

AUTOMAÇÃO RESIDENCIAL COM ARDUÍNO: UTILIZANDO BLUETOOTH E ANDROID

Dayvison Domingos da Silva
Marcelo Fernandes de Sousa
Humberto Barros de Alencar Junior

 **iesp**
faculdades

ISBN: 978-85-5597-099-3

**Automação Residencial com Arduíno: Utilizando Bluetooth e
Arduíno**

**Dayvison Domingos da Silva
Marcelo Fernandes de Souza
Humberto Barros de Alencar Júnior**
(Autores)

Instituto de Educação Superior da Paraíba - IESP

Cabedelo
2019



INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAÍBA – IESP

Diretora Geral

Érika Marques de Almeida Lima Cavalcanti

Diretora Acadêmica

Iany Cavalcanti da Silva Barros

Diretor Administrativo/Financeiro

Richard Euler Dantas de Souza

Editores

Cícero de Sousa Lacerda

Hercilio de Medeiros Sousa

Jeane Odete Freire Cavalcante

Josemary Marcionila Freire Rodrigues de Carvalho Rocha

Corpo editorial

Antônio de Sousa Sobrinho – Letras

Daniel Vitor da Silveira da Costa – Publicidade e Propaganda

Hercilio de Medeiros Sousa – Computação

José Carlos Ferreira da Luz – Direito

Marcelle Afonso Chaves Sodré – Administração

Maria da Penha de Lima Coutinho – Psicologia

Rafaela Barbosa Dantas – Fisioterapia

Rogério Márcio Luckwu dos Santos – Educação Física

Thiago Bizerra Fideles – Engenharia de Materiais

Thiago de Andrade Marinho – Mídias Digitais

Thyago Henriques de Oliveira Madruga Freire – Ciências Contábeis

Copyright © 2019 – Editora IESP

É proibida a reprodução total ou parcial, de qualquer forma ou por qualquer meio. A violação dos direitos autorais (Lei nº 9.610/1998) é crime estabelecido no artigo 184 do Código Penal.

O conteúdo desta publicação é de inteira responsabilidade do(os) autor(es).

**Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Padre Joaquim Colaço Dourado (IESP)**

S586a

Automação residencial com arduino: utilizando bluetooth e android [recurso eletrônico] / Dayvison Domingos da Silva, Marcelo Fernandes de Souza, Humberto Barros de Alencar Junior. - Cabedelo, PB: Editora IESP, 2019.

72 p.

Tipo de Suporte: E-book

Modo de Acesso: Digital via página web

ISBN: 978-85-5597-099-3

1. Automação residencial. 2. Sistemas - Computação. 3. Arduínos. 4. Computação. I. Sousa, Marcelo Fernandes de. II. Alencar Junior, Humberto Barros de. III. Título.

CDU: 004

Bibliotecária: Angélica Maria Lopes Silva – CRB-15/023

Editora IESP

Rodovia BR 230, Km 14, s/n,
Bloco Central - 2 andar - COOPERE
Morada Nova. Cabedelo - PB.
CEP 58109-303

SUMÁRIO

1. INTRODUÇÃO	12
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	15
1.3 JUSTIFICATIVA	15
1.4 METODOLOGIA.....	17
2. FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 AUTOMAÇÃO RESIDENCIAL	18
2.1.1 Princípios da Domótica.....	21
2.1.2 Classificação da Domótica	22
2.1.3 Arquitetura da Domótica.....	23
2.2 CONTROLADOR LÓGICO PROGRAMÁVEL.....	24
2.3 ENTENDENDO A PLATAFORMA DE PROTOTIPAGEM ELETRÔNICA ARDUINO.....	25
2.3.1 Breve histórico do Arduino	25
2.3.2 O que é o Arduino ?	27
2.3.3 Sinal Analógico e Digital do Arduino	28
2.3.3.1 Modulação por largura de pulso (PWM)	29
2.3.4 Modelos Clássicos do Arduino	29
2.3.4.1 Arduino Uno	30
2.3.4.2 Arduino Duemilanove	31
2.3.4.3 Arduino Ethernet	32
2.3.4.4 Lilypad Arduino.....	33
2.3.4.5 Arduino Mega	34
2.3.5 Ambiente de desenvolvimento do Arduino	36
2.3.6 Vantagens e desvantagens em se usar Arduino	39
2.4 ARQUITETURA E FUNCIONAMENTO DA PROTOBOARD	41
2.5 RESISTORES E A LEI DE OHM	42
3. DESENVOLVIMENTO DO PROJETO	45
3.1 PLATAFORMA DE DESENVOLVIMENTO MIT APP INVENTOR 2 BETA	46
3.2 DESCRIÇÃO DO APLICATIVO ANDROID LARDUINO PRIME	48
3.2.1 Visão Geral dos Utilitários e Blocos de Código do Aplicativo	50

3.2.2 Detalhamento da Implementação do Aplicativo Larduino Prime	52
3.3 ARQUITETURA DO CIRCUITO ELETRÔNICO	54
3.3.1 Módulo Bluetooth	55
3.3.2 Módulo Relé	57
3.3.3 Módulo Rfid Mfrc522	59
3.3.4 Sensor de Temperatura Termistor NTC	61
3.3.5 Resistor LDR	62
3.3.6 Display LCD 16x2.....	63
3.3.7 Sensor de Gás MQ-05.....	64
3.4 DESCRIÇÃO DO SKETCH DESENVOLVIDO PARA O PROJETO.....	65
3.5 TESTE E RESULTADOS	69
4. CONCLUSÃO	71
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	72

1. INTRODUÇÃO

Sabemos que os projetos fazem parte do dia-a-dia de distintas Organizações, privadas e públicas, dos mais variados segmentos de atuação. Desde as épocas mais remotas, ainda que sem os artifícios e as metodologias atuais, as pessoas se preocupavam no controle de materiais, custos, recursos, prazos, bem como, na avaliação dos riscos.

Com o advento da tecnologia, o homem começou a criar e a desenvolver novas ferramentas que lhe proporcionassem melhorias, no que diz respeito à qualidade de vida, trazendo para o ambiente residencial todo conforto e controle que a tecnologia pode proporcionar. Sistemas que anteriormente eram utilizados com exclusividade nos ambientes corporativos das indústrias e comércio, passaram a ser projetados e utilizados também nos ambientes domésticos, através da integração e supervisão de equipamentos concentrados em um único ponto com o fito de facilitar, melhorar e simplificar a qualidade de vida das pessoas, através do gerenciamento técnico, conforto, economia, prevenção de acidentes, falhas de equipamentos, e também segurança aos usuários, das diversas ações realizadas no cotidiano, que de tão frequentes nos acostumamos, mas que ainda nos tomam tempo e energia.

O gerenciamento do consumo de energia e água, os controles de iluminação, acesso, climatização, comunicação, informática etc, integrados e comandados por um sistema de automação, tem demonstrado a possibilidade de tornar o ambiente de trabalho do edifício mais produtivo, saudável e eficiente. Esses benefícios, que, contribuem para o aumento da produção, para assim reduzir os custos operacionais, reflete-se em resultados financeiros, razão pela qual a tecnologia da automação foi incorporada, com mais facilidade, às edificações industriais e comerciais que às habitacionais (DIAS, 2004, p. 28).

Segundo (BOLZANI, 2004), a automação residencial pode ser definida como um conjunto de tecnologias que ajudam na gestão e execução de tarefas domésticas cotidianas. A sua utilização tem por objetivo proporcionar um maior nível de conforto, comodidade e segurança além de um menor e mais racional consumo de energia.

Entretanto, para (WORTMEYER; FREITAS; CARDOSO, 2005) automação residencial representa o emprego de tecnologias ao ambiente doméstico (incluindo residências, condomínios, hotéis), com o objetivo de propiciar conforto, praticidade,

produtividade, economia, eficiência e rentabilidade, com valorização da imagem do empreendimento e de seus usuários.

Por outro lado, podemos deduzir de maneira sucinta que a automação residencial busca resolver problemas de ordem funcional, tal como desligar e ligar luzes, utilizar sensores para acionar dispositivos, fechar e abrir portas e etc., ou seja, objetiva integrar equipamentos e serviços de modo em que eles fiquem centralizados em um “sistema inteligente” e programável, através do qual sejam possíveis o controle e supervisão de diversas tarefas de modo automático.

Outrossim, ao longo do tempo, a computação vem ajudando pessoas comuns a fazerem seus trabalhos de forma cada vez mais rápida e eficiente. Com o advento da automação, tarefas repetitivas podem ser realizadas por máquinas.

Segundo Dias, 2004, a domótica, por meio de seu conjunto multidisciplinar de aplicações, bem integrada às residências, é capaz de aumentar a qualidade de vida de quem nelas habita.

O neologismo domótica resultou da junção da palavra latina “Domus” (casa) com “Robótica” (controle automatizado), sendo este usado pela primeira vez na França por volta de 1980.

Depois de o público conhecer uma residência automatizada, não haverá como retroceder, toda a cadeia de concepção da moradia, (a arquitetura construção etc.), evoluirá, e, principalmente, o ocupante do imóvel. Assim, deverão ser necessários vários profissionais que, interagindo, permitirão o real desenvolvimento das técnicas da domótica (WERNECK, 1999, p. 132).

É de praxe o emprego das palavras automação residencial, automação predial e domótica, como sendo sinônimos, já que o sistema domótico é composto de uma rede de comunicação que permite a interconexão de uma variedade de dispositivos, equipamentos e outros sistemas, com o fito de obter informações sobre o ambiente residencial e o meio em que ele se insere, e efetuando determinadas ações a fim de supervisioná-lo ou gerenciá-lo.

“Em meio à correria é muito bom contar com a ajuda dos recursos tecnológicos e ainda poder ter uma casa mais bonita e valorizada” (MEDRADO, 2008).

Todavia, apenas uma pequena parcela da sociedade mundial faz uso de sistemas domóticos, pois seus entraves quanto a sua aplicação ocorrem

principalmente devido à falta de conhecimento e o custo da tecnologia usada, que ainda é considerado demasiadamente alto.

Neste Trabalho de Conclusão de Curso buscamos apresenta os principais aspectos da Automação Residencial, tendo como foco a descrição de seus elementos básicos, tais como atuadores, controladores, barramentos, *interfaces* e sensores. Procuramos ainda levar em conta os ambientes de aplicação, ressaltando os benefícios que uma residência automatizada pode proporcionar, como economia, segurança, praticidade e conforto, bem como, os aspectos proeminentes em um estudo de caso de residência automatizada hipotética com um esquema de infraestrutura de comunicação para suportar os diversos elementos da automação residencial.

Para tanto, visando implementar a automação residencial do projeto, tomamos como base a plataforma de prototipagem eletrônica *open-source* Aduino Mega (ATmega2560), um *smartphone* com sistema operacional *Android*, um módulo relé com 8 canais 5v e um o módulo *Bluetooth* Rs232 3.3v para fazer o pareamento entre microcontrolador e o *smartphone*.

Destarte, para o gerenciamento dos dispositivos estabelecidos no projeto da automação residencial, desenvolvemos o aplicativo *Larduino Prime* através da plataforma *App Inventor 2 Beta*, que poderá ser encontrada no site <http://appinventor.mit.edu/explore/teach.html>, que nada mais é do que uma ferramenta desenvolvida pela Google que permite a criação de aplicativos para *smartphones* e *tablets* que rodam o sistema operacional *Android*, sem que seja necessário conhecimento em programação, ele é responsável por enviar pelo *smartphone* as requisições ao *bluetooth* que são processadas na placa do Arduino Mega, assim ativando e desativando os atuadores e sensores ligados a ele, por meio de comandos remotos disponíveis em uma Tela *Home*, sem a necessidade de locomoção do operador.

1.1 OBJETIVO GERAL

Propor a aplicação da automação residencial, tomando como base o microcontrolador Aduino Mega (ATmega2560), como instrumento de facilitação e simplificação da vida cotidiana de pessoas portadoras de paraplegia ou não, valendo-se da integração do *smartphone* via *bluetooth*, e uso simultâneo da eletricidade, eletrônica e informática, para assim ativar e desativar os atuadores e sensores ligados a ele.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar Curso profissionalizante de Robótica com Arduino;
- Realizar pesquisa bibliográfica, para fundamentação do trabalho;
- Elaborar e gerenciar um projeto que represente um ambiente residencial automatizado acessível ao público.
- Identificar os sensores e atuadores compatíveis com o microcontrolador Aduino Mega (ATmega2560), bem como suas utilizações e complexidades;
- Elaborar um aplicativo *Android* para o *smartphone* e desenvolver a programação da plataforma de prototipagem eletrônica Arduino;
- Descrever as etapas e processos relacionados ao funcionamento, aplicação e função do aplicativo;
- Construir uma maquete de demonstração;
- Demonstrar a aplicabilidade do Projeto;

1.3 JUSTIFICATIVA

No desenvolvimento deste projeto, procuramos estudar a área não só pela sua contribuição acadêmica, mas principalmente pela sua contribuição social, uma vez que a automação residencial poderá ser considerada uma grande ferramenta de inclusão social, no que diz respeito à acessibilidade e pelo apoio tecnológico da autonomia antes não alcançada.

Não obstante a Automação Residencial, doravante AR, não deve ser vista exclusivamente como artigo de luxo, disponível apenas para poucos segmentos da sociedade abastada. Destarte, como qualquer inovação, a automação residencial, inicialmente, é tida como sinônimo de status e modernidade. No entanto, o conforto, a conveniência e a acessibilidade por ela proporcionados passam a ser decisivos.

De acordo com a Associação Brasileira de Automação Residencial (AURESIDE), o custo da implantação do sistema da esperada “casa do futuro” já é algo próximo da realidade, isso porque um sistema de automação custa, em média, entre 1% e 10% do valor da obra, em contrapartida, essas cifras delimitam o mercado a pessoas com maior poder aquisitivo. Na verdade, a partir de R\$ 6 mil reais é possível incrementar os ambientes com alguns recursos do gênero, como o controle automático de iluminação e de persianas, por exemplo. Mas se o orçamento permitir, um projeto sofisticado pode atingir a valores de até R\$ 100 mil, quantia que garante mordomias hollywoodianas, de acordo com especialistas.

Ao analisarmos os métodos já existentes para implementar a AR, constatamos que estes além de propiciar conforto facilitam a vida das pessoas, entretanto, sua aquisição apresenta custos elevados.

Ainda que as maiorias dos segmentos da sociedade não consigam assimilar essas novas tecnologias, ou mesmo não às enxerguem como uma ostentação desnecessária é notório o crescimento no mercado quando nos referimos a seu emprego, pois sua aludida modernização propicia o aumento de segurança, o bem estar social e ainda que pareça um contrassenso, a redução dos custos.

Entretanto, já existem tecnologias (controladores, atuadores e sensores) que poderiam ser implementadas nestas residências com um custo bem mais baixo que as existentes, o mercado tem aumentado gradativamente o lançamento e uso de novos dispositivos de automação nas residências, como uma das ferramentas que, além de personalizarem o ambiente, torna-os mais atrativos, confortáveis e acessíveis a todos os segmentos da sociedade, com isso, a metodologia proposta poderia fazer mudanças significativas na vida das pessoas.

Apesar de ser uma área com iminente sucesso, ainda apresenta escassez de material bibliográfico, estudos na área facilitariam principalmente a vida de pessoas com necessidades especiais, como a paraplegia, propiciando aos mesmos mais qualidade a seus modos de viver.

1.4 METODOLOGIA

Como toda pesquisa implica o levantamento de dados de variadas fontes, iremos realizar pesquisas de campo e laboratoriais, através das técnicas de observação direta intensiva e de observação direta extensiva. Todavia, serão realizadas ainda pesquisas documentais e bibliográficas com o fito de fundamentar o projeto, através de livros, revistas e artigos inerentes ao mote.

Entretanto, para simular uma residência, será criada uma maquete nos moldes de uma casa real com todo circuito eletrônico em uma *protoboard*, permitindo a montagem de circuitos apenas com o encaixe dos componentes sem a necessidade de soldagem, e em seguida a ligação desse circuito na placa do Arduino com o módulo relé 8 canais 5v e *bluetooth*, cujos sensores e atuadores passarão a ser gerenciados através do aplicativo *Android* por nós desenvolvido (*Larduino Prime*), que será instalado no *smartphone*, o mesmo enviará as requisições ao Arduino Mega (ATmega2560) via *bluetooth*, no qual processará a requisição e executará a função. Nesse contexto, o sistema apresenta uma *interface* com um design amigável, intuitivo e simples, facilitando assim a vida do usuário.

Contudo, propusemos a plataforma *MIT App Inventor* para construção do referenciado aplicativo por este além de possuir uma *interface* simples e fácil de usar, o programa foge das linhas de programação normal, tudo graças ao recurso *drag and drop*, ou seja, a programação das aplicações acontece de forma simples e intuitiva o que possibilita até mesmo usuários comuns de lançarem seus aplicativos, não se fazendo necessário instalar nada no computador, pois a aplicação funciona de modo inteiramente *online*, diretamente de seu navegador. Para utilizá-la, basta ter uma conta de usuário da Google.

Para tanto, estudaremos os princípios básicos do Arduino e da plataforma sugerida, em seguida será feita a codificação e testes do microcontrolador com o módulo *bluetooth*, bem como a construção do aplicativo, visando assim a interação entre os citados com o *smartphone*.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 AUTOMAÇÃO RESIDENCIAL

De acordo com a história, o conceito de Automação surgiu por volta de 3500 a 3200 A.C., com a invenção da roda, tendo como intuito facilitar e agilizar o execução das atividades cotidianas. Relata (SILVEIRA; SANTOS, 1998), que o objetivo deste termo é sempre o mesmo substituir o trabalho braçal por outros meios “mais inteligentes”, liberando o tempo perdido para outras atividades, de intelecto, das artes e lazer.

Segundo Camargo, 2002 a palavra automação faz alusão à utilização de sistemas automáticos e equipamentos, particularmente os sistemas de fabricação ou sistemas de processamento de dados, que exigem pouca ou nenhuma intervenção humana em suas operações normais.

Entretanto, para (MORAES; CASTRUCCI, 2001), a expressão Automação do inglês *Automation*, tem seus primórdios na Indústria por volta da década de 60, cuja finalidade estava relacionada a essência dos computadores na linha de produção dos produtos.

Assim sendo, podemos deduzir que com o passar do tempo à definição da referenciada palavra sofreu mudanças, pois tanto os *hardwares* como os *softwares* se tornaram mais confiáveis e sofisticados, passaram assim a apresentar uma maior capacidade de processamento, bem como uma *interface* homem-máquina (IHM) mais amigável e acessível. Nesse contexto, a automação vem ganhando um novo nicho de atuação com a inclusão das residências no seu raio de ação, permitindo ao usuário operar todas as funções da casa através de um aplicativo desenvolvido para dispositivos Adroid, cujo objetivo é tão somente facilitar o dia a dia das pessoas, buscando reger quase que sempre a melhoria na segurança, conforto, e economia de recursos, antes privativo apenas as grandes indústrias e empresas.

É afirmado por (CANDITO, 2007), que o termo “Domótica” resulta da junção da palavra *Domus* que significa residência com Robótica, sendo atribuída a robótica a responsabilidade de alimentar o sistema com informações que simplifiquem a vida das pessoas no ambiente residencial, buscando assim satisfazer as necessidades básicas, ou seja, é a junção de sistemas informáticos, mecânicos, arquitetônicos,

eletrônicos e de telecomunicações, aplicados a melhorias da segurança, comunicações, gestão energética e conforto.

A Automação Residencial surgiu na França por volta do século XX, o início da implantação e utilização da mesma se deu durante a construção dos primeiros edifícios, bem como, da necessidade de controlar e interligar as funções de climatização, segurança e iluminação. Conforme (MOYA; TEJEDO 2010), essa origem remete-se aos anos 70, quando surgiram os primeiros dispositivos de automação de edifícios, baseados na tecnologia X-10. O X-10 é um protocolo que permite controlar dispositivos através de uma rede elétrica já pré-existente, não se fazendo necessário instalar novos cabos. Em contrapartida, era um sistema de fácil instalação, contudo bastante instável, haja vista que os dispositivos poderiam falhar ou mesmo danificar em decorrências de descargas elétricas ou falta de energia.

Entretanto, conforme versa a Figura 1, por volta dos anos 80 com a chegada dos primeiros computadores pessoais, dotados de interfaces gráficas, possibilitou o surgimento da automação residencial. Contudo, o grande avanço da Automação Residencial foi dado no final da década de 90, quando ocorre uma queda dos preços para aquisição de computadores pessoais, como também a evolução dos Sistemas de Informação, que passaram a utilizar as últimas tecnologias disponíveis para desenvolvimento, tanto de sistemas, quanto de telecomunicações.

Por outro lado, os dispositivos foram evoluindo com o passar dos anos e várias descobertas surgiram. Porém, na contemporaneidade a evolução e aplicação da Automação Residencial ainda se encontram em um estágio embrionário, em decorrência da pouca disseminação dos benefícios que ela pode trazer.

A automação residencial propõe uma alteração da infraestrutura da residência para centralizar os diversos tipos de serviços e de dispositivos que executam tarefas em um único equipamento, o integrador (BOLZANI, 2004). Atualmente a Domótica ou Automação Residencial utiliza esses novos recursos de forma integrada, como por exemplo, em sistemas de controle e monitoramento móvel, através de celulares, tablets, ou via web (OLIVEIRA, 2012).

Todavia é abordado por (INTILLE, 2002), que a palavra tecnologia, em toda sua magnitude, apresenta certo misticismo, pois diversos usuários acreditam que dispositivos assim, tornariam os dia-a-dia mais frustrantes.

Sem embargo, atualmente o conceito de Automação Residencial está mudado, pois temos uma torrente de possibilidades econômicas e práticas que fazem o uso da Automação Residencial, desde a básica até a mais robusta, sem jamais se desfazer do encanto, conforto e valorizando o ambiente.

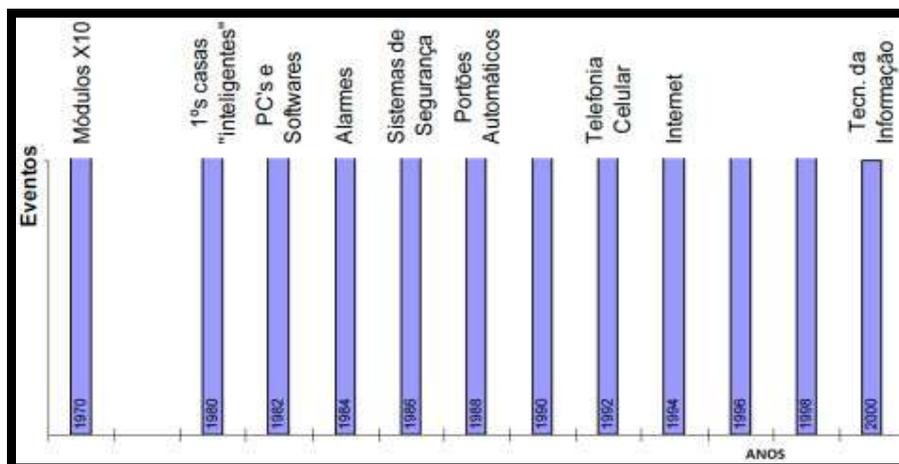


Figura 1: História da Automação Residencial

Fonte: <http://s4.static.brasilecola.com/img/2014/05/figura1.jpg>

Não obstante, para aplicar e implantar o sistema de Automação Residencial se faz necessário a elaboração de um projeto de automação, visando identificar os possíveis pontos eletrônicos no ambiente a ser automatizado, como por exemplo, luzes, internet, sistemas de segurança, portas, portões, janelas e etc., buscando levar em conta a necessidade dos habitantes daquele nicho residencial. Outra nomenclatura utilizada é a Casa Inteligente, o que sugere uma habitação totalmente adaptada ao sistema e programada para auxiliar os moradores nas funções diárias.

Apesar da aplicação dos investimentos inerentes aos recursos da Automação Residencial não ser acessível a todos, com a queda nos preços dos componentes eletrônicos e o surgimento de novas tecnologias paralelas, ocorrerá uma maior aceitação destes por parte dos jovens, possibilitando uma maior divulgação e utilização da AR em sistemas de comunicação, que poderão ser acessados e gerenciados via painéis eletrônicos.

Em suma, a Automação Residencial pode ser abrangida como uma rede que busca controlar e integrar a residência de forma digital, buscando adequar as necessidades das pessoas e aperfeiçoar as funções elétricas, tecnológicas e sustentáveis.

2.1.1 Princípios da Domótica

A Automação residencial, nada mais é do que um anexo de serviços fornecidos por sistemas tecnológicos integrados cujos princípios estão relacionados ao conforto habitacional, segurança, gestão energética e comunicação, para melhor suprir as necessidades de seus usuários, conforme Figura 2.

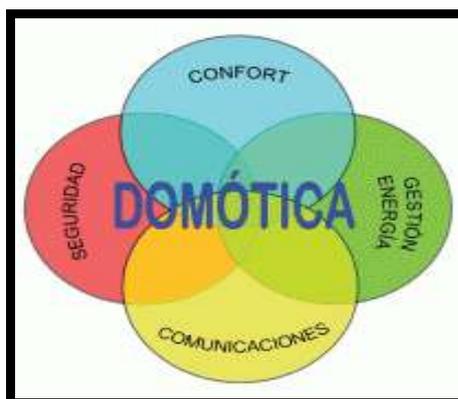


Figura 2: Benefícios da Domótica ao usuário final

Fonte: <http://casadomotica-brasil.wordpress.com/tag/conforto/>

Podemos afirmar que a segurança é um dos principais pilares da automação residencial, tudo isso em decorrência do crescente grau de periculosidade nas mais distintas regiões, preocupados com seus bens, as pessoas buscam sistemas que permitam detectar a invasão de meliantes em seu raio de atuação. Por outro lado, em meio à modernidade, esses sistemas não devem apenas detectar meliantes, mas prevenir e detectar incêndios, inundação, ou mesmo uma possível detonação em um sistema de gás, monitorando tudo via computadores e circuito interno de TV, buscando em caso de sinistro, mandar uma mensagem via *sms* ou *e-mail* para os usuários do sistema.

Entretanto, o conforto que ela propicia deverá certamente ajudar as atividades corriqueiras das pessoas com paraplegia ou não, proporcionando as mesmas uma sensação de comodidade pelo fato de algumas tarefas serem realizadas através da programação de horários, ou até mesmo sem a necessidade de ordens diretas permitindo, por exemplo, o controle de iluminação, regulação automática de temperatura, descentralização de equipamentos e centralização de controle, áudio e vídeo, e controle a distância.

Em se tratando de comunicação, a automação residencial objetiva integrar os subsistemas externos e internos, podendo ser através do acionamento remoto ou mesmo via dispositivos como celulares por intermédio da internet ou *bluetooth*, ou seja, em sua pluralidade os subsistemas e dispositivos podem ser conectados a uma única rede. Por outro lado, em épocas bem remotas estes subsistemas desempenhavam suas funções de forma independente, todavia, com a conseqüente revolução tecnológica e o avanço da domótica, a comunicação entre dispositivos vem sendo realizada por meio de um gerenciador central, que busca integrar os demais sistemas.

Por fim, os Sistemas gerenciadores de consumo de energia estão relacionados com a medição e racionamento de consumo, visando um aproveitamento eficiente e eficaz dos recursos energéticos da residência.

2.1.2 Classificação da Domótica

De acordo com a AURESIDE nos últimos anos o mercado de automação residencial brasileiro, vem apresentando uma crescente de aproximadamente 35% ao ano no que se refere a números de projetos, países da Europa e América do Norte, por exemplo, passaram por uma crise recente o que favoreceu o ambiente econômico nacional, por outro lado, a maioria dos imóveis vendidos recentemente no Brasil é designada a jovens antenados no setor de tecnologia conforme apresentado na Tabela 1.

<i>Tecnologia</i>	<i>2003</i>	<i>2004</i>	<i>2005</i>	<i>2006</i>	<i>2015(*)</i>
Cabeamento estruturado	42%	61%	49%	53%	80%
Monitoramento de segurança	18%	28%	29%	32%	81%
Multirroom audio	9%	12%	15%	16%	86%
Home Theater	9%	8%	11%	12%	86%
Controle de iluminação	1%	2%	6%	8%	75%
Automação integrada	0	2%	6%	6%	70%
Gerenciamento de energia	1%	5%	11%	11%	62%

Tabela 1 – Evolução da adoção de algumas tecnologias

Fonte: National Association of Home Builders (NAHB) <<http://www.nahb.org/>>

(*) Previsão

Segundo (MURATORI, 2004), a Domótica faz o uso de funções que podem estar agregadas a outras, bem como a outros dispositivos agregados ao sistema, o modo como ocorre à execução dessas funções define a classificação da Automação Residencial em sistemas: autônomos, integrados ou complexos:

- **Integrados:** possuem múltiplos subsistemas integrados a um único controlador. A limitação deste sistema está no fato que cada subsistema deve ainda funcionar unicamente na forma a qual seu fabricante pretendia. Basicamente trata-se apenas de controle remoto estendido a diferentes locais (TEZA, 2002, p.32). O processamento pode ser centralizado na central de automação ou distribuído pela rede (TERUEL, 2008, p.28);

- **Complexos:** possuem como grande diferencial a possibilidade da personalização de produtos manufaturados de modo que atenda as necessidades do proprietário. É dependente de comunicação de mão dupla e realimentação de status entre todos os subsistemas (TEZA, 2002, p.33).

- **Autônomos:** possuem funcionalidades somente de acionar ou desligar dispositivo ou subsistemas. Nessa classificação nenhum dispositivo ou subsistema tem relação um com o outro (TERUEL, 2008, p.28);

2.1.3 Arquitetura da Domótica

Nesta seção, apresentaremos uma visão geral e sucinta dos diversos elementos básicos (atuadores, barramentos, controladores, sensores e interfaces) envolvidos na arquitetura da Automação Residencial, que vai de simples sensores até complexas centrais de automação, conforme visualizado na Figura 3.

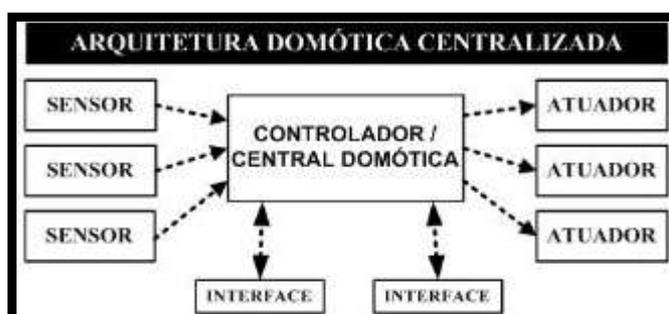


Figura 3: Arquitetura da Automação Residencial

Fonte: <http://www.engenhariae.com.br/tecnologia/o-clp-e-suas-linguagens/>

Os Controladores são dispositivos que controlam os sensores e atuadores, monitorando as informações contidas nos sensores, além disso, o referenciado pode enviar comandos para que um atuador ative ou desative algum equipamento. Nada obstante, os controladores apresentam interfaces independentes, na forma de um controle remoto, ou são sofisticadas centrais de automação.

Por outro lado, os sensores são elementos que buscam detectar excitações no ambiente por meio do monitoramento das grandezas físicas, convertendo-as em um valor passível de manipulação por sistemas computacionais.

São os sensores que encaminham as informações aos controladores sobre algum evento, para que os controladores possam enviar os comandos adequados para os atuadores, explica (ALMEIDA, 2009).

Outrossim, os atuadores são periféricos eletromecânicos de acionamento ligados entre a rede elétrica e os equipamentos, ou seja, são os dispositivos que recebem os comandos do sistema para ativar os equipamentos automatizados.

As Interfaces são os dispositivos ou mecanismos (navegador de internet, *smartphone*, painéis, controles remotos, interruptores etc.) que permitem ao usuário visualizar as informações e interagir com o sistema de automação.

Por fim, conforme (TANENBAUM, 2001), barramentos são estruturas físicas nas placas eletrônicas de controle que servem para fazer a interface de comunicação entre os diversos periféricos que compõem um sistema computacional.

2.2 CONTROLADOR LÓGICO PROGRAMÁVEL

Conforme (HENRIQUE, 2008), o controlador lógico programável (CLP) surgiu dentro da Empresa General Motors, em 1968, devido a uma grande necessidade de controlar painéis de comando a cada mudança na linha de montagem.

Deste modo, conforme referenciado, podemos deduzir que com o passar do tempo à medida que a eletrônica evoluía os dispositivos passavam a contar com sensores, bem como, circuitos ganhavam a capacidade de realizar funções lógico aritméticas, com a obtenção de sinais de entrada, processamento e saída.

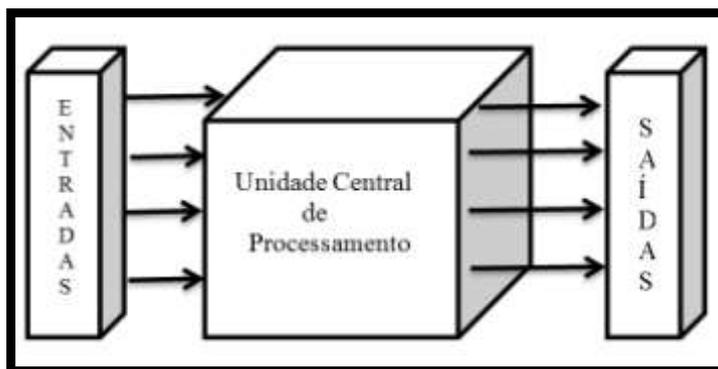


Figura 4: Estrutura Básica de um CLP

Fonte: <http://www.engenhariae.com.br/tecnologia/o-clp-e-suas-linguagens/>

Automatizar uma empresa tornou mais simples, integrando todos os sistemas, assim surgiram os controladores lógicos programáveis (HENRIQUE, 2008).

Toda esta evolução nos levou a sistemas compactos, com alta capacidade de controle, que permitem acionar diversas saídas em função de vários sinais de entrada combinados logicamente. Outra etapa importante desta evolução é que toda a lógica de acionamento pode ser desenvolvida através de software, que determina ao controlador a sequência de acionamento a ser desenvolvida (HENRIQUE, 2008, p.2).

Com base no exposto, bem como na Figura 4, deduzimos que a vantagem de se utilizar um CLP, é que o mesmo além de ser programável, é reutilizável, permitindo a comunicação com outros Controladores Lógicos Programáveis e computadores, e ainda apresenta uma considerável redução de custo de implantação, bem como, uma maior confiabilidade, flexibilidade, rapidez na elaboração de projetos e acima de tudo um menor consumo de energia.

2.3 ENTENDENDO A PLATAFORMA DE PROTOTIPAGEM ELETRÔNICA ARDUINO

2.3.1 Breve histórico do Arduino

Do que se tem registro, o Arduino teve suas origens por volta do ano de 2005 através de Massimo Banzi, ora professor do *Interaction Design Institute*, localizado na cidade de Ivrea - Itália. Este docente buscava uma alternativa barata e de fácil

aprendizagem para estudantes do curso de *design* trabalhar com tecnologia, entretanto encontrava muita dificuldade em decorrência dos alunos não serem da área de informática nem de eletrônica, ou mesmo, à inexistência de placas robustas e de baixo custo no mercado, que viabilizassem essa aprendizagem.

Com base nisso, Massimo Banzi apresentou seu problema ao engenheiro eletrônico David Cuartielles, ora pesquisador visitante da Universidade de Malmö - Suécia, cujo estava à procura de uma solução similar. Decidiram então projetar sua própria placa, um microcontrolador que pudesse ser utilizado pelos seus estudantes de arte e *design* nos mais variados projetos, não obstante, as exigências primordiais para esse microcontrolador eram que o preço almejado não ultrapassasse o valor que um estudante pagaria em uma pizza e que fosse uma plataforma que qualquer usuário pudesse utilizar.

David Cuartielles desenhou a placa, e um aluno de Massimo Banzi, David Mellis, programou o software para executar a placa. Massimo contratou um engenheiro local Gianluca Martinho, que também trabalhou no Design Institute ajudando alunos com seus projetos. Gianluca concordou em produzir uma tiragem inicial de duzentas placas. A nova placa foi chamada de Arduino em referências a um bar local frequentado por membros do corpo docente e alunos do instituto (EVANS; NOBLE; HOCHENBAUM, 2013, p. 25)

Segundo (EVANS; NOBLE; HOCHENBAUM, 2013), “a popularidade do Arduino cresceu em decorrência da placa ser um sistema de fácil utilização, baixo custo e que poderia ser usado em seus próprios projetos, além de ser uma excelente introdução para programação de microcontroladores”.

Por fim, a escola de *design* fechou, entretanto a ideia do Arduino promovida pelos pesquisadores ascendeu. Atualmente o Arduino original é fabricado pela empresa italiana *Smart Projects*. Por outro lado, há algumas placas Arduino fabricadas pela empresa americana *SparkFun Electronics*. Até hoje foram produzidas e comercializadas várias versões do hardware do Arduino.

2.3.2 O que é o Arduino ?

Arduino é uma plataforma de prototipagem eletrônica baseada em uma simples placa de entrada e saída e um ambiente de desenvolvimento que possibilita a escrita de linhas de comando na placa. Não obstante, conforme relatado na Seção 2.3.1, o Arduino foi criado com o fito de ser acessível a todos e admitir o controle e desenvolvimento de sistemas interativos de baixo custo. A placa é constituída de seu próprio ambiente de desenvolvimento, com linguagem de programação baseada em C/C++, conforme será melhor detalhado na Seção 2.3.5.

Entretanto, como em sua essência histórica o microcontrolador Arduino apresenta código e arquitetura abertos, seus algoritmos podem ser reutilizados e a placa pode ser reproduzida sem a necessidade de pagamento de direitos autorais.

Em termos práticos, um Arduino é um pequeno computador que pode ser programado para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele conforme a Figura 5. O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (MCROBERTS, 2013, p. 22).

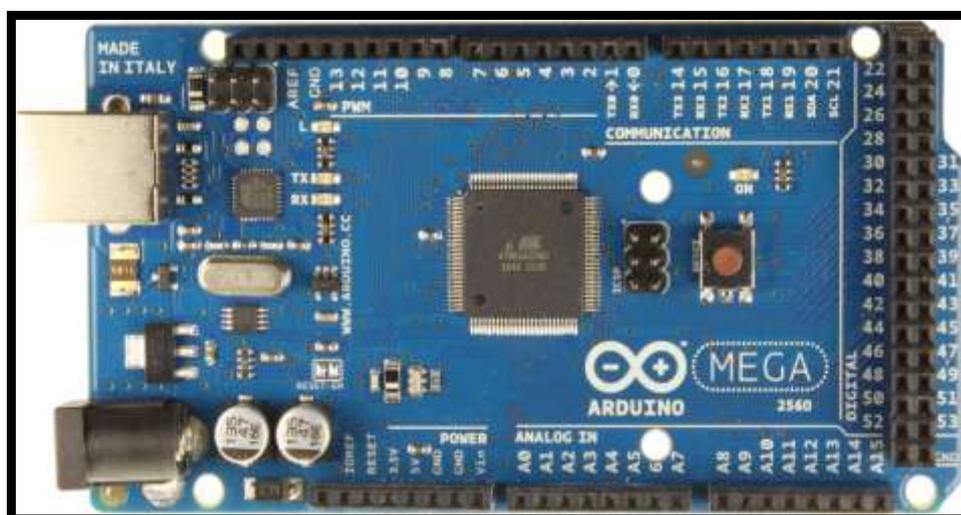


Figura 5: Arduino Mega 2560

Fonte: <http://arduino.cc/en/Main/ArduinoBoardMega2560>

De acordo com (MCROBERTS, 2013), o “Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles”.

2.3.3 Sinal Analógico e Digital do Arduino

Os pinos digitais do Arduino podem funcionar como entrada ou saída de dados, entretanto quando um pino é programado para funcionar como entrada digital ao ser executado, o mesmo efetua a "leitura" da tensão aplicada a ele, na prática, o programa identifica se um pino está alimentado com 0 ou 5 volts.

Não obstante, o pino digital do Arduino é geralmente usado para identificar se um botão está pressionado, ou um sensor está "sentindo" alguma coisa no mundo externo. Com base no exposto, observamos que a função de entrada digital apenas emprega o 0 lógico (sem tensão) ou o 1 lógico (com tensão).

Em contrapartida, através da entrada analógica do Arduino é possível saber se existe ou não uma tensão aplicada em seu pino e sua respectiva medição, valendo-se do uso de sensores que convertem algumas grandezas físicas em um valor de tensão, nos quais são lidos pela entrada analógica.

Então qual a diferença entre os mundos analógico e digital?

No mundo digital, tudo possui dois estados; uma chave pode apenas estar ligada ou desligada, um LED pode estar aceso ou não, você está acordado ou dormindo. Estes estados podem ser descritos de várias maneiras, como uns ou zeros, ligado ou desligado, alto ou baixo. O pino digital do Arduino trabalha da mesma forma; quando ajustado para saída, ele é de 0 ou 5 volts, com a tensão 0 sendo "zero" lógico e 5 volts sendo "um" lógico. No mundo analógico, as coisas possuem um intervalo de valores. As músicas possuem notas que abrangem uma gama de frequências, um carro acelera através de uma gama de velocidades, uma onda senoidal flui suavemente entre os valores máximos e mínimos, e a temperatura entre máxima e mínima. Nós frequentemente queremos explorar o mundo analógico e o Arduino possui seis entradas que podem nos ajudar com isso. Porém o Arduino é ainda um dispositivo digital, então você necessita de um meio para converter o sinal de entrada a uma representação digital. Isso é feito com um conversor analógico-para-digital (ACD) (EVANS; NOBLE; HOCHENBAUM, 2013, p. 70).

No entanto, conforme a Figura 6 apresentada na Seção seguinte é notória que a grande diferença entre estes sinais se dá por conta da captação dos mesmos, o sinal analógico oferece muitas interferências e às vezes pode ser captado uma frequência distinta da desejada, o que é denominado de ruído, por sua vez, o sinal digital é apresenta uma maior robustez em decorrência da lógica digital aplicada, possuidora de apenas duas constantes binárias, o 0 (falso) ou 1 (verdadeiro), desta forma o sinal recebido não sofre uma variação, podendo ser captado ou não.

2.3.3.1 Modulação por largura de pulso (PWM)

A modulação por largura de pulso (*pulse width modulation*), doravante PWM, é uma técnica usada para obter resultados analógicos com o uso de meios digitais. Conforme apresentado na Figura 6 o sinal digital é usado para criar uma onda quadrada, alternando entre ligado (*HIGH*) e desligado (*LOW*), com isso, o sinal digital do Arduino oscila entre 0 e 5 volts. Isso é feito alterando a quantidade de tempo em que a saída permanece alta (*HIGH*) e baixa (*LOW*). A duração do tempo em que ela permanece ligada é conhecida como a largura do pulso.

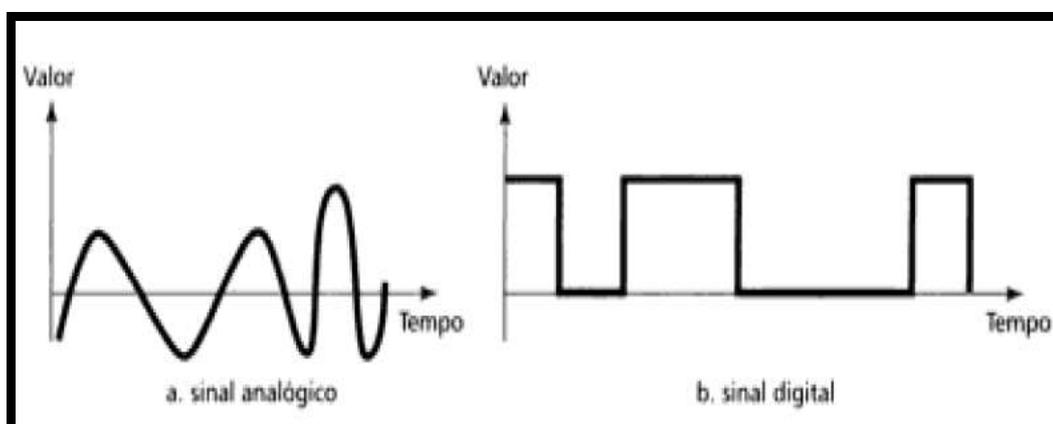


Figura 6: Sinal analógico x Sinal digital

Fonte: http://blogdeumbruno.blogspot.com.br/2013_04_01_archive.html

Em suma, essa operação tem alto grau de importância, por criar uma variação intermediária entre os estados máximo e mínimo, ou seja, é um mecanismo que permite controlar o período cíclico da frequência da alimentação, pois permite obter uma tensão analógica a partir de um sinal digital.

2.3.4 Modelos Clássicos do Arduino

Em se tratando do microcontrolador Arduino, de um modo geral, sua arquitetura é composta de um microcontrolador Atmel AVR, um regulador linear de 5 volts e um cristal ou oscilador responsável por enviar pulsos a uma frequência específica para o Arduino operar na velocidade correta. Contudo, conforme o modelo utilizado à plataforma de prototipagem poderá apresentar uma saída

Universal Serial Bus (USB), na qual permitirá conectá-lo a um *personal computer* (PC) para *upload* ou recuperação dos dados.

Em contrapartida, a mais contemporânea placa do Arduino (Arduino Uno R3), difere das versões antepassadas por não mais fazer o uso do chip FTDI, responsável por fazer a condução da USB para Serial, propiciando a mesma diversas prerrogativas quando comparada a placas antecessoras no que diz respeito a custo e desempenho.

Entretanto, com a adesão do novo chip USB, o processo de fabricação de placas clone do Arduino Uno ficou mais difícil.

Temos até o momento uma série de versões do Arduino, todas baseadas em um microcontrolador de 8 bits Atmel AVR reduced instruction set computer (RISC). A primeira placa foi baseada no ATmega8 rodando a uma velocidade de clock de 16 MHz com memória flash de 8 KB; mais tarde, placas tais como Arduino NG plus e Diecimila (nome italiano para 10.000) usava o ATmega168 com uma memória flash de 16 KB. As versões mais recentes do Arduino, Duemilanove e Uno, usam o ATmega328 com memória flash de 32 KB e podem comutar automaticamente entre Universal Serial Bus (USB) e corrente contínua (DC). Para projetos que exigem mais Entradas/ Saídas e memória, há o Arduino Mega1280, com memória de 128 KB, ou o mais recente Arduino Mega2560, com memória de 256 KB (EVANS; NOBLE; HOCHENBAUM, 2013, p. 26).

Há uma gama de variantes quando nos referimos a modelos de placas Arduino (Uno, Duemilanove, Ethernet, Mega, Lilypad, Nano), dentre os modelos clássicos existentes, podemos destacar o Arduino Uno e Duemilanove, os quais lideram a vasta maioria dos projetos para Arduino na Internet.

Todavia, com a adição do Arduino Mega 2560, o microcontrolador passou a oferecer mais memória e um número maior de pinos de entrada/saída. As novas placas utilizam um novo bootloader, o Optiboot, que libera mais 1,5 KB de memória flash e permite uma inicialização mais rápida.

2.3.4.1 Arduino Uno

O Arduino Uno é um microcontrolador baseado no ATmega328. De acordo com a Figura 7, o mesmo é composto por 14 pinos digitais de entrada / saída, dos quais 6 podem ser usados como saídas *Pulse Width Modulation* (PWM), 6 entradas

analógicas, uma conexão USB, um botão de *reset* e um *jack* de alimentação auto chaveada.

A placa pode operar com o fornecimento de uma tensão externa de 6 a 20 volts. Contudo, se for fornecido menos de 7 V, o pino de 5 V poderá fornecer menos de cinco volts e a placa. Por outro lado, se for usado mais que 12 V, o regulador de voltagem pode superaquecer e danificar a placa, ou seja, o intervalo recomendado é de 7 a 12 volts.

A maior diferença entre o Uno e seus antecessores é a inclusão de um microcontrolador programado Atmega8U2 com um conversor USB-para-serial, substituindo o chip FTDI obsoleto usado nas versões anteriores. O Atmega8U2 pode ser reprogramado para fazer o Arduino se parecer com outro dispositivo USB, tal como mouse, teclado ou joystick. Outra diferença é que ele possui uma tensão integrada de 3,3 V mais confiável, o que ajuda na estabilidade de algumas proteções que causavam problemas no passado (EVANS; NOBLE; HOCHENBAUM, 2013, p. 27).



Figura 7: Arduino Uno R3

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

2.3.4.2 Arduino Duemilanove

O Arduino Duemilanove ("2009") é um microcontrolador baseado no Atmega168 ou ATmega328. De acordo com a Figura 8, tal qual o Arduino Uno, o mesmo é composto por 14 pinos digitais de entrada / saída, dos quais 6 podem ser usados como saídas PWM, 6 entradas analógicas, uma conexão USB, um botão de *reset* e um *jack* de alimentação auto chaveada.

Porém, fundamentado em (EVANS; NOBLE; HOCHENBAUM, 2013), podemos afirmar que caso o usuário seja possuidor de uma versão Duemilanove, considere atualizar para o Uno se for usar uma tensão de 3,3 V mais estável ou mesmo queira fazer uma programação mais avançada com o Atmega8U2.

O Duemilanove se caracteriza pela seleção de potência de comutação automática entre o externo e o USB e usa o processador ATmega328, embora os modelos anteriores a março de 2009 usassem o ATmega168. Seu layout de pinos e capacidade é idêntico ao Uno e usa o chipset FTDI para comunicação USB para Serial (EVANS; NOBLE; HOCHENBAUM, 2013, p. 28).

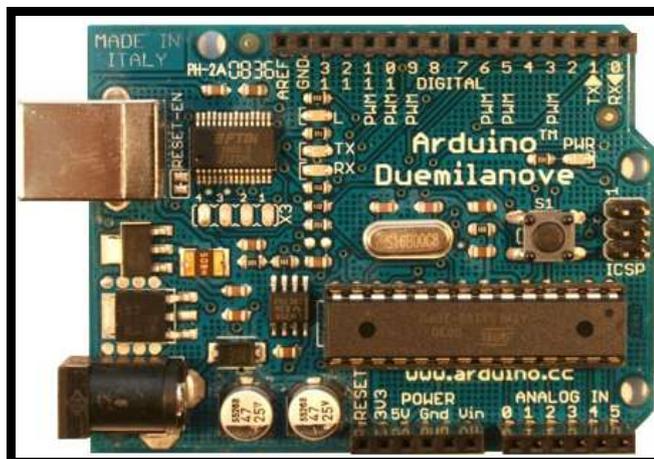


Figura 8: Arduino Duemilanove

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

2.3.4.3 Arduino Ethernet

O Arduino Ethernet é uma placa de microcontrolador baseado no ATmega328. De acordo com a Figura 9 abaixo arrolada, a mesma é composta por 14 pinos digitais de entrada / saída, 6 entradas analógicas, um conector RJ45, um botão de *reset* e um *jack* de alimentação auto chaveada.

Em contrapartida, comparada com as demais placas, os pinos digitais 10, 11, 12 e 13 são reservados para a interface com o módulo Ethernet e por isso não devem ser utilizados para outros fins, reduzindo no número de pinos digitais disponíveis para 9, dentre os quais quatro podem ser usados como saídas PWM.

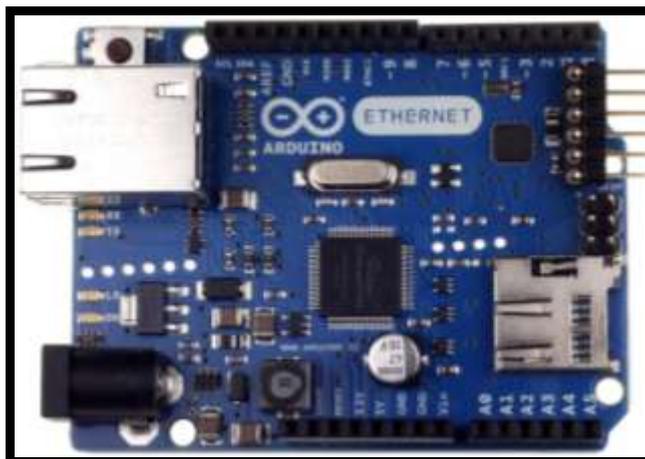


Figura 9: Arduino Ethernet

Fonte: <http://arduino.cc/en/uploads/Main/ArduinoEthernetFront.jpg>

O arduino Ethernet é uma versão de baixa potência do Arduino anunciada ao mesmo tempo em que o Uno. As principais diferenças entre uma versão e a outra são que o Arduino Ethernet possui um conector RJ45 integrado para uma conexão Ethernet e um leitor de cartão micro SD. O Arduino Ethernet não possui um chip controlador USB para serial integrada, mas possui um conector de seis pinos que pode ser conectado a um cabo FTDI ou a uma porta serial USB para fornecer um link de comunicação para que a placa possa ser programada. Ela pode também ser alimentada por um módulo opcional Power over Ethernet (POE), o qual permite ao Arduino Ethernet retirar sua energia de um cabo Ethernet de par trançado Categoria 5 conectado (EVANS; NOBLE; HOCHENBAUM, 2013, p. 28).

Fundamentado em (EVANS; NOBLE; HOCHENBAUM, 2013), podemos deduzir que o Arduino Ethernet é a opção ideal para monitoramento remoto e estações de registros de dados com leitor de cartão micro SD integrado e uma conexão com uma rede Ethernet com fio para alimentação.

2.3.4.4 Lilypad Arduino

É de conhecimento geral, que em resposta as adversidades encontradas nos projetos, o Arduino em sua essência propiciou um grande número de variações, notadamente em resposta a uma necessidade.

Conforme apresentado na Figura 10, o Lilypad Arduino foi desenvolvido pela *SparkFun Electronics* e pela Leah Buechley, visando resolver projetos voltados para área têxtil.

O LilyPad Arduino apresenta em sua arquitetura, conexões que podem ser costuradas em tecidos por intermédio dos kits especiais de costura, sua versão de baixa potência pode ser lavável, todavia sem jamais esquecer de retirar as baterias.

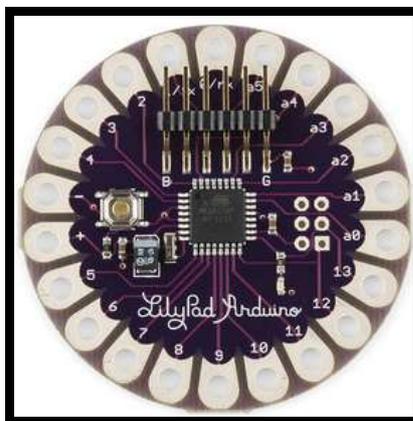


Figura 10: LilyPad Arduino

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

Conforme (EVANS; NOBLE; HOCHENBAUM, 2013), a principal diferença entre o LilyPad Arduino e outros Arduinos é uma velocidade de processamento menor de 8 MHz, em contrapartida à normal de 16 MHz. Uma coisa a se observar: a tensão de entrada não deve exceder 5,5 V.

2.3.4.5 Arduino Mega

Dentre os vários representantes da família Arduino, o Mega, usa um microprocessador de maior superfície de montagem, onde cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno de 20 a 50 KΩ.

O Arduino Mega fornece um aumento significativo na funcionalidade de entrada-saída em relação ao Arduino padrão; portanto, com o aumento da memória, ele é ideal para aqueles projetos maiores que controlam uma grande quantidade de LEDs, possuem um grande número de entradas e saídas ou necessitam de mais de uma porta serial de hardware – o Arduino Mega possui quatro. As placas possuem 54 pinos digitais e entrada-saída, 14 dos quais podem fornecer saída analógica PWM, e 16 pinos de entrada analógica. A comunicação é feita com até quatro portas seriais de hardware. A comunicação SPI e o suporte para dispositivos I2C/TWI estão também disponíveis. A placa também inclui um conector ICSP e um botão de reset. Um ATmega8U2 substitui o chipset FTDI usado pelo seu antecessor e processa a comunicação serial USB (EVANS; NOBLE; HOCHENBAUM, 2013, p. 29).

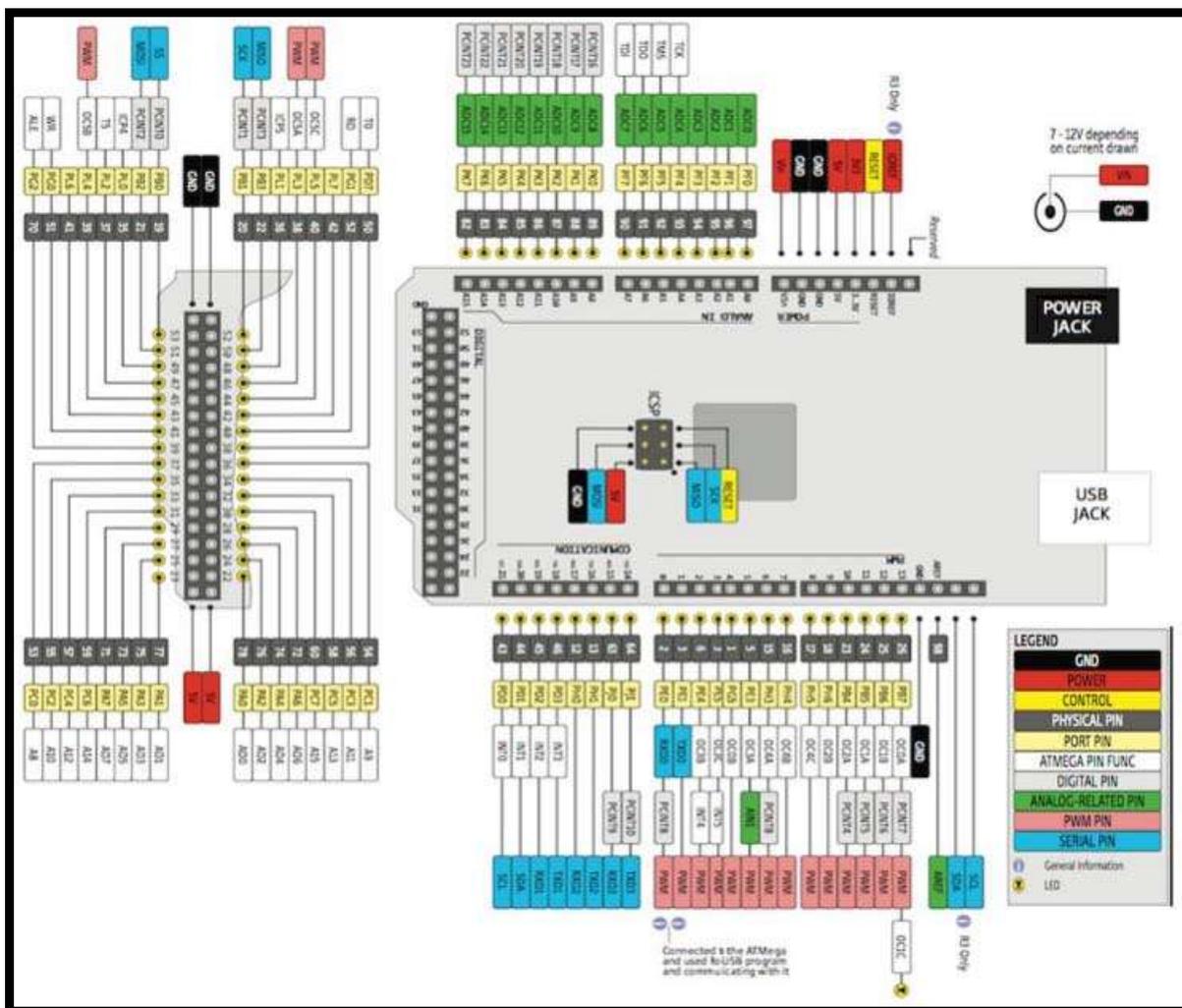


Figura 11: Arquitetura do Arduino Mega 2560

Fonte: Adaptado de, <http://s1.static.brasilecola.com/img/2014/05/figura15.jpg>

Para este projeto, adquirimos o microcontrolador Arduino Mega 2560 conforme Figura 11 acima catalogada, por além de possuir um microcontrolador baseado na tecnologia ATmega2560 com 256 KB de memória *flash* para armazenamento de código, dos quais 8 KB são usados pelo *bootloader*, 8 KB de SRAM e 4 KB de EEPROM, o mesmo apresenta em sua Arquitetura uma maior robustez quando comparado as demais placas Arduino, bem como, uma grande compatibilidade com a maioria dos *shields* projetados para o Arduino Uno e Duemilanove.

De acordo com a Tabela 2, o Arduino Mega pode ser alimentado pela conexão USB (5 volts) ou com uma fonte de alimentação externa auto chaveada. Nesse contexto, podemos deduzir que a placa pode operar com o fornecimento de uma tensão externa de 6 a 20 volts, tal qual o Arduino Uno.

Não obstante, se for fornecido menos de 7 volts, o pino de 5 V poderá fornecer menos de cinco volts e a placa pode ficar instável. Em contrapartida se for usado mais que 12 V, o regulador de voltagem pode superaquecer e danificar a placa, ou seja, o intervalo recomendado é de 7 a 12 volts.

Micro controlador	Atmega 2560
Tensão de funcionamento	5V
Tensão de entrada (recomendado)	7 a 12V
Tensão de entrada (máxima)	6 a 20V
Pinos de entrada e saída digital	54 (dos quais 14 podem ser saídas PWM)
Pinos de entradas analógicas	16
Valor máximo de corrente fornecida por pino	40mA
Valor de corrente para pino 3,3V	50mA
Memória flash	256KB
SRAM	8KB
EEPROM	4KB
Velocidade de clock	16MHz

Tabela 2: Especificações do Arduino Mega 2560

Fonte: <http://arduino.cc/en/Main/ArduinoBoardMega2560>

Um microcontrolador, ao contrário de um microprocessador, é desenhado e construído de forma a integrar diversos componentes num único circuito integrado, evitando, assim, a necessidade de adicionar componentes externos ao microcontrolador, que permitiriam as suas funcionalidades.

2.3.5 Ambiente de desenvolvimento do Arduino

Quando referenciamos o software da plataforma do Arduino, estamos nos referindo ao *Integrated Development Environment* (IDE) do mesmo, que nada mais é do que um ambiente de desenvolvimento gratuito cuja função basicamente é o desenvolvimento dos *sketchs* nas linguagens C e/ou C++ e enviá-los ao microcontrolador para que possam ser executados.

É dito por (EVANS; NOBLE; HOCHENBAUM, 2013) que o termo *sketch* vem de *processing*, uma linguagem frequentemente ensinada a estudantes de design e artes, e na qual o Arduino IDE é baseado.

Para realizar o *download* do *software* do Arduino, existente apenas para os Sistemas Operacionais (SO) Windows, Linux, e Mac, se faz necessário ir até a página oficial da plataforma de prototipagem em questão, que poderá ser encontrada no site <http://www.arduino.cc/>, em seguida escolher o seu SO e baixar.

De acordo com a Figura 12, a plataforma de desenvolvimento do Arduino é dividida em cinco partes, a *Toolbar* no topo (barra de botões), o código ou a *Sketch Window* no centro (área de programação), o Menu principal, a Barra de estado e a janela de mensagens na base (console do compilador).

Em se tratando do ambiente de desenvolvimento do Arduino (Figura 12), no Menu principal é possível encontrar todas as funções da IDE no que diz respeito ao seu gerenciamento, já na Barra de botões estão localizados alguns atalhos para facilitar a vida do desenvolvedor, por outro lado, a Barra de estado tem a função de informar qual placa arduino foi configurada para uso no momento, por sua vez, a área destinada a programação é voltada para o desenvolvimento do *sketch* e por do meio Console do compilador que se tem uma apresentação de todas as mensagens do compilador, que são tanto mensagens de informação ou aviso sobre erros no programa.

O que o IDE, realmente faz com seu código?

Quando você pressiona o botão “carregar”, ele verifica o código procurando erros e executa algumas traduções mínimas para converter o sketch para um C++ válido. O código é então compilado, o que significa que é convertido numa forma que pode ser entendido pelo Arduino. O arquivo resultante é então combinado com as bibliotecas padrão do Arduino antes de ser carregado no seu hardware (EVANS; NOBLE; HOCHENBAUM, 2013, p. 41).

Posto que o monitor serial objetiva auxiliar na programação da placa enviando dados através da porta de comunicação, basicamente se faz necessário conectar a placa no computador e através desta tela, como pode ser observado na Figura 13, podemos ver as informações enviadas ou recebidas pela placa, para tanto se faz necessário ainda iniciar o monitor serial, pressionando o botão *Serial Monitor* como apresentado na Figura 12.



Figura 12: Ambiente de desenvolvimento

Fonte: http://www.robotizando.com.br/curso_arduino_software_pg2.php

Como pode ser verificado na Figura 13, a parte superior da janela do monitor serial é usada para enviar dados ao Arduino, por outro lado, a parte principal da janela mostra os dados de saída de caracteres ASCII da plataforma de prototipagem eletrônica pela serial (o cabo USB) para o PC, onde o monitor serial os exibe. Não obstante, notamos que através do monitor serial podemos definir a taxa de transmissão usada para a comunicação, bem como a rolagem automática de texto e a forma de término de linha que é anexada aos dados enviados ao Arduino. Por outro lado, observamos ainda no canto inferior direito, conforme mostra a Figura 13, a taxa de transmissão (*Baud Rate*) por segundo em que alterações de estado ou *bits* (dados) são enviadas para o Arduino.

Destaca (MCROBERTS, 2013), que a configuração padrão é 9.600 *baud*, o que significa que, se você quisesse enviar um livro pela linha de comunicação serial (nesse caso, seu cabo USB), 1.200 letras ou símbolos de texto seriam enviados por segundo ($9.600 \text{ bits} / 8 \text{ bits por caractere} = \text{bytes ou caracteres}$).

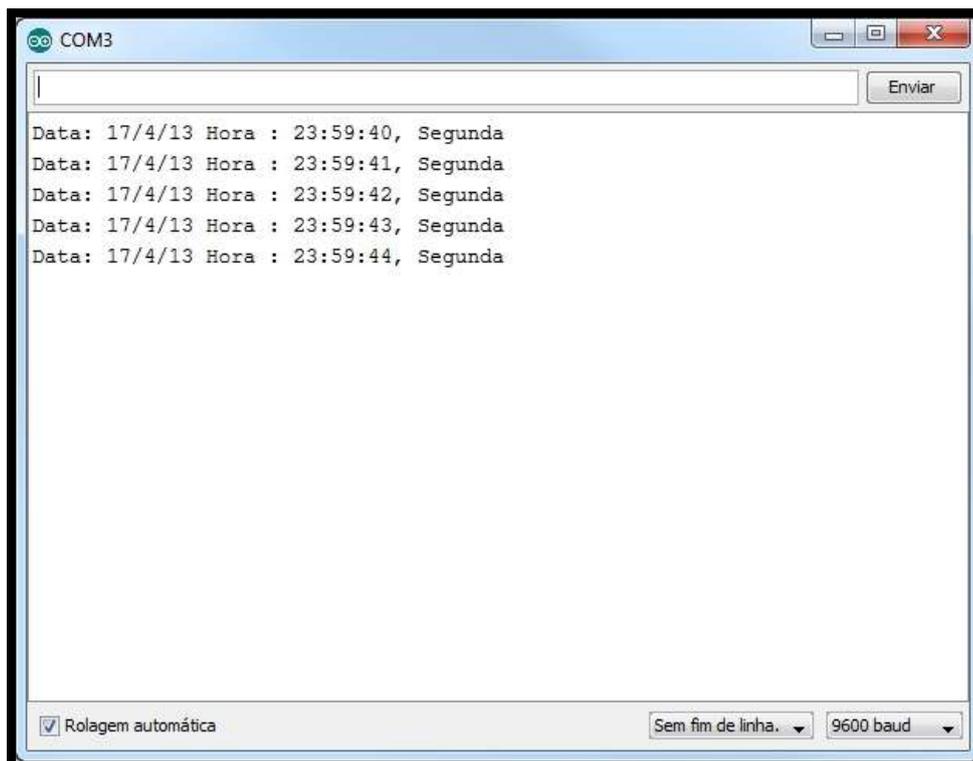


Figura 13: Saída de dados no Monitor Serial

Fonte: <http://www.arduinoecia.com.br/2013/04/real-time-clock-rtc-ds1307.html>

Com base no exposto, podemos deduzir que o monitor serial é um instrumento de bastante utilidade quando nos referimos a receber e enviar dados para o microcontrolador, principalmente quando relacionado à depuração e testes de código, permitindo o monitoramento dos dados entre o Arduino e o sistema do computador central via cabo USB.

2.3.6 Vantagens e desvantagens em se usar Arduino

O mundo da Automação Residencial apresenta uma gama de soluções disponíveis no mercado, contudo a plataforma de prototipagem eletrônica open-source Arduino foi a que mais nos intrigou, não apenas pela sua capacidade de adaptação nos mais variados tipos de projetos, mas principalmente por se apresentar como uma solução de baixo custo para conectar componentes, bem como por sua facilidade de programação.

A grande vantagem em se utilizar o Arduino, levando-se em conta as demais plataformas microcontroladoras e o poder aquisitivo dos segmentos menos abastados, está relacionada ao fato do Arduino ser *open-source* e *open-hardware* (fonte aberta), ou seja, isso significa que o código, os esquemas, o projeto etc. podem ser utilizados livremente por qualquer usuário e com qualquer propósito, conforme relatado na Seção 2.3.2.

Dessa forma, há muitas placas-clone e outras placas com base no Arduino disponíveis para compra, ou que podem ser criadas a partir de um diagrama. De fato, nada impede que você compre os componentes apropriados e crie seu próprio Arduino em uma matriz de pontos ou em sua *Printed Circuit Board* (PCB) feita em casa. A única ressalva que a equipe do Arduino impõe é que você não utilize a palavra "Arduino". Esse nome é reservado a placa oficial. Dai a existência de nomes para as placas-clone como Freeduino, Roboduino etc.. Como os projetos são de fonte aberta, qualquer placa-clone é 100% compatível com o Arduino e, dessa forma, qualquer *software*, *hardware*, *shield* etc. também será 100% compatível com o Arduino genuíno (MCROBERTS, 2011, p. 24).

Outra grande vantagem do Arduino é a sua facilidade de manuseio, ou seja, até mesmo pessoas que não são da área de tecnologia podem em um intervalo de tempo muito pequeno aprender os primórdios do Arduino e assim criar seus próprios projetos, além disso, na grande rede mundial de computadores há um grande número de comunidades que fazem o uso do Arduino e partilham de seus códigos e diagramas de circuito, com o fito de auxiliar outros desenvolvedores em seus projetos.

Em contrapartida, em se tratando de desvantagens além da plataforma trabalhar com uma tensão de 5 volts (V), apresenta "pouca robustez" quando comparada a grandes placas. Outra grande desvantagem relaciona-se ao fato de sua plataforma de hardware já ser modelada, ou seja, caso o usuário queira mudar para outros tipos de arquitetura vai encontrar muita dificuldade.

Não obstante, apesar da admirável quantidade de informação disponível para iniciantes na Internet, a maioria desses dados está espalhada em várias fontes, fazendo com que seja complicado rastrear as informações necessárias.

Por fim, os fatores preponderantes para se aprender a manipular a plataforma de prototipagem são o entusiasmo e a disposição em aprender, já que o Arduino foi projetado com uma arquitetura simples e não dispendiosa.

2.4 ARQUITETURA E FUNCIONAMENTO DA PROTOBOARD

A *protoboard*, ou mesmo *breadboard* (Figura 14), nada mais é do que uma base, muito utilizada para de construção de protótipos eletrônicos durante a fase de desenvolvimento de um projeto, por ela facilitar a movimentação de componentes ou a adição de outros. Por sua vez, a placa consiste em uma série de furos em uma grade; sob a placa, esses furos são conectados por uma tira de metal condutivo, a grande vantagem apresentada por ela é a não necessidade de se soldar dispositivos, ou seja, é reutilizável, fator no qual torna mais fácil de usar para construção de circuitos mais complexos, ou mesmo protótipos temporários.

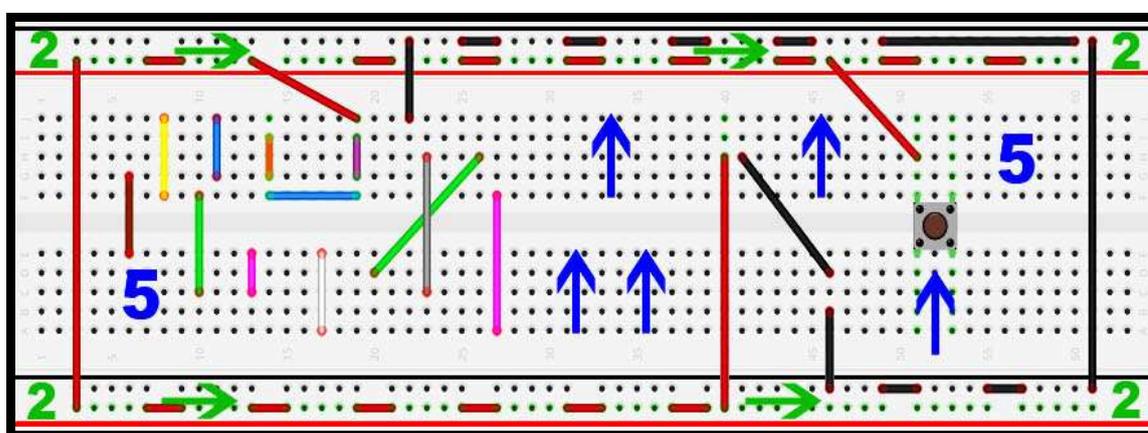


Figura 14: Protoboard - Matrizes 5/2 (Transmissão)

Fonte: próprio Autor

A *protoboard* conforme observado na Figura 14, normalmente apresenta uma faixa central com tamanho específico para componentes eletrônicos pequenos, bem como, dois tipos de matrizes principais, uma com duas colunas, doravante chamada de matriz 2 e outra com cinco colunas, de agora em diante chamada de matriz 5, que diferem no que diz respeito ao modo de transmissão de energia e dados; os *bornes* de alimentação (matriz 2) devem ser conectados a fonte que será utilizada no experimento, entretanto eles não energizam automaticamente os barramentos de alimentação, sendo necessário o uso de fios *jumpers* (geralmente com pontas moldadas para facilitar sua inserção na *protoboard*) utilizados para ligar provisoriamente os dispositivos eletrônicos na *breadboard*.

As tiras ao longo do topo e da base correm em paralelo a placa, e são projetadas para carregar o barramento de alimentação e o barramento do terra. Os componentes no meio da placa convenientemente conectam com os 5V (ou a voltagem que você estiver utilizando) ou com o terra. Algumas protoboards tem uma linha vermelha e outra preta correndo paralelas a esses furos, para mostrar qual é a alimentação (vermelho) e qual é o terra (preto). Em protoboards maiores, o barramento de alimentação as vezes tem uma divisão, indicada por uma quebra na linha vermelha. Isso torna possível enviar voltagens diferentes para partes distintas de sua placa. Caso você esteja utilizando apenas uma voltagem, um pequeno pedaço de fio jumper pode ser colocado de um lado a outro desse espaço, para garantir que a mesma voltagem seja aplicada em todo o barramento. As tiras no centro correm a 90 graus dos barramentos de alimentação e do terra, e há um espaço vazio no meio para que você possa colocar Circuitos Integrados, de modo que cada pino do chip vá para um conjunto diferente de furos e, portanto, para um barramento diferente (MCROBERTS, 2013, p. 46).

Em suma, as *protoboards* ou matriz de contatos, por serem projetadas para fins experimentais apresentam limitações de ordem prática em sua arquitetura, como baixa capacidade de corrente (cerca de 1A), capacitância e resistência dos contatos internos consideráveis, susceptibilidade à captação de ruídos e interferências. Desta forma, buscaremos fazer o uso da mesma para proposta de automação Residencial em questão, e uma vez comprovado o funcionamento do circuito, o mesmo deverá ser montado em uma placa de circuito impresso, para utilizá-lo em definitivo.

2.5 RESISTORES E A LEI DE OHM

É sabido que resistores são elementos que oferecem resistência à passagem de corrente elétrica que permanece constante independentemente da tensão ou corrente elétrica que circular pelo dispositivo, causando uma queda na voltagem em seus terminais, e ainda possui um formato cilíndrico e faixas coloridas que definem o seu valor em Ohms, cujo símbolo é a letra grega Ômega (Ω).

Podemos pensar em um resistor como sendo um tubo de água com um diâmetro menor do que o cano conectado a ele, ou seja, à medida que a água (ou a corrente elétrica) entra no resistor, o cano se torna mais fino e o volume da água (corrente) saindo na outra extremidade é reduzido.

Conta (SILVA, 2009) que George Simon Ohm foi um físico alemão que viveu entre os anos de 1789 e 1854 e verificou experimentalmente que existem resistores nos quais a variação da corrente elétrica é proporcional à variação da diferença de potencial (ddp).

Conforme (SILVA, 2009) Simon realizou inúmeras experiências com diversos tipos de condutores, aplicando sobre eles várias intensidades de voltagens, contudo, percebeu que nos metais, principalmente, a relação entre a corrente elétrica e a diferença de potencial se mantinha sempre constante.

A resistência elétrica designa a capacidade que um condutor tem de se opor à passagem de corrente elétrica, ou seja, a função da resistência elétrica é de dificultar a passagem de corrente elétrica.

Com base no exposto, podemos deduzir matematicamente que a voltagem aplicada nos terminais de um condutor é proporcional à corrente elétrica que o percorre.

O valor do resistor é calculado usando a seguinte fórmula (Figura 15):

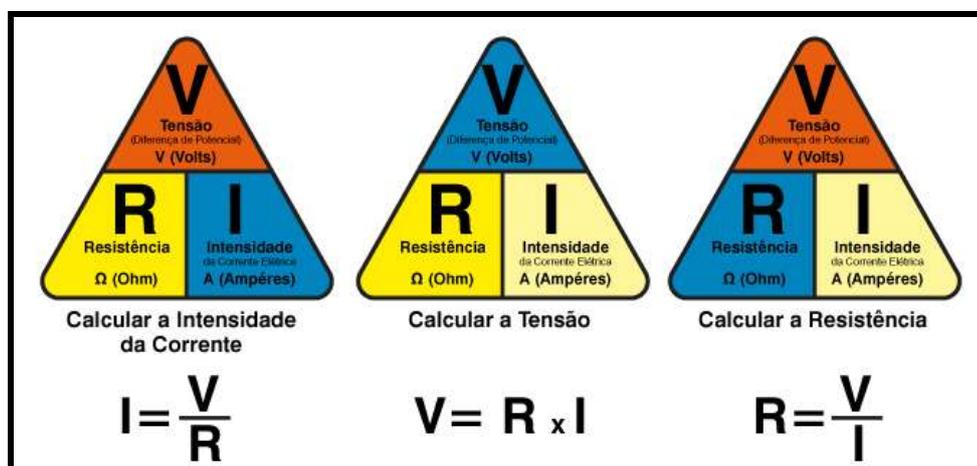


Figura 15: Fórmula da Lei de Ohm

Fonte: http://www.dreaminc.com.br/sala_de_aula/lei-de-ohm/

Conforme da Tabela 3, o valor de um resistor pode ser identificado analisando as cores que apresenta em torno dele ou então usando um Ohmímetro (instrumento de medição de resistores).

Cores	1ª Faixa	2ª Faixa	3ª Faixa	4ª Faixa Multiplicativa	5ª Faixa Tolerância
Preto	–	0	0	–	–
Marrom	1	1	1	0	±1% F
Vermelho	2	2	2	00	±2% G
Laranja	3	3	3	000	–
Amarelo	4	4	4	0000	–
Verde	5	5	5	00000	±0,5% D
Azul	6	6	6	000000	±0,25% C
Violeta	7	7	7	0000000	±0,1% B
Cinza	8	8	8	00000000	±0,05% A
Branco	9	9	9	000000000	–
Ouro	–	–	–	÷10	5% J
Prata	–	–	–	÷100	10% K
Incolor	–	–	–	–	20% M

Tabela 3: Código de Cores e Valores dos Resistores

Fonte: <http://eelusf2014.blogspot.com.br/>

Por apresentar tamanho reduzido, seria inviável imprimir nos resistores as suas respectivas resistências, por isso optou-se então pelo código de cores. Ao redor do resistor (Figura 16) você tipicamente encontrara quatro faixas de cores, e através da Tabela 3 descobrir o valor do mesmo.

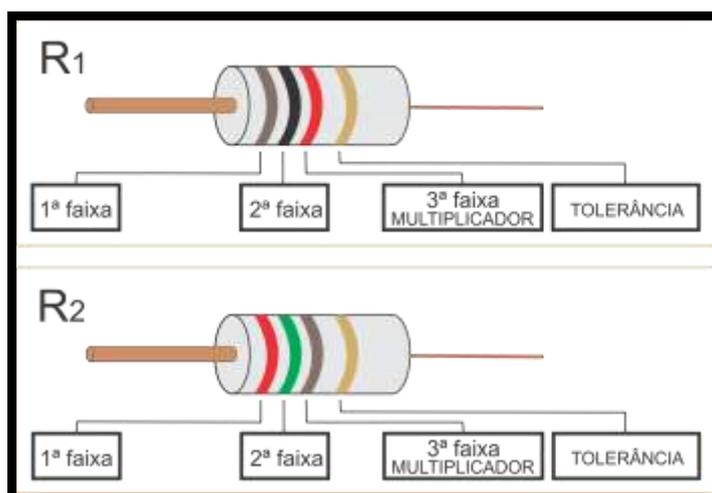


Figura 16: Modelo de Resistores

Fonte: http://osfundamentosdafisica.blogspot.com.br/2013/06/cursos-do-blog-eletricidade_19.html

3. DESENVOLVIMENTO DO PROJETO

Neste projeto, realizar tarefas de controle, voltadas para Automação Residencial de baixo custo como meio facilitador da vida cotidiana das pessoas portadoras de paraplegia ou não é o ponto principal, para tanto, é necessário contar com uma interface de manipulação, uma estrutura de comunicação e por fim um sistema de acionamento. Entretanto, este conjunto de ferramentas deve estar alinhado para trabalhar perfeitamente de forma conjunta e ordenada.

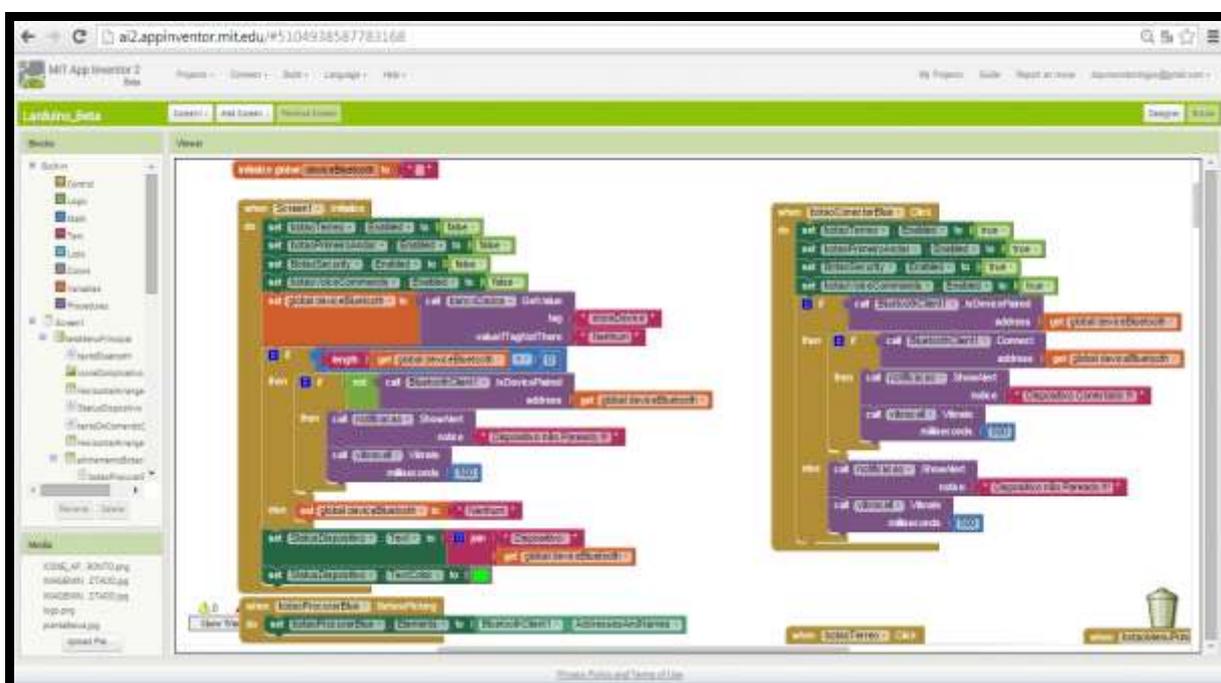


Figura 17: *Print Screen* da tela de desenvolvimento do aplicativo Larduno Prime

Fonte: Próprio Autor

Destarte, para transformar um simples click na tela do *smartphone* em um acender e apagar de luzes, ou mesmo abrir e fechar de um portão eletrônico, o aplicativo *Larduno Prime* (Figura 17), conta com o envio de dados previamente padronizados no aplicativo *android*, através do *bluetooth* via radiofrequência, estes dados são entregues ao Arduino. Uma vez recebidos, o Arduino interpreta os comandos e realiza as tarefas nele programadas para tal.

3.1 PLATAFORMA DE DESENVOLVIMENTO MIT APP INVENTOR 2 BETA

O *App Inventor* é uma ferramenta baseada em *cloud computing*, o que significa que você pode construir aplicativos diretamente no seu navegador web.

No entanto, conforme mencionado na Seção 1.4 deste Trabalho de Conclusão de Curso, para construção da interface de comunicação do *smartphone* com a plataforma de prototipagem Arduino, propusemos a plataforma *MIT App Inventor 2 Beta* (Figura 18), que poderá ser encontrada no site <http://appinventor.mit.edu/explore/teach.html>, este site oferece todo o suporte que você precisa, para aprender e construir seus próprios aplicativos, por outro lado, o que torna o *MIT App Inventor* especial é a forma como a comunidade global de usuários adotaram a ferramenta implementando uma variedade de maneiras.

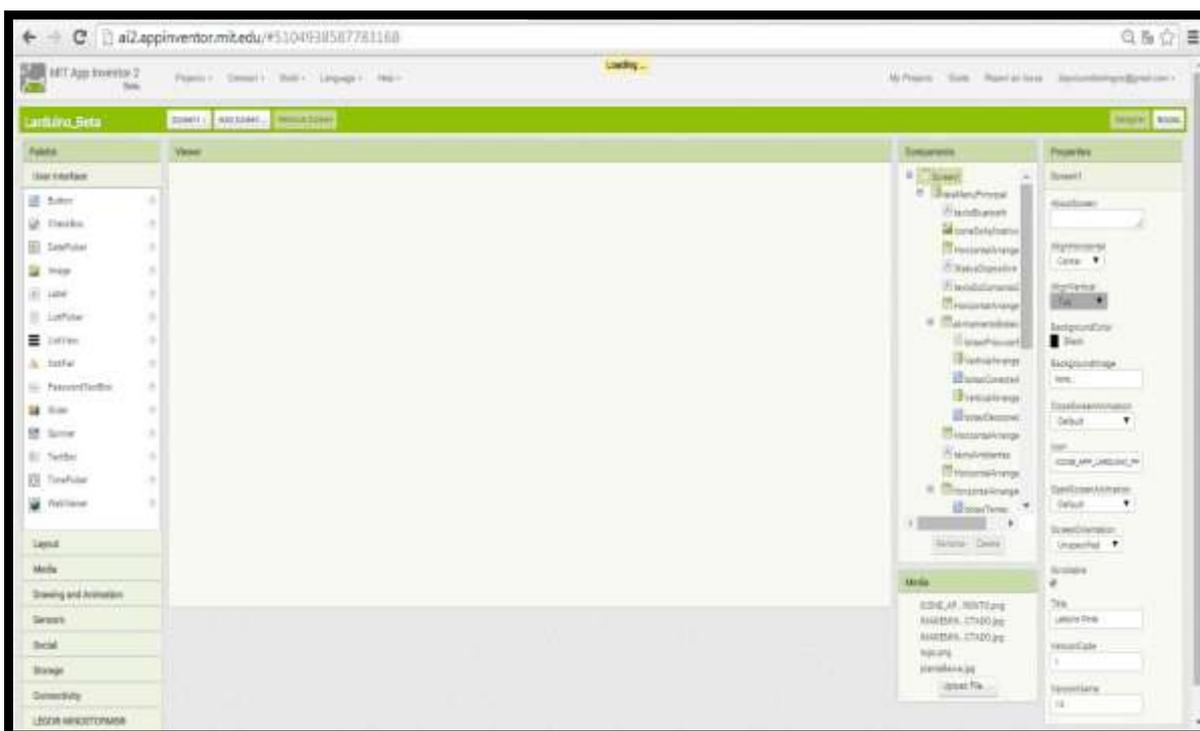


Figura 18: Plataforma de desenvolvimento MIT *app Inventor 2 Beta*

Fonte: Próprio Autor

Entretanto, para que toda estrutura da proposta de Automação Residencial funcione, se faz necessário utilizar um dispositivo que permita a interação com todo o sistema, para tanto, desenvolvemos o aplicativo *Larduino Prime*.

Não obstante, com o surgimento dos *smartphones*, o controle para sistemas de automação passou a ser mais difundido.

Com base no exposto, optamos por *smartphones* (Moto g), por suprirem com eficiência e eficácia as exigências do projeto de Automação Residencial em questão, haja vista que são equipamentos de fácil manuseio, com excelente desempenho e de custos relativamente baixos em relação aos seus concorrentes, foi a forma por nós encontrada para dar dinamismo ao sistema, já que através deles quaisquer acionamentos poderão ser feitos através de um simples toque na tela.

Assim sendo, com estes dispositivos os usuários poderão interagir com os equipamentos configurados e instalados em pontos específicos de uma residência, ora representada por uma maquete (Figura 19), que simulará os moldes de uma casa real com todo circuito eletrônico em uma *protoboard*, permitindo a montagem de circuitos apenas com o encaixe dos componentes.



Figura 19: Maquete nos moldes de uma casa real

Fonte: Próprio Autor

Além dos requisitos anteriormente citados, para que o usuário possa interagir com o sistema se faz necessário que o mesmo tenha instalado em seu *smartphone*, o aplicativo *Larduino Prime*. Contudo, para que toda esta estrutura funcione, é necessário fazer o pareamento do *bluetooth* do *smartphone* com o módulo *bluetooth* conectado ao Arduino, após isso o usuário poderá utilizar o sistema, realizando os acionamentos dos dispositivos a uma distância de aproximadamente 10 metros.

3.2 DESCRIÇÃO DO APLICATIVO ANDROID LARDUINO PRIME

Conforme apresentado na Figura 20, o *Larduino Prime* é um aplicativo *android* que concentra em um único dispositivo (*smartphone*), toda a gestão dos dispositivos conectados ao microcontrolador Arduino Mega 2560, ou seja, através da comunicação via *bluetooth* o aplicativo em questão propicia ao usuário a possibilidade de controle de iluminação, temperatura ambiente, condicionador de ar, sistema de segurança, acionamento de portão eletrônico e Sensor detector de vazamento de gás butano (C₄H₁₀), por meio de *tablets* ou *smartphones* que contenham o sistema operacional *Android*.

O aplicativo (Figura 20) foi desenvolvido com uma *interface* com *design* amigável, intuitivo e simples, facilitando assim a vida do usuário, de acordo com seus desejos e necessidades.



Figura 20: *Smartphone* Moto g com aplicativo *Larduino Prime* instalado

Fonte: Próprio Autor

Não obstante, desenvolvido na IDE MIT app *Inventor 2 Beta*, o aplicativo *Android Larduino Prime*, conforme o designer das telas mostrado na Figura 21, possui cinco etapas de criação concluídas, são elas: Menu Principal, Controle do Ambiente Térreo (Térreo), Controle do Ambiente Primeiro Andar (Primeiro Andar), Controle do Sistemas de Segurança (*Security Systems*) e Acionamento das Funcionalidade por Comando de Voz (*Voice Commands*).



Figura 21: Telas do Aplicativo *Larduino Prime*

Fonte: Próprio Autor

O Menu Principal do *Larduino Prime*, controla as demais interfaces do aplicativo, sendo por nós considerado o ponto crítico do Sistema, já que foi a funcionalidade mais complexa de ser desenvolvida e conseqüentemente a mais interessante, é através desta tela que o usuário estabelece a comunicação entre o *smartphone* e o Arduino Mega 2560.

Em contrapartida, para que as demais funcionalidades contidas no Menu Principal sejam desbloqueadas, e o usuário possa usufruir dos privilégios fornecidos pelo *Larduino Prime*, como navegar pelo sistema e escolher qual interface secundária deseja abrir, se faz necessário estabelecer via aplicativo uma conexão entre o *bluetooth* do *smartphone* e o módulo *bluetooth* conectado ao Arduino Mega 2560.

3.2.1 Visão Geral dos Utilitários e Blocos de Código do Aplicativo

Conforme mencionado na Seção anterior, a construção do código para estabelecer a conexão *bluetooth* do aplicativo *Larduino Prime*, abaixo arrolado na Figura 22, foi a mais interessante de se desenvolver, pois através desta conexão é possível em tempo real variar a intensidade de funcionamento de inúmeros equipamentos conectados ao Arduino em um raio de aproximadamente 10 metros.

```

initialize global deviceBluetooth to
when (Screen1) Initialize
do
  set BotaoTerno.Enabled to false
  set BotaoPrimeiroAndar.Enabled to false
  set BotaoSegundo.Enabled to false
  set BotaoVoz.Enabled to false
  set global deviceBluetooth to call BancoDados.GetValue
  tag storeDevice
  valueIfTagNotThere Nenhum
  if length get global deviceBluetooth > 0
  then
    if not call BluetoothClient.IsDevicePaired
    address get global deviceBluetooth
    then
      call notificacao.ShowAlert
      notice Dispositivo não Pareado !!!
      call Vibracao.Vibrate
      milliseconds 500
    else
      set global deviceBluetooth to Nenhum
  set StatusDispositivo.Text to join Dispositivo
  get global deviceBluetooth
  set StatusDispositivo.TextColor to

when BotaoProcuraBlue.BeforePicking
do
  set BotaoProcuraBlue.Elements to BluetoothClient.AddressesAndNames

when BotaoProcuraBlue.AfterPicking
do
  set global deviceBluetooth to BotaoProcuraBlue.Selection
  call BancoDados.StoreValue
  tag storeDevice
  valueToStore get global deviceBluetooth
  set StatusDispositivo.Text to join Dispositivo
  get global deviceBluetooth
  set StatusDispositivo.TextColor to

when BotaoConectaBlue.Click
do
  set BotaoTerno.Enabled to true
  set BotaoPrimeiroAndar.Enabled to true
  set BotaoSegundo.Enabled to true
  set BotaoVoz.Enabled to true
  if call BluetoothClient.IsDevicePaired
  address get global deviceBluetooth
  then
    if call BluetoothClient.Connect
    address get global deviceBluetooth
    then
      call notificacao.ShowAlert
      notice Dispositivo Conectado !!!
      call Vibracao.Vibrate
      milliseconds 500
    else
      call notificacao.ShowAlert
      notice Dispositivo não Pareado !!!
      call Vibracao.Vibrate
      milliseconds 500
  else
    call BluetoothClient.Disconnect
  set BotaoTerno.Enabled to false
  set BotaoPrimeiroAndar.Enabled to false
  set BotaoSegundo.Enabled to false
  set BotaoVoz.Enabled to false
  call notificacao.ShowAlert
  notice Dispositivo Desconectado !!!
  call Vibracao.Vibrate
  milliseconds 500
  set StatusDispositivo.Text to Dispositivo: Nenhum
  set StatusDispositivo.TextColor to
  
```

Figura 22: Código para construção do *bluetooth*

Fonte: próprio Autor

Para tanto, conforme apresentado na (Figura 23), para construção do código acima (Figura 22) fizemos o uso de alguns utilitários tais como: **BluetoothClient**, **TinyDB**, **Player** e **Notifier**.



Figura 23: Componentes Utilitários do Aplicativo *Larduino Prime*

Fonte: Próprio Autor

Destarte, para estabelecer a conexão, se faz necessário clicar no botão Procurar, que nada mais do que um botão do tipo **ListPicker**, usado para dispor listas de dispositivos, no nosso caso a lista de dispositivos *bluetooth* que estão no alcance. Escolhido o dispositivo *bluetooth* a ser pareado, e clicado no botão conectar (Figura 20), o utilitário de armazenamento *TinyDB* (Figura 23) salvará permanentemente os dispositivos *bluetooth* pareados com aplicativo, ou seja, caso o usuário saia e entre uma outra vez no aplicativo, não se fará mais necessário procurar o *bluetooth* a ser pareado, pois o mesmo estará salvo no *TinyDB*, restando apenas clicar no Botão Conectar para o *BluetoothClient* intermediar a conexão.

Neste momento, através do utilitário *Player* (Figura 23), cuja principal função é tocar música ou vídeo, entretanto não usaremos para este fim, e sim para fazer o *smartphone* vibrar por 500 milissegundos, informando ao usuário através do utilitário *Notifier* (Figura 23) a seguinte mensagem na tela (**Dispositivo Conectado !!!**), que nesse momento, desbloqueará as demais funcionalidades do aplicativo.

Pensando em uma maior comodidade para o usuário, disponibilizamos no aplicativo o comando de acionamento por voz, para tanto, foi necessário criar o botão **Voice Commands** (Figura 20) e fazer o uso do utilitário **SpeechRecognizer** (Figura 23) para converter o que foi falado pelo usuário em texto.

Entretanto, não seria interessante se o comando por voz fosse executado sem a necessidade de se apertar o botão criado para tal?

Com base no exposto, fizemos o uso do utilitário **AccelerometerSensor** (Figura 23), pois através deste o usuário poderá balançar o *smartphone* e ele abrirá o comando de voz sem a necessidade de apertar o botão do (**Voice Commands**).

3.2.2 Detalhamento da Implementação do Aplicativo Larduino Prime

Inicialmente, conforme apresentado na Figura 22, foi criada a variável Global **deviceBluetooth**, visando nela guardar o endereço e nome do dispositivo *bluetooth* pareado, em seguida, cada vez que o aplicativo for iniciado a primeira coisa a ser executada são as instruções dentro do bloco (**When Screen1.Initialize do**).

Entretanto, a primeira vez que o aplicativo é rodado, o bloco (**When Screen1.Initialize do**) não tem muito sentido, haja vista que não há nada salvo no banco de dados (*TinyDB*). Outrossim, objetivando verificar se a variável (**deviceBluetooth**) tem ou não algum valor guardado é feito um teste usando o bloco **length get global deviceBluetooth**, já que o bloco **length** retorna o tamanho da *string*, ou seja, se o resultado for maior que zero é porque tem algo gravado nela.

Embora, feito o teste na variável **deviceBluetooth**, ainda será verificado através do bloco (**not call BluetoothClient1.IsDevicePaired address get global deviceBluetooth**) se o **BluetoothClient1** não está conectado, caso o dispositivo não esteja pareado, o bloco (**call notificacao.ShowAlert notice**) irá garantir que seja emitido um alerta ao usuário de que o dispositivo não está pareado, e nesse momento a variável **deviceBluetooth** recebe o texto nenhum, na cor vermelha, já que não conectou a nada, caso contrário será colocado no status do dispositivo o endereço e nome do dispositivo pela variável recebido, na cor verde.

Deste modo, através do bloco (**when botaoProcurarBlue.BeforePicking do**) será mostrado a lista de *bluetooth* próximo disponíveis a serem selecionados. Depois de selecionado o *bluetooth* a ser pareado será executado o bloco (**when botaoProcurarBlue.AfterPicking do**), no qual irá “setar” a variável **deviceBluetooth** para guardar o dispositivo que foi selecionado, posteriormente com o *TinyDB* iremos verdadeiramente guardar no banco de dados o dispositivo selecionado através da *tag* definida por nós como (**storeDevice**), ou seja, é nessa *tag* que será guardado o valor que está na variável **deviceBluetooth**.

Enfim, com o dispositivo *bluetooth* já selecionado, é hora de conectar, para tanto, usaremos o bloco (**when botaoConectarBlue.Click do**) para fazer a conexão, onde inicialmente é mais uma vez feito o teste se o dispositivo está pareado ou não, tudo através do evento (**BluetoothClient1.IsDevicePaired address get global deviceBluetooth**), feita a conexão, o usuário será informado com a seguinte mensagem na tela (**Dispositivo Conectado !!!**).

Por outro lado, no que diz respeito ao desenvolvimento do código que estabelece o comando de acionamento por voz do aplicativo *Larduino Prime* (Figura 24), foi feito o uso do bloco (**when *SpeechRecognizer.AfterGettingText result***), pois a partir do momento que é aberto pelo usuário o comando por voz, é gerado resultados, resultados estes que passarão por tratamento e análise.

```

when botaoVoiceCommands Click
do
  call SpeechRecognizer1 .GetText

when AccelerometerSensor1 Shaking
do
  call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .AfterGettingText result
do
  set textoDoComandoDeVoz .Text to get result
  if get result = "ligar lâmpada sala"
  then
    if BluetoothClient1 .IsConnected
    then
      call BluetoothClient1 .SendText
      text "A"
  else
    if get result = "desligar lâmpada sala"
    then
      if BluetoothClient1 .IsConnected
      then
        call BluetoothClient1 .SendText
        text "B"
    else
      if get result = "ligar lâmpada 4º"
      then
        if BluetoothClient1 .IsConnected
        then
          call BluetoothClient1 .SendText
          text "C"
      else
        if get result = "desligar lâmpada 4º"
        then
          if BluetoothClient1 .IsConnected
          then
            call BluetoothClient1 .SendText
            text "D"
  
```

Figura 24: Código para construção do Comando de acionamento por Voz

Fonte: Próprio Autor

Nada obstante, conforme apresentado na Figura 24, para que o comando de voz possa ser aberto no aplicativo se faz necessário o uso do bloco (**call *SpeechRecognizer1.getText***), ou seja, por esse motivo, objetivando acionar o comando por voz tanto pelo botão **Voice Commands**, como pelo sacudir do *smartphone* para cima e para baixo, o referenciado bloco foi colocado dentro das funções (**when *botaoVoiceCommands.Click do***) e (**when *AccelerometerSensor1.Shaking do***) respectivamente.

Por fim, a voz do usuário será convertida em texto e guardada na variável **result**, em seguida, através do bloco (**set.textoDoComandoDeVoz.Text to get result**) o usuário poderá verificar na tela do **Menu Principal** o que foi falado em forma de texto, em paralelo ao ocorrido será feito um teste entre o conteúdo da variável **result**, ora falada pelo usuário e convertida em texto, com um texto pré-programado por nós como, por exemplo, ligar lâmpada sala, se as premissas forem verdadeiras o aplicativo verificará se o *bluetooth* está realmente conectado, e em caso de conectado, através do bloco (**call Bluetooth.Client1.SendText**) será enviado o caractere que foi por nós pré-programado para o módulo *bluetooth* conectado ao Arduino, cujo processará o dado recebido e executará a ação equivalente ao caractere recebido.

3.3 ARQUITETURA DO CIRCUITO ELETRÔNICO

Conforme mencionado no item 2.3.2, o Sistema como um todo, foi codificado na plataforma de prototipagem eletrônica *open source* Arduino na linguagem de programação baseada em C/C++, que juntamente com o módulo *bluetooth* forneceram um excelente desempenho na comunicação no que diz respeito às taxas de processamento dos acionamentos dos dispositivos ligados ao microcontrolador Arduino Mega 2560. Destarte, no momento que é acionada alguma funcionalidade no aplicativo, o mesmo enviará uma requisição via conexão *bluetooth* para ser processada no Arduino.

Por outro lado, o usuário final não terá acesso ao código que foi introduzido no Arduino, todavia poderá interagir com a plataforma através da interface do aplicativo *Android Larduino Prime*.

Segundo (CUNHA, 2007), colocar capacidade computacional dentro de um circuito integrado, equipamento ou sistema é a forma mais apropriada de se definir o que venha a ser um sistema embarcado.

Nada obstante, a Figura 25 representa como foram feitas as conexões entre a Arduino e os demais dispositivos, cujas foram realizadas por intermédio de fios *jumpers* em uma protoboard, em contrapartida, em alguns dispositivos foi necessário utilizar solda para uma conexão mais robusta.

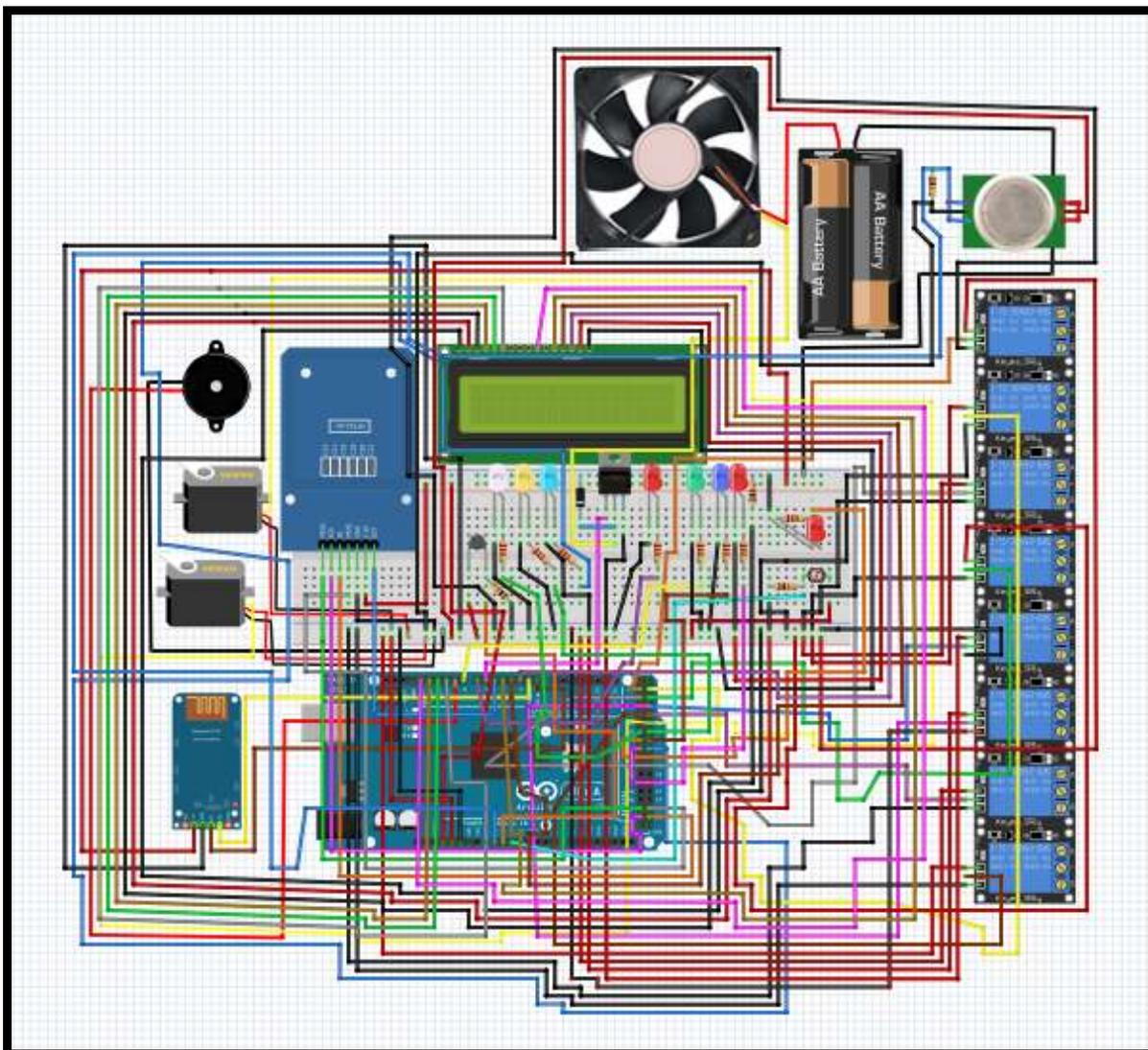


Figura 25: Conexão entre os dispositivos da Central Eletrônica

Fonte: Próprio Autor

Destarte, conforme apresentado na Figura 25, a Central Eletrônica da proposta de Automação Residencial em questão integra equipamentos e sistemas dentro de uma casa. Por exemplo: iluminação, alarme, controle de acesso, sensor de gás butano, sensor de temperatura, climatização e etc.

3.3.1 Módulo Bluetooth

No entanto, conforme pode ser observado na Figura 26, o Módulo *bluetooth* RS232, ora usado no projeto, foi desenvolvido com o fito de interagir com circuitos eletrônicos microprocessados ou microcontrolados e, portanto operam com níveis de

tensão TTL, ou seja, 3.3V ou 5V. Não obstante, o mesmo pode operar nos modos *Master* e *Slave*, quando no modo *Master* o Arduino pode receber a conexão de outros Arduinos, e no modo *Slave* é mais indicado para um Arduino se conectar a um smartphone, computador ou outro Arduino que esteja com módulo *Master*.

Outrossim, pode ser observado na Figura 26 que a conexão do *Bluetooth* com o Microcontrolador, geralmente se dá através de 4 fios *jumpers*: 3.3V (alimentação), GND (terra), RX e TX sendo estes do tipo *Crossover*, ou seja, o pino TX do Microcontrolador deve ir ao pino RX do Módulo *Bluetooth* e vice-versa.

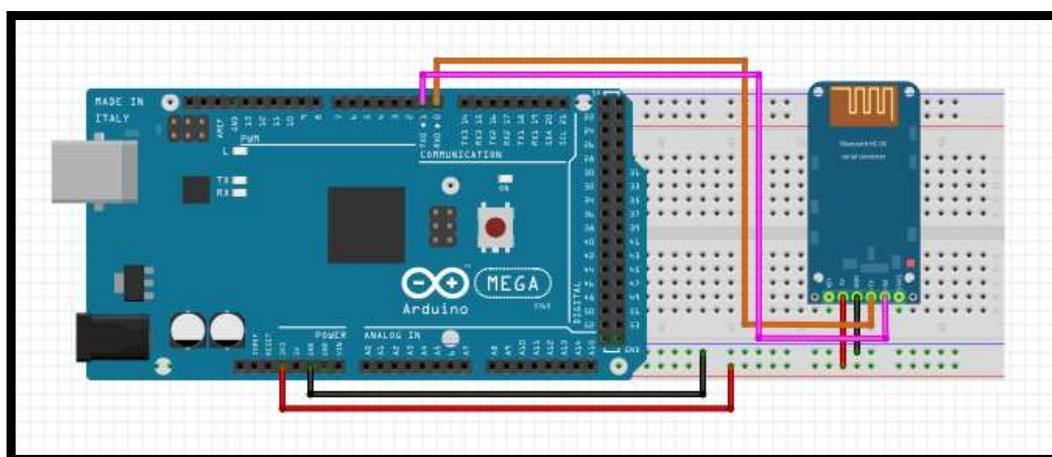


Figura 26: Conexão entre o Módulo *bluetooth* e o Arduino Mega 2560

Fonte: Próprio Autor

Entretanto, o sistema utiliza uma frequência de rádio de 2.4 GHz para criar uma comunicação de curto alcance entre os dispositivos habilitados, ou seja, permite apenas a comunicação entre dispositivos próximos.

Versa (TANENBAUM, 2003) que a estrutura de camadas do *Bluetooth* não segue o modelo OSI, o modelo TCP/IP, o modelo 802 ou qualquer outro modelo conhecido.

A unidade fundamental de um sistema Bluetooth é uma piconet, que consiste em um nó mestre e até sete nós escravos ativos localizados dentro de uma distância de 10 metros. É possível utilizar-se de diversas piconets dentro da mesma sala e elas podem até mesmo serem conectadas por um nó de ponte. Quando se tem diversas redes piconets conectadas umas as outras a estrutura passa a ser chamada de scatternet (TENENBAUM, 2003, p.331).

Posto que já tenham classes de *Bluetooth* com alcance de 100 metros, a grande maioria conta com uma abrangência de 1 a 10 metros, o que por um lado

seja considerada uma desvantagem, por outro ajuda na segurança dos usuários, uma vez que, antes de efetuar trocas de dados e arquivos entre dispositivos possuidores do *Bluetooth*, normalmente se faz necessário ativar algumas funcionalidades através das configurações dos aparelhos.

3.3.2 Módulo Relé

O Módulo relé provê uma maneira fácil de controlar dispositivos que apresentem alta tensão, na Figura 27 pode ser observado a disposição dos pinos do módulo relé de 8 canais, ora usado na proposta de Automação Residencial em questão, com o microcontrolador Arduino. Não obstante, cada canal do relé apresenta leds indicadores que mostram o estado do mesmo, variando entre ligado e desligado, bem como, módulo tolera cargas de até 10 amperes, em 125 VAC, 250 VAC ou 30 VDC. Por outro lado, o módulo já contém todo o circuito de proteção para evitar danos ao microcontrolador, e possui baixa corrente de operação.

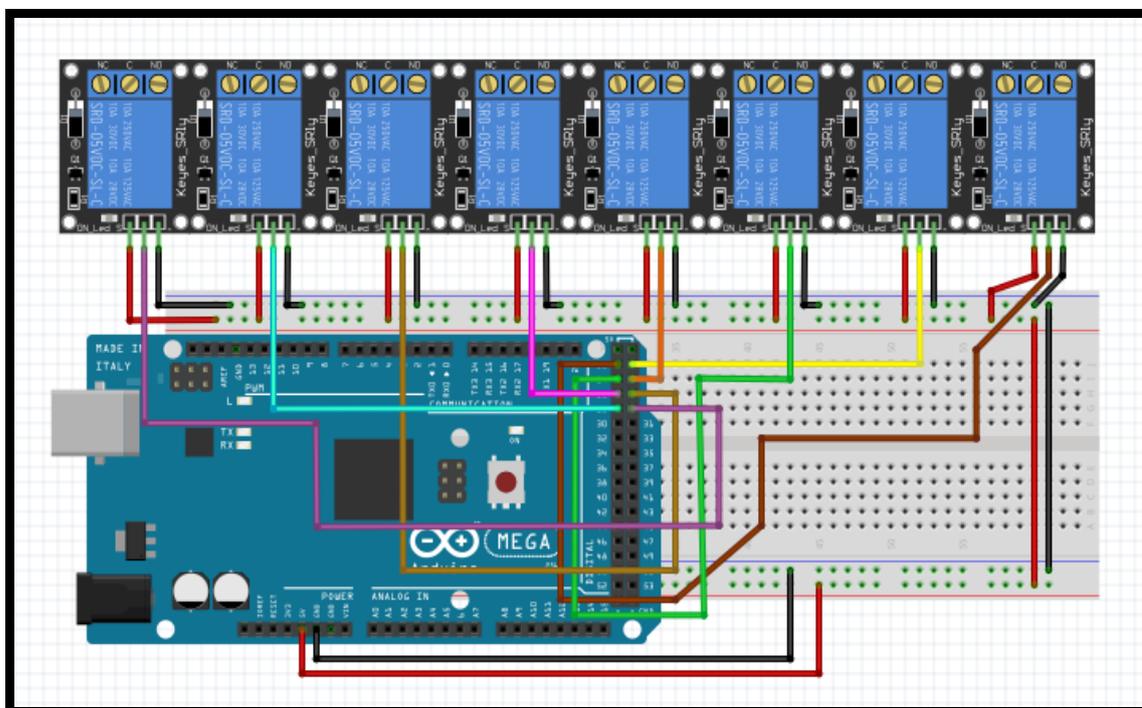


Figura 27: Conexão entre o Módulo Relé 8 canais e o Arduino Mega 2560

Fonte: Próprio Autor

Nada obstante, conforme versa a Figura 28, o relé possui pinos que estão relacionados à comunicação, ou seja, é possível observar que do lado esquerdo superior os pinos JD-Vcc, Vcc e GND, permitem que seja conectada uma fonte externa de 5V ao módulo, logo abaixo, os pinos GND, IN1, IN2 e VCC respectivamente estão relacionados ao *ground* do Arduino, acionamento do canal do relé 1, acionamento do canal do relé 2 e os 5 volts do Arduino.

Os relés podem ter diversas configurações quanto aos seus contatos, na Figura 28 é possível observar os contatos NC (Normal Fechado), C (Comum), e NO (normal aberto). Os contatos NO são os que estão abertos enquanto a bobina não está energizada e que fecham quando a bobina recebe corrente, em contrapartida, os NC abrem-se quando a bobina recebe corrente, ou seja, o contato NC é inversamente proporcional ao contato NO. Todavia em se tratando do contato C ou comum, o mesmo é responsável por estabelecer a condução no momento em que for fechado o contato com NO, e o oposto com o NC.



Figura 28: Pinagem do Módulo relé 2 canais

Fonte: <http://blog.filipeflop.com/modulos/controle-modulo-rele-arduino.html>

Por fim, o relé nada mais é do que um dispositivo eletromecânico ou não, com inúmeras aplicações possíveis em comutação de contatos elétricos. Servindo para ligar e desligar dispositivos. Entretanto, no que diz respeito ao seu acionamento, se faz necessário apenas mudar o estado das portas digitais ligadas aos pinos. Todavia, o relé será acionado quando o estado da porta digital estiver em **LOW**, ou seja, os canais do módulo relé **são ativados em nível baixo**.

3.3.3 Módulo Rfid Mfrc522

Não obstante, conforme pode ser observado na Figura 29, o Módulo Rfid (*Radio Frequency Identification*) baseado no chip Mfrc522, ora usado nesse projeto, tem como fito prover uma maior segurança a Residência automatizada, fornecendo um maior controle de acesso e identificação de pessoas no ambiente, seja por meio de crachás do tipo S50 ou mesmo via tags do tipo chaveiro. Na contemporaneidade, é de praxe encontrar a referenciada tecnologia em prédios comerciais, indústrias, pedágios, ou em cartões tipo bilhete único, utilizado em várias cidades brasileiras para acesso ao transporte coletivo.

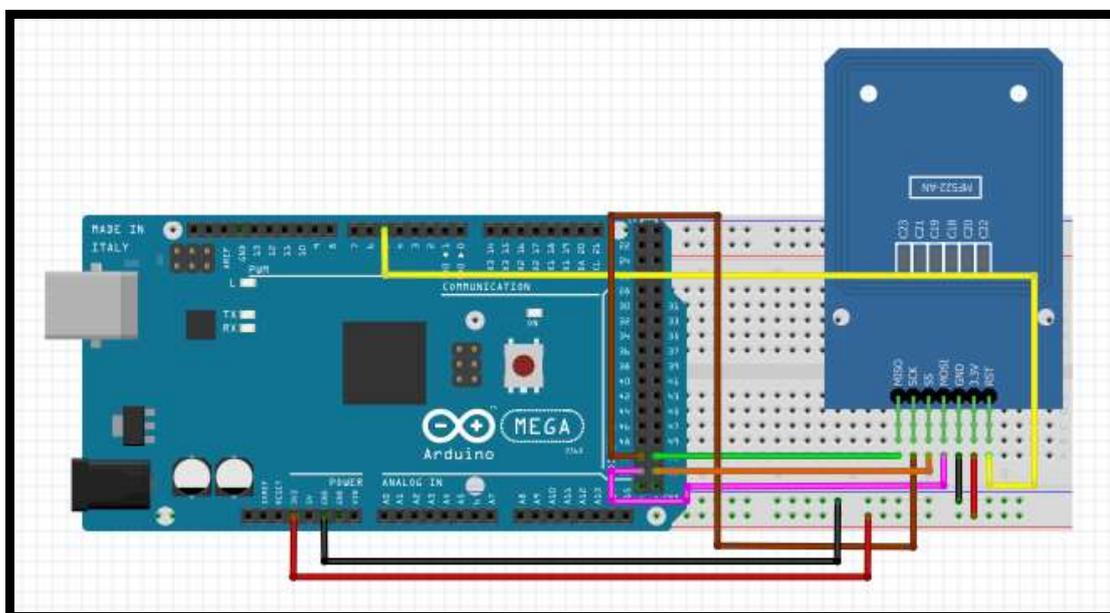


Figura 29: Conexão entre o Módulo Rfid e o Arduino Mega 2560

Fonte: Próprio Autor

Entretanto, os crachás ou as *tags* Rfid, além de serem de pequeno tamanho e consumo “desprezível”, são utilizados na comunicação sem contato a uma frequência de 13,56MHz. Outrossim, cada crachá/*tag* do leitor Rfid apresenta em sua composição um número hexa decimal, o qual faz alusão a sua própria identificação (UID), e é por intermédio desta que será controlado o acesso de cada usuário na Residência, onde o UID do cartão exibirá as informações de acesso em um display LCD do tipo 16X2 tal qual representado na Figura 25.

Deste modo, como pode ser observado na Tabela 4, o leitor Rfid é composto por 8 pinos, dos quais 7 são conectados ao Arduino .

O Módulo utiliza tensão de 3.3 V, através da interface SPI (*Serial Peripheral Interface*), e pode ser conectado facilmente ao microcontrolador. Por sua vez, o SPI é um protocolo de dados em série síncrona, utilizado para comunicação entre um ou mais dispositivos periféricos ao longo de distâncias curtas, podendo ainda ser usado para comunicação entre dois microcontroladores.

Sinal	Pino Arduino Uno	Pino Arduino Mega	Pino Rfid
Reset	9	5	RST
SPI SS	10	53	DAS
SPI MOSI	11	51	MOSI
SPI MISO	12	50	MISO
SPI SCK	13	52	SCK
IRQ	Não conectado	Não conectado	IRQ
3.3 Volts	3.3 V	3.3 V	3.3 V
GND	GND	GND	GND

Tabela 4: Pinagem do Módulo Rfid Mfrc522

Fonte: Próprio Autor

Baseado na Figura 30 é possível notar que o MISO, MOSI e SCK estão disponíveis em um local físico sobre o pinagem ICSP do microcontrolador; isto é útil, por exemplo, na concepção de se usar um *shield* para trabalhar com a placa.



Figura 30: Pinos ICSP

Fonte: <http://arduino.cc/en/uploads/Reference/ICSPHeader.jpg>

Por fim, o custo dessa tecnologia vem caindo demasiadamente com o passar dos anos, a ponto dos leitores Rfid custarem aproximadamente 30 reais. Da mesma forma, não é difícil conectá-lo ao Arduino. Como resultado, muitos projetos interessantes podem ser criados com eles.

3.3.4 Sensor de Temperatura Termistor NTC

Para proposta de Automação Residencial em questão, um dos fitos era obter a temperatura do ambiente. Com base no exposto, foi feito um levantamento relacionado aos termistores do tipo PTC (*Positive Temperature Coefficient*) e TNC (*Negative Temperature Coefficient*). Entretanto, no que diz respeito as suas funcionalidades, o PTC tem o coeficiente de temperatura positivo, ou seja, a resistência aumenta com o aumento da temperatura, por outro lado no TNC quando ocorre o aumento da Temperatura a resistência diminui, e com isso este sensor oferece estabilidade mecânica, térmica e elétrica, juntamente com um elevado grau de sensibilidade. A excelente combinação de baixo custo e desempenho levou ao amplo uso do Termistor NTC nas mais diversas aplicações que envolvam medição de temperatura.

O Sensor de Temperatura NTC, conforme versa a Figura 31, foi construído a partir de um artefato resistor com coeficiente negativo e termicamente sensível, cujas misturas de manganês, cobalto óxidos de metais de transição, cobre e níquel, apresentaram variação de resistência ôhmica em relação à temperatura submetida.

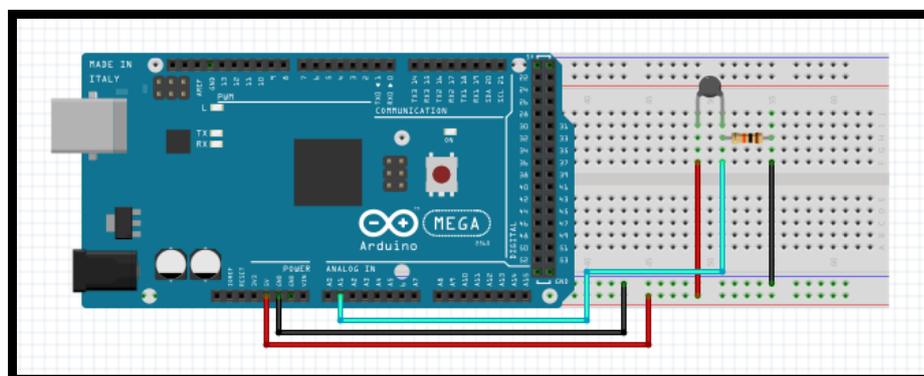


Figura 31: Conexão entre o Termistor TNC e o Arduino Mega 2560

Fonte: Próprio Autor

Destarte, para fazer a leitura da temperatura do interior da Residência, ora representada pela maquete (Figura 19), foi necessário utilizar o Termistor TNC com uma resistência de 10 k Ω , cujo valor nominal é dado normalmente a 25 $^{\circ}$ Celsius e a variação do mesmo está compreendida entre o intervalo de - 40 $^{\circ}$ a 125 $^{\circ}$ Celsius. Todavia, este é um tipo de resistência que altera seu valor em razão da temperatura onde o componente é colocado.

3.3.5 Resistor LDR

O sensor de luminosidade LDR (*Light Dependent Resistor*), conforme listado na Figura 32, é um tipo de resistor feito de sulfeto de cádmio (CdS) ou seleneto de cádmio (CdSe) cuja resistência altera em decorrência da magnitude de radiação eletromagnética que é incidida sobre ele, ou seja, como implica seu nome, esse dispositivo é um resistor que depende de fótons (luz).

Assim sendo, o transdutor de entrada (LDR) que converte fótons em valores de resistência é demasiadamente utilizado em fotocélulas que controlam o acionamento das luzes dos postes de iluminação, bem como as luzes em residências, sendo ainda bastante utilizado em sensores fotoelétricos.

Cada um dos terminais do LDR vai para um eletrodo. Entre um material mais escuro, formando uma linha sinuosa entre os eletrodos, está o material fotocondutor. O componente tem um revestimento de plástico transparente ou vidro. Quando a luz atinge o material fotocondutor, ele perde sua resistência, permitindo que mais corrente flua entre os eletrodos (MCROBERTS, 2013, p. 117).

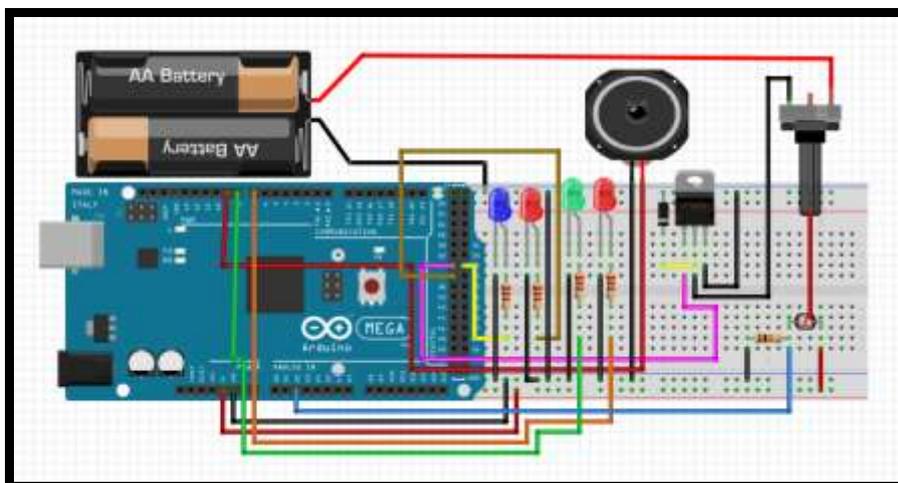


Figura 32: Conexão entre o Resistor LDR, Laser, Speaker e o Arduino Mega
Fonte: Próprio Autor

Entretanto, para este projeto, o resistor LDR será utilizado no Sistema de Segurança, onde no momento de ativação do Alarme via aplicativo (estado armado) será disparado um laser vermelho no referenciado resistor, que detectará quando uma pessoa interromper o feixe de fótons (laser), ora por ele recebido, disparando assim um alarme sonoro através do *Speaker* conectado ao Arduino Mega 2560, com o fito de reprimir o invasor. A figura 32 apresenta o modelo utilizado no projeto.

3.3.7 Sensor de Gás MQ-05

O sensor de Gás MQ-05 foi utilizado neste projeto por além de se apresentar como uma solução de baixo custo, quando o assunto é detecção dos mais variados tipos de gases combustíveis, o mesmo apresenta um alto grau de sensibilidade aos gases metano (CH_4), propano (C_3H_8) e butano (C_4H_{10}), que comumente são utilizados nas cozinhas de ambientes domésticos e industriais. A figura 34 aborda o modelo utilizado no projeto.



Figura 34: Sensor de Gás MQ-05

Fonte: https://www.futurlec.com/Gas_Sensors.shtml

Entretanto, tomando como base o tempo de resposta do sensor em questão, medidas podem ser tomadas o mais rápido possível, evitando assim um iminente sinistro no local. Além disso, conforme abordamos a Figura 35, a sensibilidade do NQ-05 pode ser ajustada de acordo com o valor do resistor utilizado.

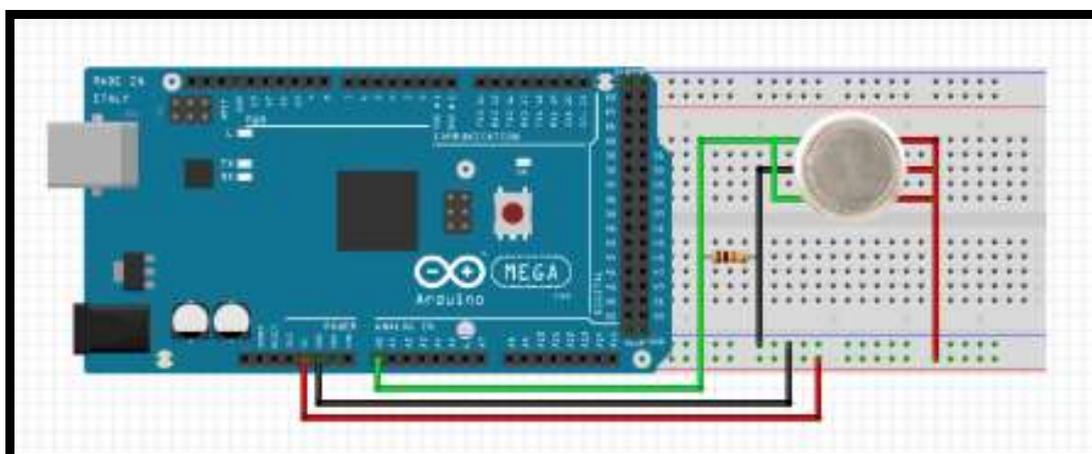


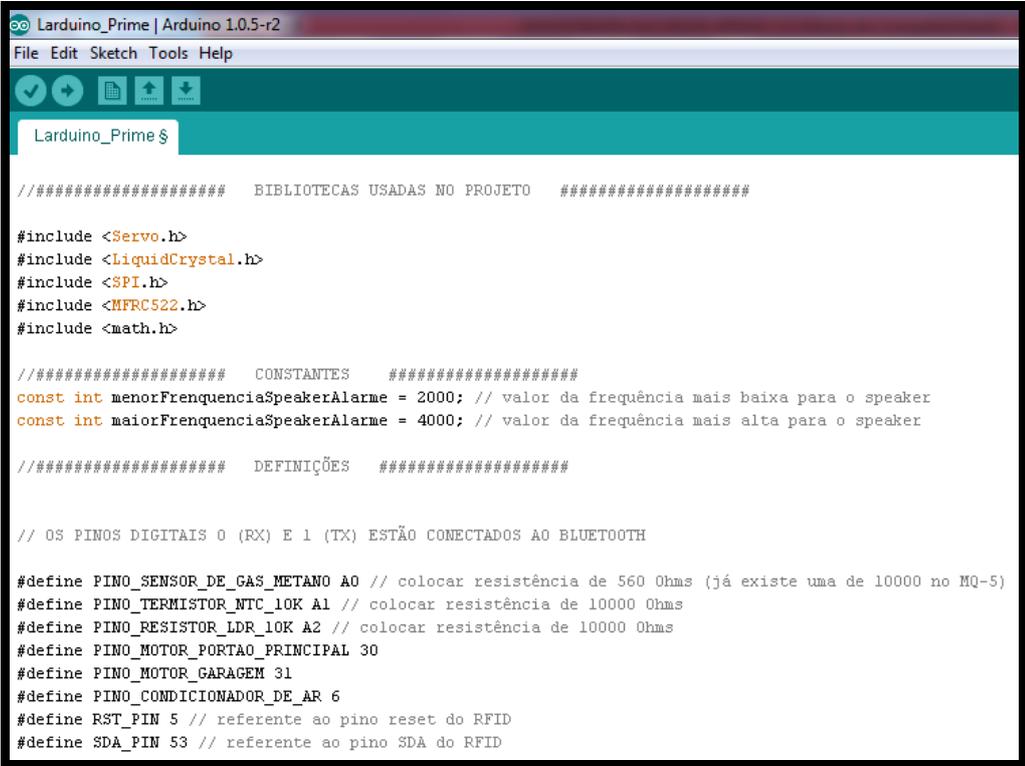
Figura 35: Conexão entre o Sensor de Gás MQ-05 e o Arduino Mega 2560

Fonte: Próprio Autor

Destarte, na nossa proposta de Automação Residencial, caso o sensor NQ-05 detecte algum vazamento de gás butano no ambiente, será emitido um alerta sonoro e visual para o usuário, por consequência o sistema de fornecimento de gás (C4H10) será desativado. Parece cenário de ficção científica mais na verdade não, a chamada Automação Residencial veio para ficar.

3.4 DESCRIÇÃO DO SKETCH DESENVOLVIDO PARA O PROJETO

Conforme versa a Figura 36, o *sketch* desenvolvido para este projeto passará a gerir toda a parte física dos dispositivos conectados ao microcontrolador. No entanto, através do uso de módulos de *hardware* e bibliotecas de *software*, foi possível expandir as funcionalidades do projeto, pois as bibliotecas disponibilizaram funcionalidade, tais como escrever um texto em uma tela LCD, movimentar um Servo Motor, ou mesmo interagir com periféricos SPI.



```

Larduino_Prime | Arduino 1.0.5-r2
File Edit Sketch Tools Help

Larduino_Prime $

##### BIBLIOTECAS USADAS NO PROJETO #####

#include <Servo.h>
#include <LiquidCrystal.h>
#include <SPI.h>
#include <MFRC522.h>
#include <math.h>

##### CONSTANTES #####
const int menorFrequenciaSpeakerAlarme = 2000; // valor da frequência mais baixa para o speaker
const int maiorFrequenciaSpeakerAlarme = 4000; // valor da frequência mais alta para o speaker

##### DEFINIÇÕES #####

// OS PINOS DIGITAIS 0 (RX) E 1 (TX) ESTÃO CONECTADOS AO BLUETOOTH

#define PINO_SENSOR_DE_GAS_METANO A0 // colocar resistência de 560 Ohms (já existe uma de 10000 no MQ-5)
#define PINO_TERMISTOR_NTC_10K A1 // colocar resistência de 10000 Ohms
#define PINO_RESISTOR_LDR_10K A2 // colocar resistência de 10000 Ohms
#define PINO_MOTOR_PORTAO_PRINCIPAL 30
#define PINO_MOTOR_GARAGEM 31
#define PINO_CONDICIONADOR_DE_AR 6
#define RST_PIN 5 // referente ao pino reset do RFID
#define SDA_PIN 53 // referente ao pino SDA do RFID

```

Figura 36: Sketch Desenvolvido para o projeto de Automação Residencial

Fonte: Próprio Autor

Destarte, existem três tipos diferentes de bibliotecas do Arduino: padrão, essencial (*core*) e de terceiros. Entretanto, algumas destas podem ser utilizadas por si próprias, enquanto outras necessitam ser usadas com componentes eletrônicos adicionais, frequentemente na forma de módulos ou *shields*.

Assim sendo, a biblioteca essencial além de mascarar muito da complexidade tradicionalmente envolvida quando se trabalha com o microcontrolador é aquela constituída dentro da IDE do Arduino.

Não obstante, as bibliotecas padrão são aquelas que já estão introduzidas por padrão nos projetos, pois já se encontram incluídas na IDE do Arduino no momento de sua instalação, ou seja, são as que a equipe de desenvolvimento do Arduino deduz ser de suma importância para o desenvolvimento dos projetos da maioria das pessoas. Em contrapartida, as bibliotecas de terceiros são aquelas disponibilizadas por usuários do Arduino, porém não são distribuídas como padrão com a IDE do Arduino.

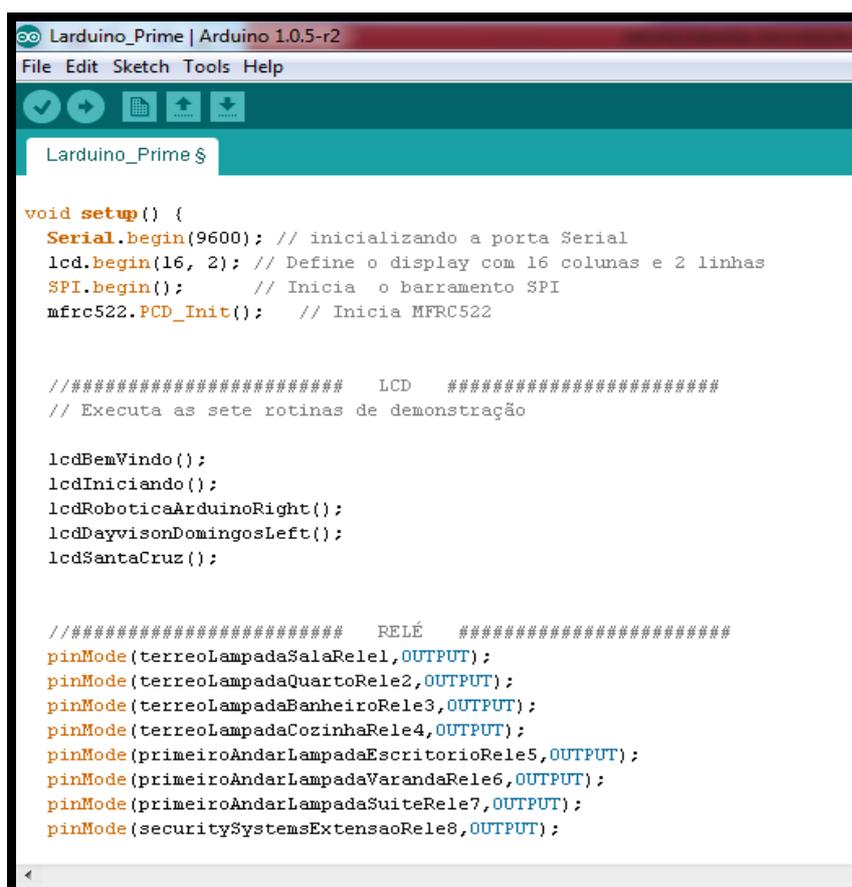
Por outro lado, como pode ser observado na Figura 36, fizemos o uso de algumas bibliotecas, dentre as quais destacamos: **Servo.h**, a qual propicia ao Arduino Mega controlar até 48 Servos Motores, todavia, em nosso projeto a biblioteca foi utilizada para controlar apenas dois servos motores, presentes respectivamente na garagem e portão principal do protótipo; já a biblioteca **LiquidCrystal.h**, foi utilizada no display visando propiciar um feedback do que estaria acontecendo no ambiente. Para o módulo Rfid MFRC522, fizemos o uso da biblioteca **MFRC522.h** por ser a mais indicada no que diz respeito a manipulação do *hardware*. Por fim, destacamos a biblioteca **SPI.h** (*Serial Peripheral Interface*), por ser responsável pela comunicação entre dois dispositivos, além de poder ser empregada para comunicar com uma variedade de periféricos ou sensores, cujo protocolo usa quatro fios, três dos quais são comuns a cada dispositivo, e um que é escolhido como sendo o escravo.

Neste projeto utilizamos a linguagem de programação baseada em C/C++, na qual o Arduino revela o lado de software, e explica seu desenvolvimento constituído a partir de funções contidas em seu próprio ambiente.

Deste modo, conforme pode ser observado nas Figuras 37 e 38 respectivamente, o sketch do Arduino apresenta como funções principais a *setup()* e a *loop()*, as quais são imprescindíveis para o funcionamento do programa.

A função `setup()`, por meio da inserção de instruções gerais, tais como a definição dos modos dos pinos, ou mesmo a taxa de transmissão serial, tem como objetivo preparar todo o programa antes que o *loop* principal seja executado.

Neste projeto, a função `setup()` além de inicializar o barramento SPI, o MFRC522 e a porta Serial com a taxa de transmissão 9.600 *baud* por segundo em que alterações de estado ou *bits* (dados) são enviadas para o Arduino, define o *display* LCD como tendo 16 colunas e 2 linhas. Outrossim, de acordo com a Figura 37, é possível notar que função além de executar 7 rotinas de demonstração com o visor LCD, define o modo como o pinos digitais e analógicos utilizados no projeto estão disponíveis para o microcontrolador, bem como, via resistor LDR o nível de calibração de luminosidade do meio ambiente para o Sistema de Segurança.



```

Larduino_Prime | Arduino 1.0.5-r2
File Edit Sketch Tools Help

Larduino_Prime $

void setup() {
  Serial.begin(9600); // inicializando a porta Serial
  lcd.begin(16, 2); // Define o display com 16 colunas e 2 linhas
  SPI.begin(); // Inicia o barramento SPI
  mfrc522.PCD_Init(); // Inicia MFRC522

  //##### LCD #####
  // Executa as sete rotinas de demonstração

  lcdBemVindo();
  lcdIniciando();
  lcdRoboticaArduinoRight();
  lcdDayvisonDomingosLeft();
  lcdSantaCruz();

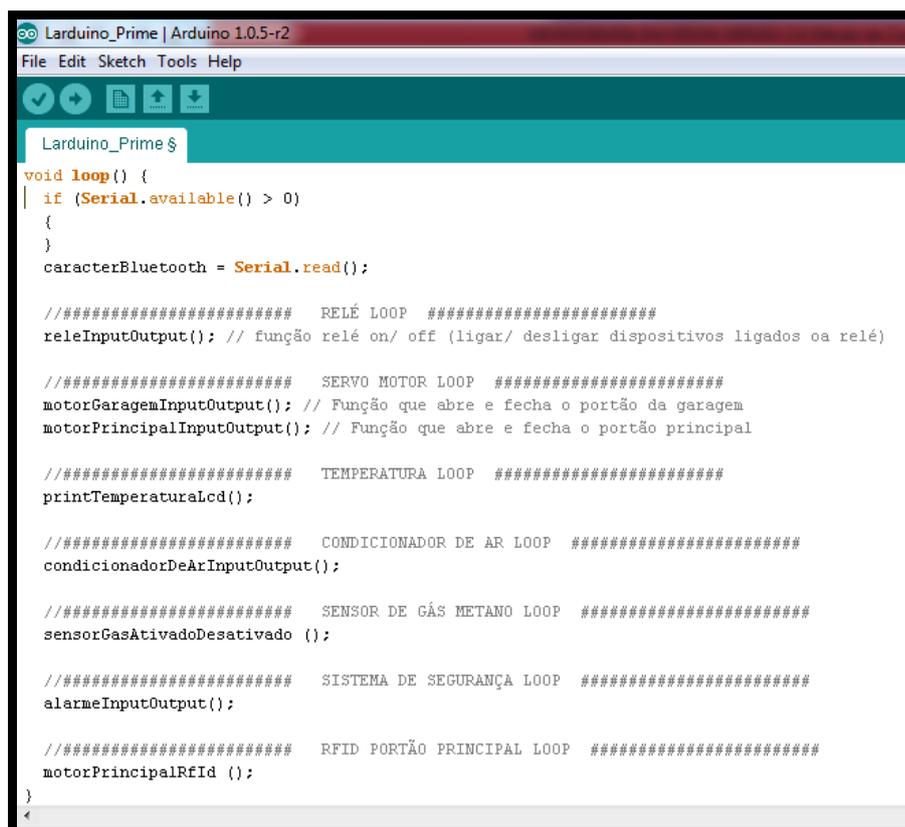
  //##### RELÉ #####
  pinMode(terreoLampadaSalaRele1,OUTPUT);
  pinMode(terreoLampadaQuartoRele2,OUTPUT);
  pinMode(terreoLampadaBanheiroRele3,OUTPUT);
  pinMode(terreoLampadaCozinhaRele4,OUTPUT);
  pinMode(primeiroAndarLampadaEscritorioRele5,OUTPUT);
  pinMode(primeiroAndarLampadaVarandaRele6,OUTPUT);
  pinMode(primeiroAndarLampadaSuiteRele7,OUTPUT);
  pinMode(securitySystemsExtensaoRele8,OUTPUT);

```

Figura 37: Função `void setup` do sketch `Larduino_Prime`

Fonte: Próprio Autor

A função *loop()*, é considerada a função principal do programa, haja vista, que todas as declarações contidas dentro do escopo serão executadas uma de cada vez, até que se alcance o fim da função; nesse instante, todas as instruções contidas no loop serão reiniciadas desde o começo e assim infinitamente, ou até que o botão *reset* seja pressionado ou o Arduino desligado.



```

Larduino_Prime | Arduino 1.0.5-r2
File Edit Sketch Tools Help

Larduino_Prime §
void loop() {
|  if (Serial.available() > 0)
|  {
|  }
|  caractereBluetooth = Serial.read();

|  ##### RELÉ LOOP #####
|  releInputOutput(); // função relé on/ off (ligar/ desligar dispositivos ligados ao relé)

|  ##### SERVO MOTOR LOOP #####
|  motorGaragemInputOutput(); // Função que abre e fecha o portão da garagem
|  motorPrincipalInputOutput(); // Função que abre e fecha o portão principal

|  ##### TEMPERATURA LOOP #####
|  printTemperaturaLcd();

|  ##### CONDICIONADOR DE AR LOOP #####
|  condicionadorDeArInputOutput();

|  ##### SENSOR DE GÁS METANO LOOP #####
|  sensorGasAtivadoDesativado ();

|  ##### SISTEMA DE SEGURANÇA LOOP #####
|  alarmeInputOutput();

|  ##### RFID PORTÃO PRINCIPAL LOOP #####
|  motorPrincipalRfid ();
}

```

Figura 38: Função *void loop* do sketch *Larduino_Prime*

Fonte: Próprio Autor

Com base na Figura 38, é possível notar que a função *loop()* faz a chamada de outras funções nas quais estão relacionadas diretamente com as funcionalidades dos dispositivos conectados ao microcontrolador.

Diante do exposto, as funções: ***releInputOutput()***, faz o acionamento dos relés ligando as lâmpadas da maquete, ***motorGaragemInputOutput()***, tem como objetivo abrir e fechar o portão da garagem do protótipo, ***printTemperaturaLcd()*** retorna para o usuário via *display* LCD a temperatura em graus Célsius e Fahrenheit, ***motorPrincipalInputOutput()*** tem como objetivo abrir e fechar o portão principal, ***condicionadorDeArInputOutput()*** além de retornar a temperatura em graus Célsius

via display LCD, liga e desliga o *cooler*, ora representando um condicionador de ar instalado na residência, ***sensorGasAtivadoDesativado()*** faz o uso do sensor de Gás MQ-05 para detectar vazamento de gás butano (C₄H₁₀) na residência, a ***motorPrincipalRfid()*** tem como fito controlar o acesso e identificação de pessoas no ambiente por intermédio de crachás ou tags do tipo chaveiro, liberando ou não a entrada do usuário no portão principal, em contrapartida, a ***alarmeInputOutput()*** está diretamente relacionada a ativação do Sistema de Segurança no que diz respeito a detecção de intrusos no ambiente.

3.5 TESTE E RESULTADOS

Uma residência automatizada não é aquela que simplesmente apaga ou acende lâmpadas por intermédio de um dispositivo móvel, mas sim aquela em que os dispositivos são capazes de identificar as mutabilidades no ambiente e com isso gerenciar as informações, resultando em economia.

Objetivando expor os conceitos e demonstrar que uma Automação Residencial alternativa de baixo custo é possível, foi de suma importância à realização de testes, buscando adaptar a programação do *sketch* do Arduino para que junto com seus componentes fosse aproveitada toda a sua eficácia.

Nada obstante, no limiar do projeto, o fito era apenas fazer o uso do microcontrolador Arduino Uno para acender e apagar algumas lâmpadas na maquete, entretanto ao decorrer da implementação do protótipo, cujo se encontra nos moldes de uma casa real, foi constatado que o Arduino na sua versão Uno, apresenta algumas limitações, quando o assunto é quantidade de portas analógicas e digitais.

Outrossim, para solucionar este empasse se fez necessário a adoção do Arduino Mega 2560, haja vista a quantidade de portas analógicas e digitais disponíveis para controlar toda a parte externa e interna do protótipo, sem jamais esquecer a robustez que o mesmo propicia no que diz respeito ao seu desempenho, quando comparado ao Arduino Uno.

No entanto, durante os testes com o módulo *bluetooth* RS232 HC - 06, verificamos que não foi possível conectar ao mesmo tempo mais de um dispositivo *Android* com o módulo em questão, ou seja, para conectar outro aparelho se fez

necessário desemparelhar o dispositivo ora conectado ao bluetooth, fator no qual nos motivará para expandir a proposta para internet em estudos futuros.

Em seguida, além da implantação de um sistema de iluminação, pensando no controle de acesso e identificação de pessoas no ambiente, foi cogitada a ideia de se colocar um sensor de distância (Ultrassônico HC-SR04) com o módulo de identificação por rádio frequência (Rfid Mfrc522), ou seja, quando o usuário chegasse a uma distância menor ou igual a 50 centímetros do local indicado para colocar os crachás do tipo S50 ou mesmo as tags do tipo chaveiro o Display LCD exibiria uma mensagem ao usuário para colocar as identificações no local indicado.

Entretanto, devido à capacidade constante de comunicação entre o sensor Ultrassônico HC-SR04 e o Arduino Mega 2560, o programa como um todo passou a apresentar um delay de aproximadamente 5 segundos, no que diz respeito ao seu tempo de resposta quando acionada alguma funcionalidade. Contudo como uns dos objetivos desta proposta de Automação Residencial são o aumento da segurança, comodidade, praticidade e conforto, independente de status social, um retorno rápido é fundamental.

Diante do exposto, se fez necessário retirar o Sensor Ultrassônico do projeto, no qual ficará para estudos futuros, contudo mantivemos a ideia do controle de acesso e identificação de pessoas via rádio frequência, bem como, toda a parte na qual propiciará ao usuário a possibilidade de controle de iluminação, temperatura ambiente, condicionador de ar, sistema de segurança, acionamento de portão eletrônico e Sensor detector de vazamento de gás butano (C4H10).

Pensando em uma maior comodidade para o usuário, disponibilizamos no aplicativo o comando de acionamento por voz, todavia, ao ser testado em um ambiente com barulho em excesso como, por exemplo, em uma sala de estar com várias outras pessoas falando ao mesmo tempo, o comando de acionamento por voz se fez ineficaz, devido ao ruído das demais vozes do ambiente.

Por fim, após as conexões dos circuitos eletrônicos e o desenvolvimento da programação do software para o *sketch* da plataforma de prototipagem utilizada no projeto e para o aplicativo *Android Larduino Prime*, os testes confirmam as hipóteses discutidas neste Trabalho de Conclusão de Curso. O protótipo foi desenvolvido em etapas e testado conforme o avanço dos resultados obtidos do mesmo. Os testes foram de suma importância para validar a integração das diferentes partes da Proposta de Automação Residencial, evitando assim, problemas de funcionamento.

4. CONCLUSÃO

Deste modo, diante do exposto apresentado, iremos chegar à conclusão que o desempenho será ótimo, tendo em vista os materiais de baixo custo utilizados. O aplicativo desenvolvido será intuitivo e fácil de usar. O sistema como um todo ficará simples, objetivo e rápido fazendo que o usuário possa utilizar sem problemas.

As soluções comerciais de automação residencial são **tipicamente proprietárias**, baseadas em um dispositivo de controle específico com um software embarcado e que possuem custo bastante elevado. Em algumas destas soluções, somente o hardware de controle tem custo superior a R\$ 10.000,00 (dez mil reais).

A finalidade deste foi o desenvolvimento de um Sistema de Automação Residencial de baixo custo, no qual propiciaria aos portadores de paraplegia ou não aumentar a segurança, comodidade e conforto independente de status social.

O conceito de Automação Residencial está mudado. Temos ao nosso alcance uma gama de possibilidades práticas e econômicas que utilizam a automação, desde a básica até a mais otimizada, em sistemas de integração para diversos ambientes. O resultado é um ambiente prático, agradável, mais bonito, valorizado e seguro, tudo isso de acordo com o interesse do usuário, visando sempre a praticidade, simplicidade e objetividade dos comandos. Todas estas funções sem se desfazer da beleza e valorização o ambiente.

Por fim, além das etapas já concluídas nesta proposta de Automação Residencial, é esperado para estudos futuros implementar inúmeras melhorias e complementos. Realizar manipulações mais complexas com eletrônicos, interação com multimídia e internet, melhorar a experiência do usuário, e sobretudo no quesito referente ao domínio e utilização dos equipamentos são metas tangíveis para próximas etapas de desenvolvimento.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, R. **A tecnologia por trás da mágica**. Disponível em: <<http://quicaze.com/126/atecnologia-por-tras-da-magica/>>. Acessado em: 23 de outubro 2014.

ANGEL, P. M. **Introducción a la domótica; Domótica: controle e automação**. Escuela Brasileño - Argentina de Informática. EBAI. 1993.

AURESIDE (Associação Brasileira de Automação Residencial). Temas técnicos: **Segurança, Sistemas de Segurança**. Disponível em: <<http://www.aureside.org.br/temastec/default.asp?file=seguranca.asp>>. Acesso em: 14 de junho de 2014.

AURESIDE (Associação Brasileira de Automação Residencial). Temas técnicos: **Software, Introdução aos Sistemas de Comando de Voz**. Disponível em: <<http://www.aureside.org.br/temastec/default.asp?file=softwares02.asp>>. Acesso em: 14 de junho de 2014.

BANZATO, Marco O. **Controle de Iluminação e suas Aplicações**. São Paulo, 2002.

BOLZANI, Caio Augustus Moraes. **Residências inteligentes - domótica, redes domésticas, automação residencial**. Editora e Livraria da Física. São Paulo, 2004.

CAMARGO, A. R. **As Interfaces Telemáticas dos Serviços Urbanos da Urbanização Virtual: Estudo das Redes de Infra-estrutura, dos Edifícios Inteligentes, dos Ambientes Cognitivos e Comunicativos de Trabalho para as Unidades Terminais dos Serviços Urbanos**. artigo proposto ao ENTAC-98. S. Carlos, 1998.

CANDITO, Decio A. **Utilização de Conceitos de Integração de Sistemas Direcionados a Domótica - Estudo de Caso para Automação Residencial**. Dissertação (Mestrado em Engenharia Mecânica). Universidade Estadual de Campinas. São Paulo, 2007.

CHAMUSCA, A. **Domótica e Segurança Electrónica : a inteligência que se instala**. Ingenium Edições, Portugal, 2006.

CUNHA, A. F.. **Saber Eletrônica**. Disponível em: <<http://techtrainig.eng.br/conteudo/ARTIGO-SIST-BEM.pdf>>. Acesso em: 15 de novembro de 2014.

DIAS, César Luiz de Azevedo. **Domótica: aplicabilidade às edificações residenciais**. (Dissertação de mestrado) Universidade Federal Fluminense. Niterói, 2004.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. Editora Novatec. São Paulo, 2013.

HENRIQUE, Paulo P. **Funcionamento de um Controlador Lógico Programável (CLP)**. Disponível em: <http://www.pharmaster.com.br/artigos/docs/20080703_7439_Funcionamento%20de%20um%20CLP.pdf>. Acessado em 10 de setembro de 2014.

LANG, Mariana. **Sistemas de automação residencial transferem para tablet e celular o controle da casa**. Folha de São Paulo. Disponível em: <<http://lightingnow.com.br/blog/sistemas-de-automacao-residencial-transferem-para-tablet-e-celular-o-controle-da-casa>>. Acesso em: 18 de julho de 2014.

MCROBERTS, Michael. **Arduino Básico**, Editora Novatec, São Paulo, 2013.

MEDRADO, V. **Assessoria de Imprensa Serifa Comunicação**. Disponível em: <http://www.farolcomunitario.com.br/uberlandia_100_0046.htm>. Acesso em: 20 de julho de 2014.

MORAES, C. C.; CASTRUCCI, P. L. **Engenharia de automação industrial**. Editora LTC. Rio de Janeiro, 2001.

MOYA, J. M.; TEJEDO, R. J. **Manual de Domótica**. Disponível em: <http://books.google.com.br/books?id=V6lzqqDcfF8C&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em: 16 de junho de 2014.

MURATORI, José Roberto. **Automação Residencial: Histórico, definições e conceitos**. Disponível em: <http://www.instalacoeseltricas.com/download/Automacao_residencial1.pdf>. Acessado em: 14 de junho de 2014.

OLIVEIRA, J.P. **Domótica: Perspectiva da Plataforma Arduino**. Monografia (Conclusão de Curso) Universidade Estadual de Goiás. Goianésia, 2012.

SGARBI, Julio André; TONIDANDEL, Flavio. **Domótica Inteligente: Automação Residencial baseada em Comportamento**. Centro Universitário da FEI. São Bernardo do Campo - SP, 2007.

SILVA, Marcos Aurélio da. **A lei de Ohm**. Disponível em: <<http://www.brasilecola.com/fisica/a-lei-ohm.htm>>. Acesso em: 15 de outubro de 2014.

SILVEIRA, Paulo R. da; SANTOS, Winderson E. . **Automação e Controle Discreto**. Editora Érica. São Paulo, 1998.

TANENBAUM, Andrew S. **Redes de Computadores**. Ed. Elsevier. Tradução autorizada do idioma inglês da edição publicada por Pretince Hall. Rio de Janeiro, 2003.

TERUEL, E. C. **Uma proposta de framework para sistemas de automação residencial com interface para WEB.** Dissertação (Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação). Centro Estadual de Educação Tecnológica Paula Souza. São Paulo, 2008.

TEZA, V. R.. **Alguns Aspectos Sobre A Automação Residencial – Domótica.** Programa de Pós-graduação em ciência da Computação. UFSC. Florianópolis-SC, 2002.

WERNECK, Siva Bianchi de Frontin. **Domótica: União de arquitetura e tecnologia da informação na edificação residencial urbana.** (Tese de Mestrado em Arquitetura) Universidade Federal do Rio de Janeiro. Rio de Janeiro, 1999.

WORTMEYER, C.; FREITAS, F.; CARDOSO, L. **Automação residencial: Busca de tecnologias visando o conforto, a economia, a praticidade e a segurança do usuário.** II Simpósio de Excelência em Gestão e Tecnologia SEGeT2005. Rio de Janeiro, 2005.