

**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ
CAMPUS DE ITAITUBA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ELIEZER COSTA SILVA
JANDERSON SANTOS BENTO**

**DOMÓTICA: Uma visualização com microcontrolador
Arduino**

**ITAITUBA-PA
2014**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ
CAMPUS ITAITUBA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ELIEZER COSTA SILVA
JANDERSON SANTOS BENTO

DOMÓTICA: UMA VISUALIZAÇÃO COM MICROCONTROLADOR
ARDUINO

Trabalho Acadêmico de Conclusão de Curso apresentado ao Colegiado de Informática do Instituto Federal de Educação, Ciência e Tecnologia do Pará – IFPA Campus Itaituba, como requisito para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Esp. Michel Marialva Yvano.

ITAITUBA-PA

2014

Dados Internacionais de Catalogação na Publicação (CIP)

S586d Silva, Eliezer Costa

Domótica: uma visualização com microcontrolador Arduino. / Eliezer Costa Silva, Janderson Santos Bento. 2014.

94 f. il; color, 31 cm.

TAC (Graduação) – Instituto Federal do Pará. Campus Itaituba, Coordenação de Informática – Curso de Tecnologia em análises e desenvolvimento de sistemas, Itaituba, 2014.

Orientador Profº. Esp. Michel Marialva Yvano.

Inclui bibliografia.

1. Domótica. 2. Gerenciamento. 3. Arduino. I. Yvano, Michel Marialva, orientador.

II. Bento, Janderson Santos. III. Título

CDD: 004.383

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ
CAMPUS ITAITUBA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ELIEZER COSTA SILVA
JANDERSON SANTOS BENTO

DOMÓTICA: UMA VISUALIZAÇÃO COM MICRO CONTRALADOR
ARDUINO

Data da Defesa: ____/____/____

Conceito: _____

Banca Examinadora

Prof. Orientador: Esp. Michel Marialva Yvano – IFPA

Prof. . Esp. Sorivan Albuquerque Pena - IFPA – Itaituba

Prof. Me. Jefferson Alves Teixeira – IFPA – Itaituba

Dedicamos primeiramente a Deus, porque Ele é a fonte eterna de todas as coisas e Ele nos deu a oportunidade de chegamos até aqui.

Dedicamos esta, como todas as demais conquistas das nossas vidas aos nossos amados Pais. Aos nossos irmãos, familiares, amigos pessoais e de curso, que motivaram e ajudaram para o alcance dessa vitória.

Dedicamos a vocês!

AGRADECIMENTOS

Agradecemos primeiramente a Deus, pois o que seria de nós sem a Fé que temos Nele. Ao orientador Professor Esp. Michel Marialva Yvano pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia. A todos os professores do curso, que foram importantes na vida acadêmica e aos amigos e colegas, pelo incentivo e pelo apoio constante.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”.

Marthin Luther King

RESUMO

Este trabalho consiste de uma visualização da Domótica, através do microcontrolador Arduino. A Domótica proporcionar conforto, segurança, acessibilidade dentre outras comodidades, além de adequar eficiência na utilização dos recursos energéticos e hídricos contribuindo com o desenvolvimento sustentável. A falta de gerenciamento no consumo energético em algumas atividades domésticas acarreta alto consumo, que por sua vez representa a maior fatia de gastos a nível global, ultrapassando os transportes e a indústria. Como toda nova tecnologia a Domótica inicialmente foi vista pela grande maioria como símbolo de *status*, porém, esse pensamento deve ser deixado para trás, principalmente pelos benefícios que a mesma traz para a sociedade. Ela é capaz ainda de otimizar variáveis do ambiente externo e interno, transformando a residência em um local confortável, trazendo benefícios vinculados a uma boa iluminação, climatização adequada e monitoramento da residência, dentre outras utilidades. Para o desenvolvimento de sistemas na área da Domótica é necessário utilizar microcontroladores, com isso optou-se pela utilização do Arduino que é bastante utilizado pela facilidade de utilização, programação e seu baixo custo.

PALAVRAS-CHAVE: Domótica. Gerenciamento. Arduino. Segurança.

ABSTRACT

This work consists of a preview of Home Automation via the Arduino microcontroller. The Home Automation provide comfort, safety, accessibility, among other amenities, in addition to adapting the efficient use of energy and water resources contributing to sustainable development. The lack of management in the energy consumption in some household activities entails high consumption, which in turn represents the largest share of spending globally, surpassing the transport and industry. As with any new technology Home Automation was initially seen by most as a status symbol, but that thought should be left behind, especially the benefits that it brings to society. She is also able to optimize variables of the external and internal environment, transforming the house into a comfortable place, with benefits linked to good lighting, air conditioning and adequate monitoring of residence, among other utilities. For the development of systems in the area of Home Automation is necessary to use microcontrollers, it was decided to use the Arduino which is widely used for its ease of use, programming and its low cost.

KEYWORDS: Home Automation. Management. Arduino. Security.

LISTA DE ILUSTRAÇÕES

Figura 01 – Inteligência: Natural e Artificial	25
Figura 02 – Placa de automação – Placa gravadora.....	29
Figura 03 – Placa de automação – Placa RCOM.....	29
Figura 04 – Placa de automação – Placa CNC android	30
Figura 05 – Placa de automação – Placa de portão automático	30
Figura 06 – Placa de automação – Placa de controle de velocidade	31
Figura 07 – Sistemas embarcados em veículos.....	35
Figura 08 – <i>Data Logger</i> para temperatura do ar	36
Figura 09 – Aparelhos de interação com o usuários	37
Figura 10 – Controle de manipuladores robóticos.....	38
Figura 11 – MATLAB, <i>software</i> interativo de processamento de sinais	39
Figura 12 – Diagrama de bloco de uma cadeia de processamento utilizando o Arduino.....	41
Figura 13 – Placa, Arduino Uno	42
Figura 14 – Placa, Arduino Ethernet	43
Figura 15 – Placa, Arduino ADK.....	43
Figura 16 – Placa, Arduino LilyPad	44
Figura 17 – Placa, Arduino Leonardo.....	44
Figura 18 – Placa, Arduino Yun.....	45
Figura 19 – Placa, Arduino Ter.....	45
Figura 20 – Placa, Arduino Micro	46

Figura 21 – Cobertura Arduino GSM.....	47
Figura 22 – Cobertura Arduino Ethernet	47
Figura 23 – Cobertura Arduino Wi-Fi.....	48
Figura 24 – Cobertura Arduino SD sem fio	48
Figura 25 – Cobertura Arduino Motor	49
Figura 26 – Cobertura Arduino sem fio Proto	50
Figura 27 – Componente, pinagem	53
Figura 28 – Componente, LEDs	54
Figura 29 – Componente, potenciômetro	54
Figura 30 – Componente, <i>push-button</i>	55
Figura 31 – Componente, sensor de luminosidade	55
Figura 32 – Componente, sensor de temperatura.....	56
Figura 33 – Componente, sensor infravermelho	56
Figura 34 – Componente, servo motor	57
Figura 35 – Componente, <i>displays</i>	57
Figura 36 – Microcontrolador Arduino Uno R3. Parte superior e inferior	63
Figura 37 – Shield (Cobertura) de internet. Parte superior e inferior	64
Figura 38 – Cabo UTP, com conector Rj45 e Roteador	66
Figura 39 – Folha de acrílico	66
Figura 40 – Pistola e silicone (cola quente).....	67
Figura 41 – Fios reciclados	67
Figura 42 – LEDs	68
Figura 43 – Protótipo da casa	68

Figura 44 – <i>Softwares</i> Arduino IDE 1.5.5	69
Figura 45 – <i>Software</i> iArduino	70
Figura 46 – Resultado obtido – comunicação	71
Figura 47 – Resultado obtido – protótipo em funcionamento	72
Figura 48 – Resultado obtido – Vista superior do protótipo.....	72
Figura 49 – Resultado obtido – ligação dos filamentos na placa Arduino	73

LISTA DE SIGLAS

CAN – *Controller Area Network* (Controlador de Área de Rede)

CDT – Centro de Desenvolvimento Tecnológico

CI – Circuito Integrado

CSE – Colégio Santa Emília

DSP – *Digital Signal Processor* (Processador Digital de Sinais)

DVD – *Digital Versatile Disc* (Disco Digital Versátil)

EUA – Estados Unidos da América

GPS – *Global Positioning System* (Sistema de Posicionamento Global)

IA – Inteligência Artificial

IBGE – Instituto Brasileiro de Geografia e Estatística

IDE – *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento)

LabMAR – Laboratório de Monitoramento Ambiental Remoto

LED – *Light Emitting Diode* (Diodo Emissor de Luz)

MIT – *Massachusetts Institute of Technology* (Instituto de Tecnologia de Massachusetts)

NASA – *National Aeronautics and Space Administration* (Administração Nacional da Aeronáutica e do Espaço)

PC – *Computer Personal* (Computador Pessoal)

PCB – *Printed Circuit Boards* (Placas de Circuito Impresso)

PDA – *Personal Digital Assistant* (Assistente Pessoal Digital)

PWM – *Pulse Width Modulation* (Modulação por Largura de Pulso)

RAM - *Random Access Memory* (Memória de Acesso Aleatório)

RJ45 – *Registered Jack 45*

ROM – *Read Only Memory* (Memória Apenas de Leitura)

SEE – Sistemas Eletrônicos Embarcados

TV - Televisão

UFMS – Universidade Federal do Mato Grosso do Sul

UFPR – Universidade Federal do Paraná

UFPEL – Universidade Federal de Pelotas

UTP – *Unshielded Twisted Pair* (Par Trançado sem Blindagem)

USB – *Universal Serial Bus* (Barramento Serial Universal)

SUMÁRIO

INTRODUÇÃO	17
1. REFERENCIAL TEÓRICO	20
1.1. DOMÓTICA	20
1.1.1. Histórico	20
1.1.2. Conceitos	21
1.1.3. Aplicações	22
1.1.4. Inteligência Artificial	23
1.2. SISTEMAS EMBARCADOS	25
1.2.1. Definição e contextualização histórica	25
1.2.2. Aplicações	28
1.2.3. Funcionalidades e Características	31
1.3. MICROCONTRALADOR ARDUINO	40
1.3.1. Descrição e análise evolutiva	40
1.3.2. Componentes	50
1.3.2.1. Hardware	52
1.3.2.2. Software	58
1.3.3. Arduino e o Sistema Embarcado	58
1.4. PROPOSTA DE PESQUISA E INOVAÇÃO	60
2. RESULTADOS	62
2.1. MATERIAS UTILIZADOS NO DESENVOLVIMENTO DO PROJETO	62
2.1.1. Componentes de comunicação	62
2.1.1.1. Arduino Uno e Shield de Internet – Visão global	63
2.1.1.2. Roteador e cabo UTP com conector RJ45	65
2.1.2. Componentes utilizados no protótipo da casa	66
2.1.3. Softwares utilizados	68

2.2.	RESULTADOS OBTIDOS	70
3.	DISCUSSÃO	74
	CONCLUSÃO	76
	REFERÊNCIAS BIBLIOGRÁFICAS	78
	APÊNDICE	
	ANEXOS	

INTRODUÇÃO

Domótica é a tecnologia que utiliza métodos pelo qual é possível controlar eventos sem um pensar consciente, ela integra o controle e a supervisão de dispositivos domésticos usando a informática e técnicas de comunicação com o intuito de obter uma casa mais confortável, segura e econômica.

Como qualquer novidade a Automação Residencial ou Domótica, inicialmente, é percebida pela população como um símbolo de *status* e modernidade. A Associação Brasileira de Automação Residencial (AURESIDE) afirma que no momento seguinte, o conforto e a conveniência por ela proporcionados passam a ser decisivos. E por fim, ela se tornará uma necessidade vital e um fator de economia.

Lagido (2009) afirma que os desperdícios de energia elétrica em atividades domésticas ocasionam altos consumos energéticos, chegando a ultrapassar os consumos dos transportes e a indústria, a nível global. Mas existem formas de gerenciar esse consumo, às vezes pequenos gestos podem ocasionar redução de consumo de energia, em outros casos demandam altos investimentos, que apesar de tudo terão retorno no longo prazo, diminuindo as faturas de energia e poupando o meio ambiente.

É necessário conscientizar a população sobre o uso racional, bem como das opções tecnológicas que auxiliam na redução do consumo energético sem tirar o conforto. A Domótica além de proporcionar conforto, segurança e acessibilidade, melhora a eficiência na utilização dos recursos energéticos e hídricos contribuindo com o desenvolvimento sustentável a médio e longo prazo, se revelando uma ferramenta muito importante no processo de economia de recursos. A evolução tecnológica e a possibilidade de controlar vários equipamentos de uma residência aperfeiçoando assim os recursos utilizados na casa causam reduções significativas no consumo de energia.

A Associação Brasileira de Automação Residencial (AURESIDE) afirma que uma das principais preocupações dos projetistas e instaladores de sistemas de Automação Residencial deve ser a integração entre eles. Os produtos modernos, embora muitas vezes de complexa tecnologia, dispõem de interfaces “amigáveis” para que possam ser operados com certa facilidade pelo usuário final.

Ainda de acordo com a AURESIDE, quando uma série de produtos trabalha sem comunicação entre si, o resultado na maioria das vezes é uma grande confusão operacional.

O Arduino é uma unidade de processamento capaz de suportar para que produtos trabalhem com comunicação entre si, através de sensores e atuadores ligados aos seus terminais de entradas, com as informações processadas computacionalmente, pode atuar no controle ou no acionamento de algum outro elemento conectado ao terminal de saída. O Arduino é uma boa ferramenta de prototipação rápida e de projeto simplificado, em um ambiente acadêmico, pode ser perfeitamente utilizado como ferramenta educacional, já que não requer do usuário conhecimentos profundos de eletrônica nem programação de dispositivos digitais específicos.

O Arduino faz parte do conceito de *hardware* e *software* livre e está aberto para o uso e contribuição de toda a sociedade. O conceito Arduino surgiu na Itália, em 2005, com o objetivo de criar um dispositivo que fosse utilizado em projetos/protótipos arquitetados de uma forma menos dispendiosa do que outros sistemas disponíveis no mercado. É uma plataforma de computação física ligada a sensores e atuadores, possibilitando construir sistemas que percebam a realidade e responder com ações. (BRASIL e RENNA, 2013)

Existe uma revolução acontecendo dentro das casas, elevando os conceitos até então conhecidos como lar e habitação. E para visualizar essa revolução foi desenvolvido um protótipo como microcontrolador Arduino para demonstrar a Domótica e o futuro das casas. Para alcançar o proposto foi revisada a literatura existente sobre automação e controle com Arduino, correlacionamos os dados da pesquisa sobre Domótica e algumas disciplinas, verificando os campos de uso e os processos da montagem até a finalização. Analisamos os dados obtidos e apresentamos a proposta de criação de um protótipo programado para a implantação de ações através de comandos materializados.

Este trabalho acadêmico é organizado e estrategicamente dividido em subtítulos onde inicialmente é acometida a introdução acerca do que será trabalhado; Posteriormente são abordadas as referências teóricas sobre a Domótica, histórico, conceitos e aplicabilidades da mesma, visualizando também um relato sobre a inteligência artificial. Sistemas Embarcados, sua contextualização histórica, aplicações, funcionalidades e características, e sobre o Microcontrolador Arduino,

apontando sua análise evolutiva, componentes de *hardware* e *software*, e um apanhado sobre exemplos aplicações simples com o Arduino em sistemas embarcados que se utilizam Arduino enquanto sistema embarcado e por fim, uma proposta de pesquisa e inovação.

Em seguida são abordados os resultantes do desenvolvimento do projeto, materiais utilizados no desenvolvimento do projeto, bem como os componentes de comunicação e os utilizados no protótipo da casa. E por fim, é realizado um trabalho metódico que evidencia o resultado final com a produção do protótipo da aplicação de forma simplificada. Consecutivamente descreve-se uma discussão à cerca da realização do projeto. E por fim a conclusão, nesta fazer-se uma análise geral sobre todo o projeto.

1. REFERENCIAL TEÓRICO

1.1 DOMÓTICA

1.1.1 Histórico

A história da Automação é tão antiga quanto à história da própria civilização. A Automação teve seu surgimento ainda nos primórdios da humanidade, ausente de uma data que se caracterize como marco.

Segundo Bravo, Alves e Saar (2009)

Considera-se automatização qualquer processo que auxilie o ser humano nas suas tarefas do dia-a-dia, sejam elas comerciais, industriais, domésticas ou no campo. Como exemplo pode-se citar o uso da roda d'água na automatização do processo de moagem, serrarias, ferrarias e trituração de grãos em geral.

No decorrer do último século, devido a um forte desenvolvimento tecnológico, a ideia de edifício estático desaparece, emergindo na sua linha evolutiva o conceito de edifício dinâmico, em que a tecnologia se conjuga com a informação residente, otimizando as capacidades do edifício na oferta de serviços. Lopez e Neto (2004) asseguram que nesta linha de pensamento distinguem-se três períodos

Primeiro até 1900, os edifícios são caracterizados por uma estrutura fixa, em que os principais materiais usados são pedra e madeira, e os serviços são reduzidos aos básicos, tais como a distribuições de águas, esgoto e iluminação. Segundo entre 1900 e 1945 aparecimento de estruturas em cimento armado e de equipamentos elétricos mais aperfeiçoados, destacando-se o uso generalizado de eletrobombas e motores elétricos. Após 1945 até a atualidade a nível estrutural não se verificam grandes alterações, nas habitações. Pelo contrário nos edifícios fornecedores de serviços, assiste-se ao aparecimento de estruturas dinâmicas, tais como paredes falsas, locais de trabalho em espaço abertos.

Nos anos 60 o aparecimento dos primeiros sistemas de conforto centralizado nos edifícios foi um fator evolutivo que teve aplicações práticas nos equipamentos de climatização. Nos anos 70 houve a divulgação dos microprocessadores tendo aplicações no desenvolvimento de equipamentos de supervisão mais sofisticados. E nos anos 80 surgem novos requisitos de conforto, segurança, transmissão de dados,

processamentos de informação tendo aplicações práticas nos sistemas de automação, sistemas de telecomunicações e sistemas computacionais. (NETO, LOPES e FREITAS, 2004)

Em todos os aspectos a sociedade tem sofrido evoluções e transformações nos últimos anos. Tais mudanças juntamente com a necessidade de oferecer flexibilidade, de se adaptar a novas tecnologias e a novos requisitos, deram origem ao aparecimento do conceito de edifício inteligente.

Nesta linha Lopes e Neto (2004) afirmam que

Um conjunto muito elevado de áreas ligadas à arquitetura exteriores e interiores, automação, conforto e gestão de edifícios, telecomunicações e infraestrutura de suporte, informática, ecologia, entre muitas outras que não interessa referir agora, precisam adaptar-se as novas tecnologias.

O que se denota é que dá inter-relação entre estas diferentes áreas, podem-se criar locais de trabalho mais confortáveis, económicos, seguros, ergonómicos e estimulantes, propiciando formas de interação mais estimulantes que levem à prestação de serviços mais eficazes e produtivos.

1.1.2 Conceitos

O conceito de edifício inteligente surgiu na década de 1980, como referido anteriormente, está associado sobre tudo ao setor terciário. O principal objetivo seria formar um sistema integrado tornando o nível de vida mais facilitado, nomeadamente nas áreas de conforto, segurança, gestão de energia e das comunicações. (NETO, LOPES e FREITAS, 2004)

Os projetos das residências convencionais, segundo Dias e Pizzolato (2004) não satisfazem por completo aos anseios dos moradores, o que se constitui num contra censo, pois habitação, por atender às necessidades básicas do ser humano como as de proteção, segurança e bem estar, é considerada como um dos bens de consumo de maior importância para a maioria das famílias. A moradia, o abrigo, o lar, deve ser prazeroso, eficiente, dignificante e, por ser um bem de grande vida útil, deve ser flexível às transformações sociais e tecnológicas. Dias (2004), acrescenta

que a Domótica, por meio do seu conjunto de multidisciplinar de aplicações, bem integrada às residências, é capaz de aumentar a qualidade de vida nelas habitada.

De acordo com Dias e Pizzolato (2004)

A automação alcançou as edificações corporativas. A cada dia, novos componentes que agregam tecnologias a ela relacionadas são aplicados as instalações prediais. A aplicação da automação predial tem demonstrado que é possível proporcionar ou ampliar benefícios em fatores como: gerenciamento técnico, conforto, economia, prevenção de acidentes e falhas de equipamentos, e também segurança aos usuários.

A Domótica é um ramo da inteligência artificial com sistemas capazes de executarem atividades à distância por dispositivos de uso contínuo apertando ou não algum botão como: ligar o ar-condicionado com uma simples chamada de telefone e utilizando aplicativos no celular ou uma página da internet acionar alarmes quando invasores tentarem entrar na residência, tudo isso com o mínimo ou nenhum esforço, tais comandos são executados por máquinas capazes de perceber o que está acontecendo e tomar decisões.

1.1.3 Aplicações

A Domótica pode ser aplicada praticamente a todas as atividades realizadas no âmbito doméstico como:

- a) Ligar, desligar e regular a iluminação do ambiente;
- b) Ligar, desligar e regular a climatização;
- c) Comando de veneziana, porta e portões elétricos;
- d) Ligar, desligar TV;
- e) Comando e controle dos eletrodomésticos;
- f) Comandar e controlar sistemas de alarmes e controle de acessos;
- g) Controlar parâmetros ambientais e atmosféricos como umidade, vento, chuva, sol;
- h) Detectar incêndios, vazamentos de gás e perda de água;
- i) Auxílio para idosos e deficientes físicos.

Essa tecnologia integra e conecta entre si todos esses tipos e instalações com monitoramento e controle dos dispositivos de dentro de uma residência. Graças a Domótica é possível deixar as residências mais acolhedoras e agradáveis para quem as reside, sendo possível regular e controlar parâmetros que indicam sensivelmente sobre a boa qualidade de vida de uma pessoa, que tem sua habitação como um local de descanso e relaxamento, garantindo também a segurança e comodidade.

Nunes (2004) garante que no tocante ao conforto, as possibilidades são imensas quando se refere a controlar a iluminação, aquecimento ou ar-condicionado, outra vertente relevante refere-se à gestão racional da energia.

Com a popularização da Domótica, é possível que todos possam usufruir de benefícios que esta tecnologia traz, contudo para que ocorra esta utilização em ampla escala é necessário que haja o trabalho conjunto de profissionais de diversas áreas do conhecimento.

Werneck (1999) afirma que

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.

Sendo assim a Domótica com seus métodos e aplicações é uma área muito interessante que tem atraído um grande número de estudiosos e adeptos, facilitando desta maneira a ampliação do desenvolvimento tecnológico e a utilização desta nova maneira de viver confortavelmente.

1.1.4 Inteligência Artificial

O homem tem a capacidade única de raciocínio e durante milhares de anos, ele procurou entender como funciona o pensar, isto é, como um mero punhado de matéria pôde compreender, perceber, prever e manipular um mundo muito maior e muito mais complexo que ele próprio. O campo da inteligência artificial (IA) vai ainda mais além, pois tenta não apenas compreender, mas também construir entidades inteligentes. (GOMES, 2010)

O termo “inteligência artificial” nasceu em 1956 no famoso encontro de Dartmouth. No final dos anos 50 e início dos anos 60, os cientistas Newell, Simon, e J. C. Shaw introduziram o processamento simbólico. Ao invés de construir sistemas baseados em números, eles tentaram construir sistemas que manipulassem símbolos. A abordagem era poderosa e foi fundamental para muitos trabalhos posteriores. (CÂMARA, 2001). Osório (1999) relata que

O termo “*Artificial Intelligence*” (A.I. – I.A. em português) foi usado pela primeira vez em 1956 por McCarthy (e desenvolvido por grandes pesquisadores como Marvin Minsky e Herbert Simon), e nada mais é do que uma tentativa de formalizar o eterno sonho da criação de um “cérebro eletrônico” (termo muito usado na ficção científica e mesmo na época inicial do desenvolvimento dos computadores).

Ao sobrevoar-se o local da IA, pode-se observar um terreno instável, entrecortado por diversas disciplinas (neurociência, informática, linguística, psicologia, filosofia, entre outras), que, ao mesmo tempo em que buscam uma contribuição interdisciplinar, se chocam em velhas certezas paradigmáticas, características das comunidades que representam. A partir desse cenário, pode-se logo concluir a fertilidade dos temas aí semeados e a necessidade por novos e constantes insumos que permitam às ideias que aí florescem acompanharem as variações do ambiente em questão. (PRIMO e COELHO, 2002)

Teixeira (1990) em seu artigo comenta quais as áreas do conhecimento que mais sofreram influência e mudanças, devido ao uso da Inteligência Artificial no seu contexto

Para a linguística, a IA significava uma revolução: agora seria possível criar um programa de computador onde estivessem representadas as estruturas gramaticais das diversas línguas humanas. Se isto pudesse ser feito, um sonho muito antigo seria realizado: teríamos descoberto a raiz comum de todas as línguas humanas e uma máquina universal de tradução tornar-se-ia possível. Mas foi realmente sobre a filosofia que o impacto da IA foi maior: criar uma máquina pensante significa desafiar uma velha tradição que coloca o homem e sua capacidade racional como algo único e original do universo. Mais do que isto, criar uma máquina pensante significa dizer que o pensamento pode ser recriado artificialmente sem que para isto precisemos de algo como uma “alma” ou outra marca divina.

Osório (1999) defende a ideia de que a implementação dos sistemas inteligentes parte do pressuposto que iremos “copiar” a inteligência humana e colocá-la a nosso serviço através de implementações que automatizem este tipo de processo/comportamento.

A figura 01 reproduzir o tão sonhado objetivo da inteligência artificial.



Figura 01: Inteligência: Natural e Artificial
Fonte: OSÓRIO, 1999

A inteligência artificial é considerada uma das ciências, mas recentes, ela teve seu aparecimento no final da Segunda Guerra Mundial, e como já foi referido ela abrange subcampos em variadas áreas do conhecimento. Uma destas áreas de conhecimento é justamente a área de estudo deste trabalho a Automação Residencial ou simplesmente Domótica, pois pode-se agregar tomadas de decisões aos sistemas melhorando assim o desempenhos do processo automatizado de gerenciamento de recursos, tais como energia e água, tudo isso partindo da construção de árvores de decisão baseadas no estudo do comportamento dos usuários de sistema.

1.2 SISTEMAS EMBARCADOS

1.2.1 Definição e contextualização histórica

Com o advento da era digital, recursos e equipamentos tecnológicos vêm evoluindo diariamente mudando a dinâmica do nosso dia-a-dia. E os sistemas eletrônicos embarcados (SEE) são notadamente mais presente nesse contexto, proporcionando facilidade e praticidade na vida dos usuários que se utilizam dessas tecnologias.

A expressão em inglês *Embedded Electronic System* costuma ser traduzida para o português como sistema eletrônico embarcado, ou sistema eletrônico

embutido. Esta denominação genérica indica um sistema eletrônico que tem como base um microprocessador, mas que diferentemente de um computador para uso genérico, possui um *software* completamente dedicado ao dispositivo ou sistema que ele controla. (UFPR, 2008) Esta afirma ainda que

Nos primeiros anos dos computadores digitais na década de 1940, os computadores eram por vezes dedicados a uma única tarefa. Eram, entretanto, muito grandes para serem considerados embarcados. O conceito de controlador programável foi desenvolvido algum tempo depois. O primeiro sistema embarcado reconhecido mundialmente foi o *Apollo Guidance Computer*, desenvolvido nos EUA por Charles Stark Draper no MIT para a NASA.

O termo sistema embarcado tem sua origem no fim da década de 1960, e desde suas primeiras aplicações os SEE vêm reduzindo seu preço. Nessa época o que existia era um pequeno programa de controle funcional de telefones. Posteriormente com o advento de microprocessadores especialistas, foi possível desenvolver *software* específico para os variados tipos de processador. Os programas eram escritos em linguagem de máquina. Ainda que na década de 1970 principiassem o surgimento de bibliotecas de códigos direcionados para sistemas embarcados com processadores específicos. Atualmente os sistemas embarcados podem ser programados em linguagens de alto nível e possuem sistemas operacionais. (CHASE, 2007)

Nos últimos anos temos visto uma crescente utilização de *softwares* embarcados em praticamente todos os objetos construídos pelo homem. Taurion (2005) delibera os sistemas ou *software* embarcados como um *software* que é embutido em um equipamento, como em um sistema de injeção eletrônica de um automóvel, permitindo que este equipamento atue com maior funcionalidade e flexibilidade. Antes apenas utilizados em sistemas complexos como sistemas industriais, aeronaves e navios, hoje vemos *softwares* embarcados em geladeiras, televisores e fornos microondas. Estes equipamentos tornam-se cada vez mais sofisticados, demandando mais e mais complexidade no seu *hardware* e *software* embarcado.

Oliveira (2006) afirma que se calcula nos Estados Unidos uma média de oito dispositivos baseados em microprocessadores por pessoa. Apesar de esse parecer um número muito grande, basta observar os equipamentos eletrônicos: aparelhos de

TV e de DVD, tocadores de MP3, telefones convencionais, sistemas de alarme, relógios e telefones celulares são alguns dos exemplos de aparelhos que utilizam sistemas embarcados.

Com o crescimento do mercado de equipamentos eletrônicos, a difusão do acesso à Internet e a equipamentos multimídia, hoje os equipamentos dedicados necessitam reagir à maior quantidade de periféricos externos, como *displays* gráficos, câmera de aquisição de foto e vídeo, captura e reprodução de som, acionamento de motores, uma grande quantidade de botões e diversas formas de comunicação. (TAURION, 2005)

A importância dos *softwares* embarcados é demonstrada de forma inequívoca por alguns números, Tourin (2005) afirma que em alguns setores, como o automotivo, as estimativas apontam que mais de 70% das inovações serão baseadas em tecnologias de *softwares* e não mais nas partes mecânicas. Atualmente os modelos de automóveis mais sofisticados incorporaram mais de 100 milhões de linhas de código de *software* em cada veículo.

Wolf (2001) assegura que os sistemas embarcados apresentam características em comum com os sistemas computacionais, mas não possuem a mesma uniformidade. Cada aplicação apresenta requisitos diferentes de desempenho, consumo de potência e área ocupada. Isso pode acarretar em uma combinação diferente de módulos de *hardware* e *software* para atender a esses requisitos.

Neste raciocínio, segundo pesquisas divulgadas pela UFPR (2008) pode-se afirmar

Sistemas eletrônicos embarcados são desenvolvidos para uma tarefa específica, ao contrário do computador comum, que pode ser usado em diferentes aplicações. O *software* escrito para sistemas embarcados é muitas vezes chamado *firmware* e fica armazenado em uma memória ROM ou memória *flash* ao invés de um disco rígido. Por vezes o sistema também é executado com recursos computacionais limitados, por exemplo, sem teclado, sem tela e com pouca memória. Podem possuir desde nenhuma interface do utilizador a uma interface de utilizador completa, similar à dos sistemas operacionais *desktop*. Sistemas mais simples utilizam botões, LEDs ou telas bastante limitadas, geralmente mostrando somente números ou uma fila pequena de caracteres. Tem sido comum também a adoção de interface baseada em *browsers*, como é o caso de modems e roteadores, no qual já existe uma conexão padrão *Ethernet*. Isso evita o custo de uma tela sofisticada, ainda que seja fornecida uma interface complexa e completa a ser acessada em um computador.

Quem programa e desenvolve sistemas embarcados costuma pensar neles como entradas, saídas, processamentos, memórias e o ambiente onde será utilizado. Já o usuário final se preocupa apenas em como utilizar este sistema e o que ele agrega de valor a um produto, em termos de redução de custos, aumento de funcionalidade e aumento de desempenho. (CUNHA, 2013)

1.2.2 Aplicações

A tecnologia da computação tem se tornado onipresente no dia-a-dia da população na contemporaneidade. Henzinger e Sifakis (2007) apud Damke e Moraes (2008) em uma visão, mas ampla afirmam que os sistemas embarcados ou *softwares* embarcados hoje controlam a comunicação, os transportes, os sistemas médicos e diversos outros campos do meio ambiente contemporâneo.

Os Sistemas embarcados estão sendo utilizados cada vez mais, agregando diversas funcionalidades aos equipamentos utilizados no cotidiano das pessoas. Segundo Chase (2007) mais de 90% dos microprocessadores fabricados mundialmente são destinados a máquinas que usualmente não chamadas de computadores. Dentre alguns destes dispositivos estão aparelhos celulares, fornos microondas, automóveis, aparelhos de DVD e PALM's. O que diferencia este conjunto de dispositivos de um computador "convencional" (*Desktop* ou *Notebook*), conhecido por todos é o seu projeto baseado em um conjunto dedicado e especialista, construído por *Hardware*, *Software* e Periféricos – um Sistema Embarcado.

Pela sua natureza especialista um Sistema Embarcado pode ter inúmeras aplicações e estão inseridos em milhares de dispositivos comuns, como em eletrodomésticos, aparelhos de áudio e vídeo, celulares, *paggers*, calculadores e diversos outros dispositivos.

Almeida (2008) dá alguns outros exemplos de tecnologias desenvolvidas através de sistemas embarcados: Alarmes automotivos; Sistemas de *airbag* e ABS; Computadores de bordo; Injeção eletrônica; GPS; Máquinas de lavar; Microondas, DVDs; TV digital; Calculadoras; Videogames, *PDA*s; Semáforos, Câmeras de

vigilância, Radares; Equipamentos para rede (roteadores, *switches* e modems); Telefones celulares; Centrais telefônicas; Equipamentos de controle industriais e vários outros.

Existem diversos modelos de placas de automação disponível no mercado, desenvolvidas para diversas finalidades. Mostraremos alguns modelos de placas:

PLACA GRAVADORA (*GravaPen*) – Com a placa demonstrada na figura 02 é possível reproduzir e gravar quaisquer tipos de sons em formato wav diretamente em um *pendrive*. Pode-se utilizar o próprio microfone embutido na placa para capturar o som a ser gravado.



Figura 02: Placa de automação – Placa gravadora
Fonte: ROGERCOM, 2014

PLACA RCOM – *HOME BEE ROGERCOM*: A figura 03 expõe a placa pela qual é possível ligar/desligar lâmpadas, aparelhos eletro-eletrônicos, fechaduras elétricas, irrigação de jardins, abrir/fechar portas, portões, cancelas etc.



Figura 03: Placa de automação – Placa RCOM
Fonte: ROGERCOM, 2014

PLACA DE AUTOMAÇÃO CNC ANDROID – Placa compatível com o sistema android para que você controle lâmpadas, motores, eletrodomésticos e qualquer equipamento elétrico a partir de telefone celular ou *tablet* que possua a plataforma android. É possível visualizar a mesma através da figura 04.

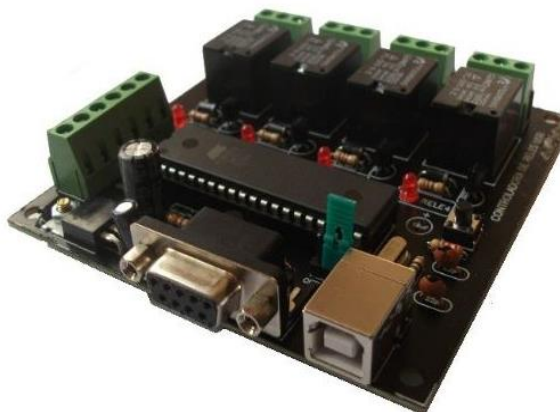


Figura 04: Placa de automação – Placa CNC android
Fonte: MERCADOLIVRE, 2014

PLACA DE PORTÃO AUTOMÁTICO: A figura 05 mostra uma placa utilizada em algumas aplicações que envolvem a automatização de portões e portas.



Figura 05: Placa de automação – Placa de portão automático
Fonte: HOTFROG, 2010 apud MARCHESAN, 2012

PLACA DE CONTROLE DE VELOCIDADE: A placa vista na figura 06 é utilizada para o controle de velocidade para motor de corrente contínua em

aplicações voltadas para a automação e robótica de motores que não necessitam da inversão de rotação.

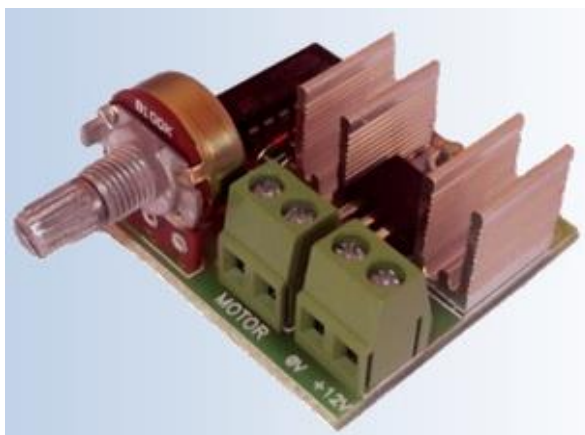


Figura 06: Placa de automação – Placa de controle de velocidade
Fonte: MERCADOLIVRE, 2014

1.2.3 Funcionalidades e Características

Tendo em vista que os *softwares* ou sistemas embarcados estão se tornando cada vez mais complexos, existe a necessidade de operarem em ambientes de total confiabilidade, exigindo que métodos aprimorados de engenharia sejam adotados. Alguns desses sistemas contêm milhões de linhas de códigos tornando-os de altíssima qualidade. Taurion (2005) assegura que os sistemas que operam em um marca-passo, por exemplo, não pode contêm nenhum erro, já que está lidando diretamente com vida humana.

Existem dois modos de funcionalidade em sistemas embarcados, eles são fundamentais para determinar-se a programação e o funcionamento na aplicação ao qual foi submetida. Estes tipos são:

Reativo: o funcionamento se dá como resposta a eventos externos, que podem ser efetuados por períodos (caso de sistemas rotacionais ou de controles de loop) ou assíncronos (pressionamento de um botão por parte do usuário). Existe, então, uma necessidade de entrada de dados para que aconteçam as ações de funcionamento. Porém, a saída, deve ser realizada exatamente após os sinais de entrada começar a atuar.

Controle em tempo real: existe tempo limite para ser executada cada tarefa (leitura de sensor, emissão de sinais para um atuador, atualização de *display*, entre outros métodos de entrada). Por este motivo, nem sempre tempo real é igual ao modo mais rápido de executar uma tarefa. Estes modos de operação, por serem cíclicos, não dependem da entrada de sinais para executar as atividades, sendo capaz de tomar decisões referentes à ausência dos mesmos (CUNHA, 2013). Os sistemas de tempos real são classificados em:

Soft Real Time: As tarefas podem são executadas em um intervalo de tempo específico, não ocasionado consequências graves se este limite de tempo não for cumprido. Um exemplo é um sistema bancário, onde apenas uma mensagem de erro aparecerá se determinada tarefa não for realizada dentro do tempo pré-determinado (CUNHA, 2013).

Hard Real Time: As tarefas devem são executadas em um período de tempo específico, com consequências graves se qualquer tarefa não for cumprida. Como exemplo pode-se pensar nos sistemas de controle de um avião, onde uma falha pode resultar em queda e perdas de vidas (CUNHA, 2013).

Os sistemas embarcados não possuem interface para com os usuários, eles possuem características peculiares próprias, por serem desenvolvidos para funções específicas, mas existem algumas típicas de todos os sistemas, é que esses manipulam dados dentro de sistemas ou produtos maiores. Algumas dessas características são:

- a. Sistemas embarcados são projetados para realizar uma função ou uma gama de funções e não para serem programados pelo usuário final, como os computadores pessoais. O usuário pode alterar ou configurar a maneira como o sistema se comporta, porém não pode alterar a função que este realiza. (PETER, 2004 apud SANTOS, 2006)
- b. Sistemas embarcados normalmente interagem com o ambiente em que se encontram, coletando dados de sensores e modificando o ambiente utilizando atuadores. (PETER, 2004 apud SANTOS, 2006)
- c. Eles devem ser confiáveis. Muitos destes sistemas realizam funções críticas, onde falhas podem causar catástrofes. A principal razão para que estes sistemas sejam a prova de falhas, é que eles interagem com o meio, causando impactos a

este. Dizer que um sistema é confiável significa que este possui certas características. (PETER, 2004 apud SANTOS, 2006)

- Estabilidade: é a probabilidade que um sistema não irá falhar.
- Recuperação: é a probabilidade de que uma falha no sistema será corrigida em um curto intervalo de tempo.
- Disponibilidade: é a probabilidade de que um sistema estará disponível em certo tempo. Alta estabilidade e recuperação levam a uma alta disponibilidade.
- Segurança: um sistema deve ser seguro em dois aspectos. Ele deve ser seguro para o meio ambiente, ou seja, uma falha não acarreta em danos ao meio ou as pessoas que utilizam este sistema, e ele devem manter as informações confidenciais dentro dele, sem permitir que pessoas não autenticadas manipulem estas informações.

d. Sistemas embarcados possuem outras métricas de eficiência além das já conhecidas por projetista de sistemas desktop e servidores. (PETER, 2004 apud SANTOS, 2006)

- Consumo de Energia: Tendo em vista que muitos sistemas embarcados são móveis, e são alimentados por baterias, estes devem ser projetados para consumir o mínimo de energia possível. A tecnologia na fabricação de baterias evolui em ritmo muito menor do que as aplicações que fazem uso destas baterias, logo a energia disponível deve ser aproveitada da melhor maneira possível.

- Tamanho de código: Todo código da aplicação que é executada em um sistema embarcado deve estar presente neste, quase sempre em memória. Muito raramente sistemas embarcados possuem dispositivos de armazenamento magnético, como discos rígidos, para armazenar código, logo a disponibilidade de memória é muito limitada, por isso o desperdício de memória deve ser evitado pelos programadores de *software* embarcados.

- Execução eficiente: O uso de recursos de *hardware* deve ser restrito ao que é realmente necessário para implementar uma certa função. Os requisitos de tempo devem ser satisfeitos empregando-se o mínimo de recursos possível e minimizando o consumo de energia. Visando reduzir o consumo de energia, a frequência do *clock* e a tensão de alimentação devem estar presentes.

- Peso: Sistemas móveis devem ser leves. Os celulares de primeira geração mostram que sistemas móveis pesados tendem a desaparecer.

- Custo: O uso eficiente de componentes de *hardware* pode baixar o custo final de um sistema embarcado, no mercado de eletrodomésticos, por exemplo, a competitividade é importantíssima, e o custo é um fator decisivo para dizer se um produto possui mercado.

e. Grande parte dos sistemas embarcados não possui teclados, mouse, monitores ou outros dispositivos encontrados em computadores pessoais para realizar interfaceamento com o usuário. Eles também possuem interfaces dedicadas, como botões, LEDs e chaves. Por isso dificilmente o usuário reconhece a informação sendo transmitida ou processada dentro deles. (PETER, 2004 apud SANTOS, 2006)

f. Muitos possuem requisitos de tempo real. Não completar uma tarefa em um tempo determinado pode resultar em perda de dados e consequentemente de qualidade (aplicações multimídia) ou causar danos. O não cumprimento de um requisito de tempo real pode resultar em catástrofe.

Sistemas de tempo real não devem utiliza componentes ou técnicas que diminuam o tempo de processamento na média, como memórias *cache*. Em sistemas de tempo real, uma resposta do sistema deve ser explicada e comprovada sem argumentos estatísticos. (PETER, 2004 apud SANTOS, 2006)

g. Muitos sistemas embarcados são híbridos, pois são compostos por partes analógicas e partes digitais. As partes analógicas utilizam sinais contínuos em valores de tempo contínuos, e as partes digitais usam sinais discretos no tempo discreto. (PETER, 2004 apud SANTOS, 2006)

h. Tipicamente, sistemas embarcados são reativos ao ambiente, ou seja, eles estão em interação contínua com o ambiente e executam em um ritmo determinado por este. Pode-se dizer que um sistema reativo encontra-se em um estado, esperando por uma entrada. Para cada entrada recebida, ele realiza o processamento da informação e gera uma saída. Autômatos são exemplos de sistemas reativos. (PETER, 2004 apud SANTOS, 2006)

Verifica-se então que alguns sistemas possuem características comuns, eles são especialistas em determinada função e consequentemente, o número e as particularidades dos sistemas operacionais correspondentes variam na mesma proporção.

Os sistemas embarcados estão inseridos em milhares de dispositivos comuns utilizados no dia a dia como em eletrodomésticos, aparelhos de áudio e vídeo, celulares e outros. Chase (2007) exemplifica alguns tipos de aplicabilidades:

a. Setor Automobilístico

Um veículo top de linha é um excelente exemplo de um complexo sistema literalmente “embarcado”. Centenas de sensores fornecem informações sobre todo o funcionamento do veículo. Várias unidades de processamento independentes atuam em regiões diferentes e se comunicam entre si, captando os sinais destes sensores e fazendo com que as ações referentes a cada caso sejam tomadas. Esta comunicação geralmente se dá através de redes, cujo protocolo CAN tem se tornado o padrão. Isto acontece desde a central que memoriza a posição dos bancos, espelhos, volante, etc. Para cada usuário do veículo até a central que gerencia o funcionamento do motor. (CHASE, 2007) A figura 07 enfatiza o exposto anteriormente.



Figura 07: Sistemas embarcados em veículos
Fonte: CHASE, 2007

b. Aquisição de Dados – *Data Logger*

A figura 08 trata da aquisição de dados, que é um exemplo de aplicação mais utilizada em sistemas embarcados. Consistem de sistemas que através de sensores (temperatura, umidade, pH e outros) capturam as variáveis ambientes a serem analisadas e são gravadas em memória para consultas posteriores. O Sistema além de monitorar o ambiente, com adição de atuadores ao projeto, pode ter a capacidade de controlar as variáveis de ambiente com base em um critério estabelecido pelo projetista do sistema.



Figura 08: Data Logger para temperatura do ar
Fonte: CHASE, 2007

c. Propósito Geral

São as aplicações mais parecidas com os computadores de mesa, mas em embalagens embarcadas. Nelas costuma haver grande interação entre os usuários e o sistema, geralmente através de terminais de vídeo ou monitores. Como exemplo tem-se os videogames, os sistemas de acesso biométrico, impressoras, *smartphones* e PDAs, caixas de banco, sistemas de monitoramento médico etc. Alguns exemplos dessas aplicações são evidenciados na figura 09.



Figura 09: Aparelhos de interação com os usuários
Fonte: CHASE, 2007

d. Sistemas de Controle

Geralmente são aplicações mais robustas, com placas dedicadas e múltiplos sensores de entrada e saída, como controles em malha fechada com realimentação em tempo real.

Muitas vezes fornecem pouca interação com o usuário, mostrando sinalizações através de LEDs. Usados nos motores de automóveis, processos químicos, controle de vôo, usinas nucleares, aplicações aeroespaciais e monitoramento e controle de variáveis ambiente (temperatura, umidade, pH do ar). A figura 10 mostra-nos um exemplo de manipuladores robótico muito utilizado.

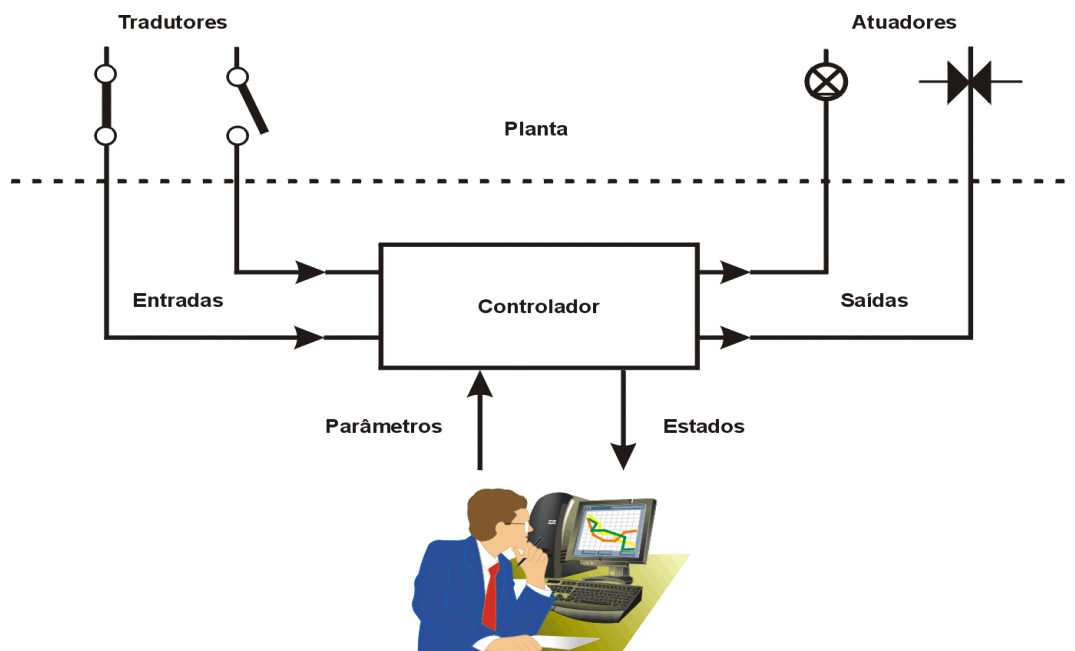


Figura 10: Controle de manipuladores robóticos
Fonte: SILVA, 2013

e. Processamento de Sinais.

Um exemplo de software interativo utilizado para o processamento de sinais é visto na figura 11. Esses tipos de aplicabilidades envolvem um grande volume de informação a ser processada em curto espaço de tempo. Os sinais a serem tratados são digitalizados através de conversores Analógico/Digital, processados e novamente convertidos em sinais analógicos por conversores Digital/Analógico. Casos de tratamento de áudio, filtros, *modems*, compressão de vídeo, radares e sonares, etc. Existem os DSP (*Digital Signal Processor* – Processador Digital de Sinais) os microcontroladores dotados deste recurso são os *Blackfin* da *Analog Devices* e o *DsPIC* da *Microchip*.

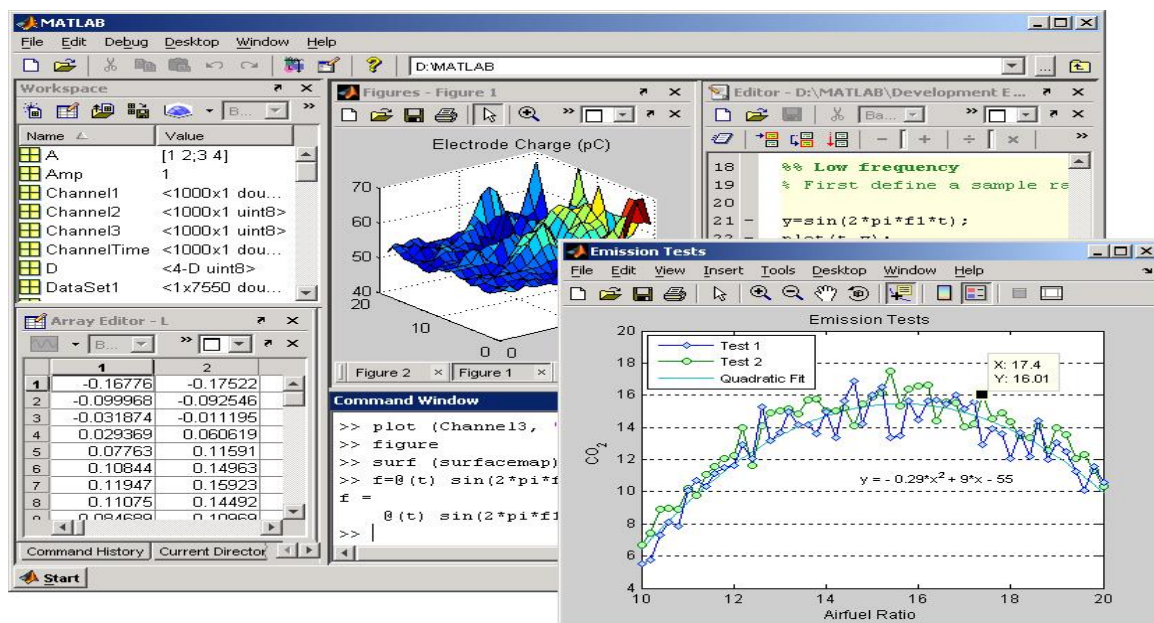


Figura 11: MATLAB, software interativo de processamento de sinais
Fonte: BARRA, 2013

A indústria eletrônica tem crescido nos últimos anos a uma taxa impressionante e um dos principais motivos para tal crescimento é a incorporação de sistemas eletrônicos numa grande variedade de produtos tais como automóveis, eletrodomésticos e equipamentos de comunicação pessoal. Sistemas de computação estão presentes em todo lugar e não é surpresa que anualmente são produzidos milhões de sistemas destinados a computadores pessoais (desktop), estações de trabalho, servidores e computadores de grande porte. (BARROS e CAVALCANTE, 2008)

Segundo o professor Denardim (2009) em um passado recente, o alto custo dos dispositivos eletrônicos limitou o uso dos microcontroladores apenas aos produtos domésticos considerados de alta tecnologia (televisão, vídeo e som). Porém, com a constante queda nos preços dos circuitos integrados, os microcontroladores passaram a ser utilizados em produtos menos sofisticados do ponto de vista da tecnologia, como máquinas de lavar, microondas, fogões e refrigeradores. Assim, a introdução do microcontrolador nestes produtos cria uma diferenciação e permite a inclusão de melhorias de segurança e de funcionalidade. Alguns mercados chegaram ao ponto de tornar obrigatório o uso de microcontroladores em determinados tipos de equipamentos, impondo um pré-requisito tecnológico.

Dentre os vários tipos de microcontroladores existente no mercado, o Arduino preencheu nossas expectativas. A facilidade de manuseio, simplicidade na programação, juntamente com a gama de componentes que podem ser agregados a mesmo, levaram-nos a escolha do mesmo para o desencadeamento do nosso projeto.

1.3 MICROCONTRALADOR ARDUINO

1.3.1 Descrição e análise evolutiva

O Arduino teve seu surgimento derivado de um projeto acadêmico, e devido ao sucesso alcançado ao longo do tempo, existe uma enorme comunidade de seguidores/utilizadores em todo o mundo. Tal razão para isso são suas funcionalidades, simplicidades, possibilidade de utilização em várias plataformas e o seu baixo custo.

A Universidade Federal do Mato Grosso do Sul – UFMS (2012), através do seu Grupo de Robótica, afirma que

O Arduino surgiu em 2005, na Itália, com um professor chamado Massimo Banzi, que queria ensinar eletrônica e programação de computadores aos seus alunos de *design*, para que eles usassem em seus projetos de arte, interatividade e robótica. Porém, ensinar eletrônica e programação para pessoas que não são da área não era uma tarefa simples, e outra dificuldade era a inexistência de placas poderosas e baratas no mercado.

Arduino foi projetado com a finalidade de ser de fácil entendimento, de fácil programação e de fácil aplicação, sobre tudo o mesmo é multiplataforma, podendo ser configurado em ambientes Linux, Mac OS e Windows. Além disso, um grande diferencial deste dispositivo é ser mantido por uma comunidade que trabalha na filosofia *open-source*, desenvolvendo e divulgando gratuitamente seus projetos. (DI RENNA *et al*, 2013)

Santos (2009) confirma que o estudo do Arduino abre portas à compreensão de uma importante ferramenta de desenvolvimento através de uma aprendizagem

simples, mas dedicada, onde podemos fazer desde *robots* a Domótica entre muitas outras aplicações, bastando simplesmente ter imaginação.

O Arduino pode perceber o ambiente por receber informação de uma grande variedade de sensores, e pode estimular o ambiente controlando luzes, motores, e outros atuadores. O CSE (2011) ainda afirma ainda que

A parte de *hardware* do projeto, uma placa que cabe na palma da mão, é um computador como qualquer outro: possui microprocessador, memória RAM, memória *flash* (para guardar o *software*), temporizadores, contadores, dentre outras funcionalidades. Atualmente, o projeto está na versão *Uno*, porém muitos Arduinos encontrados hoje são da versão *Duemilanove* (2009, em italiano), que possui um *clock* de 16 MHz, 2 Kb de memória RAM, 32 Kb de memória flash, 14 portas digitais e 6 entradas analógicas.

Arduino é basicamente um kit de desenvolvimento, que pode ser visto como uma unidade de processamento capaz de mensurar variáveis do ambiente externo, transformadas em um sinal elétrico correspondente, através de sensores ligados aos seus terminais de entrada. De posse da informação, ele pode processá-la computacionalmente. Porém, ele pode ainda atuar no controle ou no acionamento de algum outro elemento eletro-eletrônico conectado ao terminal de saída. (DI RENNA *et al*, 2013) A figura 12 explana um Diagrama de bloco de uma cadeia de processamento que utiliza o microcontrolador Arduino.



FIGURA 12: Diagrama de bloco de uma cadeia de processamento utilizando o Arduino
Fonte: DI RENNA *et al*, 2013

Os projetos em Arduino podem ser únicos ou podem comunicar com outros circuitos, ou até mesmo com outros *softwares* em um computador (por exemplo, Java, Flash, Processing, MaxMSP, PHP, entre outros). As placas podem ser montadas à mão ou serem compradas montadas, e o *software* pode ser obtido gratuitamente. Como o Arduino é um projeto aberto, diversas empresas fabricam e disponibilizam suas placas no mercado, como o Freeduino, Seeduino, Roboduino, Pinguino e as brasileiras Severino e Brasuino. (CSE, 2011)

Deste modo, em um ambiente profissional, as características do Arduino fazem dele uma boa ferramenta de prototipação rápida e de projeto simplificado. Por outro lado, em um ambiente acadêmico, ele pode ser perfeitamente utilizado como ferramenta educacional, uma vez que não requer do usuário conhecimentos profundos de eletrônica digital nem da programação de dispositivos digitais específicos. (DI RENNA *et al*, 2013)

As placas de microcontrolador Arduino podem ser montadas pelo próprio usuário, mas, os mantenedores possuem um serviço de venda do produto pré-montado, através deles próprios e também por distribuidores oficiais com pontos de venda mundiais. Existe uma variedade de Arduino disponível, com configurações e características de *hardware* e *software* diferentes, as figuras a seguir listam os principais tipos de Arduinos.

ARDUINO UNO

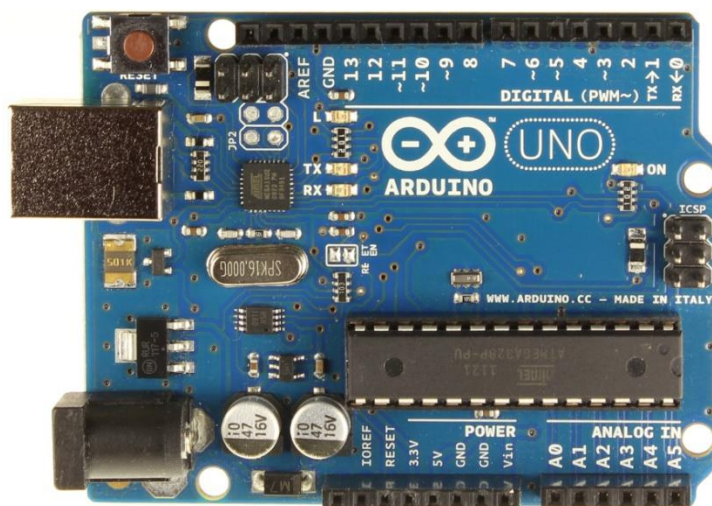


Figura 13: Placa, Arduino Uno
Fonte: ARDUINO.CC, 2014

O Arduino Uno exposto na figura 13 contém tudo o necessário para apoiar o microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC ou bateria-to-DC para começar. O Uno é diferente de todas

as placas anteriores em que não usam o chip controlador USB-to-serial FTDI. (ARDUINO.CC, 2014)

ARDUINO ETHERNET

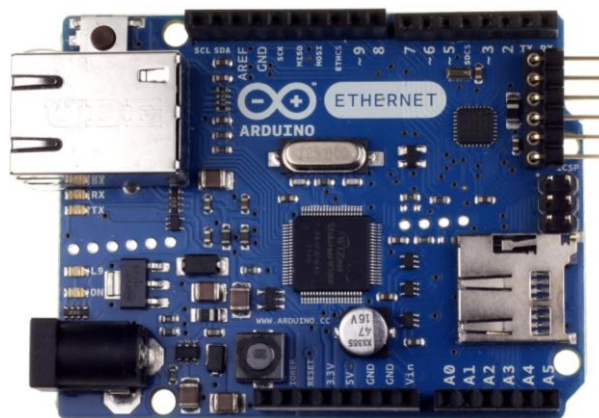


Figura 14: Placa, Arduino Ethernet
Fonte: ARDUINO.CC, 2014

O Arduino Ethernet (figura 14) é diferente de outras placas na medida em que não tem um chip controlador USB-to-serial onboard, mas tem uma interface *Wiznet Ethernet*, um leitor de cartão micro SD de bordo, que pode ser usado para armazenar arquivos para servir através da rede, é acessível através da Biblioteca SD. (ARDUINO.CC, 2014)

ARDUINO ADK



Figura 15: Placa, Arduino ADK
Fonte: ARDUINO.CC, 2014

O Arduino ADK (figura 15) tem uma interface host USB para se conectar com telefones baseados no Android, com base na MAX3421E IC. Ele tem 54 pinos de entrada/saída. (ARDUINO.CC, 2014)

ARDUINO LILYPAD

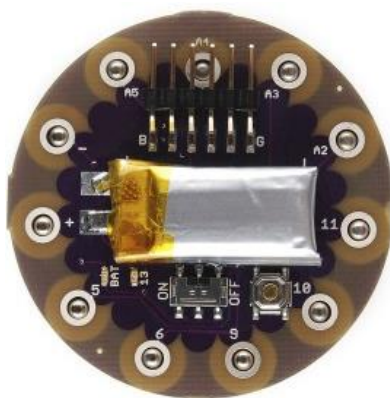


Figura 16: Placa, Arduino LilyPad
Fonte: ARDUINO.CC, 2014

O Arduino LilyPad (figura 16) é uma placa de microcontrolador projetado para wearables e e-têxteis. Tem 9 pinos para entrada/saída. Além disso, ele tem um circuito de carga construído para a bateria. A placa é baseada no ATmega328. (ARDUINO.CC, 2014)

ARDUINO LEONARDO



Figura 17: Placa, Arduino Leonardo
Fonte: ARDUINO.CC, 2014

O Arduino (figura 17) Leonardo tem 20 pinos de entrada/saída digital e conexão micro USB. Para utilizá-lo basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC ou bateria-to-DC para começar a utilizar. (ARDUINO.CC, 2014)

ARDUINO YUN



Figura 18: Placa, Arduino Yun
Fonte: ARDUINO.CC, 2014

O Arduino Yun visualizado na figura 18, é uma placa de microcontrolador baseado no ATmega32u4. O processador Atheros suporta uma distribuição Linux baseada no OpenWRT chamado Linino. (ARDUINO.CC, 2014)

ARDUINO TER

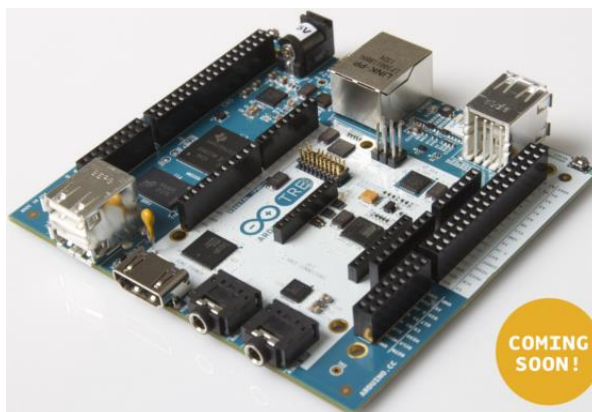


Figura 19: Placa, Arduino Ter
Fonte: ARDUINO.CC, 2014

Arduino TER visto na figura 19, tem um desempenho que abre as portas para aplicações movidas a Linux mais avançados. Podem-se executar aplicativos de desktop de alto desempenho, algoritmos de processamento intensivo ou comunicações de alta velocidade. (ARDUINO.CC, 2014)

ARDUINO MICRO



Figura 20: Placa, Arduino Micro
Fonte: ARDUINO.CC, 2014

O Arduino Micro (figura 20) tem 20 pinos de entrada/saída digital. Ele tem um formato que permite que ele seja facilmente colocado em uma placa de ensaio. O micro é semelhante ao Arduino Leonardo em que o ATmega32u4 foi construído em comunicação USB, eliminando a necessidade de um processador secundário. (ARDUINO.CC, 2014)

Existem alguns componentes como o *Shields* (cobertura) que são placas que podem ser conectados em cima do PCB - *Printed Circuit Board* (Placa de Circuito Impresso) da placa Arduino, que servem para estender a sua capacidade. Por serem fáceis de montar e barato de produzir são utilizadas para diferentes finalidades, como:

- ✓ Obter acesso à internet;
- ✓ Controlar motores;
- ✓ Transferir imagens;
- ✓ Capturar rádiofreqüência e outras finalidades.

As figuras posteriores mostram algumas das diferentes coberturas existentes.

SHIELD – COBERTURA ARDUINO GSM



Figura 21: Cobertura Arduino GSM
Fonte: ARDUINO.CC, 2014

A cobertura Arduino GSM (figura 21) conecta seu Arduino à Internet usando a rede sem fio GPRS. Basta ligar este módulo para sua placa Arduino, conecte um cartão SIM de uma cobertura GPRS e seguir algumas instruções simples para começar a controlar o seu mundo através da internet. Você também pode fazer/receber chamadas de voz e enviar/receber mensagens SMS. (ARDUINO.CC, 2014)

SHIELD – COBERTURA ARDUINO ETHERNET

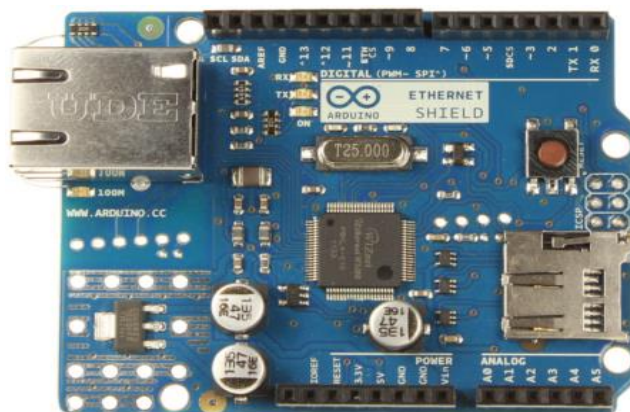


Figura 22: Cobertura Arduino Ethernet
Fonte: ARDUINO.CC, 2014

A cobertura Arduino Ethernet (figura 22) conecta seu Arduino à internet em poucos minutos. Basta ligar este módulo para sua placa Arduino, conecte-o à sua rede com um cabo Rj45. (ARDUINO.CC, 2014)

SHIELD – COBERTURA ARDUINO Wi-Fi

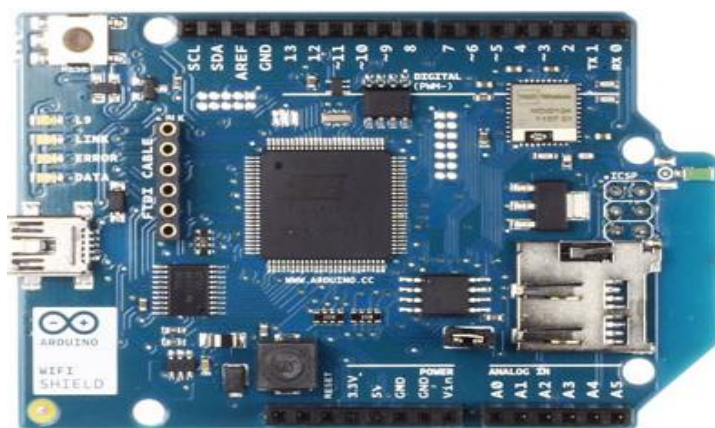


Figura 23: Cobertura Arduino Wi-Fi
Fonte: ARDUINO.CC, 2014

A cobertura Arduino Wi-Fi (figura 23) permite que uma placa Arduino conecte-se à internet usando a especificação 802.11 sem fio (Wi-Fi). (ARDUINO.CC, 2014)

SHIELD – COBERTURA ARDUINO SD SEM FIO

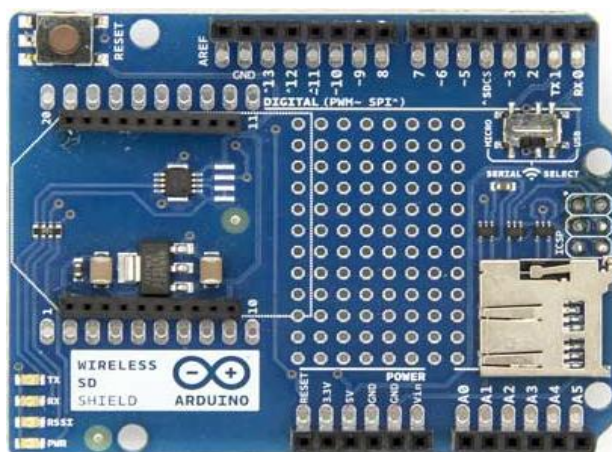


Figura 24: Cobertura Arduino SD sem fio
Fonte: ARDUINO.CC, 2014

A cobertura SD (figura 24) sem fio permite que uma placa Arduino se comunique sem fios com um módulo sem fios. O módulo pode comunicar até 100 metros em ambientes internos ou 300 pés ao ar livre. Ele pode ser usado como um substituto serial/USB ou você pode colocá-lo em um modo de comando e configurá-lo para uma variedade de transmissão e melhores opções de rede. (ARDUINO.CC, 2014)

SHIELD – COBERTURA ARDUINO MOTOR

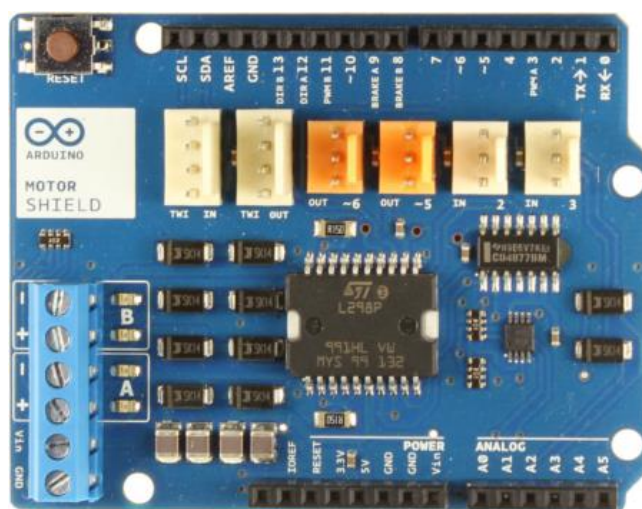


Figura 25: Cobertura Arduino Motor
Fonte: ARDUINO.CC, 2014

A cobertura Arduino Motor (figura 25) é baseada na L298 (folha de dados), que é um driver de ponte completa duplo projetado para conduzir cargas indutivas tais como relés, solenóides, DC e motores de passo. Ele permite que você dirija dois motores DC com sua placa Arduino, controlando a velocidade e a direção de cada um de forma independente. (ARDUINO.CC, 2014)

SHIELD – COBERTURA ARDUINO SEM FIO PROTO

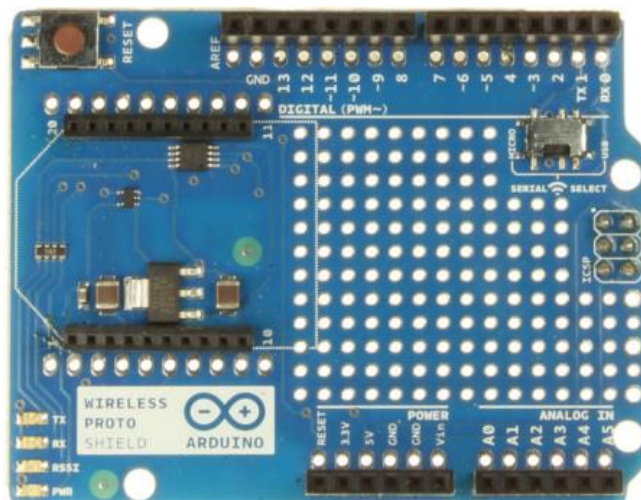


Figura 26: Cobertura Arduino sem fio Proto

Fonte: ARDUINO.CC, 2014

A cobertura sem fio Proto (figura 26) permite que uma placa Arduino se comunique sem fios. O módulo pode comunicar até 100 metros em ambientes internos ou 300 pés ao ar livre (com linha de visada). Este cobertura não tem o soquete SD. (ARDUINO.CC, 2014)

1.3.2 Componentes

O *hardware* e o *software* do Arduino são ambos de fonte aberta, o que significa que o código, os esquemas, o projeto e tudo o que se relaciona ao mesmo, podem ser utilizados livremente por qualquer pessoa e com qualquer propósito. 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. (McROBERTS, 2011)

O Arduino é considerado um computador de pequeno porte. A GlobalCode (2009) menciona que sendo um computador de pequeno porte ele utiliza um microcontrolador (computador completo em um *chip*) e não um microprocessador tradicional que precisa de memórias e demais recursos externos para funcionar. HERNANDEZ e KALIL (2013) confirmam esta informações

Define-se como microcontrolador um componente dotado de microprocessador, memória e outros elementos que variam conforme o objetivo de sua elaboração, tais como memórias secundárias e portas de entrada e saída. A maioria dos equipamentos eletrônicos modernos é dotada de ao menos um microcontrolador em seu interior. Temos como exemplo os eletrodomésticos, que necessitam desse componente para execução de suas tarefas básicas. É vantajoso o uso de um microcontrolador como Arduino pelo preço acessível e pelo baixo consumo de energia.

Vejamos um comparativo na tabela 01, segundo a GlobalCode (2009) entre um computador tradicional (PC Intel, Apple, RISC) e uma placa com microcontrolador programável (Program-ME, Arduino e vários outros).

Computador Tradicional (PC Intel, Apple, RISC)	Placa microcontrolada de pequeno porte
Alto consumo de energia	Baixo consumo de energia (muito baixo!)
Muita memória	Pouca memória (muito pouca!)
Caro	Muito barato
Complexo	Simple
Utiliza um sistema operacional	Não utiliza um sistema operacional
Vários programas ao mesmo tempo	Um programa por vez
Pseudo tempo real	Tempo real (hard real-time)
Conexão de <i>hardware</i> por USB/fire-wire	Conexão de <i>hardware</i> via portas digitais e analógicas
Saída de vídeo padrão	Sem saída de vídeo padrão

*Considerando uso de microcontroladores de pequeno porte

TABELA 01: Diferença entre computador tradicional e microcontrolador programável
Fonte: GLOBALCODE, 2009

A interface do Arduino com outros dispositivos está mais perto do meio físico que a de um PC, pode ler dados de sensores (temperatura, luz, pressão etc.) e controlar outros circuitos (lâmpadas, motores, eletrodomésticos etc.), dentre outras

coisas que não conseguíamos diretamente com um PC. A grande diferença com relação ao uso desses dispositivos, no caso do Arduino, é que, na maior parte das vezes, nós mesmos construímos os circuitos que são utilizados, ou seja, não estamos limitados apenas a produtos existentes no mercado: o limite é dado por nosso conhecimento e criatividade. (JUSTEN, 2012)

Existem diversos componentes que podem ser ligados no Arduino, com diferentes finalidades, *displays* LCD; comunicação sem fio – rádio-frequência; infravermelho; *bluetooth*; acelerômetro; bússola; GPS; detector de cor; leitor de impressão digital; circuitos integrados em geral; sensores de luz, temperatura, toque, umidade, presença; dentre outros componentes.

Apesar de o Arduino ser um computador independente, em alguns casos podemos aproveitar o uso de um PC para explorar outra funcionalidade muito eficiente, já que o Arduino consegue conversar com o computador através da porta *USB*. Isso nos permite desenvolver um *software* que roda no PC e se comunica com o *software* que roda no Arduino, o que nos abre um universo de possibilidades. (JUSTEN, 2012)

Então para melhor entendemos a plataforma Arduino, faz-se necessário perceber o funcionamento do mesmo, para isso tentaremos descrever tal funcionamento abordando a parte de *hardware* e *software*. Acreditando-se que assim facilitará a base de sua compreensão, Santos (2009), Justen (2012) e Jacquet *et al* (2013), exemplifica os principais componentes de *hardware* e *software* dos Arduino:

1.3.2.1 *Hardware*

- **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. Os microcontroladores utilizados têm sua quantidade de memória

variável dependendo dos modelos. As mas utilizadas pelos Arduino são demonstrada na tabela 02:

	ATMega 168	ATMega 328	ATMega 1280
Flash	16 Kbytes (2 KBytes Bootloader)	32 KBytes (2 KBytes Bootloader)	128 KBytes (4 KBytes Bootloader)
SRAM	1024 Bytes	2048 Bytes	8192 Bytes
EEPROM	512 Bytes	1024 Bytes	4096 Bytes

TABELA 02: Quantidade de memória disponível em cada modelo de microcontrolador
Fonte: SANTOS, 2009

- **Protoboard:** A figura 27 mostra um exemplo de *protoboard*, onde há duas fileiras horizontais de pinagens, uma na parte superior e outra na inferior. Os furos das fileiras superiores estão interligados entre si (o mesmo vale para os furos das fileiras inferiores). Por isso, geralmente, tais fileiras são usadas para ligar a alimentação.

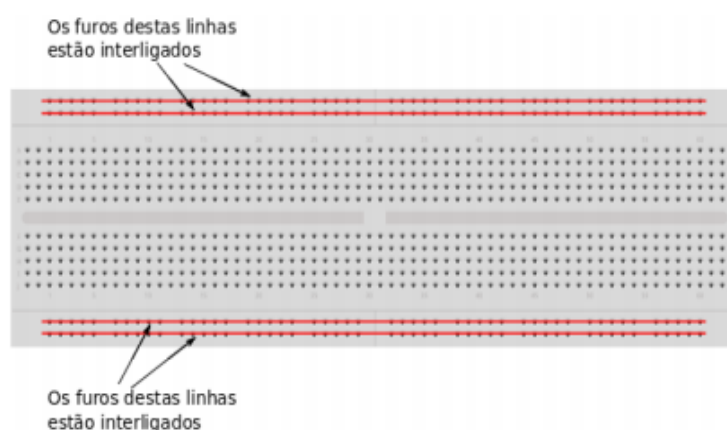


FIGURA 27: Componente, Pinagem
Fonte: JACQUET et al, 2013

- **LED's:** como visto na figura 28, é um diodo emissor de luz que quando alimentado corretamente, permite o fluxo de energia apenas em um sentido. É um componente polarizado (com pólo positivo e negativo) e deve ser corretamente conectado para funcionar.

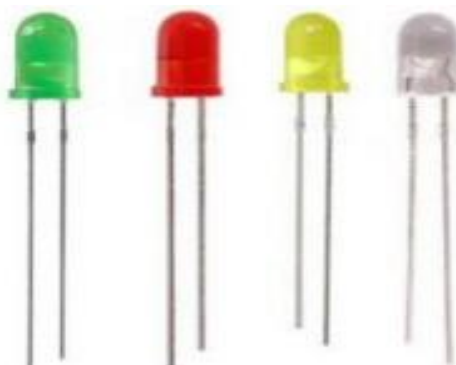


FIGURA 28: Componente, LEDs
Fonte: JACQUET et al, 2013

- **Potenciômetro:** A figura 29 traz um modelo de potenciômetro, que é um componente com uma haste giratória, cuja função é variar a resistência. Normalmente, o valor gerado pelo mesmo é lido pelo Arduino como um valor analógico em uma entrada analógica. Uma aplicação bastante simples desse componente é controlar a frequência com que um LED pisca, por meio da variação da resistência ao girar o botão do potenciômetro.



FIGURA 29: Componente, potenciômetro
Fonte: JACQUET et al, 2013

- **Push-button:** É um componente que conecta dois pontos no circuito ao ser pressionado (como ligar um *led* ao pressioná-lo). A figura 30 evidencia um exemplo de *push-button*.



FIGURA 30: Componente, push-button
Fonte: JACQUET et al, 2013

- **Sensor de Luminosidade:** O sensor de luminosidade (figura 31) é basicamente um sensor que permite detectar a luz do ambiente variando seu valor de resistência (em Ω), dependendo da intensidade da luz em sua superfície. Eles são baratos, fáceis de adquirir em vários tamanhos e especificações, mas também são muito imprecisos.

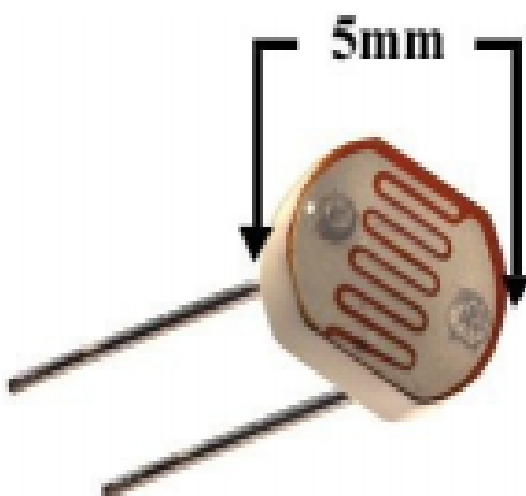


FIGURA 31: Componente, sensor de luminosidade
Fonte: JACQUET et al, 2013

- **Sensor de temperatura e umidade:** O sensor visto na figura 32 faz exatamente o que o nome sugere, lê a temperatura e a umidade do ar. É um sensor básico e geralmente lento, mas são ótimos para se fazer algumas coletas básicas de dados.

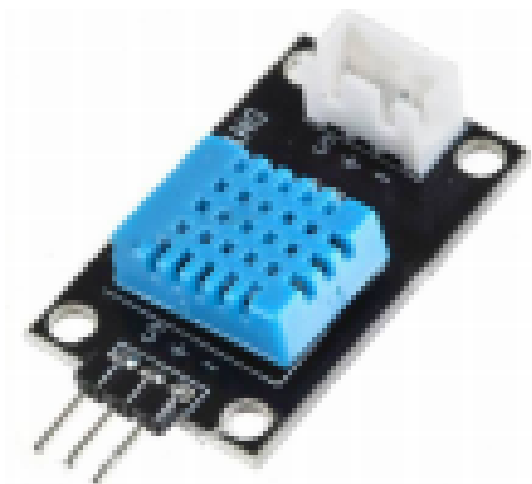


FIGURA 32: Componente, sensor de temperatura
Fonte: JACQUET et al, 2013

- **Sensor Infravermelho:** utilizado para fazer leituras em infravermelho para detectar a distância do sensor até objetos, isto é, através da leitura em infravermelho é possível fazer a detecção de objetos que estejam à frente do sensor, muito útil caso se construa um robô que necessite desviar de obstáculos por exemplo. A figura 33 traz um exemplo desse tipo de sensor.



FIGURA 33: Componente, sensor de infravermelho
Fonte: JACQUET et al, 2013

- **Servo Motor:** é um motor com amplitude de angulação inferior aos demais motores, em geral, conseguindo realizar um deslocamento de 180° . Um servo motor (figura 34) recebe um sinal de controle, verifica a posição que está e então se move para a posição desejada. Eles são muito mais precisos do que os demais motores para atingir a posição final desejada.



FIGURA 34: Componente, servo motor
Fonte: JACQUET et al, 2013

- **Display / Display de Cristal Líquido (LCD):** O *Display* de 7 segmentos é um dispositivo muito útil quando se quer mostrar alguma informação numérica. Já o *Display* de Cristal Líquido (*Liquid Cristal Display*) é mais utilizado para mostrar informações na forma de texto. Na figura 35 é exposto um exemplo de cada tipo de display citado.

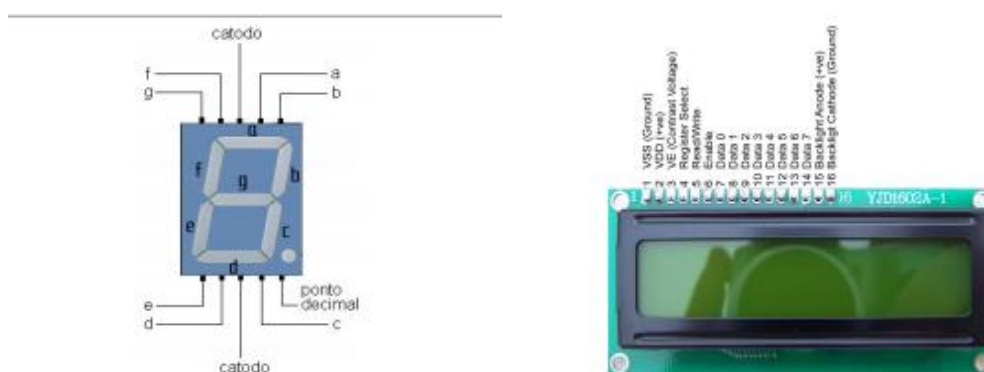


FIGURA 35: Componente, displays
Fonte: JACQUET et al, 2013

1.3.2.2 Software

O *Software* de desenvolvimento Arduino é bastante fácil e intuitivo de utilizar, não havendo qualquer nível de dificuldade. Foram estruturados passos de forma a simplificar a sua utilização e instalação.

Como qualquer computador, o Arduino precisa de um *software* para executar comandos. Esse *software* será desenvolvido na Arduino IDE em nosso PC, utilizando a linguagem *Wiring*, bem próxima a sintaxe da linguagem C. Após escrever o código, o compilaremos e então faremos o envio da versão compilada à memória *flash* do Arduino, através da porta USB. A partir do momento que o código é gravado no Arduino não precisamos mais do PC, pois como o Arduino é um computador independente, conseguirá sozinho executar o código que criamos desde que seja ligado a uma fonte de energia.

1.3.3 Arduino e o Sistema Embarcado

Nos últimos anos, a área de Sistemas Embarcados tem avançado de forma muito dinâmica. Esse dinamismo dá-se por diversos aspectos, ora pela convergência das redes de telecomunicações, ora pela convergência digital, ora pela evolução da microeletrônica e, por conseguinte, dos produtos advindos desse meio (sensores e atuadores), além da evolução das linguagens de programação e da própria forma de programar esses sistemas. (SILVA *et al*, 2012.)

Em uma pesquisa realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostra que 21,4% da população brasileira se sentem insegura no domicílio em que reside. Em virtude disso houve um crescimento considerável no setor de automação residencial, instalação de sensores com diversas finalidades, como detectar movimentos, fumaça, acionamento de lâmpadas, portões eletrônicos e outros benefícios que fazem do lar um local seguro, prático e confortável.

Marchesan (2012) expõem que essas porcentagens revelam uma tendência cada vez maior na busca por alternativas de segurança domiciliar, que vão desde

métodos simples, como grades, olho mágico, até procedimentos mais sofisticados, como afirma Pereira (2012) apud Marchesan (2012) “instalação de alarmes, câmeras de filmagem, seguranças particulares, entre outros”.

Há primórdio tais soluções eram vista como luxo pela sociedade, devido ao seu valor de custo, que variam de acordo como o que será implantado, impedindo assim a aquisição dos mesmos pela população de baixa renda. Nesse meio termo surge o Arduino, um sistema embarcado, simples, porém de muitas funcionalidades e finalidade. O uso do Arduino vem crescendo incalculavelmente, devido suas características que o tornam atraente, seu fácil manuseio e seu valor de mercado.

Desenvolver-se uma solução independente de outros sistemas, de baixo custo, com uma interface única, simples e intuitiva, para fins de configuração, monitoramento e acionamento dos dispositivos. O usuário deverá conectar os dispositivos eletrônicos na plataforma, ligá-la na rede, fazer a devida configuração na interface, e usufruir das funcionalidades do sistema tudo isso sem complicação (MONTEIRO, 2010 apud ZANDONÁ e VALIM, 2012)

Se analisarmos os aspectos da sociedade moderna, pode-se concluir que a maioria das pessoas passa grande parte do dia fora da sua residência, surge aí então a necessidade de equipamentos de segurança e outras comodidades que se utilizam dos sistemas embarcados, muitos usando o Arduino. Mas há a necessidade de vínculo desta residência com a grande mobilidade no mundo atual, através de interface gráfica online, geralmente acessada através de dispositivos móveis, e espera-se ainda que ela esteja sempre disponível para acesso ao utilizador. Ou seja, não apenas em casa, mas em qualquer local em que o utilizador sinta a necessidade de verificar ou controlar o estado de qualquer dispositivo na sua habitação.

Para chegarmos onde estamos hoje, a construção de sistemas digitais evoluíram desde o uso das grandes válvulas eletrônicas, passando pelos transistores até chegamos aos atuais circuitos integrados (CI) digitais. Essa evolução foi responsável por permitir a implementação de sistemas digitais cada vez mais complexos em um menor espaço físico, utilizando técnicas precisas e sofisticadas durante a fabricação dos CIs. (MOORE, 1998 apud PEIXOTO, 2013)

Em um projeto desenvolvido pela UFPEL - Universidade Federal de Pelotas (2013) através do Centro de Desenvolvimento Tecnológico - CDT, foi desenvolvido um sistema embarcado para rastreamento veicular, cuja principal funcionalidade estava diretamente pautada para dar assistência à gestão de tráfego aos transportes coletivos oferecidos aos alunos da instituição. O protótipo desse projeto utilizava a plataforma Arduino, devido ao seu custo e ampla documentação disponível. Vale lembrar que o mesmo encontra-se em operação, utilizando a plataforma Arduino como núcleo principal de processamento.

Sistemas embarcados, utilizando o microcontrolador Arduino vêm sendo largamente utilizado na Domótica, para diversas aplicações, como automação, iluminação, climatização, segurança, comunicação etc. Por esse motivo Brugnara (2007) apud Baumann (2008), defini a Domótica como uma tecnologia paralela a automação industrial, ela faz um gerenciamento dos equipamentos domésticos, que por sua vez são controlados por uma central computadorizada. Como já referenciado anteriormente, dessa forma permite o uso de dispositivos para automatizar as rotinas e tarefas da casa, distinguindo de controles normais por ter uma central que comanda tudo, podendo ser acoplada em um computador e/ou internet.

1.4 PROPOSTA DE PESQUISA E INOVAÇÃO

Pesquisas teóricas e experimentais para a criação de um protótipo de uma casa, valendo-se de conceitos de automação predial/residencial, onde a iluminação da mesma seja automatizada através de comunicação sem fios, utilizando recurso de baixo custo, foram realizadas. O desenvolvimento do protótipo utilizou na sua parte tangível materiais reaproveitáveis como acrílico e LEDs de celulares e um microcontrolador de baixo custo. Castrucci e Bottura (2006) apud Araujo *et al* (2012) expõem que a automação é considerada qualquer sistema apoiado em computadores que vise substituir tarefas de trabalho humano e/ou que vise soluções rápidas e econômicas para as indústrias e os serviços modernos.

O projeto foi desenvolvido em grande parte no laboratório de informática II do Instituto Federal de Educação Ciências e Tecnologia do Pará – IFPA, campus Itaituba, onde foram realizadas pesquisas para embasamento teórico acerca do assunto abordado, bem como a experimentação de algoritmos para controle e comunicação do microcontrolador. Os acadêmicos supracitados com o apoio do orientador conseguiram desenvolver um protótipo de uma casa cuja iluminação é automatizada, valendo-se do conceito de Domótica, para fins experimentais.

A realização de pesquisa teórica foi fundamental para o desenvolvimento do trabalho acadêmico, além dos experimentos realizados em laboratório. Pode ressaltar que a crescente popularização da Domótica, juntamente com a necessidade de elevar o conhecimento sobre o assunto, foram os pontos principais para a realização de pesquisas, com o intuito de obter êxito na implantação do protótipo abordado nesse projeto.

De forma bem simples Domótica é uma tecnologia que permite a gestão de todos os recursos habitacionais. O termo resulta da junção “Domus” (casa) com “Robótica” (controle automatizado de algo). Valendo-se disso foram observados custos e a facilidade de integração do código com o microcontrolador dentro das pesquisas teóricas sobre automação, para escolher-se o mais adequado a utilizar.

2 RESULTADOS

Para maior nitidez dos métodos e tecnologias utilizada para o desenvolvimento desse protótipo, esse capítulo será distribuído por seções, onde são esclarecidas todas as etapas percorridas para o almejo definido. Na primeira seção são apresentados os componentes de comunicação. Na segunda são propiciados os componentes utilizados para a elaboração do protótipo da casa utilizada no projeto. Na terceira e última seção será explanado os *softwares* utilizados para compilação de códigos e o *software* aplicativo para dispositivo móvel.

Colocar-se-á em prática todos os conhecimentos adquiridos pelas pesquisas aplicando fisicamente todos os comandos inseridos digitalmente.

Basicamente, computação física é uma forma de comunicação entre o mundo físico real e o mundo virtual do computador. A transformação de energia é o princípio básico necessário para tornar isso possível. (SEBASTIAN, 2008 apud NETO, 2011)

2.1 MATERIAS UTILIZADOS NO DESENVOLVIMENTO DO PROJETO

2.1.1 Componentes de comunicação

A escolha da plataforma Arduino para a consumação desse projeto deu-se entre vários motivos: custo/benefício; *hardware* e *software* “*open source*”; multiplataforma; por ter fácil portabilidade e devido à enorme biblioteca de projetos disponível na internet. Esses foram alguns dos critérios que levaram ao desígnio do mesmo, dentre os diversos outros modelos de plataformas que trabalham com automação no mercado.

E em meio às placas disponível que usam a plataforma Arduino, selecionou-se o Arduino Uno R3. Para conectar a placa Arduino à internet utilizar-se-á um módulo de *ethernet Shield* (Cobertura), que permite conectar-se à internet em poucos minutos, através de um cabo Rj45. Um roteador será utilizado para fazer a

comunicação entre o Arduino e o aplicativo. Um *notebook* também será utilizado para fazer a programação e servir de alimentação para a placa de automação.

2.1.1.1 Arduino Uno e Shield de Internet – Visão global

O Arduino Uno R3 (figura 36) é uma placa de microcontrolador baseado no ATmega328 (datasheet). Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas *Pulse Width Modulation* - PWM), 6 entradas analógicas, um de 16 MHz ressonador cerâmico, uma conexão USB, um conector de alimentação, um cabeçalho ICSP, e um botão de reset. Ele contém tudo o necessário para apoiar o microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC ou bateria-to-DC para começar. (ARDUINO.CC, 2014)



Figura 36: Microcontrolador Arduino Uno R3. Parte superior e inferior
Fonte: ARDUINO.CC, 2014

A tabela 03 mostra as principais características da placa de automação Arduino Uno R3.

CARACTERÍSTICAS – Arduino Uno R3

Microcontrolador	ATmega328
Voltagem Operacional	5V
Voltagem de entrada (recomendada)	7-12V
Voltagem de entrada (limites)	6-20V
Pinos E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente CC por Pino E/S	40 mA
Corrente CC para pino 3,3V	50 mA
Memória <i>Flash</i>	32 KB (ATmega328) dos quais 0,5KB são utilizados pelo <i>bootloader</i>
SRAM	2 KB (ATmega328)
EEPROM	1 KB (Atmega328)
Velocidade de <i>Clock</i>	16 MHz

Tabela 03: Características do Microcontrolador Arduino Uno R3

Fonte: ARDUINO.CC, 2014

Como já exposto, para conectar a placa de automação à internet para que haja comunicação entre a mesma e o aplicativo, foi usado um módulo de internet Arduino *Ethernet Shield*. O modelo da cobertura utilizada no projeto e visualizado na figura 37.

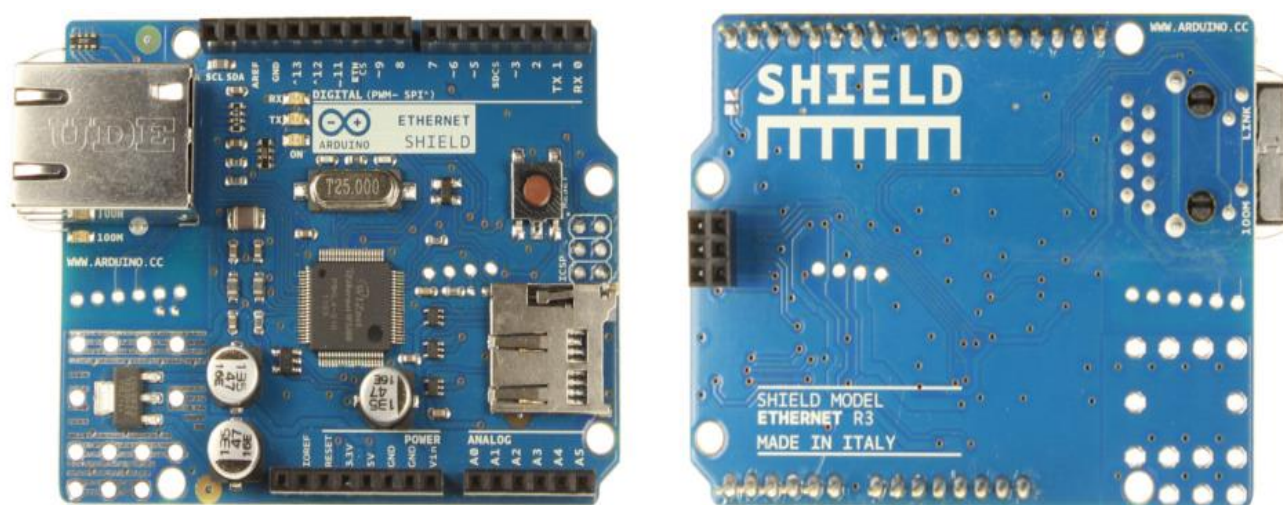


Figura 37: Shield (Cobertura) de internet. Parte superior e inferior

Fonte: ARDUINO.CC, 2014

Na tabela 04 são demonstradas em síntese as principais características da *Shield* de internet.

PRINCIPAIS CARACTERÍSTICAS – Arduino <i>Shield Ethernet</i>
Requer uma placa Arduino para seu funcionamento
5V Tensão de operação (fornecida pela placa Arduino)
Controlador Ethernet: W1500 com buffer interno 16K
Velocidade de conexão: 10/100Mb
Conexão com o Arduino na porta SPI
Conexão através de cabo de rede Rj45
Leitor de micro SD

Tabela 04: Características do shield (cobertura) de internet
Fonte: ARDUINO.CC, 2014

2.1.1.2 Roteador e cabo UTP com conector RJ45

Um roteador e um cabo UTP *Unshielded Twisted Pair* (Par Trançado sem Blindagem) com conector Rj45 (figura 38) foram utilizados para prover comunicação entre a placa Arduino através da cobertura de internet e o *smartphone* o qual possui instalado o aplicativo.



Figura 38: Cabo UTP, com conector RJ45 e Roteador
Fonte: MULTILASER e ALIEXPRESS, 2014

2.1.2 Componentes utilizados no protótipo da casa

Fazendo uma demonstração dos materiais utilizados para a criação do protótipo, podemos analisar que grande parte foram reaproveitados.

- ✓ Acrílico – A estrutura que simula as paredes, telhados e divisas da casa foram feitas com folhas de acrílicos (figura 39) sem serventias até então.



Figura 39: Folha de acrílico
Fonte: ALUFORTE, 2014

- ✓ Silicone – A chamada “cola-quente” foi utilizada para fazer as junções dos acrílicos para dar forma a casa. A figura 40 ratifica o afirmado anteriormente.



Figura 40: Pistola e silicone (cola quente)
Fonte: MERCADO LIVRE, 2014

- ✓ Fios reciclados – Os fios (figura 41) que simulam a rede elétrica da casa foram reaproveitados de carregadores de celulares.



Figura 41: Fios reciclados
Fonte: CASA BEM FEITA, 2014

- ✓ LEDs – Assim com os fios, os LEDs (figura 42) foram reutilizados de *flash* de aparelhos celulares sem utilidades.

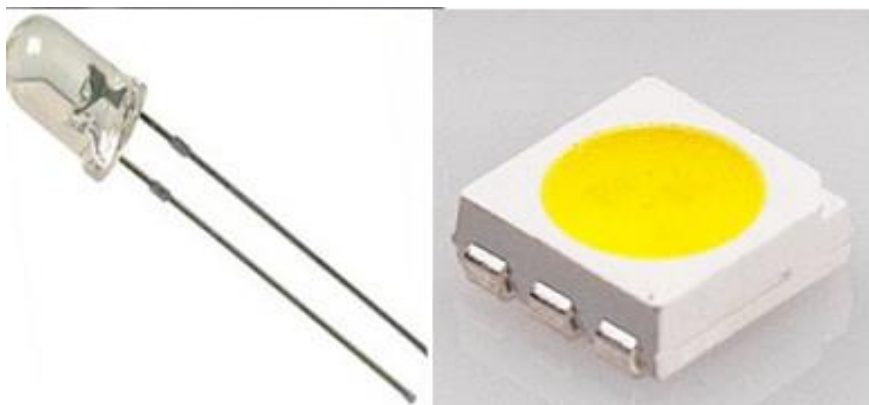


Figura 42: LEDs
Fonte: MAUEBIA e LUZELED, 2014

A figura 42 demonstra como ficou o protótipo da casa, com os materiais supracitados anteriormente.



Figura 43: Protótipo da casa
Fonte: BENTO e SILVA, 2014

2.1.3 Softwares utilizados

Os *softwares* utilizados para ao êxito desse projeto são listado a seguir:

- ✓ Arduino IDE 1.5.5 – Utilizado para a compilação e envio do código fonte para a placa Arduino Uno R3, através de cabo USB. O mesmo é visualizado na figura 44.

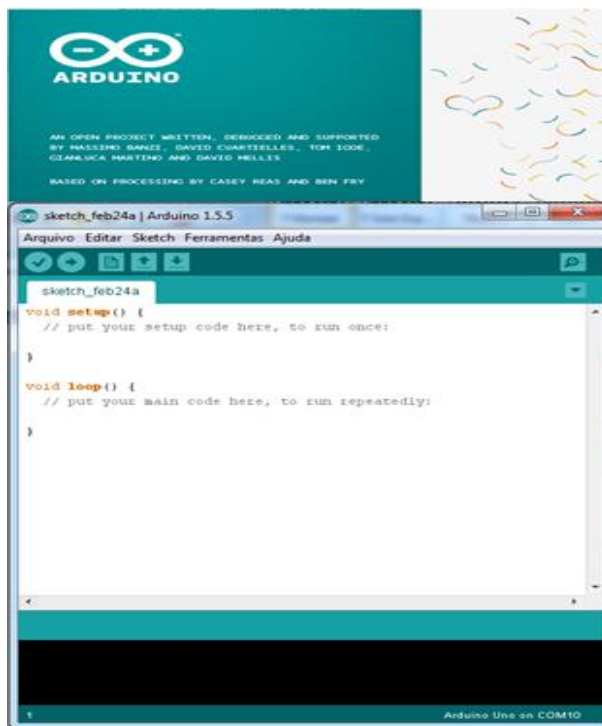


Figura 44: Software Arduino IDE 1.5.5
Fonte: ARDUINOPASSION, 2014

- ✓ iArduino – Software aplicativo iArduino (figura 45) foi instalado em um aparelho celular *iphone*, o qual serve para controlar as entradas digitais da placa Arduino através de um simples arrastar de dedo, podendo-se escolher entre as opções ligado/desligado.

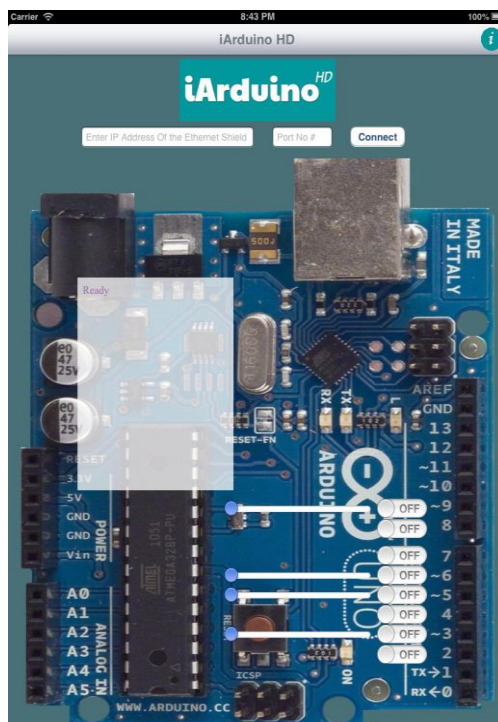


Figura 45: Softwares iArduino
Fonte: INSTRUCTABLES, 2014

2.2 RESULTADOS OBTIDOS

No decorrer do desenvolvimento obtivemos o seguinte trecho do código para interligação dos dispositivos, configurações de endereço MAC, IP, *Gateway*, Mascara de sub-rede e Porta local.

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0F, 0x09, 0xCC }; ← endereço MAC da cobertura de internet.

byte ip[] = { 192,168,0,110 }; ← Definição do endereço de IP da cobertura de internet.

//byte gateway[] = { 192,168,1,1 }; ← Definição do gateway (opcional)

//byte subnet[] = { 255,255,255,0 }; ← Endereço de sub-rede (opcional)

unsigned int localPort = 7777; ← Porta local. Deve ser igual no *software* aplicativo iArduino

IPAddress iPhoneIP(192, 168, 0, 103); ← Endereço de IP do iPhone.

A figura 46 demonstra o funcionamento da troca de códigos entre os dispositivos.

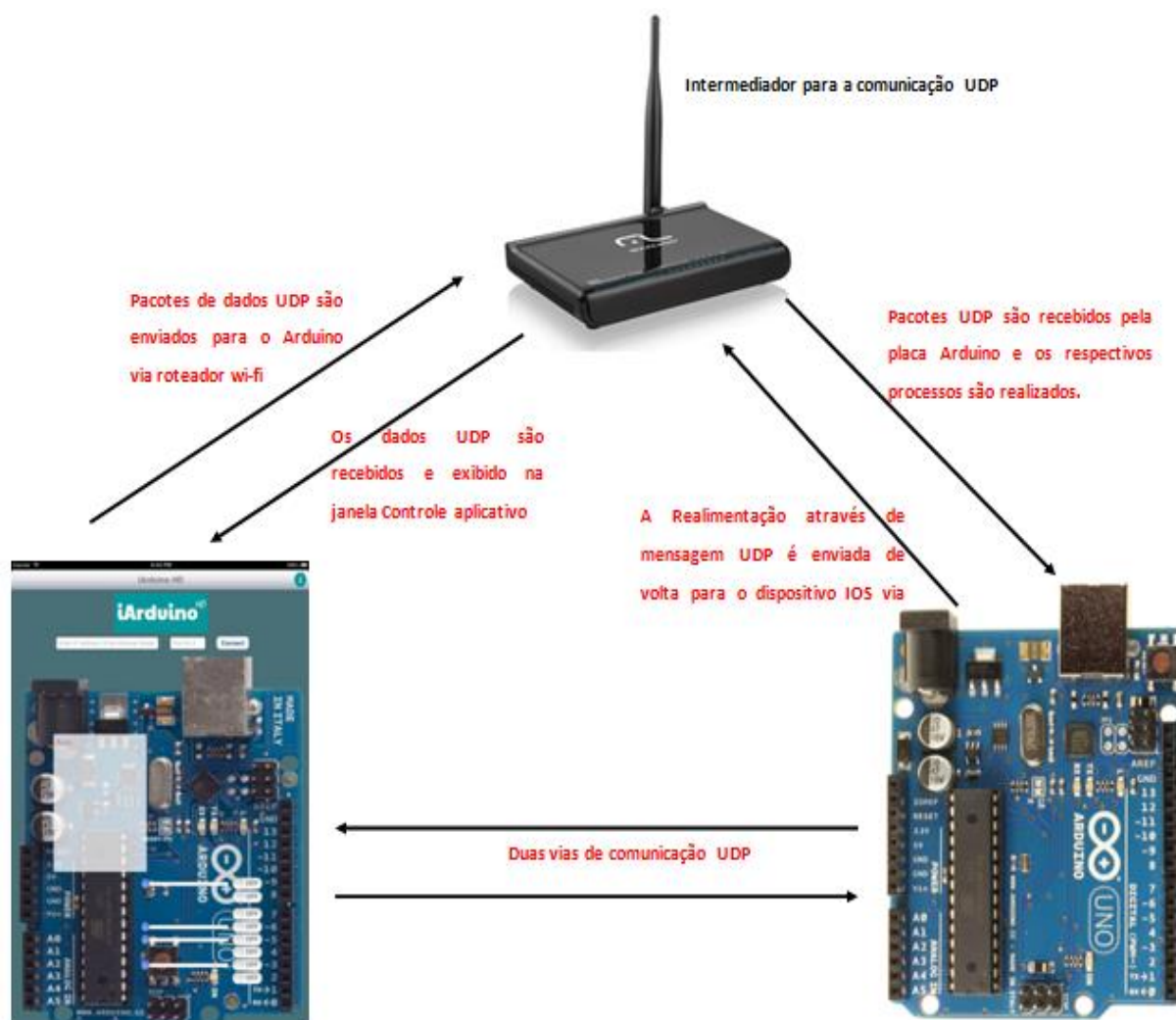


Figura 46: Resultado obtido – comunicação
Fonte: BENTO e SILVA, 2014

Para mostrá-los o funcionamento, criamos um protótipo com quatro compartimentos (cômodos) onde foi instalada a iluminação que irá ser gerenciada através de rede sem fio. A imagem 47 mostra a placa de automação Arduino Uno R3 com as ligações dos LEDs de cada compartimento do protótipo.

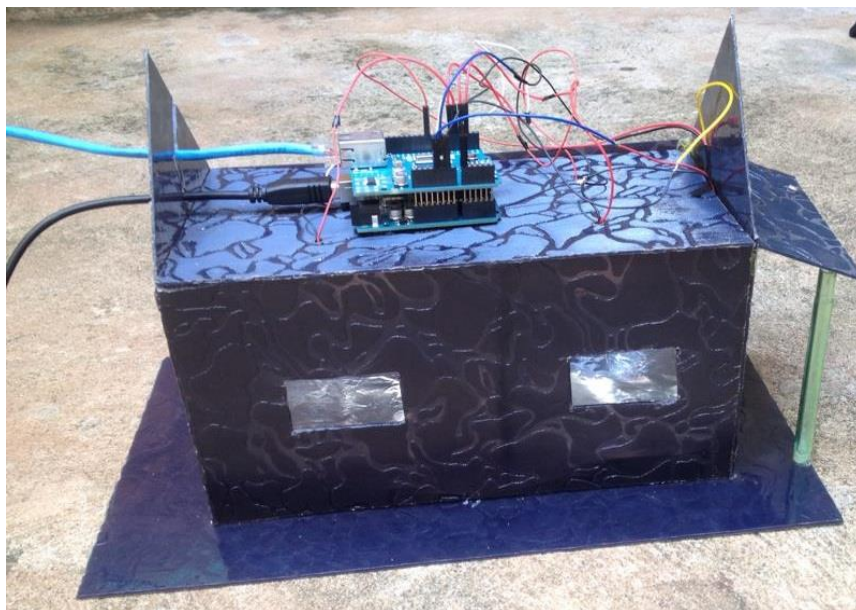


Figura 47: Resultado obtido – protótipo em funcionamento
Fonte: BENTO e SILVA, 2014

Na figura 48 é mostrada a vista superior do protótipo da casa, identificando e enumerando cada cômodo e o sensor de luminosidade.

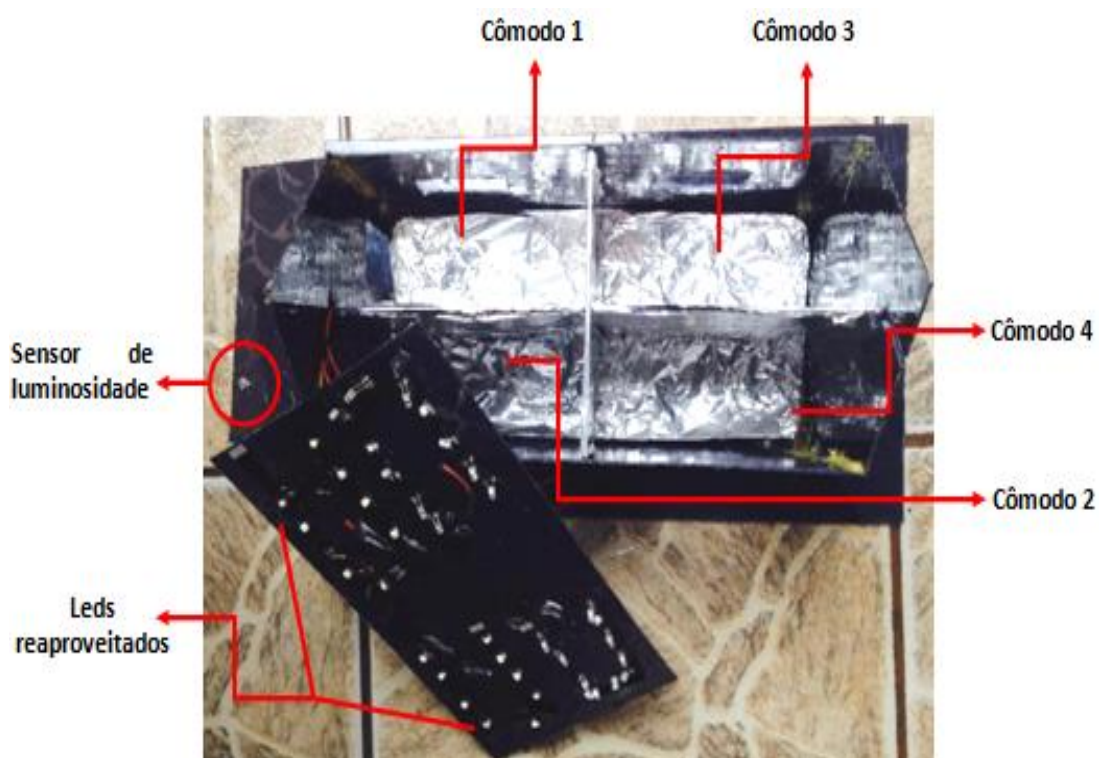


Figura 48: Resultado obtido – Vista superior do protótipo
Fonte: BENTO e SILVA, 2014

As ligações dos filamentos na placa Arduino são demonstradas na figura 49.

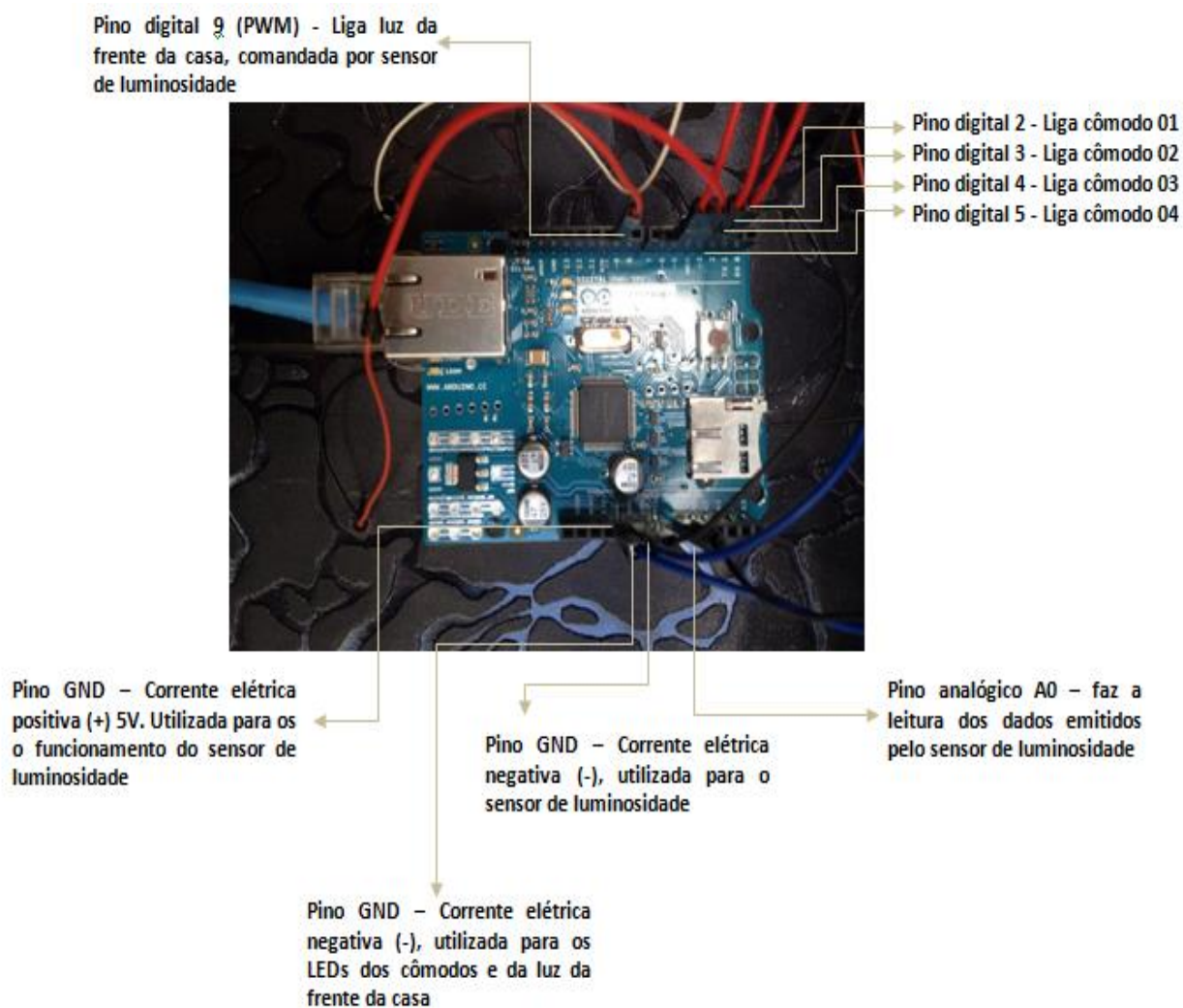


Figura 49: Resultado obtido – Ligação dos filamentos na placa Arduino
Fonte: BENTO e SILVA, 2014

3. DISCUSSÃO

As flexibilidades de aprendizado propiciadas pela plataforma Arduino foram constantemente notável no decorrer do projeto. O baixo consumo de energia torna-o próprio para uma aplicação onde a energia é um problema. O código fonte foi outro ponto positivo observado, a gama de projetos e artigos disponível na internet, propiciou a não inicialização do mesmo a partir do zero.

O microcontrolador possui sua estrutura comparável a um simples computador, colocado em um único chip todos os componentes necessários, como uma unidade de processamento central (CPU), a memória de programa e de dados e temporizadores incorporados. O microcontrolador possui flexibilidade média, assim como o tempo de projeto. A capacidade de processamento, se comparado com as demais arquiteturas citadas no artigo, é baixa. A capacidade de integração do microcontrolador é média, e o seu consumo de energia é baixo. (RODRIGUES, PEDÓ e TEDESCO, 2013)

O tempo para a realização do projeto foi considerável baixo, sendo que a integração dos códigos a fim de se chegar ao objetivo disposto foi o fato que demandou mais tempo. O aplicativo iArduino, propicia apenas o gerenciamento das portas digitais da placa de automação Arduino Uno R3, e a lâmpada da frente da casa utiliza um sensor de luminosidade, que é ligado a uma porta analógica, sabendo que a placa Arduino armazena apenas um código fonte em sua memória, houve assim a necessidade de integração dos dois códigos a fim de atingir o resultado esperado. Logo após foram realizados um série de teste analisando a efetivação da junção dos códigos e obtendo eficácia.

A princípio o objetivo foi apenas monitorar e controlar a iluminação do protótipo através de qualquer dispositivo móvel, com ou sem internet, desde computadores a celulares. Mas houve ainda a possibilidade de incorporar várias outras funcionalidades no projeto, como automação de porta de entrada da residência, sensor de temperatura, para no caso de um possível incêndio, mas embarramos na falta de conhecimento avançados no campo da eletrônica, e no momento foi acordado a não incorporação de tais funcionalidades pelo fato de

queremos ao máximo utilizar apenas o nosso conhecimento e mérito, sem menosprezar ajudas de terceiros.

No entanto, foi bastante satisfatório a realização do projeto, pela questão que ao final de tudo conseguimos nosso objetivo que era a automação da iluminação do protótipo de uma residência, com o microcontrolador Arduino e materiais reaproveitáveis.

CONCLUSÃO

Contemporaneamente o conceito de Domótica (casas inteligentes) tornar-se cada vez mais uma realidade. Esses modelos de casas utilizam-se de máquinas (microcontroladores) que gerenciam a integração dos diversos serviços acessível ao morador, através de sensores e atuadores. A constante correria, a insegurança, o racionamento de energia, a comodidade e conforto são adjetivos que conglomeram a sociedade atual, e o aparecimento de sistemas e equipamentos Domóticos revelam bastantes benefícios no âmbito decorrido.

A plataforma Arduino é considerada um sistema embarcado, muito difundido e utilizado em projetos de Domótica. Nesse estudo procurou-se abordar desde a base da Domótica, suas características, aplicabilidades, as placas utilizadas no processo de automações e suas peculiaridades. Destaca-se ainda os benefícios e facilidades que levaram a escolha da plataforma Arduino como base para esse protótipo, ressaltando também que a escolha da arquitetura a ser utilizada em determinado processo de automação altera-se de acordo com a necessidade da aplicação da mesma.

A plataforma Arduino foi desenvolvida por um professor, com o intuito de ensinar seus alunos a programarem sem ter um conhecimento aprofundando na área, ressaltasse a partir da daí tal facilidade que a mesma apresenta. Uma pessoa com uma base de programação é capaz de entender praticamente por completo qualquer código utilizado no desenvolvimento de aplicações com Arduino.

O intuito de criar um protótipo de uma residência Domótica veio com a necessidade de aprimoramento do conhecimento das novas tecnologias presente na nossa sociedade tecnológica. A simples automação da iluminação em um ambiente foi idealizada para melhor compreender a enorme gama de possibilidades que se pode criar através dessa ferramenta versátil e prática que é o microcontrolador Arduino.

Como a placa de automação Arduino só aceita um código por vez, ou seja, se uma pessoa quer usar o mesmo com diversas utilidades ao mesmo tempo, precisa-se então fazer a junção de códigos, havendo então uma pequena dificuldade

na hora de fazer tal junção. Porém o triunfo na junção do código utilizado no projeto veio após inúmeras pesquisas e análise do código fonte.

O protótipo em fase de funcionamento cumpre como seu proposto inicialmente, e a obtenção do êxito foi notável até então. Houve limitações, porém essas não foram nocivas no que diz respeito à realização do mesmo. Diversas funcionalidades podem ser acopladas no projeto, automação de portas e portões; sensores de temperatura que ajudam na proteção da residência contra incêndio; acionamento de eletroeletrônicos e eletrodomésticos com simples comandos via internet, ou via comando de voz; sensores de presença poderiam ser utilizados em algumas lâmpadas para prover seu acionamento sem ser preciso um simples toque na tela do celular, e vários outros métodos seriam de enorme valia, trazendo benefícios e enriquecendo ainda mais o protótipo.

Contudo, a Domótica além de proporcionar facilidades, conforto e segurança, colabora no quesito racionamento de energia elétrica, pois como verifica-se no protótipo desenvolvido, que o morador da residência, com um simples click na tela do seu aparelho celular, pode comandar diversos setores elétricos, evitando assim desperdício de recurso energético.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT – Associação Brasileira de Normas Técnicas. **NORMAS PARA REFERÊNCIAS, CITAÇÕES E NOTAS DE RODAPÉ (NBR 6023/NBR 10520 - 2002)**. 2002.

ALEXANDRE I. Gervini e EDGARD de F. Corrêa. **Avaliação de Desempenho, Área e Potência de Mecanismos de Comunicação em Sistemas Embarcados**. 2008. Disponível em: <<ftp://ftp.inf.ufrgs.br/pub/simoo/papers/semish03.pdf>>. Acessado em: 02 de Jan. de 2014

ALIEXPRESS. *Home page*. Disponível em: <www.aliexpress.com.br>. Acessado em: 15 de Mar. de 2014.

ALMEIDA, Marcelo Barros de. **SISTEMAS EMBARCADOS COM LINUX – PRIMEIROS PASSOS**. 2008. Disponível em: <<http://www.cirp.usp.br/arqs/4ciclo/Embarcado.pdf>>. Acessado em: 09 de Jan. de 2014.

ALUFORTE. *Home page*. Disponível em: <www.aluforte.com.br>. Acessado em: 15 de Mar. de 2014.

ARAÚJO, Ícaro Bezerra Queiroz de. *et al.* **DESENVOLVIMENTO DE UM PROTÓTIPO DE AUTOMAÇÃO PREDIAL/RESIDENCIAL UTILIZANDO A PLATAFORMA DE PROTOTIPAGEM ELETRÔNICA ARDUINO**. 2012. Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2012/artigos/103723.pdf>>. Acessado em: 17 de Fev. de 2014.

ARDUINO. *Home page*. Disponível em: <<http://www.Arduino.cc/en/Reference/Ethernet>>. Acessado em: 10 de Jan. de 2014.

ARDUINOPASSION. *Home page*. Disponível em: <www.arduino passion.com.br>. Acessado em: 15 de Mar. de 2014.

AURESIDE. **TEMAS TÉCNICO: CONCEITOS BÁSICOS, BENEFÍCIOS DA AUTOMAÇÃO**. 2005. Disponível em: <<http://www.aureside.org.br/temastec/default.asp?file=concbasicos.asp>>. Acessado em 25 de Set. de 2013.

BARRA, Fernanda. **EMPREENDEMENTO E ENGENHARIA**. 2013. Disponível em: <<http://blogdopetcivil.com/2013/01/>>. Acessada em: 05 de Jan. de 2014.

BARROS, Edna. CAVALCANTE, Sérgio. **INTRODUÇÃO AOS SISTEMAS EMBARCADOS**. 2008. Grupo e Engenharia da Computação – GRECO. Universidade Federal de Pernambuco – UFPE. Pernambuco, Brasil.

BAUMANN, Daniel. **PROTÓTIPO DE UM SISTEMA DE SEGURANÇA RESIDENCIAL COM LINUX EMBARCADO**. 2008. Disponível em: <http://www.bc.furb.br/docs/MO/2008/330370_1_1.pdf>. Acessado em: 06 de Jan. de 2014.

BENJAMIN, Téo Ferraz; NUNES, Raul Almeida e SUTTER, Dado. **SISTEMAS EMBARCADOS PARA AUTOMAÇÃO DE INSTRUMENTAÇÃO DE LABORATÓRIO**. 2008. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2008/relatorios/ctc/dcm/dcm%208.pdf>. Acessado em: 06 de Jan. de 2014.

BRASIL, Rodrigo Duque Ramos e RENNA, Roberto Brauer de. **INTRODUÇÃO AO KIT DE DESENVOLVIMENTO ARDUINO**. Programa de educação – tutorial, Universidade Federal Fluminense; Escola de Engenharia; Curso de Engenharia de Telecomunicações. Rio de Janeiro, 2013.

BRAVO, Giuliano Favaro; ALVES, Patrick Ferreira e SAAR, Thâmara Silva. Centro de Desenvolvimento Técnico “CEDTEC” – São Mateus. **AUTOMAÇÃO RESIDENCIAL**. 2009. Disponível em: <<http://www.conteudojuridico.com.br/pdf/cj029507.pdf>>. Acessado em: 06 de Jan. 2014.

CÂMARA, Marco Sérgio Andrade Leal. **INTELIGÊNCIA ARTIFICIAL: REPRESENTAÇÃO DE CONHECIMENTO**. 2001. Disponível em: <http://student.dei.uc.pt/~mcamara/artigos/inteligencia_artificial.pdf>. Acessado em: 02 de Jan. de 2014.

CASA BEM FEITA. *Home page*. Disponível em: <www.casabemfeita.com>. Acessado em: 15 de Mar. de 2014.

CHASE, Otavio. **SISTEMAS EMBARCADOS**. 2007. Disponível em: <<http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>>. Acessado em: 02 de Dez. de 2013.

CSE, Colégio Santa Emília. **APRENDA ROBÓTICA COM O ARDUINO – CURSO DE INTRODUÇÃO**. 2011. Disponível em: <<http://www.poli.br/~marcilio/Arduino/71924840-Apostila-Arduino.pdf>>. Acessado em 05: de Jan. de 2014.

CUNHA, Alessandro F. **O QUE SÃO SISTEMAS EMBARCADOS?**. 2013. Disponível em: <<http://www.techtraining.eng.br/files/uploads/2013/04/19/artigo-sist-emb.pdf>>. Acessado em: 19 de Dez. de 2013.

DAMKE, Daniele Erica e MORAES, Patrícia Freitas. **APLICAÇÃO DA ABORDAGEM GQM PARA A DEFINIÇÃO DE UM PROCESSO DE ENGENHARIA DE REQUISITOS DE SOFTWARE EMBARCADO**. 2008. Disponível em: <<http://claudiameloprof.files.wordpress.com/2010/08/aplicac3a7c3a3o-da-abordagem-gqm-para-a-definic3a7c3a3o-de-um-processo-de-engenharia-de-requisitos-de-software-embarcado.pdf>>. Acessado em: 07 de Jan de 2104

DE LA ROCHA, Fábio Rodrigues. **CURSO INTRODUTÓRIO DE MICROCONTROLADORES – PLATAFORMA ARDUINO**. 2010. Disponível em: <<http://www.pb.utfpr.edu.br/frr/pagina/Cursos/Arduino.pdf>>. Acessado em: 04 de Jan. de 2014.

DENARDIM, Gustavo Weber. **MICROCONTROLADORES**. 2009. <http://pessoal.utfpr.edu.br/gustavo/apostila_micro.pdf>. Acessado em 10 de Mar. de 2014.

DIAS, César Luiz de Azevedo. **DOMÓTICA: APLICABILIDADE ÀS EDIFICAÇÕES RESIDENCIAIS**. Dissertação de mestrado, Universidade Federal Fluminense. Niterói, 2004

DIAS, César Luiz de Azevedo e PIZZOLATO, Nélcio Domingues. **DOMÓTICA: APLICABILIDADE E SISTEMAS DE AUTOMAÇÃO RESIDENCIAL**. 2004. Disponível em: <<http://www.essentiaeditora.iff.edu.br/index.php/vertices/article/view/File/1809-2667.20040015/86>>. Acessado em: 22 de Dez. de 2013.

DI RENNA, Roberto Brauer *et al.* **INTRODUÇÃO AO KIT DE DESENVOLVIMENTO ARDUINO (Versão: A2013M10D02)**. 2013. Disponível em: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/Arduino/Tut_Arduino.pdf>. Acessado em: 04 de Jan. de 2014.

GLOBALCODE - The Developers Company. **MANUAL DO PROGRAM – ME**. 2009. Disponível em: <<http://www.eletronlivre.com.br/program-me/download/manual%20v2.pdf>>. Acessado em: 05 de Jan. de 2014.

GOOGLE. Site de pesquisa. Disponível em: <<http://www.google.com>>. Acessado em: 20 de Fev. de 2014.

GOMES, Dennis dos Santos. **INTELIGÊNCIA ARTIFICIAL: CONCEITOS E APLICAÇÕES**. 2010. Disponível em: <<http://www.olharcientifico.kinghost.net/index.php/olhar/article/viewFile/49/37>>. Acessado em 04: de Jan. de 2014.

HERNANDEZ, Helton Xavier & KALIL, Fahad. **VIABILIDADE DA PLATAFORMA ARDUINO E DO PROTOCOLO ZIGBEE NA ELABORAÇÃO DE REDES DE SENSORES SEM FIO (RSSF)**. 2013. Disponível em: <http://mic.imed.edu.br/2013/wp-content/plugins/SubmissaoMIC/files/523f7c325_057e.pdf>. Acessado em: 05 de Jan. 2014.

INSTRUCTABLES. *Home page*. Disponível em: <www.instructables.com>. Acessado em: 15 de Mar. de 2014.

JACQUET, Hewerson Antonio Perdomo *et al.* **SUBPROJETO DESTACOM (DESPERTANDO NOVOS TALENTOS EM COMPUTAÇÃO) – ATIVIDADE ARDUINO**. 2013. Disponível em: <<http://lscad.facom.ufms.br/wiki/images/3/3b/Arduino.pdf>>. Acessado em: 05 de Jan. de 2104.

JUSTEN, Álvaro. **CURSO DE ARDUINO**. 2012. Disponível em: <<http://www.cursodeArduino.com.br/apostila/apostila-rev4.pdf>>. Acessado em: 05 de Jan. 2014.

LAGIDO, Hugo. **CONSUMO ENERGÉTICO DOMÉSTICO**. 2009. Disponível em: <<http://hlagido.wordpress.com/2009/02/04consumo-energetico-domestico/>>. Acessado em: 02 de Out. de 2013.

LOPEZ, André e NETO, Cristóvão. **SISTEMA DE CONTROLE DE EDIFÍCIOS INTELIGENTES DOMÓTICO**. Engenharia da Informática. Castelo Branco, 2004.

LUZELED. *Home page*. Disponível em: <www.luzeled.com.br>. Acessado em: 15 de Mar. de 2014.

MAUEBIA. *Home page*. Disponível em: <www.mauebia.com.br>. Acessado em: 15 de Mar. de 2014.

MARCHESAN, Marcelo. **SISTEMA DE MONITORAMENTO RESIDENCIAL UTILIZANDO A PLATAFORMA ARDUINO**. 2012. Disponível em: <http://www.redes.ufsm.br/docs/tccs/Marcelo_Marchesan.pdf>. Acessado em: 27 de Dez. de 2013.

McROBERTS, Michael. **ARDUINO BÁSICO**. 2011. Disponível em: <<http://www.novateceditora.com.br/livros/Arduino/capitulo9788575222744.pdf>>. Acessado em: 05 de Jan. de 2014.

MERCADOLIVRE. *Home-page*. Disponível em: <<http://www.mercadolivre.com.br>>. Acessado em: 07 de Jan. de 2014.

MULTILASER. *Home page*. Disponível em: <www.multilaser.com.br>. Acessado em: 15 de Mar. de 2014.

NETO, Cristóvão. LOPES, André e FREITAS, Hugo. **SISTEMA DE CONTROLO DE EDIFÍCIOS INTELIGENTES DOMÓTICA**. 2004. Trabalho Prático de Engenharia de Informática. Escola Superior de Tecnologia – Instituto Politécnico de Castelo Branco. Castelo Branco, Brasil.

NETO, Pedro S. do Nascimento *et al.* **RESUMO DE MINICURSO – INTRODUÇÃO AO DESENVOLVIMENTO DE SOFTWARE EMBARCADO PARA SISTEMAS ROBÓTICOS USANDO ARDUINO**. 2011. Disponível em: <<http://www.die.ufpi.br/ercemapi2011/escolas/IntroducaoAoDesenvolvimentoDeSoftwareEmbarcadoParaSistemasRoboticosUsandoArduino.pdf>>. Acessado em: 02 de Jan. de 2014.

NETO, Reinaldo Oliveira. **AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL UTILIZANDO MICROCONTROLADOR ARDUINO E TABLET IPAD VIA WI-FI**. 2011. Tese de graduação – Centro Universitário de Brasília – UniCEUB. Disponível em: <<http://www.repositorio.uniceub.br/bitstream/123456789/3197/2/20615176.pdf>>. Acessado em 09 de Fev. de 2014.

NUNES, Renato. **ANÁLISE COMPARATIVA DE TECNOLOGIA PARA DOMÓTICA**. Instituto Superior Técnico. 2004

OLIVEIRA, Tadeu F. **DESENVOLVIMENTO DE APLICAÇÃO PARA SISTEMAS EMBUTIDOS: UM ESTUDO DA PLATAFORMA J2ME**. 2006. 71 f. Dissertação (Bacharelado em Ciências da Computação) – Universidade Tiradentes.

OSÓRIO, Fernando. **REDES NEURAIS - APRENDIZADO ARFICIAL**. 1999. Disponível em: <<http://osorio.wait4.org/oldsite/IForumIA/fia99.pdf>>. Acessado em: 30 de Dez. de 2013.

PEIXOTO, Thiago Moratori *et al.* Departamento de Ciências da Comunicação – Universidade Federal de Juiz de Fora (UFJF). **SISTEMAS EMBARCADOS: EXPLORE SUA CRIATIVIDADE CONSTRUINDO HARDWARE E SOFTWARE**. 2013. Disponível em: <http://www.ufjf.br/nrc/files/2013/05/capitulo_12.pdf>. Acessado em: 28 de Dez. de 2013.

PRIMO, Alex e COELHO, Luciano Roth. **COMUNICAÇÃO E INTELIGÊNCIA ARTIFICIAL: INTERAGINDO COM A ROBÔ DE CONVERSAÇÃO CYBELLE**. 2002. Disponível em: <<http://www.ufrgs.br/limc/PDFs/cybelle.pdf>>. Acessado em: 03 de Jan. de 2014.

RODRIGUES, Evandro. PEDÓ, Ricardo e TEDESCO, Leonel Pablo. **SISTEMAS EMBARCADOS E SUA APLICAÇÃO NA INDÚSTRIA**. 2013. Programa de pós-graduação em sistemas e processos industriais – Universidade de Santa Cruz do Sul (UNISC). Santa Cruz do Sul, Brasil.

ROGERCOM. Home-page. Disponível em: <<http://www.rogercom.com/>>. Acessado em: 07 de Jan. de 2014.

SANTOS, Nuno Pessanha. **ARDUINO – INTRODUÇÃO E RECURSOS AVANÇADOS**. 2009. Disponível em: <http://www.isegi.unl.pt/docentes/vlobo/escola_naval/MFC/Tutorial%20Arduino.pdf>. Acessado em: 05 de Jan. de 2014.

SILVA, Jorge Frederiscon de Macedo Costa da *et al.* **CONSTRUINDO VEÍCULO TELEOPERADO COM ARDUINO PARA AUXÍLIO NO ENSINO DE SISTEMAS EMBARCADOS E ROBÓTICA MÓVEL**. 2012. Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104216.pdf>>. Acessado em: 04 de Jan. de 2014.

SILVA, Rodrigo Berto da. **CONTROLE DE MANIPULADORES ROBÓTICOS**. 2012. Disponível em: <<http://www.ebah.com.br/content/ABAAAenJ8AF/controllemanipuladore-roboticos>>. Acessado em: 04 de Jan. de 2014.

TAURION, Cezar. **SOFTWARE EMBARCADO - A NOVA ONDA DA INFORMÁTICA**. Rio de Janeiro: Brasport, 2005

TEIXEIRA, João de Fernandes. **O QUE É INTELIGÊNCIA ARTIFICIAL**. 1990. Disponível em: <http://www.filosofiadamente.org/images/stories/pdf/o_que_e_inteligencia_artificial.pdf>. Acessado em: 04 de Jan. de 2014.

UFMS – Universidade Federal Mato Grosso do Sul. Grupo de Robótica. **INTRODUÇÃO AO ARDUINO**. 2012. Disponível em: <http://destacom.ufms.br/mediawiki/images/9/9f/Arduino_Destacom.pdf>. Acessado em: 05 de Jan. de 2014.

UFPEL - Universidade Federal de Pelotas – Centro de Desenvolvimento Tecnológico. **SISTEMA DE RASTREAMENTO PARA VEÍCULOS DE TRANSPORTE COLETIVO UTILIZANDO TECNOLOGIA ARDUÍNO**. 2013. Disponível em: <http://cti.ufpel.edu.br/cic/arquivos/2013/CE_02498.pdf>. Acessado em: 30 de Dez. de 2013

UFPR - Universidade Federal do Paraná – setor de Tecnologia Departamento de Engenharia Elétrica. **SISTEMAS EMBARCADOS**. 2008. Disponível em: <<http://www.elétrica.ufpr.br/graduacao/noturno/embarcados.html>>. Acessado em: 12 de Dez. de 2013.

WERNECK, Silva Bianchi de Frontin. **DOMÓTICA: UNIÃO DE ARQUETETURA E TECNOLOGIA DA INFORMAÇÃO NA EDIFICAÇÃO URBANA**. Tese de Mestrado em Arquitetura, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 1999.

WOLF, Wayne. **COMPUTERS AS COMPONENTS**. Nova Iorque: McGraw-Hill, 2001.

ZANDONÁ, Pablo Tirloni & VALIM, Paulo Roberto Oliveira. **INTERFACE HOMEM-MÁQUINA PARA DOMÓTICA BASEADA EM TECNOLOGIAS WEB EM UM SERVIDOR EMBARCADO**. 2012. Disponível em: <<http://www.aedb.br/seget/artigos12/41016425.pdf>>. Acessado em: 05 de Jan. de 2014.

ZANDONÁ, Pablo Tirloni e VALIM, Paulo Roberto Oliveira. **INTERFACE HOMEM-MÁQUINA PARA DOMÓTICA BASEADA EM TECNOLOGIAS WEB EM UM SERVIDOR EMBARCADO**. 2012. Disponível em: <<http://www.aedb.br/seget/artigos12/41016425.pdf>>. Acessado em: 05 de Jan. de 2014.

APÊNDICE

CÓDIGO FONTE UTILIZADO NO PROJETO.

```
#include <SPI.h> //Para Arduino até a versão 0018
```

```
//#include <EthernetUdp.h>
```

```
#include <Ethernet.h>
```

```
////////// INFORMAÇÕES DE TRABALHO //////////
```

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x78, 0xE0 }; //Defina o endereço MAC do seu shield Ethernet aqui - certifique-se de substituir o endereço MAC com os valores do shields
```

```
byte ip[] = { 192,168,1,102 }; //Defina o endereço IP desejado do seu shield's por aqui - verifique a sua rede para obter detalhes de configuração
```

```
//byte gateway[] = { 192,168,1,1 }; //Se você precisar defina um gateway (Opcional)
```

```
//byte subnet[] = { 255,255,255,0 }; //Altere isso para o seu endereço de sub-rede (Opcional)
```

```
unsigned int localPort = 7777; //Porta local para escutar
```

```
IPAddress iPhoneIP(192, 168, 1, 101); //Deixe-o como o seu é / / Definir o endereço IP do iPhone / iPod / iPad para enviar mensagens de volta para iArduino
```

```
unsigned int iPhonePort = 7777; //Deixe-o como o seu é. Definir como você configurou em no aplicativo iArduino
```

```
////////////////////////////////////// ATRIBUIÇÕES DOS PINOS ////////////////////////////////////////
```

```
//Não mude nada, a menos que você entenda o que esta parte do código faz
```

```
int relay2 = 2;
```

```
int relay3 = 3;
```

```
int relay4 = 4;
```

```
int relay5 = 5;
```

```
int relay6 = 6;
```

int relay7 = 7;

int relay8 = 8;

int relay9 = 9;

////////// VARIÁVEIS UDP //////////

//As duas variáveis seguintes são definidos quando um pacote é recebido

//byte remotep[4]; //Mantém originário que recebeu o pacote IP

//unsigned int remotePort; //Mantém a porta de origem do pacote recebido

//Tampões para receber e enviar dados

Char packBuff[UDP_TX_PACKET_MAX_SIZE]; //Buffer para armazenar pacote de entrada

////////////////////////////////////

EthernetUDP Udp;

void setup() {

//Iniciar o Ethernet e UDP

Ethernet.begin(mac,ip); //Você não precisa definir o seu gateway padrão ou sub-rede manualmente, use este

//Ethernet.begin(mac,ip,gateway,subnet); //Use esta linha, em vez se você definir manualmente todos os parâmetros

Udp.begin(localPort); //Setup UDP Configuração na porta definida anteriormente

Serial.begin(9600); //Comece a comunicação serial com o PC

pinMode(relay2,OUTPUT); //Designar os pinos de saída

pinMode(relay3,OUTPUT); //Designar os 67 pinos de saída

pinMode(relay4,OUTPUT); //Designar os pinos de saída

pinMode(relay5,OUTPUT); //Designar os pinos de saída

pinMode(relay6,OUTPUT);

pinMode(relay7,OUTPUT);

pinMode(relay8,OUTPUT);

pinMode(relay9,OUTPUT);

}

void loop()

{

//Inteiros que irão realizar os nossos valores PWM para uso posterior

int relay2pwmVal;

int relay3pwmVal;

int relay4pwmVal;

int relay5pwmVal;

int relay6pwmVal;

int relay7pwmVal;

int relay8pwmVal;

int relay9pwmVal;

////////////////////////////////////

//Se há dados disponíveis, ler um pacote

int packetSize = Udp.parsePacket(); //Note que este inclui o cabeçalho UDP

if(packetSize)

{

packetSize = packetSize - 8; //Subtrair o cabeçalho 8 byte

Serial.print("Packet size: ");

Serial.println(packetSize);

//leia o pacote em packetBuffer e obter o número de remetentes do endereço IP e porta

Udp.read(packBuff,UDP_TX_PACKET_MAX_SIZE);

Serial.println("Message: ");

Serial.println(packBuff);

relay2pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

relay3pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

relay4pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

relay5pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

relay6pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

relay7pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');

```
relay8pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');  
relay9pwmVal = (packBuff[3] - '0')*100 + (packBuff[4] - '0')*10 + (packBuff[5] - '0');
```

```
//////////////////////////////// PINO 2 //////////////////////////////////
```

```
    if (packBuff[0] = 'P' && packBuff[1]=='2' && packBuff[2]=='M') //Aguarde "PWMXXX"  
    e use XXX como valor para o PWM
```

```
{
```

```
    analogWrite(relay2,relay2pwmVal); //Definir Relé1 para PWM Valor
```

```
    Serial.println(relay2pwmVal); //Escrever notificação
```

```
}
```

```
else if (packBuff[0] = 'P' && packBuff[1]=='2' && packBuff[2]=='H') //Se aparecer a  
mensagem "P6H", em seguida, definir Relé1 (6) Alto
```

```
{
```

```
    digitalWrite(relay2,HIGH); //Ligue Relé1
```

```
    Serial.println("Pin 2 ON"); //Escrever notificação
```

```
    Udp.beginPacket(iPhoneIP,iPhonePort);
```

```
    Udp.write("Pin 2 is ON"); //Enviar Mensagem de volta para iPhone
```

```
    Udp.endPacket();
```

```
}
```

```
else if (packBuff[0] = 'P' && packBuff[1]=='2' && packBuff[2]=='L') //Se aparecer a  
mensagem "P6L", em seguida, definir Relé1 (6) baixo
```

```
{
```

```
    digitalWrite(relay2,LOW); //Desligue Relé1
```

```
    Serial.println("Pin 2 OFF"); //Escrever notificação
```

```
    Udp.beginPacket(iPhoneIP, iPhonePort);
```

```
    Udp.write("Pin 2 is OFF"); //Enviar Mensagem de volta para iPhone
```

```
    Udp.endPacket();
```

```
}
```

```
//////////////////////////////// PINO 3 //////////////////////////////////
```

```
if (packBuff[0] = 'P' && packBuff[1]=='3' && packBuff[2]=='M') //Aguarde "PWMXXX" e use XXX como valor para o PWM
```

```
{
```

```
    analogWrite(relay3,relay3pwmVal); //Definir Relé1 para PWM Valor
```

```
    Serial.println(relay3pwmVal); //Escrever notificação
```

```
}
```

```
if (packBuff[0] = 'P' && packBuff[1]=='3' && packBuff[2]=='H') //Se aparecer a mensagem "P6H", em seguida, definir Relé1 (6)ALTO
```

```
{
```

```
    digitalWrite(relay3,HIGH); //Ligue Relé1
```

```
    Serial.println("Pin 3 ON"); //Escrever notificação
```

```
    Udp.beginPacket(iPhoneIP,iPhonePort);
```

```
    Udp.write("Pin 3 is ON"); //Enviar Mensagem de volta para iPhone
```

```
    Udp.endPacket();
```

```
}
```

```
else if (packBuff[0] = 'P' && packBuff[1]=='3' && packBuff[2]=='L') //Se aparecer a mensagem "P6L", em seguida, definir Relé1 (6) LOW
```

```
{
```

```
    digitalWrite(relay3,LOW); // Desligue Relé1
```

```
    Serial.println("Pin 3 OFF"); //Escrever notificação
```

```
    Udp.beginPacket(iPhoneIP, iPhonePort);
```

```
    Udp.write("Pin 3 is OFF"); //Enviar Mensagem de volta para iPhone
```

```
    Udp.endPacket();
```

```
}
```

```
//////////////////////////////// PINO 4 //////////////////////////////////
```

```
if (packBuff[0] = 'P' && packBuff[1]=='4' && packBuff[2]=='M') //Aguarde "PWMXXX" e use XXX como valor para o PWM
```

```
{
```

```
    analogWrite(relay4,relay4pwmVal); //Definir Relé1 para PWM Valor
```

```
    Serial.println(relay4pwmVal); //Escrever notificação
```

```
}
```



```
if (packBuff[0] = 'P' && packBuff[1]=='4' && packBuff[2]=='H') //Se aparecer a
mensagem "P6H", em seguida, definir Relé1 (6) ALTO
```

```
{
    digitalWrite(relay4,HIGH); //Ligue Relé1
    Serial.println("Pin 4 ON"); //Escrever notificação
    Udp.beginPacket(iPhoneIP,iPhonePort);
    Udp.write("Pin 4 is ON"); //Enviar Mensagem de volta para iPhone
    Udp.endPacket();
}
```

```
else if (packBuff[0] = 'P' && packBuff[1]=='4' && packBuff[2]=='L') //Se aparecer a
mensagem "P6L", em seguida, definir Relé1 (6) LOW
```

```
{
    digitalWrite(relay4,LOW); //Desligue Relé1
    Serial.println("Pin 4 OFF"); //Escrever notificação
    Udp.beginPacket(iPhoneIP, iPhonePort);
    Udp.write("Pin 4 is OFF"); //Enviar Mensagem de volta para iPhone
    Udp.endPacket();
}
```

//////////////////////////////// PINO 5 //////////////////////////////////

```
if (packBuff[0] = 'P' && packBuff[1]=='5' && packBuff[2]=='M') //Aguarde "PWMXXX" e
use XXX como valor para o PWM
```

```
{
    analogWrite(relay5,relay5pwmVal); //Definir Relé1 para PWM Valor
    Serial.println(relay5pwmVal); //Escrever notificação
}
```

```
if (packBuff[0] = 'P' && packBuff[1]=='5' && packBuff[2]=='H') //Se aparecer a
mensagem "P6H", em seguida, definir Relé1 (6) ALTO
```

```
{
    digitalWrite(relay5,HIGH); //Ligue Relé1
    Serial.println("Pin 5 ON"); //Escrever notificação
    Udp.beginPacket(iPhoneIP,iPhonePort);
}
```

```

    Udp.write("Pin 5 is ON"); //Enviar Mensagem de volta para iPhone

    Udp.endPacket();

}

else if (packBuff[0] = 'P' && packBuff[1]=='5' && packBuff[2]=='L') //Se aparecer a
mensagem "P6L", em seguida, definir Relé1 (6) LOW

{

    digitalWrite(relay5,LOW); //Desligue Relé1

    Serial.println("Pin 5 OFF"); //Escrever notificação

    Udp.beginPacket(iPhoneIP, iPhonePort);

    Udp.write("Pin 5 is OFF"); //Enviar Mensagem de volta para iPhone

    Udp.endPacket();

}

//////////////////////////////// PINO 6 //////////////////////////////////

if (packBuff[0] = 'P' && packBuff[1]=='6' && packBuff[2]=='M') //Aguarde "PWMXXX" e
use XXX como valor para o PWM

{

    analogWrite(relay6,relay6pwmVal); //Definir Relé1 para PWM Valor

    Serial.println(relay6pwmVal); //Escrever notificação

}

if (packBuff[0] = 'P' && packBuff[1]=='6' && packBuff[2]=='H') //Se aparecer a
mensagem "P6H", em seguida, definir Relé1 (6) ALTO

{

    digitalWrite(relay6,HIGH); //Ligue Relé1.

    Serial.println("Pin 6 ON"); //Escrever notificação

    Udp.beginPacket(iPhoneIP,iPhonePort);

    Udp.write("Pin 6 is ON"); //Enviar Mensagem de volta para iPhone

    Udp.endPacket();

}

else if (packBuff[0] = 'P' && packBuff[1]=='6' && packBuff[2]=='L') //Se aparecer a
mensagem "P6L", em seguida, definir Relé1 (6) LOW.

{

```

```

    digitalWrite(relay6,LOW); //Desligue Relé1
    Serial.println("Pin 6 OFF"); //Escrever notificação
    Udp.beginPacket(iPhoneIP, iPhonePort);
    Udp.write("Pin 6 is OFF"); //Enviar Mensagem de volta para iPhone
    Udp.endPacket();
}

////////// PINO 7 //////////

if (packBuff[0] = 'P' && packBuff[1]=='7' && packBuff[2]=='M') //Aguarde "PWMXXX" e
use XXX como valor para o PWM
{
    analogWrite(relay7,relay7pwmVal); //Definir Relé1 para PWM Valor
    Serial.println(relay7pwmVal); //Escrever notificação
}

if (packBuff[0] = 'P' && packBuff[1]=='7' && packBuff[2]=='H') //Se aparecer a
mensagem "P6H", em seguida, definir Relé1 (6) ALTO
{
    digitalWrite(relay7,HIGH); //Ligue Relé1
    Serial.println("Pin 7 ON"); //Escrever notificação>
    Udp.beginPacket(iPhoneIP,iPhonePort);
    Udp.write("Pin 7 is ON"); //Enviar Mensagem de volta para iPhone
    Udp.endPacket();
}

else if (packBuff[0] = 'P' && packBuff[1]=='7' && packBuff[2]=='L') //Se aparecer a
mensagem "P6L", em seguida, definir Relé1 (6) LOW
{
    digitalWrite(relay7,LOW); //Desligue Relé1
    Serial.println("Pin 7 OFF"); //Escrever notificação
    Udp.beginPacket(iPhoneIP, iPhonePort);
    Udp.write("Pin 7 is OFF"); //Enviar Mensagem de volta para iPhone
    Udp.endPacket();
}

```

//////////////////////////////// PINO 8 //////////////////////////////////

if (packBuff[0] = 'P' && packBuff[1]=='8' && packBuff[2]=='M') // Wait for "PWMXXX" and use XXX as value for PWM <Aguarde "PWMXXX" e use XXX como valor para o PWM

{

analogWrite(relay8,relay8pwmVal); //Definir Relé1 para PWM Valor

Serial.println(relay8pwmVal); //Escrever notificação

}

if (packBuff[0] = 'P' && packBuff[1]=='8' && packBuff[2]=='H') //Se aparecer a mensagem "P6H", em seguida, definir Relé1 (6) ALTO

{

digitalWrite(relay8,HIGH); //Ligue Relé1

Serial.println("Pin 8 ON"); //Escrever notificação

Udp.beginPacket(iPhoneIP,iPhonePort);

Udp.write("Pin 8 is ON"); //Enviar Mensagem de volta para o iPhone

Udp.endPacket();

}

else if (packBuff[0] = 'P' && packBuff[1]=='8' && packBuff[2]=='L') //Se aparecer a mensagem "P6L", em seguida, definir Relé1 (6) LOW

{

digitalWrite(relay8,LOW); //Desligue Relé1

Serial.println("Pin 8 OFF"); //Escrever notificação

Udp.beginPacket(iPhoneIP, iPhonePort);

Udp.write("Pin 8 is OFF"); //Enviar Mensagem de volta para iPhone

Udp.endPacket();

}

//////////////////////////////// PINO 9 //////////////////////////////////

if (packBuff[0] = 'P' && packBuff[1]=='9' && packBuff[2]=='M') // Wait for "PWMXXX" and use XXX as value for PWM <Aguarde "PWMXXX" e use XXX como valor para o PWM

{

analogWrite(relay9,relay9pwmVal); //Definir Relé1 para PWM Valor

```

Serial.println(relay9pwmVal); //Escrever notificação
}

if (packBuff[0] = 'P' && packBuff[1]=='9' && packBuff[2]=='H') //Se aparecer a
mensagem "P6H", em seguida, definir Relé1 (6) ALTO
{

    digitalWrite(relay9,HIGH); //Turn on relay1 <Ligue Relé1

    Serial.println("Pin 9 ON"); //Escrever notificação

    Udp.beginPacket(iPhoneIP,iPhonePort);

    Udp.write("Pin 9 is ON"); //Enviar Mensagem de volta para iPhone

    Udp.endPacket();

}

else if (packBuff[0] = 'P' && packBuff[1]=='9' && packBuff[2]=='L') //Se aparecer a
mensagem "P6L", em seguida, definir Relé1 (6) LOW
{

    digitalWrite(relay9,LOW); //Desligue Relé1

    Serial.println("Pin 9 OFF"); //Escrever notificação

    Udp.beginPacket(iPhoneIP, iPhonePort);

    Udp.write("Pin 9 is OFF"); //Enviar Mensagem de volta para iPhone

    Udp.endPacket();

}

}

```

////////// CÓDIGO SENSOR DE LUMINOSIDADE //////////

//Aqui há a necessidade de integração dos códigos. Pois o iArduino controla apenas as portas digitais. E como o sensor funciona de forma analógica (captando informações do ambiente e transformado-as em dados que possam ser processados pela placa Arduino). Ou seja, tudo isso em uma das portas analógicas da placa de automação.

```

delay(20); //Tempo definido para executar o código em milésimos de segundo

ValorLido = analogRead(LDR); //Lê valor na porta analógica

if (ValorLido < 500) { //Definição da variação da luminosidade para que a
lâmpada possa ser ligada

    analogWrite(Led, pwm); //Definir que uma das extremidades da lâmpada tem
que ser conectada em uma porta PWM

```

```
    pwm++;  
    delay(100);  
}  
  
else {  
    digitalWrite(Led, LOW); //Na porta digital a lâmpada acende  
    pwm = 0;  
}  
  
    if(pwm > 255) { //Condição, se o valor lido for maior do que o estabelecido,  
então lâmpada acender  
        pwm=255;  
    }  
}  
}
```