiramente em 4.1 são definidos os conceitos, dizendo o que faz um sistema ser ou não um SE. Em 4.2 é contado um pouco da história e do surgimento deste tipo de sistema e em 4.3 e 4.4 são apresentadas as características de um SE e como é a sua estrutura.

4.1 CONCEITOS

Um sistema é dito embarcado quando este tem apenas um propósito, uma única tarefa específica. Esse tipo de sistema interage continuamente ao meio que é aplicado por meio de sensores e atuadores. Os SE's podem interagir ou não com usuários dependendo do seu objetivo. (CHASE, 2007)

A denominação “embarcado” (do inglês, *Embedded Systems*) vem do fato de que estes sistemas são projetados geralmente para serem independentes de uma fonte de energia fixa como uma tomada ou gerador. As principais características de classificação deste sistema são a sua capacidade computacional e a sua independência de operação. Outros aspectos relevantes dependem dos tipos de sistemas, modos de funcionamento e itens desejados em aplicações embarcadas. (CHASE, 2007)

Atualmente, bilhões de sistemas computacionais são produzidos anualmente e aplicados em produtos dos mais variados segmentos. Considerados como computadores dedicados, ou embutidos, tais sistemas são muitas vezes chamados de computadores pervasivos ou ubíquos, no sentido que os usuários muitas vezes não os veem. (GARCIA, 2018)

4.2 HISTÓRIA

O surgimento dos SE's ocorreu no fim da década de 60. Com as necessidades do mercado atual e com os avanços tecnológicos da microeletrônica, as grandes empresas perceberam um mercado promissor nesse

tipo de tecnologia, o que as levou a investir na otimização dos componentes eletrônicos para esse segmento.

A evolução dos microcontroladores, que são um conjunto de diferentes tipos de chips que integrados possuem quase todos os aspectos e componentes de um computador pessoal, como: processador, memória RAM, periféricos de entrada e saída, conversor Analógico\Digital, etc.

Diferentemente de um computador pessoal que pode atender diferentes tipos de necessidades do usuário, os microcontroladores por possuir recursos limitados até podem possuir execuções de diferentes operações, mas seu propósito é para geralmente executar apenas uma tarefa exclusiva.

Nessa mesma época, surgiram os microprocessadores que foram muito importantes para os SE's, que são basicamente um circuito integrado que realiza as funções de cálculo e tomada de decisões, este executa as funções que os outros componentes externos o solicitam, como o microprocessador não trabalha individualmente e nem possui memória para que possa ser programado, trabalha como um processador de informações que os outros componentes (memórias, periféricos e conversores) os envia para execução.

O primeiro SE digital que se tem conhecimento foi AGC (*Apollo Guidance Computer*). Era um sistema responsável pelo controle das espaçonaves do projeto Apollo. O AGC não possuía processador, era construído com portas NOR (operação lógica que opera com dois valores lógicos), este sistema foi responsável por levar o homem ao espaço nas décadas de 60 e 70. Sua utilização foi de extrema importância para o projeto Apollo, para que os cálculos exatos de navegação fossem realizados no espaço, independentemente dos cálculos de rotas que eram feitos na Terra. (EMBARCADOS, 2013)

A maioria dos primeiros programas eram feitos em linguagem de máquina (*Assembler*), já nas décadas seguintes começavam a surgir bibliotecas de códigos direcionados para os SE's para determinados processadores. Atualmente os sistemas podem ser desenvolvidos em linguagens de alto nível ou até mesmo possuir versões especiais de sistemas operacionais. (EMBARCADOS, 2013)

Atualmente, os sistemas embarcados são a categoria de sistemas processados de maior utilização no mundo, superando em número os PC's, *notebooks*, servidores e semelhantes. Está presente em uma categoria de sistemas de computação dita oculta, no qual esses sistemas quase não são notados pelos humanos. Os vários sistemas deste tipo podem fazer parte de diversos segmentos de extrema importância na vida humana, desde um simples sistema até os mais sofisticados.

Um exemplo do nosso cotidiano é o elevador que possui uma interface com inúmeros botões, onde cada um representa um andar ou alguma ação do elevador, o sistema do elevador então faz o controle das paradas dos andares ou interrupção emergencial, sendo que os passageiros não possuem a informação de como o sistema comanda cada ação. (MORIMOTO, 2007)

4.3 CARACTERÍSTICAS

Quando pensamos nas características de um SE, uma das mais relevantes é o seu tamanho. Afinal, como ele faz parte de algum dispositivo, costuma ser pequeno para ocupar menos espaço. Esse tamanho reduzido reflete em sua capacidade de operação, que é bastante reduzida. Muitas vezes o software administrado pelo sistema é registrado em microcontroladores. Além disso, em vez de fazer uso de softwares convencionais, esse tipo de dispositivo adota os chamados softwares de tempo real, projetados para a execução de tarefas específicas em tempo real. (NOLETO, 2020)

Chase (2007) afirma que um sistema é classificado como sendo embarcado quando este é dedicado a uma única tarefa e interage continuamente com o ambiente a sua volta por meio de sensores e atuadores. Esse tipo de sistema requer do projetista conhecimentos em programação, sistemas digitais, noções de controle de processos, sistema de tempo real e tecnologias de aquisição de dados e de atuadores por exigir uma interação contínua com o ambiente.

Algumas das áreas que os sistemas embarcados estão inseridos são no setor automobilístico, *VideoGames*, sistemas de controle industriais, processamento de sinais, robótica, automação residencial entre outras.

Chase (2007) afirma também que todo SE é composto por uma unidade de processamento, que é um circuito integrado, fixado a uma placa de circuito impresso. Esses circuitos possuem uma capacidade de processamento de informações vindas de um software que está sendo processado internamente nessa unidade, sendo assim o software está embarcado na unidade de processamento.

4.4 ESTRUTURA

Um SE normalmente é formado por um microcontrolador e um software (*firmware*) dedicado especificamente para o sistema no qual será projetado, pode ser também conectado em algum outro tipo de periférico. Alguns dos fatores importantes para definir um SE:

- Tamanho e peso: Sempre os menores possíveis. Desde o esboço inicial do projeto tem que haver atenção ao tamanho e peso do sistema desenvolvido. Com a crescente miniaturização dos equipamentos eletroeletrônicos. O tamanho e o peso são fatores importantes para a locomoção ou para o espaço físico onde será implantado o sistema. (CUNHA, 2007)

- Consumo de energia: quanto maior a autonomia do sistema, menor será sua necessidade de recarga e menor será seu consumo de energia, o que gera economia nos geradores elétricos móveis (ou células eletroquímicas) como as baterias e pilhas. (CUNHA, 2007)

- Robustez e Durabilidade: muitos sistemas são projetados para operar em condições adversas (vibrações, calor, poeira, variações de tensão de alimentação, interferências eletromagnéticas, raios, umidade, corrosão etc.). É necessário que o sistema conviva com essas condições sem sofrer interferências externas no ambiente no qual será empregado. (CUNHA, 2007)

- Custos: o desenvolvimento de hardware para aplicação de sistemas

embarcados sofre grandes imposições por redução de custos. Desde as primeiras definições sempre foi colocado em pauta a redução de custos para ser mais acessível ao consumidor final. Estas reduções de custos desencadeia uma série de restrições sobre o planejamento do desenvolvimento dos sistemas embarcados, a fim de reduzir os custos do sistema em geral. (GUTERRES, 2017)

- Usabilidade: os sistemas embarcados possuem grande apelo comercial por terem um custo reduzido, sendo utilizado em diversos segmentos de atuação. A maioria dos processadores fabricados hoje em dia são utilizados em SE's. As aplicações dos sistemas embarcados estão gradativamente presentes em nosso cotidiano, em eletrodomésticos, no setor automotivo, em sistema controle de produção, equipamentos médicos, automação residencial e muitos outros segmentos. (GUTERRES, 2017)

4.4.1 HARDWARE

SE's geralmente possuem a seguinte subdivisão de hardware: unidade central de processamento, memória e periféricos. A unidade central de processamento (do inglês, *Central Processing Unit* - CPU) também conhecida como processador, a CPU corresponde ao cérebro do computador, onde é feita a maior parte dos cálculos. É o elemento de maior importância em equipamentos eletrônicos. É responsável pelo processamento de todos os tipos de dados e pela apresentação do resultado do processamento. Inicialmente, a CPU era composta por vários componentes separados, mas evoluiu para um único circuito integrado que recebeu o nome de microprocessador. (GUTERRES, 2017)

O microprocessador é um dispositivo programável de entrada e saída de dados, que processa os dados digitais de entrada e, associando as instruções armazenadas em sua memória, fornece como saída os dados resultantes do processamento. (SIGNIFICADOS, 2013)

A CPU é fisicamente um processador, sendo composta basicamente por três componentes:

- Unidade de Controle (UC), responsável pela busca das instruções na memória principal, sua decodificação e execução, controla todos os componentes e dispositivos de um computador, emitindo sinais elétricos (pulsos de controle). (MANOLA, 2010)

- Unidade Lógica e Aritmética (ULA), possui os circuitos necessários para executar operações lógicas e aritméticas, possui também pelo menos um registrador encarregado de armazenar números a serem operados pela ULA ou resultados de operações. (MANOLA, 2010)

- Registradores e Memória cache, memória interna de pequena capacidade de armazenamento, mas de alta velocidade, usada para armazenar resultados temporários e certas informações de controle. (MANOLA, 2010)

A memória no sistema embarcado armazena dados e instruções que são vinculadas às operações da CPU. Na Arquitetura Harvard, que é mais utilizada em SEs, as instruções e os dados são fisicamente separados em memórias diferentes. Os periféricos são interfaces que recebem e enviam informações à CPU. Em sistemas embarcados duas categorias de periféricos se destacam, os sensores e os atuadores. (GUTERRES, 2017)

Na parte de periféricos temos os sensores e os atuadores. Os sensores são responsáveis por fazer a leitura de sinais e entregarem para a unidade de processamento. Já os atuadores operam comandos que controlam o ambiente com ações manuais, elétricos ou mecânicos, estes periféricos recebem uma informação da CPU e interferem no processo em controle, exemplo são: motores, *coolers*, luzes, aquecedores, etc.

O sensor faz a detecção de alguma medida física, e gera uma interrupção para a unidade de processamento. Os sensores devem estar sempre calibrados e funcionando corretamente, para fornecer informações confiáveis, sem alterar os dados do ambiente monitorado. Exemplos de sensores são, de distância (sonar, infravermelho), de temperatura (termistores), movimento (acelerômetros). Outro tipo de sensor é o transdutor, que além de poder fazer a leitura do ambiente, ainda pode

converter sinal recebido em um outro tipo de sinal. Um exemplo de transdutor é o microfone que converte o som recebido em sinais elétricos. (GUTERRES, 2017)

Dependendo da necessidade do projeto poderá utilizar um único circuito integrado, também conhecido como SOC (*System-on-Chip*), é uma miniatura de sistema computacional em um microcontrolador, consiste em uma CPU, memória e portas de entrada e saída, é um circuito único integrado mais complexo e apresenta mais recursos. Por outro lado, existe um circuito dedicado em um módulo SOM (*System-on-Module*), que consiste em um único módulo de hardware. Um SOM, combinado a um circuito que implementa interfaces específicas, forma então o hardware do sistema embarcado. (GUTERRES, 2017)

Um microcontrolador muito utilizado pela comunidade de entusiastas é o Arduíno. O Arduino é um computador simples que roda um programa por vez, (monotarefa), geralmente sendo bem simples de usar, diferente de um Raspberry Pi, que se assemelha mais a um computador de uso geral. O Raspberry Pi geralmente conta com um sistema operativo Linux, e com a possibilidade de rodar vários programas de uma vez (multitarefa). Seu uso é um pouco mais complexo que de um Arduino.

5. RASPBERRY PI

Neste capítulo é apresentado um pouco sobre o Raspberry Pi, em 5.1 é mencionado um pouco da sua história e de como foi sua idealização e propósitos. Em 5.2 algumas características do modelo 3B, modelo este que foi utilizado neste trabalho. Em 5.3, 5.4 e 5.5 são apresentadas as características que o modelo 3B possui, sendo seu processador, sua conectividade e o tipo de armazenamento.

5.1 HISTÓRIA

Raspberry Pi é um computador de baixo custo e que tem o tamanho de um cartão de crédito. Foi desenvolvido no Reino Unido pela Fundação Raspberry Pi. Foi idealizada e projetada pelo engenheiro britânico Eben Upton e sua equipe. A função básica do Raspberry Pi é oferecer uma alternativa barata, prática e acessível para que pessoas de várias idades possam explorar todas as capacidades da computação. Além disso, também visa facilitar a aprendizagem de programação em linguagens como Scratch e Python. (PORTUGAL, 2017)

O objetivo inicial do projeto era desenvolver e comercializar um computador de placa única (SoC), de tamanho reduzido e de fácil aquisição pelos interessados no assunto. Apesar do tamanho enxuto e das características pouco convencionais, o Raspberry Pi é um computador como outro qualquer. Dependendo de sua finalidade, pode até ser utilizado como um computador de uso pessoal, tendo os mesmos recursos de um computador convencional, como navegadores de internet, reprodução de conteúdo multimídia e ferramentas de edição de texto. (RASPBERRY, 2013)

Os primeiros conceitos do Raspberry Pi surgiram em 2006 por Upton e sua equipe, mas só em agosto de 2011 que os primeiros componentes começaram a ser distribuídos para o público em geral.

5.2 RASPBERRY PI 3 MODEL B

O modelo B da terceira geração do Raspberry Pi se destaca pelo aumento de

performance e pelo hardware utilizado. Atualmente já temos no mercado modelos mais recentes e com características que superam o modelo que foi utilizado neste trabalho. Ver Figura 1.



Figura 1 - Raspberry Pi 3B

Fonte: (FAZEDORES, 2018)

O Raspberry Pi 3 vem com uma CPU ARM Cortex-A53 de 1,2 GHz e quatro núcleos. Esta CPU significa que você terá muito mais poder de processamento disponível. Será interessante ver o que podemos fazer com esse poder extra. De acordo com alguns *benchmarks*(testes de performance) feitos, a CPU oferece 10x mais desempenho em relação ao Raspberry Pi 1 e um aumento de 50% em relação ao Raspberry Pi 2. A quantidade de RAM permanece a mesma, com apenas 1 gb disponível. As portas USB também permanecem inalteradas com o Pi 3, com as mesmas 4 portas USB 2.0 que vimos no antecessor. Também não há alterações na porta Ethernet, com ela ainda em 10/100 Mbps. Os requisitos de energia permanecem praticamente os mesmos, mas se você tiver dispositivos USB com “fome de energia” conectados ao Pi, é recomendável atualizar para uma fonte de alimentação que suporte 2.5A. (PROTOTIPANDO, 2020)

Devido a estas modificações e a outros melhoramentos na organização da

placa, pode-se ter um ganho de desempenho de até 50%, se comparado ao modelo antecessor. Além do processador, outra importante mudança foi a inclusão de duas conexões sem fio, uma *Wireless Lan* 802.11 bgn e a outra é o *Bluetooth* 4.1, o que torna a utilização da placa muito cômoda, conectando a placa com o *Wi-Fi* ou por conexão *bluetooth*. (PORTUGAL, 2017)

Em relação às outras interfaces de expansão, o Model B da terceira geração oferece os mesmos conectores e mesma configuração do modelo da segunda geração: GPIO de 40 pinos, quatro portas USB, interface Ethernet 10/100, saída de som, câmera interface (CSI), *display* interface (DSI), slot para o cartão micro SD, além do slot micro USB para alimentação, necessitando de uma fonte de 5,1V com 2,5A. As dimensões também são semelhantes às do modelo da segunda geração, tendo 85mm x 56mm x 17mm. (PORTUGAL, 2017)

Mesmo sendo lançado em 2016 e possuindo versões mais recentes no mercado, a performance desta placa é suficiente para quem deseja desenvolver projetos onde há a necessidade de um grande processamento e que necessite de comunicação sem fio através do *Bluetooth* ou do *Wireless*. Este foi o modelo utilizado neste trabalho. Na figura 2 pode ser observado a disposição dos componentes que o Raspberry Pi possui.

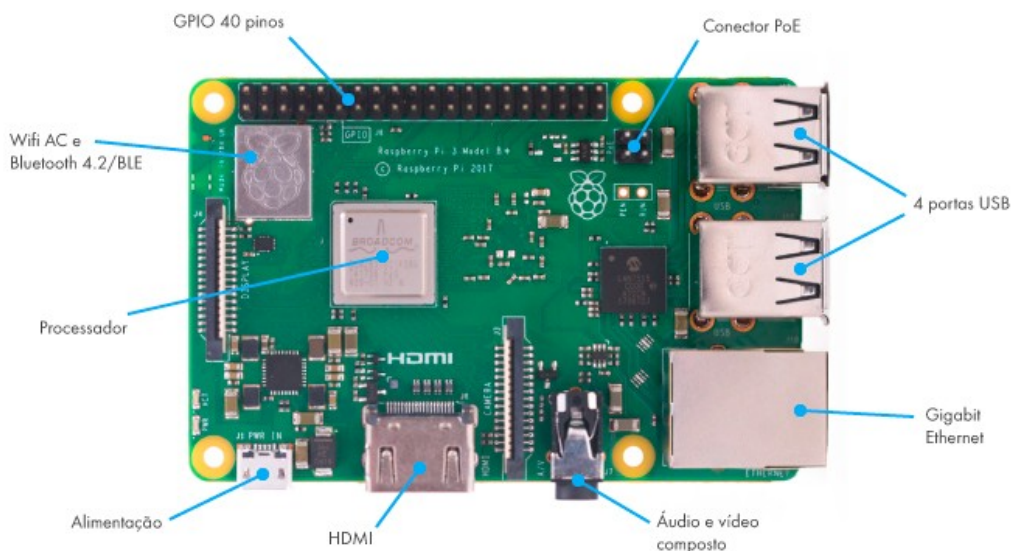


Figura 2 - Raspberry Pi 3B componentes

Fonte(FAZEDORES, 2018)

5.3 PROCESSADOR

O principal componente do Raspberry Pi 3, o chip Broadcom BCM2837 com o processador Cortex-A53, possui multiprocessamento simétrico de quatro núcleos em forma de *cluster*, cada uma, tendo uma memória cache L1 (Level 1 de 32 KB) e uma memória cache L2 (Level 2 128 KB) compartilhada, usada também pela GPU. (RODRIGUES, 2018)

O Cortex-A53 está baseada na arquitetura (de código) Armv8-A (família A), a Armv8 possui suporte de instruções 64 bits (AArch 64) e completa compatibilidade com arquitetura antecedente a Armv7 que utiliza instruções em 32 bits (AArch 32), incluindo recursos como o NEON, suporte para virtualização por e recursos de segurança (*TrustZone*). (ARM, 2017)

A tecnologia NEON é uma arquitetura de extensão SIMD (*Single Instruction Multiple Data*), que otimiza os algoritmos e as funções de processamento de sinais de áudio, vídeo, reconhecimento de voz e facial.

A *TrustZone* é uma tecnologia que fornece o isolamento de hardware para todo o sistema, criando um ambiente seguro e isolado que possa fornecer confidencialidade e integridade ao sistema.

O Cortex-A53 possui uma performance balanceada de desempenho e eficiência, construído de forma por utilizar baixa potência, mas que pode fornecer ao dispositivo um alto desempenho com restrição de energia nos ambientes.

O chip BCM 2837 ainda possui um processador gráfico (GPU) de baixo consumo, o VideoCore IV de 400MHz baseada na arquitetura DSP (*Digital Signal Processing*), dando a GPU eficiência e flexibilidade para codificar e decodificar uma série de codecs 3 para multimídia, mantendo o baixo consumo de energia, tudo isso permitindo ao Raspberry Pi uma resolução Full HD 1080p. A arquitetura DSP utiliza de processamento digital para calcular grande variedades de operações em uma sequência de amostras, esse processamento pode envolver operações lineares e não lineares. (ARM, 2017)

5.4 CONECTIVIDADE

O Raspberry Pi 3B possui porta *Ethernet* de 100MB/s de padrão RJ45, além de possuir embutido nessa versão o chip da Broadcom que serve para realizar conexão *wireless* com a placa, o módulo BCM43438 fornece Wi-Fi 802.11n (possui velocidade de conexão entre 20 e 40 Mb/s), *bluetooth* 4.1 (velocidade de conexão em torno de 25 Mb/s) e o BLE. (GUTERRES, 2017).

O BLE (*Bluetooth Low Energy*) ou *Bluetooth Smart* é uma tecnologia de rede *wireless*, que possui como principal característica um baixo consumo de energia e preço reduzido comparado ao *bluetooth* tradicional. Essa tecnologia foi desenvolvida para aplicações que necessitam enviar informações de tamanho pequeno e de distância de menor alcance para comunicação. É bastante usada para IoT (Internet das Coisas), devido às suas características. (CYPRESS, 2017)

5.5 ARMAZENAMENTO

Nenhum dos modelos do Raspberry Pi possuem um componente que funcione como um disco rígido (HD), tanto o sistema operacional como o armazenamento de arquivos devem ser gravados em um cartão de memória micro SD, que é removível.

Na formatação do micro SD deve-se utilizar uma instalação particionada do sistema operacional e do sistema de arquivo, recomenda-se para utilizar o Raspberry um micro SD com espaço de armazenamento de no mínimo de 4GB, para aplicações que utilizem um sistema operacional completo e que necessite de espaço para armazenar os demais dados, também recomenda-se utilizar um cartão de classe 10, por causa da taxa de leitura e gravação, podendo utilizar cartões de classes inferiores, mas a velocidade de execução da aplicação no Raspberry poderá ficar comprometida. (GUTERRES, 2017)

Quando o Raspberry Pi está conectado a uma fonte de alimentação, é executado um programa de inicialização chamado de *bootloader*, que lê o código

especial que está gravado no micro SD, utilizado para iniciar o sistema operacional no Raspberry Pi. Se não houver nenhum cartão SD inserido, o Raspberry não inicializará.

Pelo fato do cartão SD que possui a instalação do sistema ser facilmente removível, pode-se ter vários cartões, cada um com algum Sistema Operacional ou aplicação diferente, bastando apenas fazer a troca do mesmo para executar uma aplicação ou sistema operacional diferente.

6. SISTEMA OPERACIONAL

Neste capítulo é apresentada algumas informações sobre como devem ser os sistemas operacionais que podem ser utilizados pelo Raspberry Pi. Em 6.1 alguns exemplos de sistemas baseados em Linux que existem. Em 6.2 é apresentado o Raspberry Pi OS, sistema oficial recomendado pela Raspberry Pi Foundation. E em 6.3 é explicado como fazer a instalação do sistema em um cartão micro SD e deixá-lo pronto para ser utilizado.

6.1 DISTRIBUIÇÕES

O Raspberry Pi foi projetado para utilizar sistemas operacionais (SO's) baseados com kernel Linux, assim alguns dos SO's tradicionais para *desktop* como Ubuntu, Debian, Fedora e o Arch possuem versões de sistema operacional específicas para o Raspberry Pi. Como o Raspberry Pi é baseado em um *chipset* de dispositivo móvel, ele possui requisitos de software diferentes de um computador de mesa. Além disso os processadores da *Broadcom*, possuem características proprietárias, tendo que utilizar drivers do tipo Binary Blob 4. Os sistemas operacionais devem operar com restrições de memória RAM e espaço de armazenamento. As principais distribuições Linux são: Raspbian, Occidentalis, Arch Linux, Xbian e QtonPi. (GUTERRES, 2017)

Apesar de já existirem algumas outras distribuições não baseadas em Linux que também podem ser instaladas em um Raspberry Pi, o sistema oficial recomendado pela organização é o Raspberry Pi OS, antes chamado de Raspbian. A organização Raspberry Pi recomenda o uso deste sistema e fornece suporte.

6.2 RASPBERRY PI OS

Antigamente chamado de Raspbian o Raspberry Pi OS é um sistema Linux baseado no Debian, especialmente desenvolvido para ser utilizado por usuários de Raspberry. Ele emprega o gerenciador de janelas do Openbox e o Pi Lightweight, acompanhado de vários outros softwares pré instalados.

O Raspberry Pi OS é o sistema operacional recomendado pela Raspberry Foundation. (RASPBERRY, 2018)

6.3 INSTALAÇÃO

Para fazer a instalação do SO primeiramente devemos baixar o arquivo do site da organização. O download pode ser feito de várias formas. Após a escolha do tipo de download e a aquisição do arquivo, devemos extraí-lo para obter o arquivo imagem; Para fazer a configuração do cartão SD (recomenda-se um cartão SD de 8GB) foi usada a ferramenta Balena Etcher, selecionando o arquivo de imagem que foi baixado e de forma simples e intuitiva dentro do aplicativo Balena Etcher direcionando para a saída especificada do cartão SD. O aplicativo formatará a partição e criará um boot para o SO.

Assim está feita a instalação do sistema dentro do cartão SD. Após essas configurações podemos inserir o cartão SD na placa Raspberry Pi no local especificado e ligar. O sistema fará *boot* e iniciará a tela de configurações iniciais.

7. METODOLOGIA

Neste capítulo é falado da metodologia utilizada para o desenvolvimento deste trabalho. Primeiramente é descrito como foi o processo de levantamento de informações que deram suporte a todo o processo de escrita. Em 7.1 é descrito o processo de implementação dos métodos e também do circuito que foram utilizados para fins de teste deste trabalho. Também são apresentados alguns trechos de código.

Aplicar o conhecimento específico adquirido no processo deste trabalho foi de suma importância, pois a cada dia a busca por novas aplicações que suprem necessidades domésticas fica maior. A busca por atender necessidades específicas deste campo faz necessário pôr em prática, várias ideias e dentro dessas ideias inúmeros testes. A partir disso serão usados neste trabalho um *notebook* de uso pessoal, um Raspberry Pi 3B, ambos rodando sistemas baseados em Linux, para fins de teste.

A metodologia utilizada neste trabalho primeiramente correspondeu ao estudo e análise da literatura, buscando conhecer o funcionamento de um Raspberry Pi e seu sistema. Entender como funcionam seus pinos GPIO, fazendo para isso pesquisas e leituras de artigos e periódicos acerca do assunto.

Durante a fase de desenvolvimento, foram implementados vários testes utilizando a linguagem *Python*, com o auxílio do orientador, mediante as reuniões, e discussões, corrigindo e sugerindo modificações quando necessário.

Após as correções e ajustes na implementação do projeto foi elaborada a escrita do trabalho, apresentando a placa Raspberry Pi, suas características e o passo a passo do projeto, desde a instalação do sistema até outros tópicos que deram suporte ao regimento deste trabalho.

7.1 DESENVOLVIMENTO

A primeira parte da aplicação requer o estabelecimento de uma conexão remota. Sendo assim foram criados dois *scripts* para estabelecer uma conexão remota usando a biblioteca *socket*. Um *script* para o lado cliente e outro para o lado

servidor da conexão.

Na figura 3 podemos ver o trecho de código correspondente ao *script* criado para fazer a conexão do lado cliente. Na linha 9 podemos ver que o atributo “c” pede uma conexão ao servidor, que já deve estar esperando por conexões. O cliente deve passar como parâmetro o endereço local do servidor e a porta de comunicação. Para estabelecer uma conexão entre aparelhos distintos deve ser usado o endereço ip local do servidor nos *hosts* dos dois *scripts*. As portas usadas também devem ser as mesmas nos dois lados da comunicação, e não podem estar sendo usadas por outra aplicação. Neste trabalho a aplicação cliente é executada pelo Raspberry Pi.

```
1 import socket
2 import time
3 import string
4
5 HOST = 'localhost'
6 PORT = 5566
7
8 # criando o socket
9 c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 try:
11 # teste de conexão com o servidor
12     c.connect((HOST, PORT))
13 except Exception as erro:
14     print(str(erro))
15 print('Conectado com: \n', HOST)
16 input('Enviar sinal: \n')
```

Figura 3 – Conexão *socket* lado cliente

Fonte: Elaborado pelo autor

Na figura 4 podemos ver o trecho de código correspondente ao *script* criado para estabelecer a conexão remota a partir do lado servidor. Na linha 14 vemos que como foi mencionado, o servidor deve ficar na escuta esperando alguma conexão de um cliente. Neste trabalho o servidor é executado por um terminal executado em um computador.

```
1  import socket
2  import time
3  import string
4
5  HOST = 'localhost'
6  PORT = 5566
7
8  # cria a conexão socket
9  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
11 # associa a porta
12 s.bind((HOST, PORT))
13 # servidor entra na escuta por conexões
14 s.listen()
15 print('Esperando conexoes.. \n')
16
```

Figura 4 – Conexão *socket* servidor

Fonte: Elaborado pelo autor

Após a conexão ser estabelecida entre cliente e servidor, o servidor deve enviar uma mensagem para o cliente, essa mensagem funcionará como um tipo de menu de escolha para o cliente, a partir desta escolha o Raspberry Pi executará a ação que foi designada com o auxílio dos seus pinos de saídas GPIO. A ideia é de que o Raspberry Pi execute a aplicação após o sinal de controle ser recebido através de outro dispositivo presente na mesma rede, estabelecendo uma conexão remota antes.

Neste trabalho a simulação que o Raspberry Pi executa é o acionamento de algum dispositivo doméstico que necessite de uma corrente de energia para o seu funcionamento, uma lâmpada ou um portão elétrico podem ser pensados como exemplos.

Para realizarmos qualquer operação com a GPIO do Raspberry Pi, antes devemos configurá-los para que o hardware entenda como os pinos trabalharão, se serão pinos de entrada ou de saída. Para essas funções devemos primeiramente importar a biblioteca Rpi.GPIO, nesta biblioteca estarão todas as funções necessárias para poder configurar e acionar a GPIO do Raspberry Pi. Na figura 4 é mostrado como é feita a configuração dos pinos. O Raspberry Pi 3B possui 40 pinos configuráveis e todos eles possuem dois valores. O valor de Broadcom(BCM), que

são valores que os pinos carregam mas que não necessariamente estão em ordem ou o valor de *Board*, que são os valores que os pinos recebem em ordem de impressão na placa. Para trabalharmos com eles devemos primeiramente definir como serão ditos esses valores para que o sistema entenda qual pino está sendo utilizado. Na linha 8 da figura 5 está o exemplo, informando o tipo de numeração que será usada no código. Como os pinos servirão para saída de sinal, são setados como pinos do tipo OUT.

```
1 import socket
2 import time
3 import RPi.GPIO as GPIO
4
5 HOST = 'localhost'
6 PORT = 5566
7
8 GPIO.setmode(GPIO.BCM)
9 #Relé1
10 GPIO.setup(18, GPIO.OUT)
```

Figura 5 – Configuração dos pinos GPIO do Raspberry Pi

Fonte: Elaborado pelo autor

Feitas as configurações necessárias para trabalhar com a GPIO do Raspberry Pi, devemos saber que a tensão máxima fornecida por ele é uma tensão de 5 V, o que é pouco para acionar dispositivos domésticos que trabalham com tensões de 110 V ou 220 V. Como necessitamos de uma tensão maior do que o Raspberry fornece foi necessária a utilização de um módulo relé, que é um dispositivo usado para acionar cargas maiores que a carga fornecida pelo dispositivo emissor do sinal. O módulo relé que fará o controle da tensão para o acionamento externo do dispositivo final. Na figura 6 um exemplo de relé de dois canais. Este relé foi o utilizado neste trabalho.

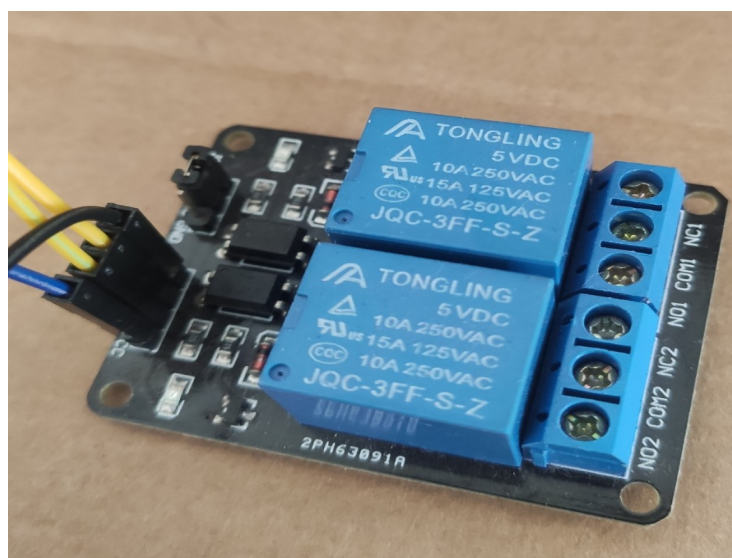


Figura 6 – Módulo relé de 2 canais

Fonte: Elaborado pelo autor

Para a conexão de todos dos componentes podem ser utilizados também uma placa de prototipagem, que permite fazer conexões com mais controle e organização e vários cabos *jumper*, que são cabos usados para fazer a ligação e transmitir o sinal entre os componentes. Na figura 7 podemos ver esses dois componentes.

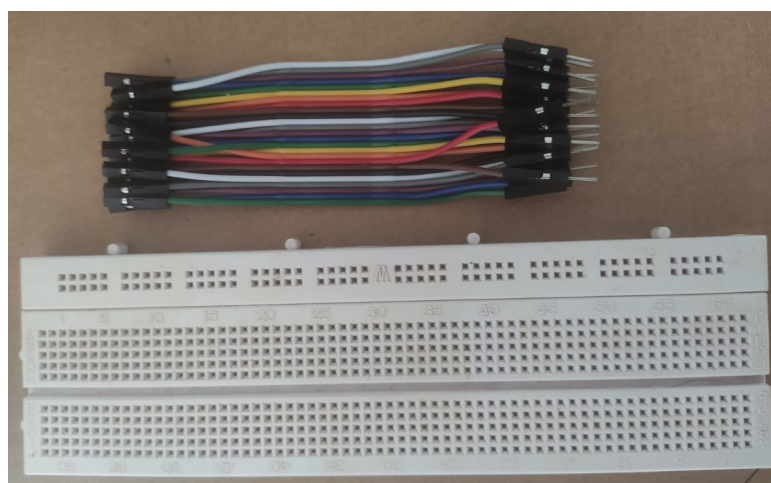


Figura 7 – Placa de prototipagem e cabos *jumper*

Fonte: Elaborado pelo autor

O módulo relé possui dois canais de saída, que podem ser vistos na figura 8, a partir dessas saídas é que será emitido a tensão necessária para acionamento do aparelho.

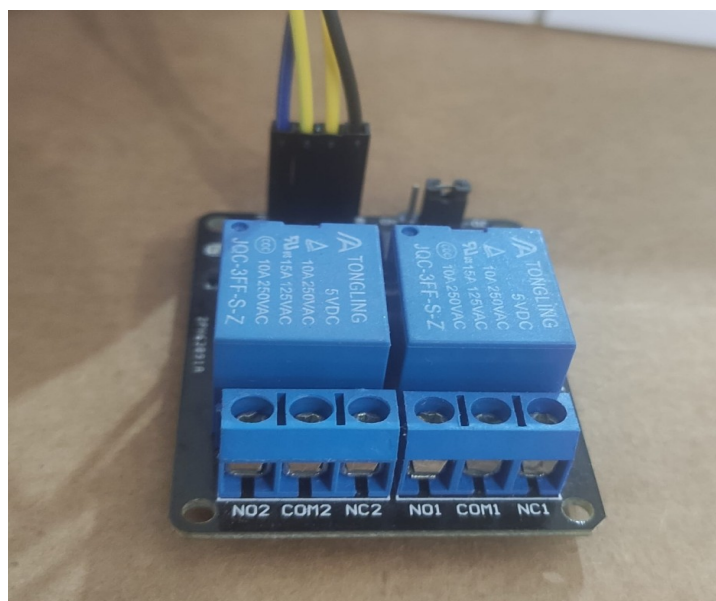


Figura 8 – Saídas do módulo relé

Fonte: Elaborado pelo autor

O módulo relé também possui dois *leds* que servem para sabermos quando está sendo transmitido o sinal. Sendo assim quando o *led* está aceso significa que o circuito está fechado e funcionando. Na figura 9 pode ser visto o lugar onde eles se posicionam dentro do componente.



Figura 9 – Leds de funcionamento do módulo relé

Fonte: Elaborado pelo autor

Na figura 10 pode ser visto a configuração do circuito básico após a montagem e conexão dos componentes que foram utilizados. Para realizar a montagem do circuito básico foram utilizados somente esses componentes descritos. Bastando apenas verificar o acionamento do módulo relé após executar os *scripts* para comprovar o seu funcionamento.

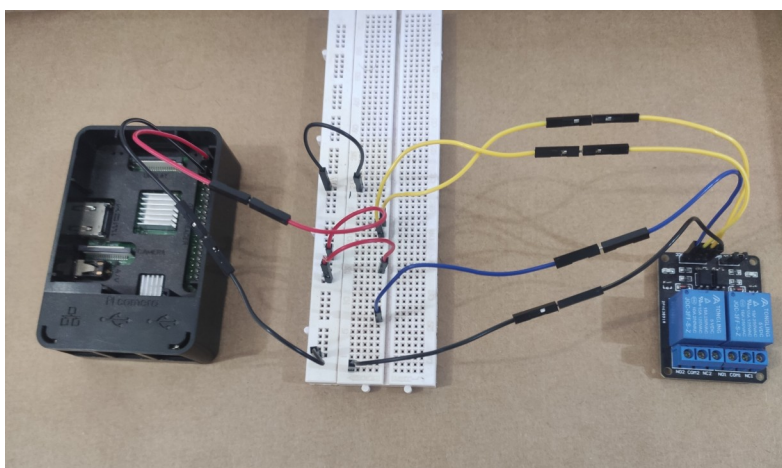


Figura 10 – Circuito utilizado

Fonte: Elaborado pelo autor

Após todos os testes iniciais foi feito também um teste prático, que simula a funcionalidade da aplicação já funcionando em um ambiente residencial. Para o teste prático realizado foi utilizado o exemplo de acionamento de uma lâmpada residencial comum de 110 V. Para isso foi adicionado ao circuito uma luminária.

Já com o relé funcionando através dos primeiros testes bastou fazer a conexão da luminária diretamente nele, sem a necessidade de grandes mudanças na montagem do circuito. Para isso é necessário usar o polo N do canal que será utilizado do relé, que será ligado à um dos polos do soquete da lâmpada e o polo COM, que será ligada a rede elétrica da residência. O polo do soquete que ficou livre também é conectado à rede elétrica junto com o polo COM do relé. Essa configuração pode ser vista na figura 11.

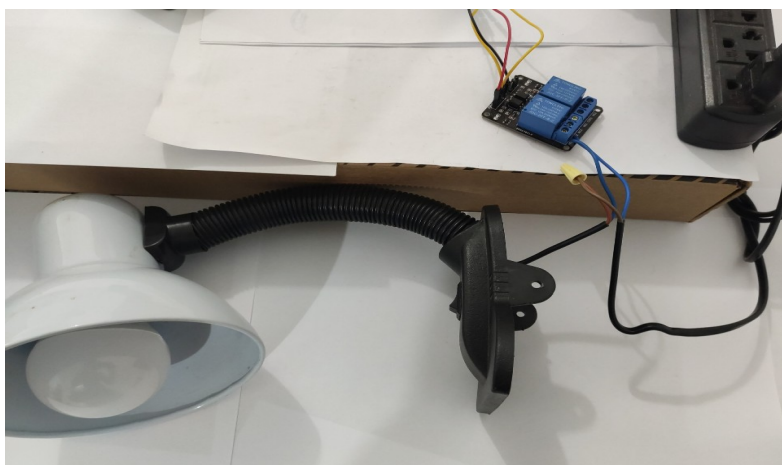


Figura 11 – Circuito do teste prático

Fonte: Elaborado pelo autor

Após fazer a conexão da luminária foi rodada a aplicação. Primeiramente iniciando o servidor no computador que ficou à espera da conexão cliente, que foi executada no Raspberry Pi. Depois de conectado o acionamento do dispositivo é feito a partir de uma resposta do teclado. O vídeo de demonstração pode ser visto no link <https://photos.app.goo.gl/pwnPbwMHAhX66PDk8>.

8. CONCLUSÃO

Hoje vivemos em um mundo onde quase tudo pode ser automatizado, por isso o campo de exploração que este trabalho pode servir de referência é muito grande. A popularização da internet das coisas tornou muito viável oferecer soluções que possuem o intuito de automatizar ou simplesmente facilitar ações a serem executadas, seja dentro de uma residência ou na indústria.

A conclusão deste trabalho atingiu os objetivos iniciais que eram o estudo dos componentes e do Raspberry Pi e após a primeira etapa montar alguns testes que simulam uma aplicação de automação, que pode futuramente ser melhorada. Foi verificada a eficácia do Raspberry Pi em controlar pequenas funções, por exemplo o acionamento de uma lâmpada residencial.

Para sugestões de trabalhos futuros pode ser pensado o desenvolvimento de uma aplicação móvel com interface que gerencie várias ações que são automatizadas e acionadas ou não, ao mesmo tempo pelo Raspberry Pi. Pode-se pensar também em introduzir técnicas de inteligência artificial que disparem o sinal de controle da ferramenta, por exemplo, um portão que destrave após a verificação da imagem da pessoa que está à sua frente. O objetivo destes trabalhos sugeridos seria aprimorar e otimizar a ideia das simulações que foram desenvolvidas neste trabalho.

9. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Hyggo. **Internet das Coisas: Tudo Conectado**. Computação Brasil - Revista da Sociedade Brasileira de Computação, 7-8. Disponível em: <https://www.sbc.org.br/images/flippingbook/computacaobrasil/computa_29_pdf/comp_brasil_2015_4.pdf> Acesso em: 10/08/2021.

ANDRADE, Bruno. **10 sistemas operacionais que você pode executar no Raspberry Pi em 2019**. 2019. Disponível em. <[10 sistemas operacionais que você pode executar no Raspberry Pi em 2019 – GNU/LINUX BRASIL \(gnulinuxbrasil.com.br\)](http://gnulinuxbrasil.com.br/10-sistemas-operacionais-que-voc%C3%AA-pode-executar-no-raspberry-pi-em-2019/)> Acesso em 20/07/2021.

ARAÚJO, Thayron. **Raspberry Pi B+: Introdução a Porta GPIO**. [S.l.]: Fazedores, 2014. Disponível em. <<http://blog.fazedores.com/raspberry-pi-b-introducao-porta-gpio/>>. Acesso em 18/07/2021.

ARM, D. **Cortex-A53**. 2017. Disponível em: <<https://developer.arm.com/ip-products/processors/cortex-a/cortex-a53>> Acesso em 26/07/2018.

CHASE, Otávio. **Sistemas Embarcados**. www.sbajovem.org.

CRUZ, Ariadne Arrais; LISBOA, Emerson Fausto. **WebHome: Automação residencial utilizando Raspberry PI**. Revista Ciência e Tecnologia. 2014. Disponível em: <<http://www.revista.unisal.br/sj/index.php/123/article/view/365>>. Acesso em: 25/07/2021.

CUNHA, A, F. **O que são Sistemas Embarcados**. 2007. Disponível em: <https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf> Acesso em: 20/07/2021.

EMBARCADOS, E. **Sistema Embarcado: O que é? Qual a sua importância?** 2013. Disponível em: <http://www.embarcados.com.br/sistema-embarcado/>>. Acesso em 20/07/2021.

FAZEDORES. **Nova Raspberry Pi 3B+.** Disponível em: <<https://blog.fazedores.com/nova-raspberry-pi-3-b-plus/>> Acesso em: 12/07/2021.

FIORIN, João Pedro; ANTONELLO Ricardo. **Domótica: Desenvolvimento De Protótipo Para Automação Residencial Com Smartphone Android E Raspberry Pi.** 2017. Disponível em: <<https://secitec.luzerna.ifc.edu.br/wp-content/uploads/sites/30/2018/04/Secitec-Artigo-FIORIN.pdf>>

GARCIA, D, Fernando. **Introdução aos Sistemas Embarcados e microcontroladores.** 2018. Disponível em: <<https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>> Acesso em: 20/07/2021.

GARETT, Filipe. **Raspberry Pi: conheça os modelos e saiba qual o mais indicado para você.** 2016. Disponível em: <[Raspberry Pi: conheça os modelos e saiba qual o mais indicado para você | Listas | TechTudo](#) > Acesso em 10/07/2021.

GUTERRES, Felipe. **Projeto de Sistemas Embarcado para Reconhecimento de Palavras Ditas.** 2017. Disponível em: <<https://www.comp.uems.br/~ricardo/PFCs/PFC%20183.pdf>> Acesso em: 05/08/2017.

HEIN, W. **Raspberry pi aplicado a projetos do mundo real.** Linux Magazine. 2013. Disponível em: <[fetch.php](#) > Acesso em 16/07/2021.

KINDERMANN, M, L. **Sistema de acionamento microcontrolado para dispositivos eletrônicos.** 2010. Disponível em: <[SISTEMA DE ACIONAMENTO MICROCONTROLADO PARA DISPOSITIVOS ELETROELETRÔNICOS - CORE](#) > Acesso em: 28/07/2021.

LUSEMBO, L, Plamedi. Protótipo de sistema de automação residencial integrado com Raspberry Pi utilizando Windows. 2017. Disponível em: <http://dsc.inf.furb.br/arquivos/tccs/monografias/2017_2_plamedi-luzolo-lusembo_monografia.pdf> Acesso em: 01/08/2021.

MANOLA, Renan. **Unidade Central de Processamento**. 2010. Disponível em: <<https://docplayer.com.br/2082375-Unidade-central-de-processamento-cpu-processor-renan-manola-introducao-ao-computador-2010-01.html>> Acesso em: 16/08/2021.

MONQUEIRO, B, C, J. **Entendendo os sistemas embarcados**. 2007. Disponível em: <<https://www.hardware.com.br/artigos/entendendo-sistemas-embarcados/>>. Acesso em: 20/07/2021.

NOLETO, Cairo. **Sistemas embarcados: o que são, características e exemplos de aplicação!** 2020. Disponível em: <<https://blog.betrybe.com/tecnologia/sistemas-embarcados/>> Acesso em: 20/08/2021.

PETERSON, I. L. and DAVIE, B. S. **Computer Networks, Fifth Edition: A Systems Approach**. 2011. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition.

PORTUGAL, C. R. P. **Raspberry Pi**. 2017. Disponível em: <<https://www.raspberrypiportugal.pt/raspberry-pi/>> Acesso em: 11/07/2021.

PROTOTIPANDO. **O que é um Raspberry Pi?** 2020. Disponível em: <<https://labprototipando.com.br/2020/05/22/o-que-e-um-raspberry-pi/>> Acesso em 14/07/2021.

RASPBERRY. **Raspberry Pi OS**. 2018. Disponível em: <<https://www.raspberrypi.com/software/>> Acesso em: 25/07/2021.

RODRIGUES, E. C. **Raspberry Pi 3 – Arquitetura e Organização de Computadores I**. 2018. Disponível em: <<https://pt.slideshare.net/edwildson/raspberry-pi-3-arquitetura-e-organizacao-de-computadores-i>> Acesso em: 16/07/2021.

SANTOS, P, Bruno et al. **Internet das Coisas: da Teoria à Prática**. 2016. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>> Acesso em: 20/07/2021.

SCHULTZ, Felix. **O que é Internet das Coisas (IoT) e como funciona?** Disponível em: <<https://blog.milvus.com.br/internet-das-coisas-iot/>>. Acesso em: 25/09/2021

SILVA, Guilherme et al. **Domótica: Controle, Monitoramento E Conforto Utilizando Raspberry Pi**. Revista Científica Semana Acadêmica. 2018. Disponível em: <<https://semanaacademica.com.br/artigo/domotica-controle-monitoramento-e-conforto-utilizando-raspberry-pi-0>> Acesso em 21/07/2021.

SIGNIFICADOS, **Significado de CPU**. 2013. Disponível em: <<https://www.significados.com.br/cpu/>> Acesso em 21/07/2021.

TANENBAUM, S, Andrew. (2002). **Computer Networks**. Prentice Hall Professional Technical Reference, 5th edition.

APÊNDICE A - *Script* do arquivo cliente.py

```
import socket
import time
import RPi.GPIO as GPIO

HOST = '192.168.100.8'
PORT = 5566

GPIO.setwarnings (False)
GPIO.setmode(GPIO.BCM)
#Relé1
GPIO.setup(18, GPIO.OUT)
#Relé2
GPIO.setup(24, GPIO.OUT)

# criando o socket
c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
# teste de conexão com o servidor
    c.connect((HOST, PORT))
except Exception as erro:
    print(str(erro))

print('Conectado com: ', HOST)
#Recebe informações disponíveis
msg = "MENU:"
c.send(msg.encode())

while True:
# testa msg recebida
    try:
```

```
        info = c.recv(1024).decode()
except Exception as err:
    break
# Decodifica mensagem em UTF-8:
if (info == '1'):
    GPIO.output(18, GPIO.HIGH)
    print("Relé1 ON")
    time.sleep(1)
    # GPIO.output(24, GPIO.HIGH)
    # print("Relé2 ON")
    # time.sleep(1)

    GPIO.output(18, GPIO.LOW)
    print("Relé1 OFF")
    time.sleep(1)
    # GPIO.output(24, GPIO.LOW)
    # print("Relé2 OFF")
    # time.sleep(1)

if (info == '0'):
    print("Encerrando ...")
    break

GPIO.cleanup()
```

APÊNDICE B - *Script* do arquivo servidor.py

```
import socket

HOST = '192.168.100.8'
PORT = 5566

# cria a conexão socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
# associa a porta
s.bind((HOST, PORT))
# servidor entra na escuta por conexões
s.listen()
print('Esperando conexoes.. \n')

# aceita alguma conexão
s_cliente, address = s.accept()
print ("Conectado em: ", address)

msg = s_cliente.recv(1024)
print(msg.decode('utf-8'))

while True:

    opt = input("Acionar (1)\nSair (0)\n")
    s_cliente.send(opt.encode('utf-8')) #Envia opção escolhida
    if(opt == '0'):
        print("Encerrando ..")
        break
```