

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL  
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

BENEDITO GABRIEL RAPOSO ROCHA JÚNIOR

**PROJETO DE AUTOMAÇÃO RESIDENCIAL COM A UTILIZAÇÃO DO  
ARDUINO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

BENEDITO GABRIEL RAPOSO ROCHA JÚNIOR

## **PROJETO DE AUTOMAÇÃO RESIDENCIAL COM A UTILIZAÇÃO DO ARDUINO**

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Automação Industrial, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Rodrigo Rodrigues Sumar

CORNÉLIO PROCÓPIO

2014

## **TERMO DE APROVAÇÃO**

BENEDITO GABRIEL RAPOSO ROCHA JÚNIOR

### **PROJETO DE AUTOMAÇÃO RESIDENCIAL COM A UTILIZAÇÃO DO ARDUINO**

Este trabalho de conclusão de curso foi apresentado como requisito parcial para obtenção do título de Tecnólogo em Automação Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Rodrigo Rodrigues Sumar  
Coordenador do Curso de  
Tecnologia em Automação Industrial

---

Prof. Dr. André Luiz Shiguemoto  
Responsável pela Atividade de Trabalho de Conclusão do Curso de  
Tecnologia em Automação Industrial

#### **BANCA EXAMINADORA**

---

Prof. Dr. Wagner Endo  
UTFPR

---

Prof. Esp. Carlos Alberto Pascoalino  
UTFPR

---

Prof. Dr. Rodrigo Rodrigues Sumar  
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

*Dedico esse trabalho primeiramente a Deus, à minha amada esposa que durante todo esse tempo sempre esteve ao meu lado, a minha mãe que me incentivou a abrir mão de muitas coisas para realizar esse curso.*

## **AGRADECIMENTOS**

Agradeço a todos os professores dessa instituição que como verdadeiros mestres sempre se dispuseram a ajudar e auxiliar no que precisei no âmbito acadêmico. Também agradeço a todos os amigos que ajudaram ou participaram de alguma forma durante o período de formação nesse curso.

## RESUMO

ROCHA JR, Benedito Gabriel Raposo. **Projeto de Automação Residencial com a Utilização do Arduino**. 2014. 64f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Automação Industrial), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

O projeto de automação residencial com a utilização do Arduino trata da possibilidade de automação e integração de componentes em uma residência. Palavras como domótica, e acessibilidade devem ser repetidas várias vezes nesse trabalho. Um protótipo funcional que deverá realizar alguns comandos básicos em uma casa será apresentado ao final. O Arduino, componente principal desse projeto será tratado e deverá ser realizada uma pesquisa específica sobre o componente. Esse trabalho realizara estudos sobre a domótica, estudo sobre as tendências sociais que aponto para a modernização das residências e sobre a possibilidade do uso de trabalhos como esse para integração em projetos de acessibilidade.

**Palavras chave:** Automação residencial. Domótica. Arduino. Acessibilidade.

## ABSTRACT

ROCHA JR, Benedito Gabriel Raposo. **Home Automation Project with Using the Arduino**. 2014. 64f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Automação Industrial), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

The project home automation using the Arduino comes to the possibility of automation and integration of components into a residence. Words such as home automation, and accessibility must be repeated several times in this work. A functional prototype that should perform some basic commands in a house will be presented at the end. The Arduino, the main component of this project will be handled and a specific research component should be performed. This work had conducted studies on automation, study on social trends that aim to modernize homes and about the possibility of using such works on projects for integration of accessibility.

**Keywords:** home automation. Domotique. Arduino. Accessibility.

## LISTA DE FIGURAS

Figura 1 Acessibilidade .....	19
Figura 2 - Arduino Mega.....	21
Figura 3 - Processamento .....	23
Figura 4 - Estrutura de Hardware.....	24
Figura 5 - Avr CPU .....	25
Figura 6 - <i>Ethernet Shield</i> e <i>Shield Zigbee</i> .....	26
Figura 7 - Software arduino .....	27
Figura 9 - Chip <i>Ethernet Shield</i> .....	32
Figura 10 - Placa <i>Ethernet Shield</i> .....	33
Figura 11 - Conexão <i>Ethernet Shield</i> com Arduino .....	33
Figura 12 - Módulo de Relê .....	34
Figura 13 - Esquema lâmpada relê .....	35
Figura 14 - Ligação Cooler .....	35
Figura 15 - Central Portão Automático .....	36
Figura 16 - Programa Eclipse.....	38



## LISTA DE TABELAS

Tabela 1 - Especificações Arduino.....	30
Tabela 2 - Orçamento.....	37

## LISTA DE QUADROS

Quadro 1 - Recorte Programa.....	37
Quadro 2 - Determinação de saída.....	38

## LISTA DE SIGLAS

IOS	<i>Iphone Operacional System</i>
GFK	<i>Gesellschaft für Konsumforschung</i>
PGNE	Pequenas Empresas e Grandes Negócios
ONU	Organizações das Nações Unidas
ONUBR	Organizações das Nações Unidas Brasil
Wi-Fi	<i>Wireless Fidelity</i>
USB	<i>Universal Serial Bus</i>
SMD	<i>Surface Mount Device</i>
CAD	<i>Computer Aided Design</i>
I/O	<i>Input/output</i>
PIR	<i>Passive Infrared Sensor</i>
PWM	<i>Pulse-Width Modulation</i>
ICSP	<i>In-Circuit Serial Programiming</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
AVR	AVR
RISC	<i>Reduced Instruction Set Computing</i>
CPU	<i>Central Processing Unit</i>
PC	<i>Personal Computer</i>
3G	3° Geração de tecnologia móvel
4G	4° Geração de tecnologia móvel

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	12
1.2 PROBLEMA .....	13
1.3 JUSTIFICATIVA .....	14
1.4 OBJETIVOS.....	15
1.4.1 OBJETIVO GERAL .....	15
1.4.2 OBJETIVOS ESPECÍFICOS .....	15
1.5 MÉTODO DE PESQUISA.....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	17
2.1 DOMÓTICA .....	17
2.2 DOMÓTICA E ACESSIBILIDADE .....	19
2.3 O MICROCONTROLADOR ARDUINO .....	21
2.4 O <i>SMARTPHONE</i> E O ANDROID .....	28
<b>3 MONTAGEM DO PROTÓTIPO</b> .....	30
3.1 ESPECIFICAÇÕES DO CONTROLADOR.....	30
3.2 DESCRIÇÃO DAS PARTES.....	31
3.3 ESPECIFICAÇÕES DA <i>ETHERNET SHIELD</i> .....	32
3.4 INTEGRAÇÃO DOS MÓDULOS DE CONTROLE E ESQUEMAS DE LIGAÇÃO .....	33
3.5 ORÇAMENTO .....	36
3.6 PROGRAMAÇÃO .....	37
<b>4 CONCLUSÃO</b> .....	39
4.1 PROPOSTA PARA TRABALHOS FUTUROS.....	41
<b>REFERÊNCIAS</b> .....	42
<b>REFERÊNCIA DE FIGURAS</b> .....	44
<b>APENDICÊ A: CÓDIGO PROGRAMADO NO ARDUINO</b> .....	45
<b>APENDICÊ B: CÓDIGO DO APLICATIVO DO <i>SMARTPHONE</i></b> .....	48
<b>ANEXO A: <i>DATA SHEET</i> ARDUINO MEGA 2560</b> .....	58
<b>ANEXO B: DIAGRAMA ESQUEMÁTICO DO ARDUINO</b> .....	62
<b>ANEXOS C: DIAGRAMA ESQUEMÁTICO DO <i>ETHERNET SHIELD</i></b> .....	63

# 1 INTRODUÇÃO

Com o crescente avanço da tecnologia, as tendências apontam para uma transformação no estilo de vida das pessoas, com isso, acaba acontecendo uma busca constante em direção ao conforto, comodidade e acessibilidade. Tomando proveito dessa situação hoje a automação residencial vem ganhando grande força no mercado.

A partir dessa ótica, empresas vêm desenvolvendo hardwares e sistemas de integração para residências, com o intuito de proporcionar a comodidade esperada aos clientes, que apenas com um clique podem vir a controlar boa parte ou todos os sistemas eletroeletrônicos da sua casa.

A intenção do presente trabalho é apresentar as tendências de modernização nessa área, a exposição de problemas que existem e podem ser explorados, também deverá ser feita a comprovação de que essa tecnologia está tornando-se mais acessível, que existe a possibilidade de criação de produtos mais baratos e inovadores nessa área. E ao final desse trabalho deve ser apresentado um protótipo funcional para automação residencial que venha estar suprindo essa necessidade de otimização, unificação de comandos e comodidade ao usuário em uma residência.

Com isso propõe-se a utilização de um micro controlador do tipo Arduino, para que através de uma interface eletrônica possa realizar-se comandos gerais em uma residência através de um aparelho celular do tipo *smartphone*. Esse aparelho terá controle tanto remoto quanto dentro da própria residência, com a utilização da internet.

A opção de utilização para o controle por telefones do tipo *smartphones*, se dá devido à popularização desta tecnologia, que hoje está disponível a mais ampla gama de públicos. A internet atualmente é um dos meios de comunicação mais utilizados e com as novas tecnologias de transmissão de dados do tipo 3G e 4G, torna-se uma opção barata e viável para esse tipo de aplicação.

Portanto, nesse contexto a realização de um projeto como este é bem viável, o trabalho no seu desenvolvimento deve envolver várias áreas de aplicação para a conclusão do mesmo, áreas como eletrônica, informática e comunicações.

## 1.2 PROBLEMA

No caso desse trabalho, como problema, pode-se encarar a necessidade de unificação dos controles em uma residência. Um dos maiores incômodos sem dúvida é o fato de o morador ter um controle para comandar cada aparelho da sua casa, ter que carregar várias chaves para poder entrar em sua residência, e, além disso, existem os casos de pessoas que simplesmente por alguma necessidade especial, cadeirantes ou idosos não podem realizar simples tarefas com tanta facilidade. Projetos de automação residencial vêm como uma boa alternativa para os dois casos, tanto proporcionando mais comodidade a qualquer usuário, tanto como proporcionando melhor acessibilidade a pessoas que tem algum tipo de necessidade especial.

O principal objetivo de desse trabalho não é o de realizar a completa integração dos componentes em uma casa, visa somente à construção de um sistema que possibilite tal integração, a parte de instalação e integração dos componentes será citada aqui somente como sugestiva. Uma abordagem teórica sobre o tema deverá ser realizada durante todo o trabalho, ou seja, o objetivo não será somente a criação do protótipo, mas sim a explanação e abordagem do tema que é tão pouco falado e carece de um estudo mais amplo, principalmente nas instituições de ensino, onde as tecnologias tanto referentes a instalações elétricas residenciais e projetos, remetem a estudos de 30 a 40 anos atrás. Projetos de modernização e automação, praticamente não são abordados em nenhum momento, a palavra domótica, por exemplo, quase não é ouvida.

Por isso, foi feita a opção de além da explanação teórica sobre o tema também realizar a construção de um protótipo funcional, nesse protótipo devemos agregar alguns equipamentos a serem controlados, lembrando, que serão apenas exemplificações do que poderá ser integrado ao sistema.

Outro problema que pode estar sendo solucionado com esse projeto é o controle remoto da residência, por exemplo, imaginasse que o morador tenha saído para ir ao mercado, nesse tempo um amigo chega a sua casa para realizar uma visita inesperada, essa pessoa liga para o morador que atende o seu telefone e através do mesmo realiza a abertura das entradas de sua casa, para que o seu amigo possa aguardá-lo dentro de sua residência. Ou mesmo em uma viagem onde

o dono de um sistema como esse pode simular a presença de alguém em sua casa apenas com o acender e desligar de luzes.

Outra opção de integração que não será abordada aqui é a utilização de sistema de câmeras, para melhor visualização e operação do sistema.

Dessa forma, o projeto de automação industrial com a utilização do Arduino, vem como uma excelente proposta para resolver alguns desses problemas que se encontram todos os dias na maioria das casas, a gama de aplicação é muito grande, por isso não serão amplamente abordadas, pois o trabalho ficaria muito extenso e provavelmente não haveria tempo hábil para a conclusão do mesmo. Lembrando que a utilização de sistemas de câmeras citado anteriormente foi apenas uma suposição teórica, não é o objetivo nesse trabalho integrar esses sistemas no protótipo, mas sim o de demonstrar as possibilidades existentes no mercado, e a acessibilidade à tecnologia do Arduino, que vem demonstrando-se ser além de um sistema com excelente custo-benefício, um sistema totalmente didático e que conta com um vasto banco de dados espalhados principalmente por suas comunidades na internet.

### **1.3 JUSTIFICATIVA**

Analisando essas afirmações, percebe-se que existe uma grande oportunidade de mercado referente à integração de sistemas elétricos residências. Por isso, o seguinte trabalho tem por objetivo a construção de um protótipo para inicialmente realizar controles básicos em uma residência, como acender lâmpadas, abertura de portas, etc.

A grande vantagem desse projeto é possibilidade de utilização de circuitos eletrônicos, que contam com tecnologias de alta miniaturização como a tecnologia SMD (*Surface Mount Device*), o que possibilita a construção otimizada do que anteriormente só seria possível com a construção de grandes painéis e complexos sistemas de chaveamento.

Algumas décadas após a introdução do transistor em 1940 tem-se percebido uma mudança muito dramática na indústria eletrônica. A minituarização crescente leva-nos a questionar sobre seus limites. Sistemas completos aparecem agora sobre um *wafer* milhares de vezes menor do que um simples elemento dos circuitos

anteriores. As vantagens associadas com os sistemas atuais, quando comparados com circuitos a válvulas dos anos anteriores, são, para a maioria, imediatamente óbvias: menores, mais leves (Boylestad/Nashelsky 2003)”.

Ou seja, utilizando a grande vantagem existente hoje com relação aos sistemas eletrônicos, as tecnologias inovadoras de pesquisa, como a da plataforma Arduino, necessidade crescente no mercado em busca de comodidade e acessibilidade. Abre-se uma boa oportunidade de investimento em módulos, sensores e sistemas integradores residências, tanto para aqueles que querem implementar e comercializar sistemas desse tipo, como para aqueles que desejam realizar uma abordagem acadêmica sobre esse assunto. Ou seja, o momento é favorável para que temas relacionados ganhem força e como isso novas pesquisas e ideias possam surgir nessa área. Tornando o tema mais popular e acessível.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GERAL**

Projetar um módulo de automação residencial utilizando o microcontrolador Arduino.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Explanar sobre o tema: Automação Residencial, abordando os problemas enfrentados e oferecendo uma solução inicialmente didática para esses problemas, solução essa que pode ser reavaliada e melhorada para projetos futuros;
- Utilizar o controlador Arduino, para a criação de um protótipo que venha a oferecer essa solução citada no parágrafo anterior;
- Programar circuitos integrados com a utilização da linguagem C ou a linguagem própria do controlador a ser utilizado, no caso o Arduino;
- Configuração, integração e adaptação dos comandos as plataformas dos *smartphones*, a plataforma abordada aqui será a do androide (sistema operacional embarcado em dispositivos móveis);



- Integração do protótipo com sistemas controláveis, por exemplo, iluminação, refrigeração, controles de acesso entre outros;

## 1.5 MÉTODO DE PESQUISA

O método utilizado será a montagem de um circuito eletrônico, com sistemas integrados ao Arduino, para isso será utilizado o Arduino mega, pois possibilitará um maior número de saídas programáveis. Estes circuitos são capazes de se encaixar em uma ampla gama de aplicações e com uma correta programação utilizando softwares próprios ou outros como c++, poderemos integrá-los com grande sucesso a outras partes do sistema eletrônico que será montado. Ou seja, será utilizado um método de coleta de informações, simulações de circuitos, montagem do protótipo, teste e por fim uma apresentação dos resultados.

- **ETAPA 1** – Pesquisa bibliográfica e análise de manuais e tutorias de programação.
- **ETAPA 2** – Coleta de dados e informações sobre componentes eletrônicos utilizados e sobre o Arduino.
- **ETAPA 3** – Adquirir o Arduino, estudá-lo, e decidir qual o melhor método de programação para o projeto através de fóruns e sites relacionados.
- **ETAPA 4** – Observar os resultados, analisar a possibilidade de melhorias e obter *know hall* para a finalização do projeto.
- **ETAPA 5** – montar o protótipo integrar com outros sistemas e avaliar os resultados.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 DOMÓTICA**

Domótica é um sistema que inclui várias funcionalidades técnicas da casa ("domus", em latim = casa), tais como iluminação, segurança, controle de iluminação, monitoramento e controle de energia. A Domótica é projetada para garantir mais conforto e melhor utilização dos recursos de segurança (tais como alarmes), automação, a gestão de energia (tais como programação de aquecimento) e de comunicação (como controles remotos) que podem ser encontrados em casa. Portanto, não é automatizar tarefas, mas para coordenar de forma que o sistema irá operar de forma mais inteligente (LATOURE, 2009).

A domótica é uma ciência jovem que surgiu em conjunto com a inteligência de ambientes e a computação ubíqua com o objetivo de estudar a interação entre o ser humano e os dispositivos eletrônicos e computacionais com os quais se relaciona. Embora tenha obtido maior popularidade nas últimas décadas, a automatização e o controle domésticos despertam o fascínio das pessoas desde que as casas começaram a ser eletrificadas muitos anos atrás. A domótica busca melhores soluções no ambiente residencial, em acordo com o panorama socioeconômico, cultural, ambiental, energético e de saúde do século XXI (BOLZANI 2013).

Com o advento de novas tecnologias, dos computadores e da Internet, o mundo moderno tem sofrido mudanças importantes em vários aspectos, principalmente os tecnológicos e sociais. Particularmente, a Habitação tem atraído forte interesse das comunidades técnicas e científicas, com foco no emprego de uma nova ciência para promover principalmente o bem-estar social, conforto e qualidade de vida: a Domótica. Ela consiste, basicamente, na automação doméstica das habitações (casa, escritório ou residência), fazendo uso da junção multidisciplinar de muitas especialidades, como eletricidade, mecânica, psicologia, telecomunicações, informática e medicina, na tradução de qualidade de vida para seus moradores e usuários, gerando conforto, segurança, lazer, comunicação e racionalização de energia. As crescentes exigências da sociedade com relação ao conforto para a habitação se somam com a possibilidade de melhorar o desempenho de questões

como a da sustentabilidade ambiental, onde a gestão de recursos naturais, principalmente a água e a energia se fazem presentes, na diminuição dos desperdícios e, conseqüentemente numa diminuição de custos financeiros. Outro fator relevante nessa integração habitação – tecnologia é a promoção da sustentabilidade social, onde a capacidade de acesso a serviços aumenta, proporcionando uma maior participação da sociedade, assim como uma maior autonomia na execução de tarefas domésticas, principalmente por parte de grupos sociais com alguma deficiência ou incapacidade, e idosos (ELOY et al., 2010).

O mercado imobiliário está começando a se adequar a essa nova realidade da Domótica, onde o edifício inteligente já aparece, otimizando a oferta dos serviços do condomínio e conjugando o mesmo com a residência. Essa inovação está cada vez mais se traduzindo em um diferencial na opção de compra de novas habitações, sem falar na valorização das atuais através de suas adaptações. Sua incorporação nos imóveis também tem contribuído para mudar desde as relações familiares até a estrutura de uma cidade. A reestruturação das habitações existentes também será essencial para a integração das novas tecnologias domóticas na procura de maior sustentabilidade, melhoria da qualidade de vida, conforto e adaptabilidade da própria habitação, além da promoção da inclusão social dos cidadãos, principalmente os idosos e pessoas com alguma incapacidade (Domingues e Pina Filho, 2012).

Atualmente, existe uma proporção pequena da população mundial que faz uso de sistemas domóticos de forma mais intensa, onde as barreiras quanto a sua aplicação ocorrem principalmente devido à falta de conhecimento e o custo da tecnologia, que ainda é considerado alto. Porém, da mesma forma como aconteceu no passado, o avanço da tecnologia e a conseqüente diminuição do preço de mercado dos novos equipamentos, devido a uma possível demanda, proporcionarão a entrada desses sistemas nos domicílios, a ponto de se tornarem itens essenciais a uma habitação num futuro não muito distante (Domingues e Pina Filho, 2012).

Nos países em desenvolvimento, como o Brasil, onde as taxas de urbanização e crescimento da população urbana são elevadas, porém com baixo padrão de vida no geral, é de se esperar que as cidades demorem por padronizar esse novo conceito de habitação. Porém, a tendência é que a sociedade brasileira opte cada vez mais por essas mudanças, para acompanhar a evolução global,

proporcionando maior inclusão social e diminuição dos impactos ambientais (Domingues e Pina Filho, 2012).

Com essas afirmações percebe-se que a automação residencial vem crescendo bastante nos últimos anos e hoje existem inúmeras soluções disponíveis no mercado para uma infinidade muito grande de públicos, os valores dos sistemas variam de acordo com a necessidade das pessoas, mas é certo dizer que esses sistemas vêm tendo os seus custos barateados e tornando-se sistemas cada vez mais acessíveis, segundo matéria jornalística publicada no site G1.com, até meados de 2013 nos últimos quatro anos a automação residencial no Brasil enfrentou um crescimento de 300%, informação dada pela Aureside, a associação brasileira do setor. Há também dados de que em 2012, o segmento de automação residencial faturou R\$ 4 bilhões e estimou-se um crescimento de mais de 30% em 2013. O mercado brasileiro até então reserva um grande potencial de expansão. Aproximadamente 1,8 milhão de residências estejam aptas a receber o sistema de automação. Os sistemas como dito anteriormente enfrentam uma popularização e barateio nos custos de implementação, alguns sistemas hoje oferecidos no mercado tem custos variando entre 10 mil a 30 mil reais (PGNE G1.com 2013). Sabe-se que existem outros com custos mais elevados e projetos semelhantes ao apresentado nesse trabalho com custos bem inferiores.

## 2.2 DOMÓTICA E ACESSIBILIDADE



**Figura 1 - Acessibilidade**

Fonte: [www.acervodocastanha.com.br](http://www.acervodocastanha.com.br)

A Domótica aplicada às habitações é uma área em franca expansão devido às potencialidades, quer a nível de conforto, quer a nível de economia de energia que advém da sua utilização. Existem, no entanto, três fatores negativos nos sistemas atuais: não são desenvolvidos com base nas limitações funcionais de pessoas idosas ou com deficiência motora (paraplégicas ou tetraplégicas), são de

custo elevado por implicarem na aquisição de produtos desenvolvidos especificamente para esse fim e baseiam-se frequentemente em soluções cabeadas centralizadas (PINHO SANTOS 2007).

A estrutura da população nos países industriais está se alterando. No futuro, existirão menos indivíduos para cuidar do número crescente de pessoas com limitações funcionais. Os custos de assistência de pessoas com limitações funcionais em centros especializados ou em habitações com assistência domiciliar são elevados e suportados em grande parte pelo Estado. Deste modo, é economicamente mais viável criar as condições adequadas para que as pessoas possam viver durante mais tempo nas suas habitações. Neste contexto, as tecnologias que permitem assistir as pessoas com limitações funcionais nas suas casas são vista como uma opção muito atrativa (PINHO SANTOS 2007).

Segundo dados da ONU, cerca de 10% da população mundial, aproximadamente 650 milhões de pessoas, vivem com uma deficiência. São a maior minoria do mundo, e cerca de 80% dessas pessoas vivem em países em desenvolvimento. Entre as pessoas mais pobres do mundo, 20% têm algum tipo de deficiência (ONUBR 2013).

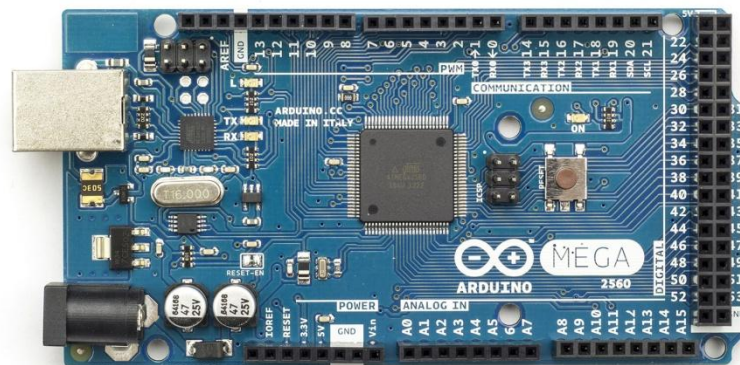
Neste contexto ainda se encontram os idosos e qualquer pessoas que necessitem de algum cuidado especial. No momento dessa pesquisa a população mundial segundo o *WordoMeters* era superior a 7,2 bilhões de pessoas e crescendo (WORDOMETERS 2014). Desse total aproximado a ONU afirma que cerca de 11,5 por cento são de pessoas com 60 anos ou mais (ONUBR 2013). No Brasil, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), cerca de 64 milhões de pessoas terão mais que 60 anos em 2050, a população total do Brasil deverá ser de aproximadamente 233 milhões de habitantes. Atualmente a população total brasileira é de algo perto de 202 milhões de habitantes (BRASIL, 2008).

Ou seja, existe um número muito grande de pessoas no Brasil e no mundo que necessitam de algum tipo de cuidado especial. É possível que o auxílio remoto promovido pelas tecnologias presentes nas residências inteligentes seja mais uma opção de tratamento e cuidado de pessoas num breve futuro, talvez uma alternativa mais adaptável à realidade das pessoas nos próximos anos e financeiramente mais atraente em comparação com os hospitais. Um retorno ao tratamento em casa, muito comum no passado (Bolzani 2013).

### 2.3 O MICROCONTROLADOR ARDUINO

Um microcontrolador é um pequeno computador com um processador e memória, que controla as funções de vários aparelhos de uso diário. Alguns microcontroladores são projetados para facilmente se conectar a um computador para programação para fins especializados. O Arduino é um exemplo de um desses microcontroladores fáceis de programar (Kimmo e Karvinen 2011).

O que facilita o uso dos microcontroladores para construção de dispositivos eletrônicos é a possibilidade de se realizar a programação de suas funções por meio de código. Microcontroladores podem controlar e interpretar formas de entrada e saída. Por exemplo, pode-se piscar um *led* ligando-o a um pino específico do Arduino, que com um o código correto pode mudar o seu estado de ligado em um segundo e desligado em outro. Também no mesmo código pode-se mudar o estado desse *led* através de uma entrada específica utilizando-se para isso de um determinado tipo de sensor ou botão. Naturalmente, a maioria dos programas fazem muitas outras funções bem mais sofisticadas. Ou seja, microcontroladores permitem em geral resolver problemas bastante complexos (Kimmo e Karvinen 2011).



**Figura 2 - Arduino Mega**  
Fonte: Arduino.cc

Segundo o site oficial, o Arduino é uma plataforma *open-source* de prototipagem eletrônica. Ela é baseada em hardware e software flexíveis, de utilização facilitada (Arduino 2014). O Arduino pode perceber o ambiente ao receber a entrada de uma variedade de sensores e pode afetar seus arredores pelo controle de luzes, motores e outros atuadores. O microcontrolador na placa é programado

usando a (linguagem de programação Arduino) baseada em *Wiring*, que segundo o seu *site* oficial: *Wiring* é um ambiente de programação de código aberto para microcontroladores (*Wiring* 2014). E o ambiente de desenvolvimento do Arduino é baseado no *Processing* que segundo o *site*: *Processing* é uma linguagem de programação, ambiente de desenvolvimento e comunidade on-line. Desde 2001, tem vindo a promover a alfabetização de Processamento de software dentro das artes visuais e alfabetização visual dentro da tecnologia. Inicialmente criado para servir como um caderno de desenho de software e para ensinar os fundamentos de programação de computadores dentro de um contexto visual, *processing* evoluiu para uma ferramenta de desenvolvimento para os profissionais. Hoje, existem dezenas de milhares de estudantes, artistas, designers, pesquisadores e amadores que utilizam *processing* para aprendizagem, protótipos, e produção (*Processing* 2014).

Projetos Arduino podem ser *stand-alone* ou podem comunicar com software rodando em um computador (por exemplo, *Flash*, *Processing*, *MaxMSP* ).

As placas podem ser construídas à mão ou comprados pré-montadas; o software pode ser baixado gratuitamente. Os designs de referência do hardware (arquivos de CAD) são disponíveis sob uma licença de código aberto, há liberdade para adaptá-los às necessidades .

O Arduino recebeu uma Menção Honrosa na seção Comunidades Digitais do *Prix Ars Electronic* 2006. Os fundadores do Arduino são: Massimo Banzi , David Cuartielles , Tom Igoe , Gianluca Martino e David Mellis (Arduino 2014).

Com isso, esse trabalho tem por objetivo a confecção de uma placa com utilização de algumas tecnologias e tem por tese a idéia da modernização e busca por conforto. Essa Idéia é baseada na “domótica que é uma área da engenharia que aplica as tecnologias de sensoriamento, controle e comunicação para um ambiente doméstico. Visando dar conforto e praticidade aos moradores” (Misuzaki 2010).

Tendo em vista esse conceito, foi optado pelo uso do Arduino “que desde o *Arduino Project*, que teve início em 2005, mais de 150.000 placas Arduino foram vendidas em todo o mundo. O número de placas-clone não oficiais sem dúvida supera o de placas oficiais, assim, é provável que mais de 500 mil placas Arduino e suas variantes tenham sido vendidas. Sua popularidade não para de crescer, e cada vez mais pessoas percebem o incrível potencial desse maravilhoso projeto de fonte

aberta para criar projetos interessantes. A maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização. Artistas, mais especificamente, parecem considerá-lo a forma perfeita de criar obras de arte interativas rapidamente. Há uma grande comunidade de pessoas utilizando Arduinos, compartilhando seus códigos e diagramas de circuito para que outros os copiem e modifiquem. A maioria dessa comunidade também está muito disposta a auxiliar outros desenvolvedores (McRoberts 2009). Por isso, a cada dia que passa a plataforma Arduino vem conquistando novos usuários. Tal sucesso é devido a sua simplicidade e ao fato de não necessitar conhecer profundamente a eletrônica e as estruturas de linguagens para criar *gadgets*, robôs ou pequenos sistemas inteligentes” (Micael Bronzatti Gaier 2011).

Mas, o quê é o Arduino? Em termos práticos, um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (ver figura 2). Por exemplo, um uso simples de um Arduino seria para acender uma luz por certo intervalo de tempo, digamos, 30 segundos, depois que um botão fosse pressionado. Nesse exemplo, o Arduino teria uma lâmpada e um botão conectados a ele. O Arduino aguardaria pacientemente até que o botão fosse pressionado; uma vez pressionado o botão, ele acenderia a lâmpada e iniciaria a contagem. Depois de contados 30 segundos, apagaria a lâmpada e aguardaria um novo apertar do botão. Pode-se utilizar essa configuração para controlar uma lâmpada em um closet, por exemplo. Esse conceito poderia ser estendido pela conexão de um sensor, como um sensor de movimento PIR, para acender a lâmpada quando ele fosse disparado. Esses são alguns exemplos simples de como se pode utilizar um Arduino (McRoberts 2009).



**Figura 3 – Processamento de sinal do Arduino**  
 Fonte [www.informaticon.com.br](http://www.informaticon.com.br)



Por causa da história e evolução do Arduino, há muitas variações sobre o que pode ser chamado de Arduino. A lista cresce mais a cada dia. A oferta oficial da equipe Arduino consiste desde o Arduino Uno, até a sua placa de maior capacidade o Arduino mega 2560 (Dale Wheate 2011). Segue a estrutura do hardware do Arduino Mega na Figura 3.

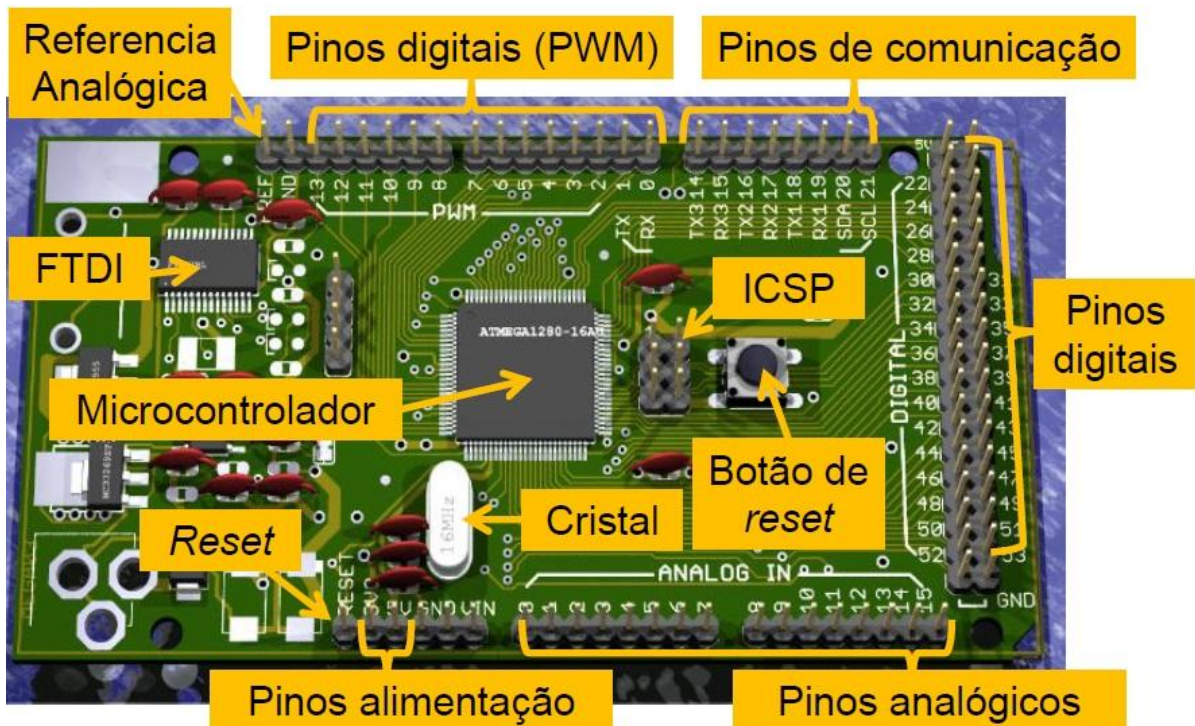


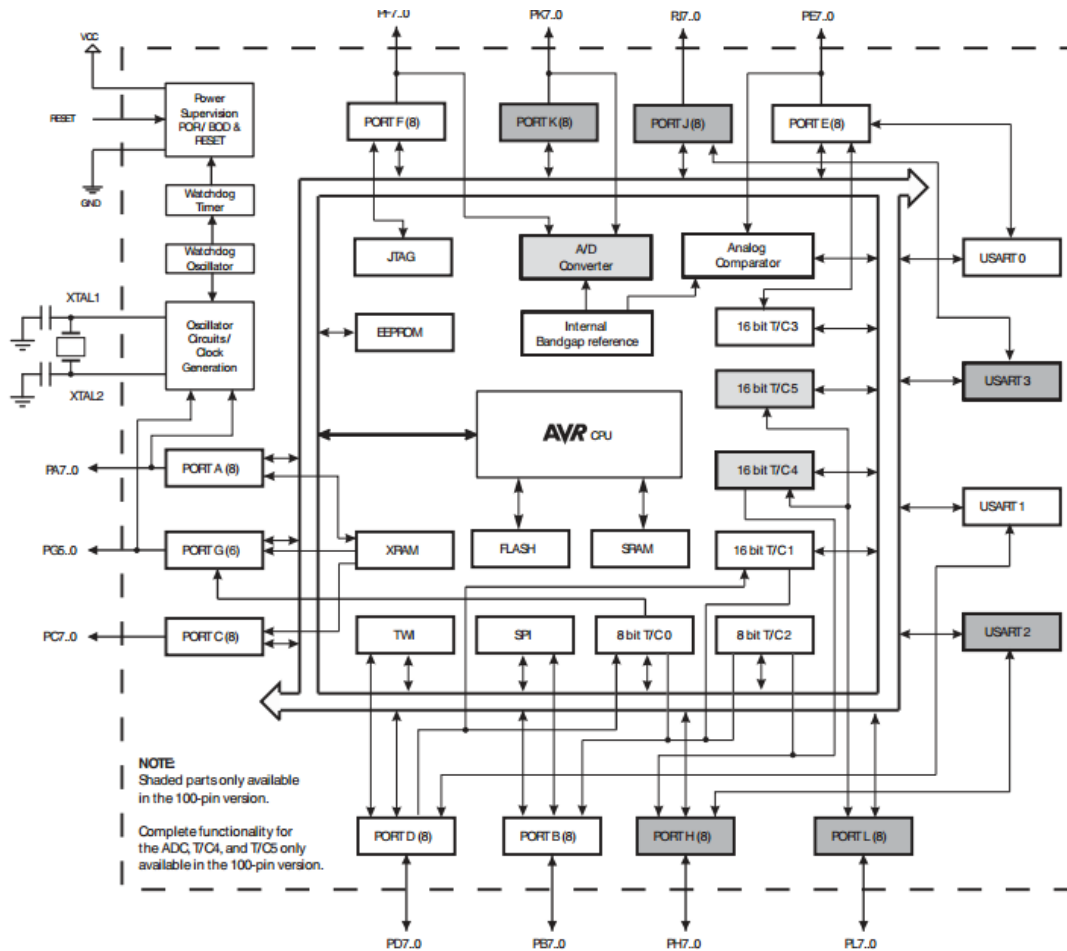
Figura 4 - Estrutura de Hardware.  
Fonte: Introdução ao arduino Pessanha Santos

No coração desse dispositivo está processador com a arquitetura AVR. Essa arquitetura foi projetada por dois estudantes em uma noite fria de inverno. Provavelmente levou mais de uma noite, mas as noites de inverno na Noruega são muito longas.

Alf-Egil Bogen e Vegard Wollan desenvolveram um CPU com um conjunto de instruções reduzindo a arquitetura (RISC), projetaram enquanto eram alunos do Instituto Norueguês de Tecnologia, em Trondheim, na Noruega. Ambos são atualmente funcionária da Atmel em Trondheim. Fabricante de inúmeros microcontroladores. A explicação oficial da Atmel sobre o significado da sigla AVR é que isso não significa nada em particular. Teorias não faltam, postulando a mais popular que ele representa é Alf RISC of Vegard, ou Risc de Alf e Vegard.

A família AVR de 8 bits da Atmel tem vários ramos. A família ATmega tem o ATmega2560, que é o CPU do Arduino Mega. Este Componente é essencialmente

um computador num chip, contendo uma unidade de processamento central (CPU), memória, matrizes, geradores de *clock*, e periféricos em um único pacote. Ver figura 4. (Dale Wheate 2011)



**Figura 5 - Avr CPU**  
**Fonte: Introdução ao Arduino Pessanha Santos**

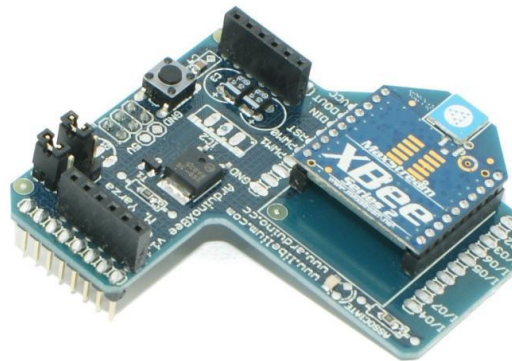
Outro detalhe importante que devemos falar sobre o Arduino é a sua capacidade de poder utilizar módulos expansores os chamados *Shields*, que são instalados nos conectores de expansão do Arduino. Os *Shields* permitem que a placa do Arduino, possa agir como um placa-mãe em miniatura de um PC, fornecendo conexões mecânicas e elétricas para circuitos adicionais. Uma ampla variedade de *Shields* estão disponíveis, oferecendo uma gama enorme de possibilidades de expansão para o Arduino (Dale Wheate 2011).

As figuras 6a e 6b demonstram dois tipos de *Shield*, uma que será utilizada nesse trabalho a *Ethernet Shield* e a *Shield Zigbee*, que pode ser utilizada como receptor de sinais de rádio de módulos *X-bee*, módulos esses bastantes utilizados

em projetos de automação pelo mundo, esses módulos são utilizados para substituição de instalações cabeadas, utilizando a transmissão de comando via rádio frequência. Lembrando que nesse trabalho será utilizada a Ethernet Shield que será detalhada a diante.



**Figura 6a - Ethernet Shield**  
 Fonte: Introdução ao Arduino Pessanha Santos



**Figura 6b - Shield Zigbee**

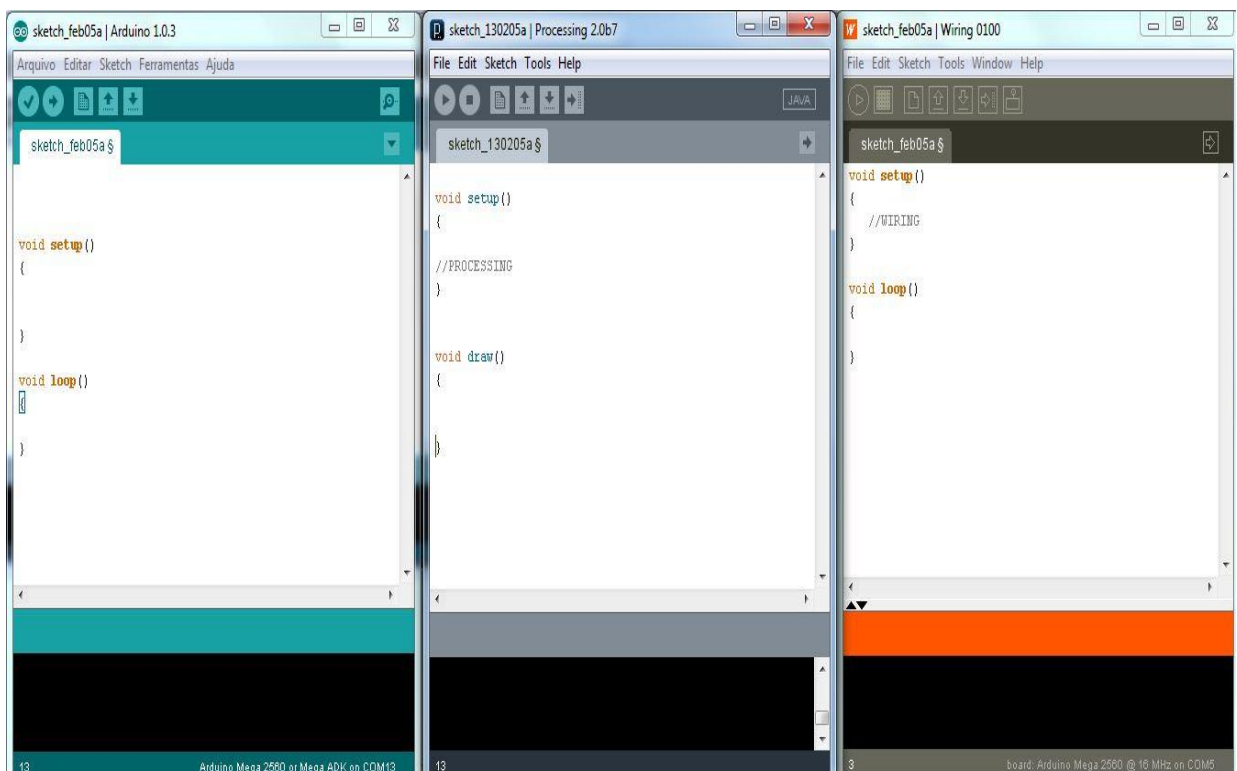
Por último e não menos importante será falada da ferramenta utilizada para a programação do Arduino e de que forma isso se dá.

Trabalhar com Arduino significa trabalhar com dois computadores de uma só vez: o microcontrolador Arduino e um PC. No mundo do desenvolvimento integrado, o PC é o computador host e o Arduino é a computador alvo. O conceito diferente é que essa comunicação ao invés de uma porta de rede acontece pela porta USB. E o Arduino, com o hardware USB adicional adequado, pode desempenhar o papel de qualquer um dispositivo USB ou um Host USB, o computador host no cenário de desenvolvimento integrado é o computador no qual o software é escrito e compilado.

O software Arduino, fornecido é projetado para funcionar na maioria dos sistemas operacionais mais populares, incluindo Microsoft Windows, Apple Mac OS, e diversas variedades de Linux.

*Software-Lots* e *Lots* de software são necessários tanto no host e nos computadores de destino. Porque as arquiteturas internas dos dois sistemas são muito diferentes uns dos outros, o processo de desenvolvimento de software de uma máquina para outra máquina diferente é muitas vezes chamado de desenvolvimento multi-plataforma, ou simplesmente desenvolvimento cruzado. Também se podem ouvir termos semelhantes, como compiladores. Ou seja, são ferramentas que ajudam a

produzir código executável para um sistema diferente daquele em que as ferramentas são executadas. (Dale Wheate 2011). Na figura 7, pode-se observar que o microcontrolador na placa do arduino é programado na linguagem Arduino em que nada mais é do que uma implementação de C++ adicionando novos comandos e métodos, sendo reconhecidos pelo *framework wiring*, e tem um ambiente de programação baseado no *processing* (Robotizando na Garagem 2013). Observa-se da esquerda para a direita os softwares do Arduino, *Processing* e *Wiring*. Nessa ordem respectiva.



**Figura 7 - Software arduino**

Fonte: [www.obotizandonagaragem.com.br](http://www.obotizandonagaragem.com.br)

Com isso, tem-se uma explanação básica do que é o Arduino, e de qual são as suas formas de trabalho e aplicação, Observa-se a grande variedade de opções que podem ser encontradas nesse projeto. Também é possível perceber que tendo o conhecimento na linguagem C, a pessoa que desejar trabalhar com esse microcontrolador não encontrará grandes dificuldades na sua programação, já que a linguagem de programação do Arduino é basicamente a linguagem C com algumas implementações, o alvo desse trabalho é a automação residencial, que envolve, como já dito anteriormente, várias áreas. O Arduino é apenas uma importante peça desse projeto, a seguir serão conhecidas outras.

## 2.4 O SMARTPHONE E O ANDROID

O *smartphone* trata-se de um aparelho que possui sistema operacional de grande capacidade de processamento e que permite se conectar a internet de todas as maneiras possíveis hoje em dia. Além disso, conta com uma loja de onde é possível baixar aplicativos (também chamados, simplesmente, de *apps*), para aumentar as funções do telefone, como jogos, gestores de correio eletrônico, navegadores, etc. (latino. msn.com 2012).

Segundo o jornal Estadão de São Paulo, em matéria divulgada na internet pelo *site* cidade verde.com e transcrita a seguir: Há alguns trimestres no mundo, as unidades vendidas de *smartphones* vêm superando as dos computadores tradicionais. Mas, em 2014, a base instalada, ou seja, o número total de dispositivos em uso dos telefones móveis com acesso à internet deve ultrapassar pela primeira vez a dos *PCs*.

A projeção foi feita pela consultoria britânica Enders, especializada no mercado de tecnologia móvel. Segundo seu estudo, a base de *PCs* conta atualmente com pouco mais de 1,5 bilhão de aparelhos em todo o mundo. Dados de uma pesquisa do *site Business Insider* apontam que o ano deve terminar com 1,4 bilhão de *smartphones* no mundo, o que dá dois aparelhos para cada nove pessoas no mundo.

A curva de crescimento dos *smartphones* ao longo dos anos é acentuada. Quatro anos atrás, 5% da população mundial tinha esse tipo de aparelho. No fim de 2014, a porcentagem deve chegar a 22%. Já os *PCs* tiveram uma queda de 8,6% no terceiro trimestre de 2013 em relação ao mesmo período no ano passado, segundo a consultoria *Gartner*.

Para muitos, o *smartphone* realiza várias tarefas que antes só podiam ser feitas no computador. Existe uma grande camada da população que não tinha acesso à internet e agora passou a ter com o *smartphone*. Segundo especialistas, no Brasil 25% dos usuários de *smartphone* usam o aparelho móvel como seu principal ponto de conexão à rede.

Há uma indicação de quem muitos consumidores vão preferir gastar seu dinheiro com um *smartphone* do que com um *PC*. Com o telefone, o usuário tem possibilidade de utilizar *Wi-Fi* grátis, navegar no ônibus, por exemplo.

No Brasil, os *smartphones* também tiveram aumento de vendas maior que os eletroportáteis (24% contra 22%) entre janeiro e agosto de 2013, segundo a consultoria GfK.

O aparecimento nas lojas de centenas de modelos a preços acessíveis, quase todos com sistema operacional Android, foi outro fator fundamental para a ascensão dos *smartphones*, em especial nos mercados em desenvolvimento.

De acordo com dados do terceiro trimestre da consultoria IDC, o sistema Android, pertencente ao Google, está presente em 81% dos aparelhos do planeta. Depois vêm o *iOS* (do *iPhone*), com 12,9%, e o *Windows Phone*, com 3,6%.

A Samsung, cujos aparelhos utilizam o sistema Android, continua liderando com folga entre as fabricantes, tendo despachado mais de 80 milhões de aparelhos de diferentes faixas de preço no terceiro trimestre deste ano.

Analistas afirmam também que com a chegada da plataforma *Firefox* no mercado, isso deve trazer ainda mais opções de aparelhos de baixo custo.

Ainda há muito mercado virgem para o *smartphone* e o *tablet*, isso se referindo a potenciais consumidores. O mercado de *PCs* já atingiu seu teto de penetração e não tem mais para onde crescer. As vendas que ocorrem hoje são principalmente de substituição. (Estadão SP. 2014).

Percebe-se, de acordo com as afirmações das pesquisas e estimativas informadas pela fonte anterior que o mercado em todo o mundo hoje propicia o uso dessa tecnologia, além de estar se tornando uma tecnologia extremamente popular e acessível, tem a vantagem de ser um dispositivo portátil, dispositivo esse que no caso do referente trabalho demonstra-se comprovadamente uma solução como painel de controle portátil para dispositivos em uma residência, já que esse dispositivo como dito anteriormente além de ser um aparelho que definitivamente está às mãos de inúmeras pessoas tem as ferramentas e tecnologias necessárias para o desenvolvimento de softwares e programas que possam estar operando em conjunto, já que como dito o *smartphone* é um aparelho de grande capacidade que pode estar operando uma série de programas ou aplicativos específicos para a sua plataforma.

### 3 MONTAGEM DO PROTÓTIPO

#### 3.1 ESPECIFICAÇÕES DO CONTROLADOR

Para a realização desse projeto como citado anteriormente foi utilizada a placa controladora Arduino, essa de fato será a central de comandos do nosso projeto. O Arduino Project oferece uma ampla variedade de placas, placas essas que podem se adequar de acordo com a necessidade do usuário. Nesse Projeto foi optado pela utilização da placa Arduino Mega 2560, o principal motivo que levou a utilização dessa placa especificamente foi o seu número de saídas e entradas programáveis, ao todo essa placa conta com 54 conexões digitais I/O programáveis, sendo que dessas 14 são do tipo *PWM*, próprias para utilização em motores e servos. Também conta com 16 entradas analógicas, entradas essas que podem ser utilizadas para a captação de variáveis ambientais. A seguir, segue a tabela com as especificações da placa.

**Tabela 1 - Especificações Arduino**

Microcontrolador	ATmega2560
Tensão de Operação	5v
Tensão de entrada (recomendada)	7-12V
Tensão de entrada (limite)	6-20V
Pinos Digitais I / O	54 (dos quais 14 oferecem saída <i>PWM</i> )
Entrada Analógica	16 pinos
Corrente DC por pino I / O	40 mA
Corrente DC para pinos de 3.3V	50 mA
Memória <i>Flash</i>	256 KB (sendo 8 KB usados para o <i>boot</i> )
SRAM	8 KB
EEPROM	4 KB
Velocidade do <i>Clock</i>	16 MHz

### 3.2 DESCRIÇÃO DAS PARTES

O projeto que será descrito mais especificamente a seguir, consiste na utilização da placa descrita anteriormente para se realizar a automação de uma residência. Nesse caso com um cunho didático e para fins de trabalho inicial, será feita a automatização de uma lâmpada de um ventilador e o controle de abertura de um sistema mecânico/eletrônico de portão. Ou seja, serão utilizadas três saídas da placa Mega 2560. Serão demonstrados os códigos para automação citada, os diagramas de ligação o código fonte para o *smartphone* e as especificações dos equipamentos utilizados.

Essa automação proposta pode de uma forma genérica e mais simplificada demonstrar o amplo potencial desse controlador para a realização de projetos desse tipo. Pois, como dito anteriormente a placa Mega 2560 é o componente da linha Arduino, mais indicado para projetos de automação, logo a um custo relativamente baixo é possível se controlar todos ou quase todos ambientes e componentes eletrônicos em uma residência. Como descrito anteriormente, são 54 portas para conexão I/O, portas essas totalmente programáveis com o uso da linguagem de programação adequada.

Os três sistemas descritos para a automação nesse trabalho são sistemas genéricos que podem ser perfeitamente imaginados e ampliados em uma residência real, ao invés do controle automático de apenas uma lâmpada, pode-se contar com o controle de todo sistema de iluminação de uma residência, o controle de todas as entradas automáticas ou mesmo controles de *on/off* de outros dispositivos, como exemplo citado do ventilador que pode ser um sistema de exaustão ou de irrigação.

É claro que ainda existem as entradas analógicas e as entradas e as saídas PWM que podem ser utilizadas para o controle de servos para automação específica, e para captação de alteração de variáveis ambientais. Por exemplo, um medidor de temperatura informa que um determinado valor *setado* foi atingido, esse valor faz com que, por exemplo, o ar condicionado ligue em uma determinada temperatura.

Existem inúmeras possibilidades a serem tentadas e criadas. Esse protótipo irá demonstrar com essas três pequenas automações as possibilidades que podem ser aproveitadas. Claro que mesmo sendo pequeno o número de componentes automatizados, já é envolvido um grau de programação relativamente complexo.



Pois, existe a necessidade de integração do código fonte em Java, código esse usado para a criação de um aplicativo no *smartphone*, com o código na linguagem Arduino, que na realidade é toda baseada na linguagem C++.

Assim, havendo o equilíbrio entre as duas programações é possível realizar uma interação completa entre os dois programas. Como consequência as possibilidades de automação tornam-se inúmeras e o usuário pode contar com uma interface simples e intuitiva ao alcance de sua mão.

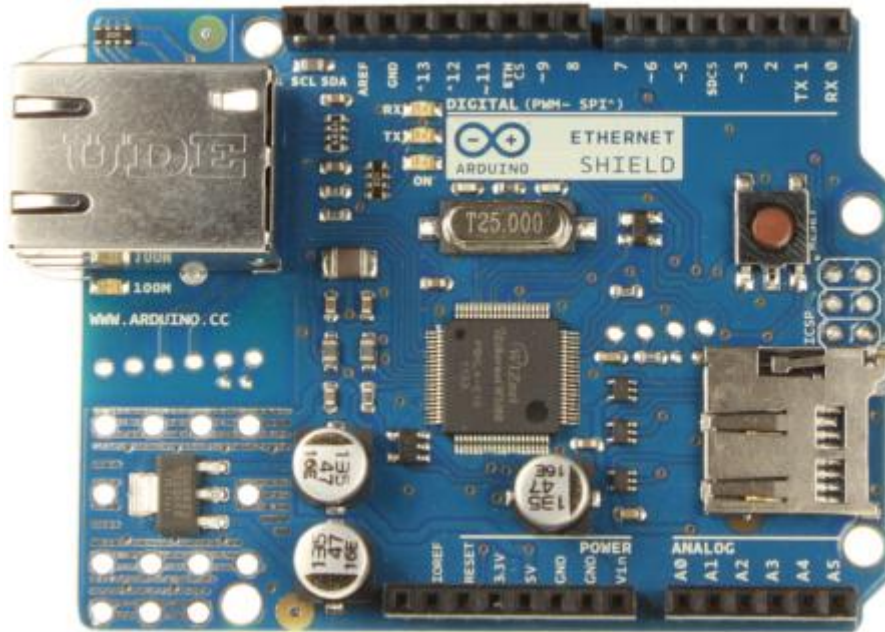
### 3.3 ESPECIFICAÇÕES DA ETHERNET SHIELD

Como principal dispositivo de conexão a rede, foi utilizado um *Ethernet shield*, o seu chip controlador é o *Wisnet W 5100*. Como visto na figura a seguir.



**Figura 1 - Chip *Ethernet Shield***  
**Fonte: datasheet *Ethernet Shield***

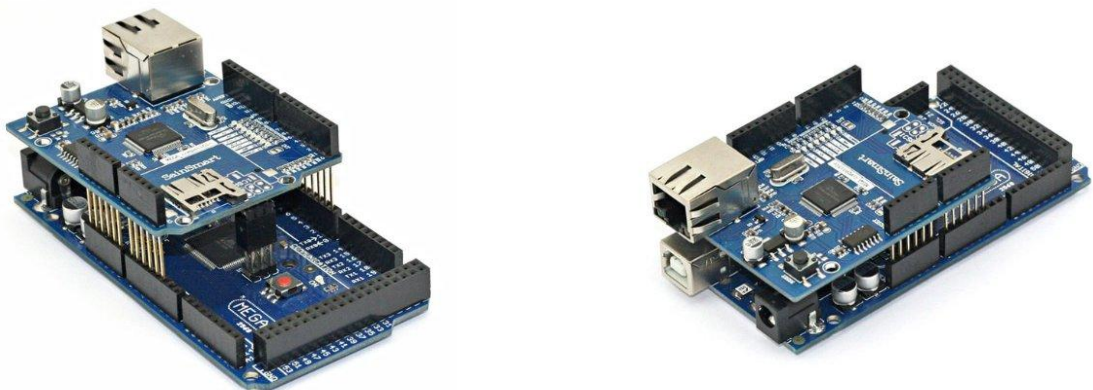
Esse chip vem embarcado em um placa com terminais prontos para serem encaixados na placa do Arduino Mega, isso facilita bastante a utilização desse componente, dispensando o uso de *protoboards* ou até mesmo torna desnecessária a fabricações de placas para a adaptação e integração dos circuitos, mais um ponto que torna o fato de trabalhar com dispositivos dessa linha bem mais prático e didático. A seguir pode-se observar a placa do *Ethernet Shield*.



**Figura 2 - Placa *Ethernet Shield***  
 Fonte: Arduino.cc

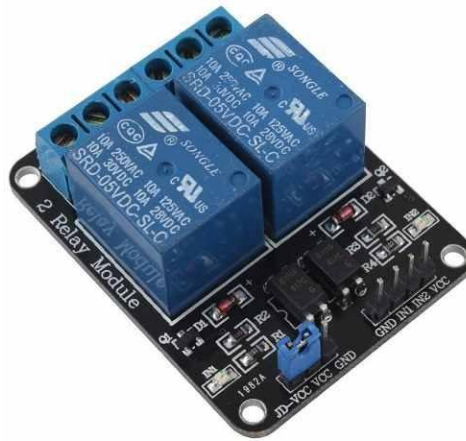
### 3.4 INTEGRAÇÃO DOS MÓDULOS DE CONTROLE E ESQUEMAS DE LIGAÇÃO

A integração das placas do Arduino Mega e do *Ethernet Shield*, se dão através dos terminais que podem ser utilizados para a conexão, inclusive, de outras *Shields* do Arduino. Mas, nesse projeto apenas será utilizada essa placa de expansão de funções. A conexão das duas placas deve ser feita da maneira demonstrada nas imagens a seguir.



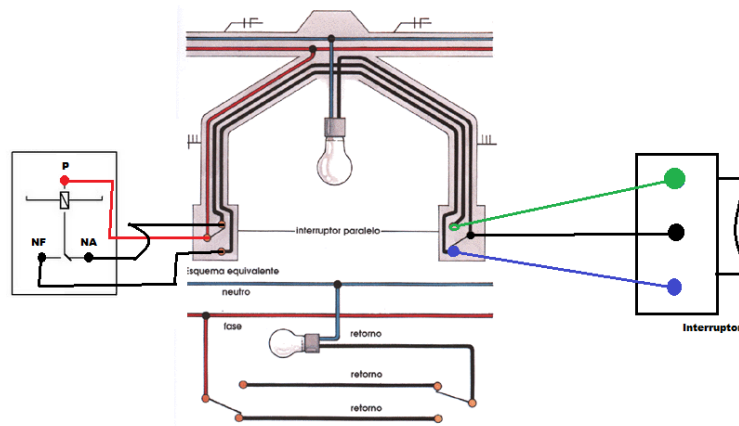
**Figura 3 - Conexão *Ethernet Shield* com Arduino**  
 Fonte: www.amazon.com

Tratando-se ainda do hardware do protótipo, foram utilizados também módulos de relês para a interligação e controle de cargas, esses módulos foram utilizados para o comando de um cooler, uma lâmpada ligada em paralelo a um interruptor e um automatizador de portões. A imagem do módulo utilizado está apresentada a seguir.



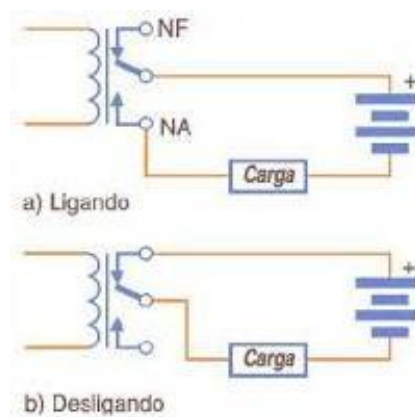
**Figura 4 - Módulo de Relê**  
Fonte: [www.mercadolivre.com.br](http://www.mercadolivre.com.br)

Este módulo de relê tem as seguintes especificações: Trabalham com bobinas de 5V, suporta tensões de 127V a 220V AC, a corrente de operação desses equipamentos máxima é de 10A, isso dá a capacidade de operarem grandes cargas com total segurança, isolando completamente o circuito principal do Arduino dos controles de potência, lembrando que a maioria dos equipamento eletroeletrônicos em uma residência trabalham dentro dessa faixa de corrente e tensão. A próxima imagem mostra o esquema de ligação de uma lâmpada sendo automatizada com a utilização de um relê.



**Figura 5 - Esquema lâmpada relê**  
 Fonte: [img3.imageshack.us](http://img3.imageshack.us)

O cooler nesse projeto foi ligado usando um esquema elétrico mais simplificado, pois o relê o aciona diretamente sem que haja interruptores em paralelo como na figura anterior. A demonstração do esquema é a próxima figura.



**Figura 6 - Ligação Cooler**  
 Fonte: [www.mecatronicaatual.com.br](http://www.mecatronicaatual.com.br)

Uma placa para automatização de um portão também foi ligada, a intenção é substituir com isso a necessidade de utilização de controles remotos para a abertura de portões, controles esses que geram grandes incômodos para os usuários, além da necessidade de trocas de baterias constantes, qualquer tombo pode tirá-los da sua frequência de trabalho. Sem contar que os controles para portões eletrônicos são como chaves, a o risco de perdê-los e qualquer pessoa que os encontrar ter acesso à residência do morador, diminuindo assim a sua segurança.

O automatizador utilizado nesse projeto é um dos mais populares hoje no mercado, um automatizador da PPA. A placa que será interligada segue com a sua imagem a seguir.



**Figura 7 - Central Portão Automático**  
Fonte: [tecgates.com.br](http://tecgates.com.br)

Podem-se observar nessa figura alguns conectores para chicotes elétricos, os brancos de pinagem macho. O conector que está próximo ao fusível é o que será utilizado nesse projeto, esse conector tem uma entrada pulso para comandos externos, entrada essa devidamente programada no programa de controle do Arduino.

### **3.5 ORÇAMENTO**

A parte de hardware do protótipo compreende vários componentes, esses componentes foram orçados e comprados, os valores estarão descritos na tabela a seguir. Um detalhe deve ser mencionado, o projeto com a utilização do Arduino, tende a ser um projeto de baixo custo, sendo uma alternativa viável a projetos já existentes, alguns componentes descritos nesse projeto podem representar um valor

elevado inicialmente, mas deve ser feita a seguinte observação, componentes como o celular, o automatizador de portão e o roteador a principio deveriam existir em uma residência convencional, em suma apenas o Arduino, a *shield* e o relê de fato são componentes que devem ser adquiridos para se iniciar a automatização da residência. É claro que em projetos futuros outros equipamentos de transmissões e comandos podem ser integrados, isso deve ser tratado em um próximo capítulo.

**Tabela 2 - Orçamento**

COMPONENTES UTILIZADOS EM PROJETO DE AUTOMAÇÃO	
ARDUINO MEGA 2560	R\$ 99,00
<i>ETHERNET SHIELD</i>	R\$ 45,00
2 RELÊS DE 2 CANAIS DE 5V	R\$ 50,00
SMARTPHONE SAMSUNG GALAXY Y	R\$ 450,00
ROTEADOR WIRELLES D-LINK	R\$ 90,00
KIT AUTOMATIZADOR PPA EUROS CUSTON	R\$ 550,00
LÂMPADA INCANDESCENTE	R\$ 2,00
INTERRUPTOR PARALELO	R\$ 5,00

### 3.6 PROGRAMAÇÃO

Após a reunião de todos os materiais citados acima, foi realizada uma pesquisa para a programação e criação do programa.

Como citado anteriormente o software para a programação do Arduino foi um software próprio de programação, com a sua base de programação em C++. O programa deverá ser anexado nesse trabalho aqui será vista apenas alguns exemplos do código.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // BYTE MAC PARA REDE
byte ip[] = { 192, 168, 0, 150 }; // NÚMERO DO IP QUE O ARDUINO ASSUMIRA NA REDE DENTRO DA FAIXA DE IP
byte gateway[] = { 192, 168, 0, 1 }; // FAIXA DE IP DO ROTEADOR
byte subnet[] = { 255, 255, 255, 0 }; //SUBNETMASK DA REDE
EthernetServer server(80); //PORTA DE SERVIÇO HTTP UTILIZADA
byte sampledata=50;
```

**Quadro 1 - Recorte do Programa**

```
if(readString.indexOf("L=1") >0){//LAÇO QUE DETERMINARA COMO A SAÍDA DO 5 DA PLACA ARDUINO DEVE SE COMPORTAR
  digitalWrite(ledPin, HIGH);
  LEDON = true;
  delay(1000);
  digitalWrite(ledPin,LOW);
  LEDON = false;
```

**Quadro 2 - Recorte determinação de saída**

Além da programação realizada no Arduino também foi feita uma programação para o Androide, essa programação foi realizada através do programa eclipse. Esse código foi escrito em Java, de fato foi necessário auxílio externo para criação desse programa, o aplicativo e o código fonte serão anexados a esse trabalho.



**Figura 8 - Programa Eclipse**  
**Fonte: Eclipse Kepler**

Por fim, toda a programação foi testada, instalada no Arduino, foi criado um aplicativo para o *smartphone*, os componentes de hardware do projeto foram integrados e colocados em funcionamento. Esse projeto deve ser demonstrado e apresentado em bancada, o funcionamento físico deve ser explicado na prática. Será mostrado o comando remoto de componentes através do *smartphone*, assim como o funcionamento de um automatizador, de uma lâmpada e um *cooler*. Todos em conjunto com esse sistema. Demonstrando, com isso, na prática tudo que foi falado nesse trabalho.

## 4 CONCLUSÃO

Este trabalho tratou de um assunto que está ganhando força na sociedade em seus dias atuais, a domótica, é uma nova ciência que surgiu com a integração de várias áreas do conhecimento. Os assuntos tratados abrangem principalmente a criação de um projeto viável. Projeto esse que deve ser tratado como uma possibilidade de inovação, por isso, o uso do Arduino.

As motivações para a realização desse trabalho são inúmeras, mas as principais foram: Criação de um projeto de automação residencial de baixo custo, sugestão para o aprimoramento para um projeto de acessibilidade futuro, um projeto de automação residencial que venha a se tornar viável para a comercialização, e trazer para o meio acadêmico um tema tão pouco tratado, em contrapartida da sua crescente popularização.

O tema ainda carece de vários estudos mais específicos, porém a ideia de que projetos como este são possíveis e podem ser trazidos para a nossa realidade foi bem tratada aqui. Afinal, também foram apresentados os meios e a construção de um protótipo. Utilizando tecnologias didáticas e comerciais. Tecnologias essas que estão barateando e tornando-se cada vez mais acessíveis.

Uma casa em breve, com certeza contara com sistemas de automação ou, utilizando uma palavra mais específica ao tema, sistemas domóticos. A domótica tem se tornado algo de grande interesse para as novas construções e um importante meio para isso são os estudos relacionados à área. As tendências apontam para que no futuro as residências sejam inteligentes, cômodas aos seus moradores, econômicas do ponto de vista energético e ecologicamente corretas. A domótica vem como uma nova ciência pronta a encarar todos esses desafios, trazer conforto e acessibilidade aos moradores em casa, prédios, comercio, etc.

A utilização do Arduino como motor principal de automação, se dá devido a sua grande e crescente popularização. A existência de um leque gigantesco de comunidades e sites relacionados ao tema facilita bastante a forma de se trabalhar com esse controlador. Pois, hoje se pode encontrar em diversas fontes, livros, manuais, fóruns, dentre outros; entusiastas que querem dividir e falar sobre as suas descobertas e contar as suas experiências. Isso tanto na área de automação residencial quanto na criação de robôs, e outros projetos eletrônicos relacionados.



As vantagens da utilização desse componente apenas por isso já se tornam evidentes. Outros controladores além de não contarem com comunidades tão abrangentes, também não tem o número de acessórios que o Arduino oferece, periodicamente são desenvolvidas e lançadas novas *Shields*, que vêm agregar e trazer mais recursos ao controlador. Além das inúmeras atualizações do próprio controlador que se tornam disponíveis periodicamente.

A ideia de construção de um protótipo inicialmente se pareceu ousada, mas ao final acabou demonstrando-se além de viável, bastante prática e assertiva. Pela escolha do tipo de controlador. O protótipo passou pela fase de aquisição de peças, estudo sobre elas, programação e teste. Talvez, a parte mais complexa tenha sido a programação. O auxílio das comunidades ajudou bastante na elaboração dos códigos. A programação em Java, matéria não é contemplada pelas disciplinas cursadas envolveu auxílio externo de outro programador da área, para a concretização do projeto. Ao final, o protótipo foi criado e constatado a sua grande funcionalidade da forma como já está. Também fora percebido já de início o seu potencial de expansão para controle de outros componentes em uma casa.

Para o futuro, há uma grande possibilidade de que trabalho como esse e talvez outros com componentes diferentes possam vir a serem desenvolvidos para o mercado. Projetos acadêmicos devem trazer à luz novas ideias e meios para se alcançar esse objetivo. A possibilidade de aumento e incremento de ideias e alternativas é vasta. Em especial, devem ser levados em conta a possibilidade de trabalhos relacionados área de acessibilidade, hoje há a existência de um grande público para esse tipo de projeto e a grande maioria dele não pode pagar pelos elevados custos que podem advir de uma possível adaptação e automação de suas casas.

Em anexos com esse trabalho estarão os *data sheets* dos componentes utilizados, diagramas dos controladores e os códigos de operação, tanto do Arduino quanto do Android.

Dessa forma, essa proposta é apresentada como uma ideia pouco praticada, mas que vêm ganhando força. A apresentação de um protótipo torna a ideia explanada aqui mais tátil e acessível. Colocando isso na nossa realidade, mostrando que é algo praticável e possível de se ver nas casas em pouco tempo, com o trabalho, empenho e novas ideias.

## 4.1 PROPOSTA PARA TRABALHOS FUTUROS

Como mencionado anteriormente nesse trabalho, há uma gama muito grande para aplicações de sistemas de automação residencial que vão desde conforto à projetos de acessibilidade.

Esse trabalho abrange várias áreas de estudo, áreas que inclusive extrapolam a área relacionada com a engenharia elétrica que foi o objeto deste estudo. Alcança áreas tanto da engenharia civil, informática, programação, análise de sistemas, áreas da saúde, acessibilidade e bem estar social.

Os temas como domótica, automação residencial com Arduino, acessibilidade deixam brechas para explorações maiores em outros trabalhos. Imagina-se um trabalho de automação residencial voltados exclusivamente para área de acessibilidade. Sendo realizados estudos de caso, construção de protótipos voltados exclusivamente para esse público, que como dito, é tão abrangente. Ou mesmo utilizando o início desse trabalho para uma automatização maior de uma casa. Foi falado que o Arduino oferece cerca de 54 possibilidade de interligação em uma casa isso possibilitaria a automação quase que completa em uma residência de médio porte, as facilidades geradas com um sistema como esse, por exemplo, para uma pessoa que não tem os movimentos das pernas ou um idoso seriam imensas.

Outra proposta interessante a ser abordada é a de criação de um plano de negócios. Ou seja, um caminho para criação de uma empresa de automação residencial. Lembrando que esse tipo de empresa ainda é tão escasso em nossa sociedade.

Por fim, verifica-se a possibilidade de criação de novos trabalhos que abrangem alunos de vários cursos, engenharias, informática, saúde e projetos de desenvolvimento e aprimoramento da vida das pessoas, desde pessoas comuns a idosos e portadores de necessidades especiais.

## REFERÊNCIAS

BOYLESTAD Robert L., Louis Nashelsky, **DISPOSITIVOS ELETRÔNICOS E TEORIA DE CIRCUITOS**, 8ª EDIÇÃO EDITORA PEARSON PRENTICE HALL. 2003

MAKE: **ARDUINO BOTS AND GADGETS**, KIMMO E TERO KARVINEN, O'REILLY MEDIA INC. 2011.

[HTTP://ARDUINO.CC/](http://ARDUINO.CC/) ACESSADO DIA 07 DE ABRIL DE 2014 ÀS 21h40min.

[HTTP://WWW.PROCESSING.ORG/](http://WWW.PROCESSING.ORG/) ACESSADO NO DIA 07 DE ABRIL DE 2014 ÀS 22h35min.

[HTTP://WIRING.ORG.CO/](http://WIRING.ORG.CO/) ACESSADO DIA 07 DE ABRIL DE 2014 ÀS 22h18min.

MIZUSAKI, LUCAS EISHI PIMENTEL, **COMPARAÇÃO DE MECÂNIISMOS DE COMUNICAÇÃO PARA A CASA INTELIGENTE**, UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, REPOSITÓRIO DIGITAL, [HTTP://HDL.HANDLE.NET/10183/25607](http://HDL.HANDLE.NET/10183/25607).

MCROBERTS, MIKE. **ARDUINO STARTER KIT MANUAL**, FIRST EDITION, EARTHSHINE DESIGN 2009.

MICHAEL BRONZATTI GAIER, **APRENDENDO A PROGRAMAR EM ARDUINO**, PET AUTO NET 2011.

DALE WHEAT, **ARDUINO INTERNAL**, SPRINGER SCIENCE+BUSINESS MEDIA LLC, 2011.

[HTTP://ROBOTIZANDONAGARAGEM.COM.BR/NOCAO-DE-PROGRAMAÇÃO/](http://ROBOTIZANDONAGARAGEM.COM.BR/NOCAO-DE-PROGRAMAÇÃO/) ACESSADO EM 12 DE ABRIL DE 2014 ÀS 11h17min.

[HTTP://TECNOLOGIA.BR.MSN.COM/ESPECIAIS/SMARTPHONES/ARTIGO.ASPX?CP-DOCUMENTID+32235572](http://TECNOLOGIA.BR.MSN.COM/ESPECIAIS/SMARTPHONES/ARTIGO.ASPX?CP-DOCUMENTID+32235572) ACESSADO EM 10 DE DEZEMBRO DE 2013 ÀS 21h00min.

[HTTP://WWW.CIDADEVERDE.COM/EM-2014-NAUMERO-DE-SMARTPHONES-DEVE-SUPERAR-O-DE-COMPUTADORES-151126](http://WWW.CIDADEVERDE.COM/EM-2014-NAUMERO-DE-SMARTPHONES-DEVE-SUPERAR-O-DE-COMPUTADORES-151126) EM 23/04/14/ÀS 23:00 FONTE DO SITE ESTADÃO.

[HTTP://DOMOTIQUE-NEWS.COM/2009/02/06/QUEST-CEQUE-LA-DOMOTIQUE/LATOUR](http://DOMOTIQUE-NEWS.COM/2009/02/06/QUEST-CEQUE-LA-DOMOTIQUE/LATOUR) 2009 ACESSADO EM 21/04/14 às 22h20min.

BOLZANI, CAIO. REVISTA FONTE Nº13, PUBLICADA POR PRODEMGE – GOVERNO DO ESTADO DE MINAS GERAIS, ARTIGO **DOMÓTICA A NOVA CIÊNCIA DO SÉCULO XXI**, PAG. 105. DEZEMBRO DE 2013.

ELOY, S.; PLÁCIDO, I.; DUARTE, J. P.; NUNES, R. Utilização de Domótica na Estratégia de Sustentabilidade Social e Ambiental. **1º Congresso Internacional de Habitação no Espaço Lusófono, ISCTE** - Instituto Universitário de Lisboa, Lisboa, Sept 2010.(retirado de artigo de Ricardo Gil Domingues e Armando Carlos de Pina Filho do 2º simpósio de pós graduação em engenharia urbana. (SIMPEGEU) novembro de 2012.

DOMINGUES, RICARDO GIL E PINA FILHO, ARMANDO CARLOS DE. II SIMPEGEU, TEMA: **A DOMÓTICA COMO TENDÊNCIA NA HABITAÇÃO**, 2012.

HTTTP://GLOBO.COM/ECONOMIA/PME/NOTICIA/2013/EMPRESA-FATURA-R-1-MILHAO-COM-SERVICOS-DE-AUTOMACAO-RESIDENCIAL.HTML **PEQUENAS EMPRESAS GRANDES NEGÓCIOS**, ACESADO EM 21/04/14 ÀS 23h00min.

**COMUNICAÇÕES EM APLICAÇÕES DE DOMÓTICA PARA APOIO A PESSOAS COM LIMITAÇÃO FUNCIONAL**, UNIVERSIDADE DE AVEIRO DEP. DE ELETROTÉCNICA, TELECOMUNICAÇÃO E INFORMÁTICA. PINHO SANTOS, VASCO MIGUEL TEÓTONEO BRITES DE. 2007.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS, **AS PESSOAS COM DEFICEIÊNCIA NO MUNDO**, RELATÓRIO 2013. <http://www.onu.org.br/a-onu-em-acao/a-onu-e-as-pessoas-com-deficiencia/> acessado em 23/04/14 às 00h00minhs.

<http://www.worldometers.info/br/> população mundial acessado em 22/04/14 às 22h00min. WORDOMETERS 2014.

(BRASIL 2008) **IBGE** (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA) 2008. [http://www.ibge.gov.br/home/estatistica/populacao/projecao\\_da\\_populacao/2008/projecao.pdf](http://www.ibge.gov.br/home/estatistica/populacao/projecao_da_populacao/2008/projecao.pdf) acessado em 23/04/14 às 23h50minhs.

## REFERÊNCIA DE FIGURAS

- (FIGURA 2) [http://arduino.cc/en/uploads/Main/ArduinoMega2560\\_R3\\_Fronte.jpg](http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Fronte.jpg)  
arduino mega
- (FIGURA 3) Videoaulas Neri - [www.informaticon.com.br](http://www.informaticon.com.br) Prof Neri Aldoir Neitzke  
Curso de Arduino. Slide 5.
- (FIGURA 4) Introdução ao Arduino Pessanha Santos ASPOF EN-AEL.
- (FIGURA 5) Introdução ao Arduino Pessanha Santos ASPOF EN-AEL.
- (FIGURA 6a) Introdução ao Arduino Pessanha Santos ASPOF EN-AEL.
- (FIGURA 6b) <http://robotizandonagaragem.com.br/nocao-de-programacao/> em  
12/04/14 às 11h17min.
- (FIGURA 7) <http://robotizandonagaragem.com.br/nocao-de-programacao/> 12/04/14  
às 20h00min.
- (FIGURA 1) [http://www.acervodocastanha.com.br/acessibilidade-em-obras-publicas-  
e-tema-de-curso/](http://www.acervodocastanha.com.br/acessibilidade-em-obras-publicas-e-tema-de-curso/) acesso em 29/04/14 às 20h00min.
- (FIGURA 9) figura chip retirada do *datasheet Ethernet Shield*.
- (FIGURA 10) <http://arduino.cc/en/Main/ArduinoEthernetShield> 29/04/14 às 21h10min.
- (FIGURA 11) [http://www.amazon.com/SainSmart-ATmega2560-Ethernet-Mega2560-  
Duemilanove/dp/B0076H2TPE](http://www.amazon.com/SainSmart-ATmega2560-Ethernet-Mega2560-Duemilanove/dp/B0076H2TPE) 29/04/14 às 20h30min.
- (FIGURA 12) [http://produto.mercadolivre.com.br/MLB-551469686-modulo-rel-  
arduino-2-canais-arduino-](http://produto.mercadolivre.com.br/MLB-551469686-modulo-rel-arduino-2-canais-arduino-) JM às 20h54min dia 1/05/14.
- (FIGURA 13) <http://img3.imageshack.us/img3/1039/esquemau.png> 01?
- (FIGURA 14) [www.mecatronicaatual.com.br/imagem ligação relê](http://www.mecatronicaatual.com.br/imagem_ligacao_relê) dia 1/5/14 às  
21h45min.
- (FIGURA 15)  
[http://tecgates.com.br/Eshop.Admin/imagens/tecgates/Thumbs/TN500\\_central-top-  
ligth.jpg](http://tecgates.com.br/Eshop.Admin/imagens/tecgates/Thumbs/TN500_central-top-ligth.jpg) 1/5/14 às 22h:00min.
- (FIGURA 16) Imagem abertura programada Eclipse Kepler.

## APENDICÊ A: Código Programado no Arduino.

```

/*CÓDIGO PROJETO DE AUTOMAÇÃO RESIDENCIAL SIMPLES*/

//PRÓXIMOS COMANDOS INCLUIRÃO AS BIBLIOTÉCAS PARA
CONEXÃO EM REDE DO ARDUINO COM A UTILIZAÇÃO DO ARDUINO
#include <SPI.h>
#include <Client.h>
#include <Ethernet.h>
#include <Server.h>
#include <Udp.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // BYTE MAC PARA
REDE
byte ip[] = { 192, 168, 0, 150 }; // NÚMERO DO IP QUE O ARDUINO
ASSUMIRA NA REDE DENTRO DA FAIXA DE IP
byte gateway = { 192, 168, 0, 1 }; // FAIXA DE IP DO ROTEADOR
byte subnet = { 255, 255, 255, 0 }; //SUBNETMASK DA REDE
EthernetServer server(80); //PORTA DE SERVIÇO
HTTP UTILIZADA
byte sampledata=50;

int ledPin = 5; // DECLARAÇÃO DE VARIÁVEIS INTEIRAS, NO CASO
ESTÃO SENDO DETERMINADAS AS SAÍDAS 5, 6 E 7
int ledPin2=6;
int ledPin3=7;

String readString = String(30); //MATRIZ DE COMPARAÇÃO

boolean LEDON = false; // VARIÁVEIS BOOLEANAS PARA
COMPARAÇÃO DE ON OU OFF
boolean LED2ON=false;
boolean LED3ON=false;

String LAMP1;
String LAMP2;
String LAMP3;

void setup(){ // NENHUMA FUNÇÃO DEVE RETORNAR AO PROGRAMA
PRINCIPAL

Ethernet.begin(mac, ip, gateway, subnet);//COMANDO PARA ENVIAR
INFORMAÇÕES PELA REDE

pinMode(ledPin, OUTPUT); //DETERMINAÇÃO DE CONFIGURAÇÃO DE
PINO ESPECIFICADO
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);

```

```
Serial.begin(9600); //VELOCIDADE DA TAXA DE TRANSFERÊNCIA
SERIAL DAS INFORMAÇÕES
}
void loop(){ //INICIA-SE A REPETIÇÃO CONTINUA DO PROGRAMA
AGUARDANDO SEMPRE ALGUM COMANDO

    EthernetClient client = server.available(); //BIBLIOTÉCA EXTERNA DO
ARDUINO, DECLARADA, BIBLIOTÉCA DE INTERCOMUNICAÇÃO COM
PROGRAMA EM JAVA DO ANDROIDE
    if (client) { // LAÇO SE INDICADO INICIO DO LOOP SE ALGUMAS
VARIÁVEIS FOREM VERDADEIRAS
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();

                if (readString.length() < 100)
                {
                    readString += c;
                }
                Serial.print(c);
                if (c == '\n') {
                    if (readString.indexOf("?") < 0)
                    {
                        }
                    else
                    if(readString.indexOf("L=1") > 0){ //LAÇO QUE DETERMINARA COMO
A SAÍDA DO 5 DA PLACA ARDUINO DEVE SE COMPORTAR
                        digitalWrite(ledPin, HIGH);
                        LEDON = true;
                        delay(1000);
                        digitalWrite(ledPin, LOW);
                        LEDON = false;
                    }
                    if(readString.indexOf("M=1") > 0){ // LAÇO QUE DETERMINA A
SAÍDA 6
                        digitalWrite(ledPin2, HIGH);
                        LED2ON = true;
                    }
                    if(readString.indexOf("N=1") > 0){ // LAÇO QUE DETERMINA A
SAÍDA 7
                        digitalWrite(ledPin3, HIGH);
                        LED3ON = true;
                    }
                    if(readString.indexOf("L=0") > 0){ // O QUE SE SEGUE NO
PROGRAMA SÃO OS FECHAMENTOS DOS LAÇOS DETERMINANDO AS
SAÍDAS E REFERÊNCIA DOS LED PARA AS VARIÉVEIS LAMP.
                        digitalWrite(ledPin, HIGH);
                        LEDON = true;
                        delay(1000);
                        digitalWrite(ledPin, LOW);
```

```

    LEDON = false;
  }
  if(readString.indexOf("M=0") >0){
    digitalWrite(ledPin2, LOW);
    LED2ON = false;
  }
  if(readString.indexOf("N=0") >0){
    digitalWrite(ledPin3, LOW);
    LED3ON = false;
  }
  if (LEDON == true){
    LAMP1 = "AC1";
  }else{
    if (LEDON == false){
      LAMP1 = "AP1";
    }
  }
  if (LED2ON == true){
    LAMP2 = ",AC2,";
  }else{
    if (LED2ON == false){
      LAMP2 = ",AP2,";
    }
  }
  if (LED3ON == true){
    LAMP3 = "AC3";
  }else{
    if (LED3ON == false){
      LAMP3 = "AP3";
    }
  }
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println();

  client.println(LAMP1);
  client.println(LAMP2);
  client.println(LAMP3);

  readString="";
  client.stop();
}
}
}
}
}

```



## APENDICÊ B: Código do aplicativo do *Smartphone*, programado pelo Eclipse.

```
//ConectHttpClient.java
```

```
public static String executaHttpGet(String url) throws Exception {
    BufferedReader bufferedReader = null;
    try {
        HttpClient client = getHttpClient();
        HttpGet httpGet = new HttpGet(url);
        httpGet.setURI(new URI(url));
        HttpResponse httpResponse = client.execute(httpGet);
        bufferedReader = new BufferedReader(new
InputStreamReader(httpResponse.getEntity().getContent()));
        StringBuffer stringBuffer = new StringBuffer("");
        String line = "";
        String LS = System.getProperty("line.separator"); // \s
        while ((line = bufferedReader.readLine()) != null) {
            stringBuffer.append(line + LS);
        }
        bufferedReader.close();

        String resultado = stringBuffer.toString();
        return resultado;
    } finally {
        if (bufferedReader != null) {
            try {
                bufferedReader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
}
```

```
//HttpClient
```

```
package br.com.unipac;
```

```
public class httpclient {
```

```
}
```

```
//MaqActivity
```

```
package br.com.unipac;
```

```

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.TextView;

public class maqActivity extends Activity implements OnClickListener {
    /** Called when the activity is first created. */

    ImageButton btOnA, btOffA, btOnB, btOffB, btOnC, btOffC;
    boolean status = true;
    TextView tvInternet, tvStatusA, tvStatusB, tvStatusC;
    Handler mHandler;
    private static String CATEGORIA = "result";
    String L, N, M;
    String hostIp = null;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Intent intent = getIntent();

        Bundle params = intent.getExtras();

        if(params != null){
            String mostraTexto = params.getString("mensagem");
            hostIp = mostraTexto.trim();
            setContentView(R.layout.tela);
        }

        mHandler = new Handler();
        mHandler.post(mUpdate);

        btOnA = (ImageButton) findViewById(R.id.btOnA);
        btOffA = (ImageButton) findViewById(R.id.btOffA);
        btOnB = (ImageButton) findViewById(R.id.btOnB);
        btOffB = (ImageButton) findViewById(R.id.btOffB);
        btOnC = (ImageButton) findViewById(R.id.btOnC);
        btOffC = (ImageButton) findViewById(R.id.btOffC);
    }
}

```

```

tvStatusA = (TextView) findViewById(R.id.tvStatusA);
tvStatusB = (TextView) findViewById(R.id.tvStatusB);
tvStatusC = (TextView) findViewById(R.id.tvStatusC);
tvInternet = (TextView) findViewById(R.id.tvInternet);

btOnA.setOnClickListener(this);
btOffA.setOnClickListener(this);
btOnB.setOnClickListener(this);
btOffB.setOnClickListener(this);
btOnC.setOnClickListener(this);
btOffC.setOnClickListener(this);
}
private Runnable mUpdate = new Runnable() {
    public void run() {
        checkNetworkStatus();
        arduinoStatus("http://" + hostIp + "/");
        mHandler.postDelayed(this, 500);
    }
};
public void onClick(View bt) {

    String url = null;
    if(bt == btOnA){
        url = "http://" + hostIp + "?L=1";
    }
    if(bt == btOffA){
        url = "http://" + hostIp + "?L=0";
    }
    if(bt == btOnB){
        url = "http://" + hostIp + "?M=1";
    }
    if (bt == btOffB){
        url = "http://" + hostIp + "?M=0";
    }
    if(bt == btOnC){
        url = "http://" + hostIp + "?N=1";
    }
    if(bt == btOffC){
        url = "http://" + hostIp + "?N=0";
    }
    String urlGetHost = url;
    String respostaRetornada = null;

    try{
        respostaRetornada =
ConectHttpClient.executaHttpGet(urlGetHost);
        String resposta = respostaRetornada.toString();
        resposta = resposta.replaceAll("\\s+", "");
    }
    catch(Exception erro){

```

```

        //Toast.makeText(maqActivity.this, "Erro.: "+erro,
Toast.LENGTH_LONG);
    }
}
public void arduinoStatus(String urlArduino){

    String urlHost = urlArduino;
    String respostaRetornada = null;

    try{
        respostaRetornada = ConectHttpClient.executaHttpGet(urlHost);
        String resposta = respostaRetornada.toString();
        resposta = resposta.replaceAll("\\s+", "");
        Log.i(CATEGORIA, resposta);

        String[] b = resposta.split(",");

        if(b[0].equals("AC1")){
            tvStatusA.setText("ON");
        }
        else{
            if(b[0].equals("AP1")){
                tvStatusA.setText("OFF");
            }
        }
        }
        if(b[1].equals("AC2")){
            tvStatusB.setText("ON");
        }
        else{
            if(b[1].equals("AP2")){
                tvStatusB.setText("OFF");
            }
        }
        }
        if(b[2].equals("AC3")){
            tvStatusC.setText("ON");
        }
        else{
            if(b[2].equals("AP3")){
                tvStatusC.setText("OFF");
            }
        }
        }
        Log.i(CATEGORIA, b[0]);
        Log.i(CATEGORIA, b[1]);
        Log.i(CATEGORIA, b[2]);
    }
    catch(Exception erro){

```

```

        //Toast.makeText(maqActivity.this, "Erro: "+erro,
        Toast.LENGTH_LONG);
    }
}

public void checkNetworkStatus(){

    final ConnectivityManager connMgr = (ConnectivityManager)
    this.getSystemService(Context.CONNECTIVITY_SERVICE);

    final android.net.NetworkInfo wifi =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

    final android.net.NetworkInfo mobile =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);

    if( wifi.isAvailable() ){

        tvInternet.setText("WIFI ONLINE");
    }
    else if( mobile.isAvailable() ){

        tvInternet.setText("3G ONLINE");
    }
    else
    {
        tvInternet.setText("OFFLINE");
    }
}

public boolean onCreateOptionsMenu(Menu menu){

    boolean result = super.onCreateOptionsMenu(menu);
    super.onCreateOptionsMenu(menu);
    menu.add(0,1,0, "Sobre").setIcon(R.drawable.ic_launcher);
    menu.add(0,2,0, "Sair").setIcon(R.drawable.exit);
    return result;
}

@Override
public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
    case 1: mensagemExibir("Sobre", "COMANDO - ARDUINO - Copyright ©
2012");
    break;
    case 2:
        Intent intent = new Intent(getApplicationContext(), tela_ip.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        maqActivity.this.finish();
        maqActivity.this.moveTaskToBack(true);
        System.exit(0);
    }
}

```

```

        break;
    }
    return super.onOptionsItemSelected(item);
}
public void mensagemExibir(String titulo, String texto){
    AlertDialog.Builder mensagem = new
AlertDialog.Builder(maqActivity.this);
    mensagem.setTitle(titulo);
    mensagem.setMessage(texto);
    mensagem.setNeutralButton("OK",null);
    mensagem.show();
}
public void onBackPressed() {
    mHandler.removeCallbacks(mUpdate);
    Intent intent = new Intent(getApplicationContext(), tela_ip.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent);
}
}
}

```

//Tela.java

```

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.TextView;

public class maqActivity extends Activity implements OnClickListener {
    /** Called when the activity is first created. */

    ImageButton btOnA, btOffA, btOnB, btOffB, btOnC, btOffC;
    boolean status = true;
    TextView tvInternet, tvStatusA, tvStatusB, tvStatusC;
    Handler mHandler;
    private static String CATEGORIA = "result";
    String L, N, M;
    String hostIp = null;
    @Override
    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

Intent intent = getIntent();

Bundle params = intent.getExtras();

if(params != null){
    String mostraTexto = params.getString("mensagem");
    hostIp = mostraTexto.trim();
    setContentView(R.layout.tela);
}

mHandler = new Handler();
mHandler.post(mUpdate);

btOnA = (ImageButton) findViewById(R.id.btOnA);
btOffA = (ImageButton) findViewById(R.id.btOffA);
btOnB = (ImageButton) findViewById(R.id.btOnB);
btOffB = (ImageButton) findViewById(R.id.btOffB);
btOnC = (ImageButton) findViewById(R.id.btOnC);
btOffC = (ImageButton) findViewById(R.id.btOffC);

tvStatusA = (TextView) findViewById(R.id.tvStatusA);
tvStatusB = (TextView) findViewById(R.id.tvStatusB);
tvStatusC = (TextView) findViewById(R.id.tvStatusC);
tvInternet = (TextView) findViewById(R.id.tvInternet);

btOnA.setOnClickListener(this);
btOffA.setOnClickListener(this);
btOnB.setOnClickListener(this);
btOffB.setOnClickListener(this);
btOnC.setOnClickListener(this);
btOffC.setOnClickListener(this);
}
private Runnable mUpdate = new Runnable() {
    public void run() {
        checkNetworkStatus();
        arduinoStatus("http://" + hostIp + "/");
        mHandler.postDelayed(this, 500);
    }
};
public void onClick(View bt) {

    String url = null;
    if(bt == btOnA){
        url = "http://" + hostIp + "?L=1";
    }

    if(bt == btOffA){
        url = "http://" + hostIp + "?L=0";
    }
}

```

```

        if(bt == btOnB){
            url = "http://" + hostIp + "/?M=1";
        }
        if (bt == btOffB){
            url = "http://" + hostIp + "/?M=0";
        }
        if(bt == btOnC){
            url = "http://" + hostIp + "/?N=1";
        }
        if(bt == btOffC){
            url = "http://" + hostIp + "/?N=0";
        }
        String urlGetHost = url;
        String respostaRetornada = null;

        try{
            respostaRetornada =
ConectHttpClient.executaHttpGet(urlGetHost);
            String resposta = respostaRetornada.toString();
            resposta = resposta.replaceAll("\\s+", "");
        }
        catch(Exception erro){
            //Toast.makeText(maqActivity.this, "Erro.: " + erro,
Toast.LENGTH_LONG);
        }
    }

    public void arduinoStatus(String urlArduino){

        String urlHost = urlArduino;
        String respostaRetornada = null;

        try{
            respostaRetornada =
ConectHttpClient.executaHttpGet(urlHost);
            String resposta = respostaRetornada.toString();
            resposta = resposta.replaceAll("\\s+", "");
            Log.i(CATEGORIA, resposta);

            String[] b = resposta.split(",");

            if(b[0].equals("AC1")){
                tvStatusA.setText("ON");
            }
            else{
                if(b[0].equals("AP1")){
                    tvStatusA.setText("OFF");
                }
            }
        }
        if(b[1].equals("AC2")){

```



```

        tvStatusB.setText("ON");
    }
    else{
        if(b[1].equals("AP2")){
            tvStatusB.setText("OFF");
        }
    }
    if(b[2].equals("AC3")){
        tvStatusC.setText("ON");
    }
    else{
        if(b[2].equals("AP3")){
            tvStatusC.setText("OFF");
        }
    }
    }
    Log.i(CATEGORIA, b[0]);
    Log.i(CATEGORIA, b[1]);
    Log.i(CATEGORIA, b[2]);
}
catch(Exception erro){
    //Toast.makeText(maqActivity.this, "Erro.: "+erro,
Toast.LENGTH_LONG);
}
}

public void checkNetworkStatus(){

    final ConnectivityManager connMgr = (ConnectivityManager)
    this.getSystemService(Context.CONNECTIVITY_SERVICE);

    final android.net.NetworkInfo wifi =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

    final android.net.NetworkInfo mobile =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);

    if( wifi.isAvailable() ){

        tvInternet.setText("WIFI ONLINE");
    }
    else if( mobile.isAvailable() ){

        tvInternet.setText("3G ONLINE");
    }
    else
    {
        tvInternet.setText("OFFLINE");
    }
}

```

```

    }
    public boolean onCreateOptionsMenu(Menu menu){

        boolean result = super.onCreateOptionsMenu(menu);
        super.onCreateOptionsMenu(menu);
        menu.add(0,1,0, "Sobre").setIcon(R.drawable.ic_launcher);
        menu.add(0,2,0, "Sair").setIcon(R.drawable.exit);
        return result;
    }

    public boolean onOptionsItemSelected(MenuItem item){
        switch(item.getItemId()){
            case 1: mensagemExibir();
            break;
            case 2:
                Intent intent = new Intent(getApplicationContext(), tela_ip.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
                maqActivity.this.finish();
                maqActivity.this.moveTaskToBack(true);
                System.exit(0);
            break;
        }
        return super.onOptionsItemSelected(item);
    }
    public void mensagemExibir(String titulo, String texto){
        AlertDialog.Builder mensagem = new
        AlertDialog.Builder(maqActivity.this);
        mensagem.setTitle(titulo);
        mensagem.setMessage(texto);
        mensagem.setNeutralButton("OK",null);
        mensagem.show();
    }
    public void onBackPressed() {
        mHandler.removeCallbacks(mUpdate);
        Intent intent = new Intent(getApplicationContext(), tela_ip.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
}

```

## **Anexo A: Data Sheet Arduino Mega 2560.**

*The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.*

*Summary:*

*Microcontroller ATmega2560*

*Operating Voltage 5V*

*Input Voltage (recommended) 7-12V*

*Input Voltage (limits) 6-20V*

*Digital I/O Pins 54 (of which 14 provide PWM output)*

*Analog Input Pins 16*

*DC Current per I/O Pin 40 Ma*

*DC Current for 3.3V Pin 50 mA*

*Flash Memory 256 KB of which 8 KB used by bootloader*

*SRAM 8 KB*

*EEPROM 4 KB*

*Clock Speed 16 MHz*

### **Power**

*The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.*

*External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.*

*The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.*

*The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. The power pins are as follows:*

**VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

**3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

**GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

**Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

**External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

**PWM: 0 to 13.** Provide 8-bit PWM output with the `analogWrite()` function.

**SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

**LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

**I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts,

though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function. There are a couple of other pins on the board:

**AREF.** Reference voltage for the analog inputs.

**Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a `.inf` file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A `SoftwareSerial` library allows for serial communication on any of the Mega2560's digital pins. The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I2C bus; see the documentation on the Wiring website for details. For SPI communication, use the `SPI` library.

## Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available in the Arduino repository. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

### Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100

nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment.

This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

## **USB Overcurrent Protection**

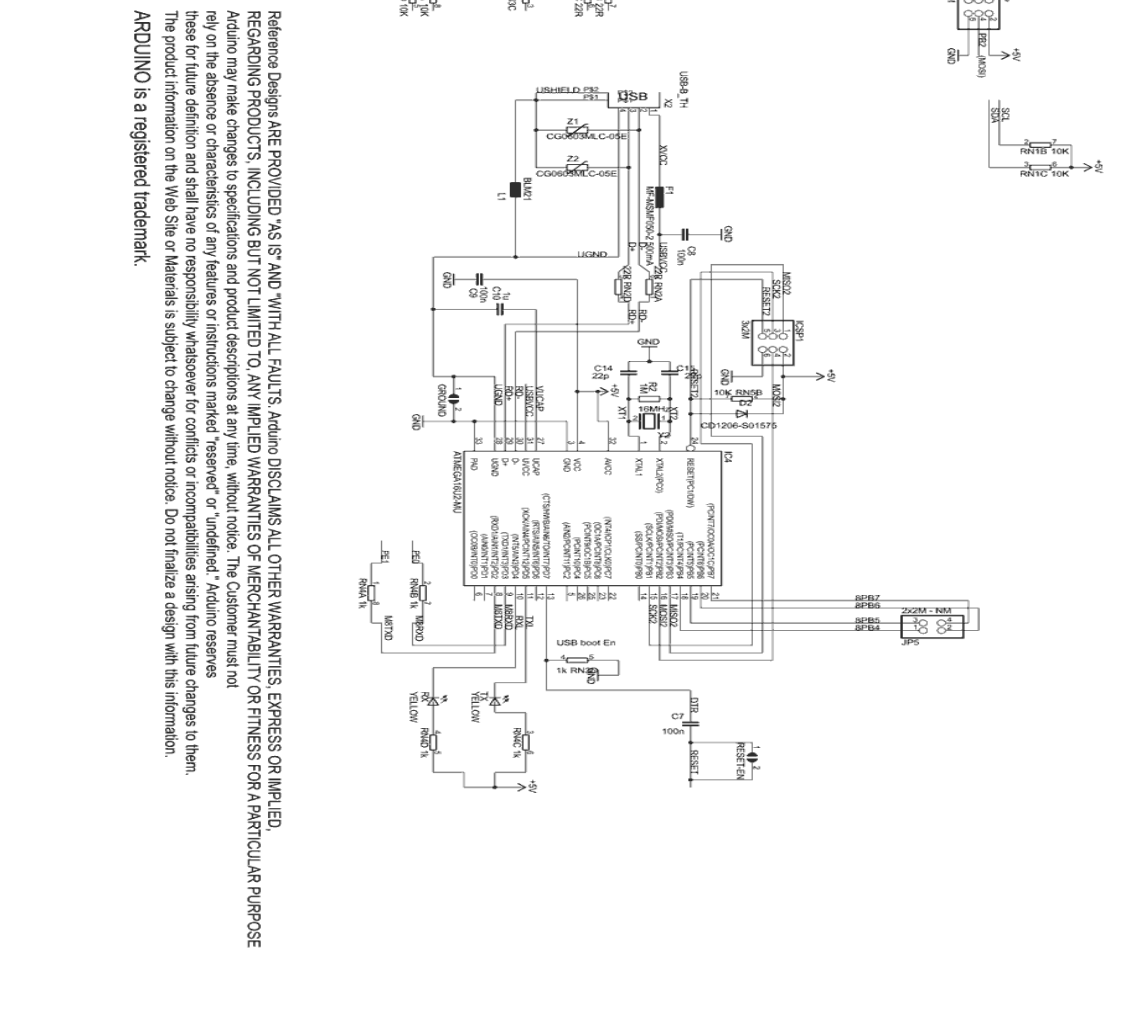
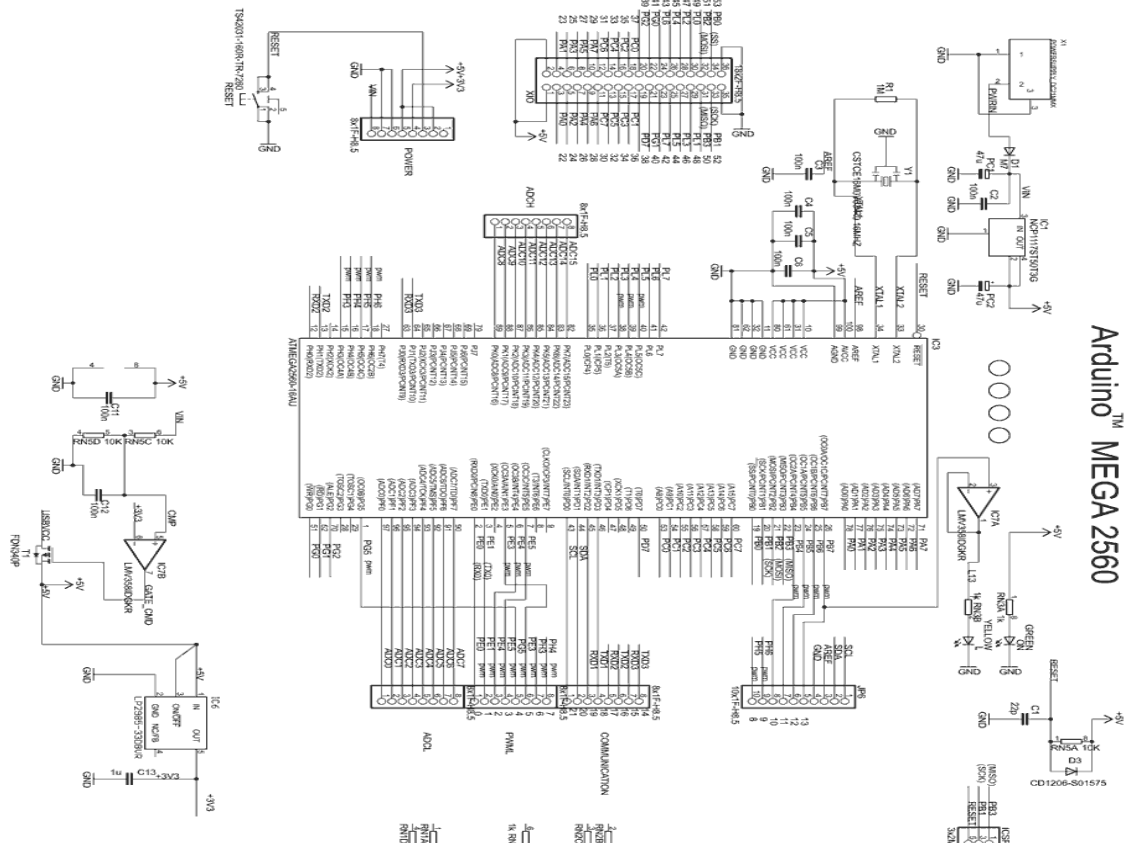
The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## **Physical Characteristics and Shield Compatibility**

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

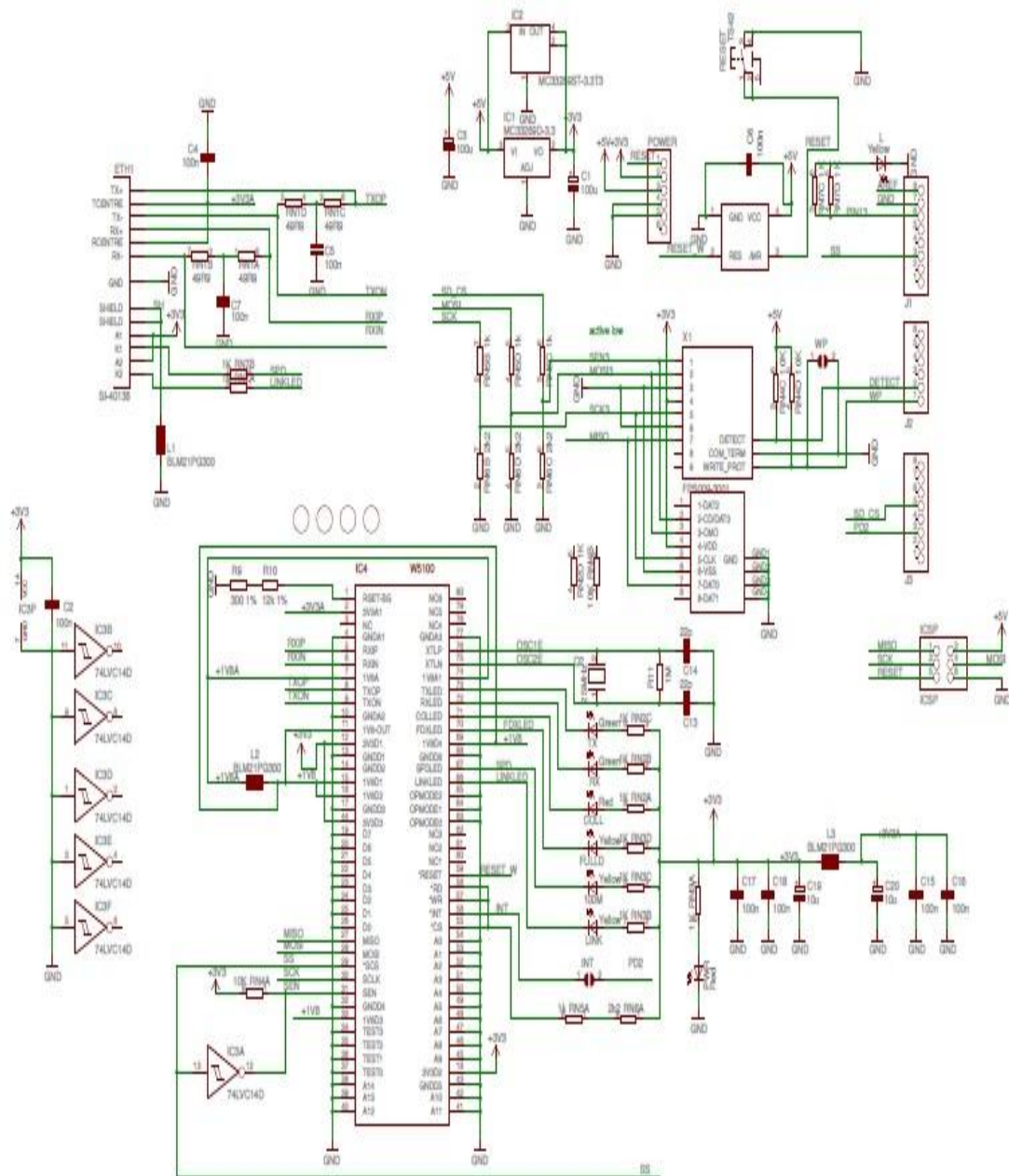
The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. Please note that I2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

# Anexo B: Diagrama Esquemático do Arduino.



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. ARDUINO DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

## Anexos C: Diagrama Esquemático do *Ethernet Shield*



### Arduino ETHERNET - shield V5

Copyright (c) 2010 Arduino  
 Released under the Creative Commons Attribution-Share Alike 3.0 License  
<http://creativecommons.org/licenses/by-sa/3.0/>