



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

WAGNER ALEXANDRE DE OLIVEIRA PORTES

**UTILIZAÇÃO DE ARDUINO E ELETRÔNICA NA AUTOMAÇÃO
RESIDENCIAL COM ACESSIBILIDADE A PESSOA PORTADORA DE
DEFICIÊNCIA**

Assis
2014

WAGNER ALEXANDRE DE OLIVEIRA PORTES

**UTILIZAÇÃO DE ARDUINO E ELETRÔNICA NA AUTOMAÇÃO
RESIDENCIAL COM ACESSIBILIDADE A PESSOA PORTADORA DE
DEFICIÊNCIA**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis como parte das exigências para graduação no Curso Superior de Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Douglas Sanches da Cunha
Área de concentração: Informática e eletrônica

Assis
2014

FICHA CATALOGRÁFICA

PORTES, Wagner Alexandre de Oliveira

Utilização de Arduino e eletrônica na automação residencial com acessibilidade a pessoa portadora de deficiência / Wagner Alexandre de Oliveira Portes. Fundação Educacional do Município de Assis – FEMA 2014

Orientador: Douglas Sanches da Cunha

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. Protótipo residencial **2.** Arduino **3.** Web Client

CDD: 001.6
Biblioteca FEMA

UTILIZAÇÃO DE ARDUINO E ELETRÔNICA NA AUTOMAÇÃO RESIDENCIAL COM ACESSIBILIDADE A PESSOA PORTADORA DE DEFICIÊNCIA

WAGNER ALEXANDRE DE OLIVEIRA PORTES

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis como parte das exigências para graduação no Curso Superior de Análise e Desenvolvimento de Sistemas analisado pela seguinte comissão examinadora:

Orientador: Me. Douglas Sanches da Cunha

Analisador: Esp. Celio Desiró

Assis
2014

DEDICATÓRIA

À minha família, que sempre acreditou e me deu forças.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por todas as oportunidades.

Agradeço à minha esposa, Simone, e ao meu filho, Pedro.

RESUMO

A domótica apresentada aqui é a aplicação do conhecimento da eletroeletrônica voltada à automação com o incremento de soluções baseadas em software e plataforma de prototipagem livre (Linux e Android), com o intuito de facilitar a vida das pessoas. O termo “domótica” é a união da palavra Domus, que deriva do latim e significa residência, e robótica, que se relaciona com a automação. Essa aplicação tem o forte intuito de apelo social, provendo acesso a atividades e equipamentos de segurança a pessoas portadoras de necessidades especiais, que não mais necessitariam da intervenção de outras pessoas. Esse projeto tem o objetivo de demonstrar que é possível, com pouco investimento, automatizar uma residência de forma adaptada para acessibilidade à pessoa portadora de deficiência. Essa automação é controlada por um hardware composto de conexões serial, USB, rede Ethernet e bluetooth, que permite monitorar qualquer equipamento à distância. O protótipo também possui uma interface de comunicação web para o gerenciamento, permitindo o acesso a dispositivos móveis.

Palavra chave: Domótica e acessibilidade.

ABSTRACT

For facilitating the everyday life of people, home automation presented here is the knowledge of electro-electronics oriented automation with the increase of software solutions based on free and prototyping platform (Android and Linux) to make life easier for them. The term "home automation" is the union of Domus word derived from Latin and means residence, and robotics that relates to automation. This application has strong order of social appeal and to provide access to activities and equipment to carry people with special needs, which no longer require the intervention of others. This project aims to demonstrate that it is possible, and with little investment automate a residence in a manner adapted for accessibility to the handicapped. A possible solution is controlled by hardware with serial connections, USB, Ethernet and Bluetooth, which allows us to control any equipment the distance. The prototype also has a web communication for the management interface allowing access to mobile devices.

Keyword: Automation and accessibility.

LISTA DE ILUSTRAÇÕES

Figura 1 - Características do Arduino.....	18
Figura 2 - Arduino Ethernet Shield	19
Figura 3 - Arduino BT Bluetooth	21
Figura 4 - Arduino Relés 5V.....	22
Figura 5 - Software Arduino IDE.....	26
Figura 6 - Botões de acesso rápido.....	27
Figura 7 - Ethernet Shield Arduino	29
Figura 8 - 1ªParte da Programação WEB.....	34
Figura 9 - 2ªParte da Programação WEB.....	35
Figura 10 - 1ªParte do Programa em C	36
Figura 11 - 2ªParte do Programa em C	36
Figura 12 - Fluxo do sistema	37
Figura 13 - Vista parcial da Maquete.....	38
Figura 14 - Vista do acionamento do Portão.....	39
Figura 15 - Parte do acionamento do CR-ROM.....	39
Figura 16 - Disposição das interfaces	40
Figura 17 - Tela do sistema de Domótica	40
Figura 18 - Ethernet Shield e Arduino Uno	41
Figura 19 – Funcionamento do portão	41
Figura 20 - Acionamento pelo celular.....	41

LISTA DE TABELAS

Tabela 1 - Características elétricas Arduino UNO	18
Tabela 2 – Características elétricas do Arduino Ethernet Shield	20
Tabela 3 – Características elétricas do Arduino BT (Bluetooth)	21
Tabela 4 – Características elétricas do Arduino Relés 5V	22
Tabela 5 - Família disponíveis endereços e protocolos.....	28
Tabela 6 – Biblioteca Ethernet.....	30
Tabela 7 - Protocolos comuns.....	31
Tabela 8 – Conexão Arduino e Servidor WEB.....	32
Tabela 9 - Tipos de socket	32
Tabela 10 - Verificação das variáveis.....	35

LISTA DE ABREVIATURAS E SIGLAS

ARM	Advanced Risc Machine
BT	Bluetooth
CLP	Controlador lógico Programável
DSP	Digital Signal processor
GSM	Global System for Mobile
HTML	Hyper text Markup Language
IDE	Integrated Development Environment
MAC	Media Access Control
MISO	Master In Slave Out
MOSE	Master Out Slave In
PIC	Programmable Interrupt Controller
PHP	Personal Home Page
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
SCK	Serial Clock
UUB	Universal Serial Bus
WEB	World Wide Web

SUMÁRIO

1 INTRODUÇÃO	13
1.1. Objetivos	14
1.2. Justificativas.....	14
1.3. Motivação.....	15
1.4. Estrutura do Trabalho	15
1.5. Perspectivas de contribuição.....	16
1.6. Metodologia de pesquisa.....	16
1.7. Recursos necessários	17
2. ARDUINO.....	18
2.1 Arduino Ethernet Shield	20
2.2 Arduino Bluetooth	21
2.3 Arduino Relés	23
2.4 Por que Arduino?	24
2.5 Domótica.....	25
2.6 Aplicações da Domótica - Automação Residencial	25
2.7 Iluminação.....	26
2.8 Segurança	26
2.9 Climatização	26
3. PROGRAMAÇÃO DO ARDUINO	27
3.1 Biblioteca SPI	28
3.2 Biblioteca Ethernet.....	29
3.3 Conexão WEB Client.....	30
3.4 Servidor WEB.	32
3.5 Programação e conceitos.....	34
4. DESENVOLVIMENTO DA MAQUETE.....	39
5. CONCLUSÃO	44
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	45

1. INTRODUÇÃO

Arduino é uma plataforma de prototipagem eletrônica de hardware livre, projetada com um microcontrolador, com suporte de entrada/saída. A principal finalidade do Arduino em um sistema é facilitar a prototipagem, implementação ou emulação do controle de sistemas interativos, no âmbito doméstico, comercial ou móvel, da mesma forma que o CLP controla sistemas de funcionamento. (ARDUINO, <<http://arduino.cc/>>).

Tecnologia assistiva é uma área do conhecimento que engloba recursos e serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas portadoras de necessidades especiais ou com mobilidade reduzida e, conseqüentemente, promover vida independente e inclusão social. A tecnologia assistiva engloba áreas como a da comunicação alternativa e ampliada; as adaptações de acesso ao computador; equipamentos de auxílio para visão e audição; controle do meio ambiente; adaptação de jogos e brincadeiras; adaptações da postura sentada; mobilidade alternativa; próteses e a integração dessa tecnologia nos diferentes ambientes como a casa, a escola, a comunidade e o local de trabalho. (Tecnologia Assistiva, <<http://www.assistiva.com.br/tassistiva.html>>).

Os Recursos são todos e quaisquer itens, equipamentos ou partes deles, produtos ou sistemas fabricados em série ou sob medida, utilizados para aumentar, manter ou melhorar as capacidades funcionais das pessoas portadoras de necessidades especiais ou com mobilidade reduzida. Os Serviços são definidos como aqueles que auxiliam diretamente uma pessoa com deficiência a selecionar, comprar ou usar os recursos acima referidos.

Este trabalho irá abordar o tema da domótica adaptada a portadores de necessidades especiais”), assunto que tem sido frequente na automação (domótica ou automação residencial). O termo vem sendo utilizado como referência à tecnologia utilizada para controle e automatização de ambientes residenciais. (Tecnologia Assistiva, <http://www.assistiva.com.br/tassistiva.html>>).

O objetivo da domótica é o melhor controle e praticidade das rotinas e tarefas

residenciais, como iluminação, climatização do ambiente, economia de recursos e controle de segurança. As origens da domótica remontam a meados da década de 80, porém o conceito de automação residencial vem sofrendo mudanças até os dias de hoje. Os meios tecnológicos vêm tornando-se cada vez mais acessíveis, tanto no sentido de tamanho e aplicação, tornando-se mais fáceis de manusear, quanto ao custo.

1.1 Objetivos

Com o aumento da demanda por profissionais com habilidade na área de automação eletrônica / computacional, o conhecimento de plataformas livres e de custo acessível, como Arduino, torna-se fundamental aos alunos e profissionais de Tecnologia em Análise e Desenvolvimento de Sistemas, Sistemas de Informação, Ciência da Computação, Engenharias e áreas correlatas à computação. O objetivo deste trabalho é desenvolver um dispositivo de domótica utilizando o Arduino, devido ao baixo custo comercial de implantação. A comunicação entre o dispositivo e o Arduino será feita por um programa desenvolvido em linguagem C, PHP, HTML. Esse trabalho será realizado 100% com aplicações Open Source. O principal objetivo é o desenvolvimento de um protótipo para o sistema de automação residencial capaz de auxiliar pessoas portadoras de necessidades especiais. Propõe-se portanto, o planejamento e construção de um projeto elétrico e cálculo para o controle das cargas elétricas a serem controladas.

1.2 Justificativas

A domótica (Automação) evoluiu muito nos últimos anos e está cada dia mais fazendo parte de nossas vidas; entretanto, o assunto tem tido pouco destaque, provavelmente, em função do alto custo. Com o Arduino, a possibilidade do projeto, torna-se algo simples e descomplicado de ser realizado e a um custo acessível. Como qualquer novidade, a domótica pode ser entendida pelo usuário como um símbolo de modernidade ou até mesmo *status*, mas o intuito desse projeto é

proporcionar conforto e segurança e até mesmo economia de recursos hídricos e elétricos, que tendem à escassez a cada dia que passa. A necessidade dessa economia ainda não é vista como urgente por muitos, pois nas próprias universidades em que se estudam meios de coibir a deterioração dos recursos, existe o desperdício, com luzes e condicionadores de ar ligados em salas vazias. Desse modo, podemos afirmar que não sabemos economizar e nem mesmo incentivar os demais cidadãos a fazê-lo. Procurando minimizar isso, utilizamos a domótica como aliada.

1.3 Motivação

A motivação para o desenvolvimento deste projeto foi a percepção de que o Arduino tornou-se uma realidade em vários projetos de automação, por ter uma fácil linguagem de desenvolvimento e um forte poder de processamento em pequenos e grandes projetos. Com isso, despertou-se no mercado a curiosidade em conhecer a programação e funcionamento do Arduino na automação residencial. Percebemos, a partir daí, que o avanço da tecnologia e o enorme crescimento do uso da internet e de aplicativos para celulares como meio de comunicação e de produtos para automação têm feito com que os computadores e dispositivos móveis se aproximem mais dos lares. Um outro efeito do aumento acelerado do mercado está na mudança dos lares de um simples ponto de acesso à internet para um provedor de serviços e aplicações em nuvem particular.

1.4 Estrutura de trabalho

O trabalho organiza-se a partir da apresentação dos módulos de hardware que fazem a integração com os equipamentos a serem comandados e controlados remotamente (Arduino).

Nesse primeiro capítulo faremos uma introdução da necessidade da domótica com a utilização do Arduino. O segundo capítulo faz referência ao hardware Arduino, procurando demonstrar a facilidade de implementação para o uso da domótica. O

terceiro capítulo aborda os objetivos específicos da domótica. O quarto capítulo refere-se à sincronia e a integração das interfaces de hardware Arduino com protótipo residencial. O quinto capítulo é demonstrado o funcionamento em maquete.

1.5 Perspectivas de contribuição

Uma das formas de aumentar o interesse dos alunos dos cursos da área de Tecnologia é buscando a união entre a teoria de uma disciplina e um problema prático, no qual possam utilizar os conhecimentos adquiridos na disciplina cursada, e sua integração multidisciplinar. Os benefícios da aplicação da domótica podem ser concentrados em quatro classes: segurança, conforto, economia de energia e comunicação. Outra perspectiva para o desenvolvimento desse projeto é apresentação do uso de uma tecnologia de baixo custo, de livre distribuição, tornando acessível para todas as camadas sociais. Por outro lado, o que aqui propomos pode também ser comercializado, pois, conforme demonstramos, o produto apresenta tecnologia viável para a automação. Com o projeto de domótica, demonstramos que é possível o trabalho e o desenvolvimento de criações de novas aplicações para computadores e internet e acreditamos contribuir para o seu uso no dia a dia.

1.6 Metodologia de Pesquisa

Como parte da metodologia, foi realizado o planejamento do protótipo, incluindo a perspectiva do projeto, tendo como atividades principais, a construção física do ambiente a ser controlado, o algoritmo básico para implantação e a estimativa de componentes e materiais a serem utilizados no desenvolvimento. A ideia consiste em: utilizar os sensores de temperatura e luminosidade nas medições dos parâmetros controlados dentro do ambiente; comparar essas medições com os parâmetros oferecidos pelo usuário; desenvolver um controle sequencial; e modificar os parâmetros no protótipo através dos atuadores (motores) para que eles fiquem o mais próximo possível do valor inserido pelo usuário.

1.7. Recursos necessários

O Arduino IDE é uma aplicação multiplataforma, escrita em Java, derivada dos projetos processing e Wiring. É ideal para introduzir a programação a iniciantes e a pessoas não familiarizadas com o desenvolvimento de software. Inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, capaz de compilar e carregar programas com um único clique. Com isso, não há a necessidade de editar makefiles ou executar programas em ambientes de linha de comando.

O aplicativo possui uma biblioteca chamada "Wiring", com capacidade para programar em C/C++, permitindo criar, com facilidade, operações de entrada e saída, com apenas duas funções no pedido, ou seja, criamos um programa extremamente funcional.

A interface IDE apresenta o Serial Monitor que possibilita visualizar as informações de status das condições dos pinos de I/O da placa Arduino. Com essa facilidade é possível monitorar e depurar o programa embarcado no hardware.

2. PLATAFORMA DE PROTOTIPAGEM ELETRÔNICA

Essa plataforma, também conhecida como Arduino possui controle de entrada de dados através de sensores como, temperatura, umidade, fumaça vibração, infravermelho, presença, e saída de dados, como motores, relés, leds, lâmpadas. É dotado de um oscilador de 16 Mhz, um regulador de tensão de 5v, plugue de alimentação, pinos conectores e porta USB.

Arduino é uma ferramenta de hardware que controla o meio físico através de sensores e atuadores. Os sensores captam as informações do ambiente definido tais como temperatura, umidade, presença de gases, presença de pessoas, luminosidade, etc. Sua plataforma de hardware é open-source e computação física, baseada em uma placa de microcontrolador simples e um ambiente de desenvolvimento para escrever o software de controle. (ARDUINO, <<http://arduino.cc/>>).

Arduino, como um projeto Open Source, não possui um fabricante oficial, (não existe um Arduino original ou produto pirata, o projeto é livre; o esquema lógico e elétrico é disponibilizado no site para quem desejar construir o seu próprio Arduino, sendo exigida apenas a atribuição do “duino” ao nome). O projeto surgiu na Itália, em 2005, e, devido à grande aceitação do público, pelo baixo custo de investimento e, conseqüentemente, baixo custo de venda, seu sucesso recebeu uma menção honrosa da Prix Ars Electronics; apesar de, oficialmente, o projeto se chamar Arduino, hoje vemos muitas outras placas Arduino, de diversos fabricantes. (ARDUINO, <<http://arduino.cc/>>)

A figura a seguir apresenta as características do Arduino Diecimila.

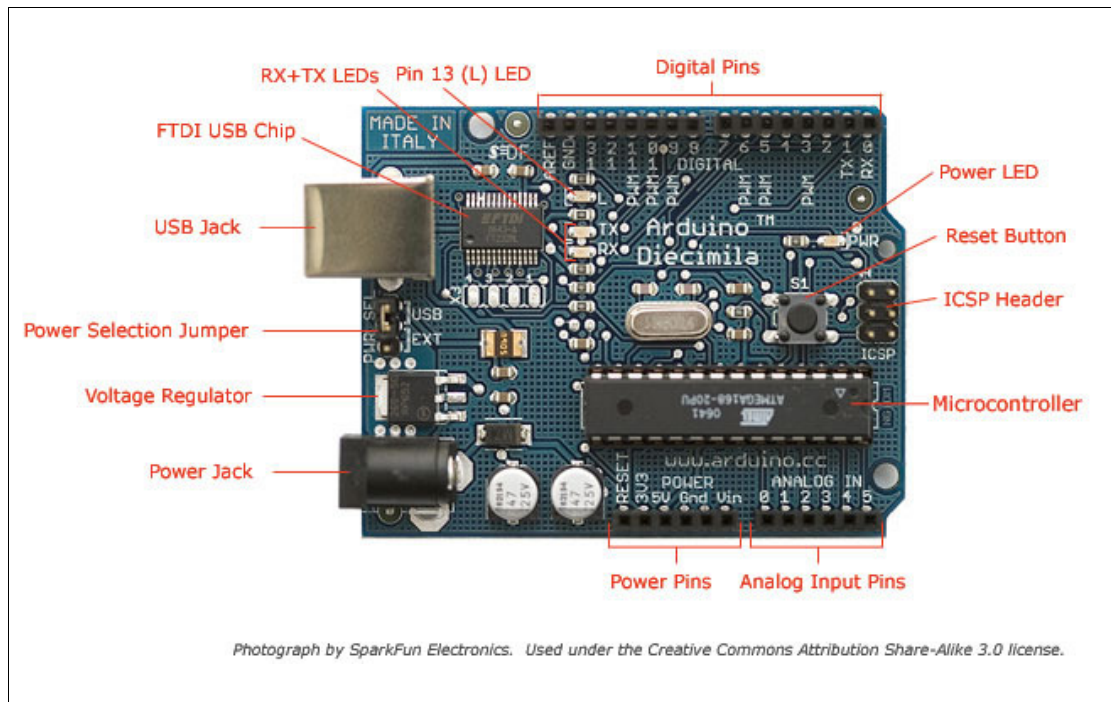


Figura 1 – Características do Arduino

O hardware de Arduino é baseado nos microcontroladores AVR da Atmel, em particular nos modelos ATmega8, ATmega168, ATmega328 e no ATmega1280. O Arduino recebe um codinome em italiano, dependendo do microcontrolador utilizado. A tabela mostra as características elétricas em relação ao microcontrolador.

Resumo	
Microcontrolador	ATmega168
Tensão de funcionamento	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada(limites)	6-20V
Digital I/O Pins	14 (dos quais 6 oferecem saída PWM)
Pinos de entrada analógica	6
Corrente DC por I/O Pin	40 mA
De corrente DC para 3.3V Pin	50 mA
Memória Flash	16 KB (dos quais 2 KB usados por bootloader)

Tabela 1 – Características elétricas Arduino UNO (ARDUINO, <<http://arduino.cc/>>).

O Arduino pode ser utilizado para desenvolver objetos interativos, tendo entradas a partir de uma ampla variedade de sensores, interruptores e controles e outras saídas físicas. Projetos com o Arduino podem ser stand-alone, ou podem se comunicar com software executando em computador. O IDE para a escrita do código é aberto e pode ser baixado gratuitamente (ARDUINO, <<http://arduino.cc/>>).

A linguagem de programação do Arduino é baseada no ambiente de programação Processing multimídia.

2.1 Arduino Ethernet Shield

O Arduino Ethernet Shield é conectado ao Arduino de forma empilhada, basta ligar este módulo à placa Arduino Diecimila. (ARDUINO, <<http://arduino.cc/>>).

O Arduino Ethernet Shield permite que uma placa Arduino conecte-se à internet. Ele baseia-se na Wiznet W5100_chip de Ethernet que, por sua vez, fornece uma rede (IP) que suporta até quatro conexões de soquetes simultâneos. A conexão é feita através de cabo Ethernet convencional par trançado Categoria 5.

A figura a seguir apresenta o Arduino Ethernet Shield

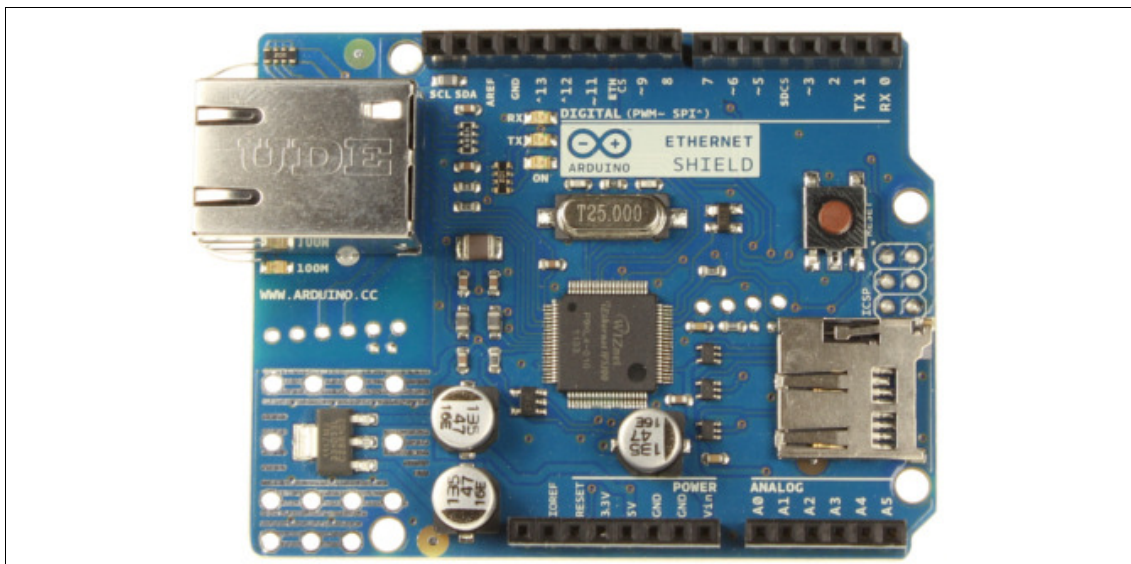


Figura 2 - Arduino Ethernet Shield (ARDUINO, <<http://arduino.cc/>>).

A tabela a seguir apresenta as características elétricas do Arduino Ethernet Shield.

IEEE802 0,3 af
Ondulação de baixa potência e ruído (100mVpp)
Faixa de tensão de entrada de 36V a 57V
Sobrecarga e proteção contra curto-circuito
9V saída
Alta eficiência DC/DC conversor: tip 75% @ carga de 50%
Isolamento 1500V (entrada e saída)
5V Tensão de operação (fornecida pela placa Arduino)

Tabela 2 – Características elétricas do Arduino Ethernet Shield

2.2 Arduino Bluetooth

O Arduino BT é uma placa de microcontrolador que foi originalmente baseada no ATmega168, mas que atualmente é fornecido com o 328. Ele suporta comunicação serial sem fio por Bluetooth (embora não seja compatível com fones de ouvido Bluetooth ou outros dispositivos de áudio). Tem 14 pinos digitais de entrada / saída, dos quais 6 podem ser usados como saídas PWM e e ainda para redefinir o WT11 módulo, 6 entradas analógicas, e uma entrada que trabalha na faixa de 16 MHz, cristal oscilador, terminais de parafuso e um botão de reset. Ele contém o necessário para suportar o microcontrolador e pode ser programado sem fio através da conexão Bluetooth.(ARDUINO, <<http://arduino.cc/>>).

A figura a seguir apresenta o Arduino Bluetooth

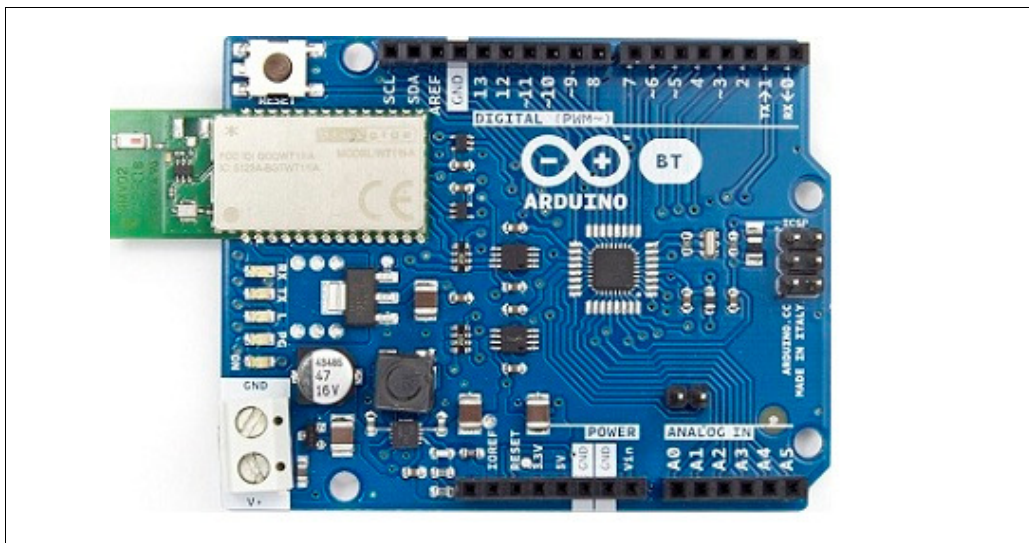


Figura 3 - Arduino BT (Bluetooth),(ARDUINO, <<http://arduino.cc/>>).

A tabela a seguir apresenta as características elétricas do Arduino BT (Bluetooth).

Resumo	
Microcontrolador	ATmega168
Tensão de funcionamento	5V
Tensão de entrada	2,5-12V
Digital I/O Pins	14 (dos quais 6 oferecem saída PWM)
Pinos de entrada analógica	6
Corrente DC por I/O Pin	40 mA
De corrente DC para 3.3V Pin	500 mA (com uma fonte de energia capaz 1,5V)
De corrente DC para 5V Pin	1000 mA (com uma fonte de energia capaz 1,5V)
Memória Flash	32 KB (dos quais 2 KB usados por bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidade do relógio 16 Mhz	
BT Module	2.1 WT11i -A- A14

Tabela 3 – Características elétricas do Arduino BT (Bluetooth) (ARDUINO, <<http://arduino.cc/>>).

2.3 Arduino Relés 5V módulo 8 Canais

Placa chaveadora com 8 relés. Funciona com 5v, a tensão mais usada em projetos digitais, com 8 saídas inversoras de alta corrente, completamente independentes e isoladas (10A). As entradas são optoacopladas, requerem baixíssima corrente para o acionamento, diretamente compatível com a maioria dos microcontroladores. (Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, lógica TTL).

A figura a seguir apresenta o Arduino Relés 5V módulo 8 Canais



Figura 4 - Arduino Relés 5V ,(ARDUINO, <<http://arduino.cc/>>).

A tabela a seguir apresenta as características elétricas do Arduino Relé 5V

Especificações:	
Canais	8
Tensão de funcionamento	5V
Tensão de carga dos Relés	Corrente de saída: 10A (250Vac - 125Vac - 30Vdc - 28Vdc)

Tabela 4 – Características elétricas do Arduino Relé 5V (ARDUINO, <<http://arduino.cc/>>).

2.4 Por que Arduino?

Há muitos outros microcontroladores e plataformas disponíveis para a computação física, como o Basic Stamp Parallax, Netmedia do BX-24, Phidgets, Handyboard do MIT, e muitos outros que oferecem funcionalidade semelhante. O Arduino também simplifica o processo de trabalho com microcontroladores, mas oferece, em relação a outros sistemas, vantagens significativas para professores, estudantes e amadores interessados. (ARDUINO, <<http://arduino.cc/>>).

Uma dessas vantagens é que Arduinos são relativamente baratos em comparação com outras plataformas de microcontroladores. O software Arduino roda em Windows, Macintosh OSX e sistemas operacionais Linux. (ARDUINO, <http://arduino.cc/>)

Com ambiente simples e programação clara, o uso do Arduino é fácil mesmo para iniciantes, mas suficientemente flexível para usuários avançados, com a possibilidade de tirar o máximo proveito. Para os professores, é conveniente o uso com base no ambiente de programação Processing, pelo qual os alunos aprendem a programar. (ARDUINO, <<http://arduino.cc/>>).

O software extensível é publicado como ferramenta de código aberto, disponível para a extensão por programadores experientes, que podem expandir a linguagem através de bibliotecas C ++. (ARDUINO, <<http://arduino.cc/>>).

O hardware extensível é baseado em Atmel ATMEGA8 e ATmega168 microcontroladores. Os planos para os módulos são publicados sob uma licença Creative Commons, para que os projetos dos circuitos possam ter a sua própria versão do módulo, ampliando-a e melhorando-a.(ARDUINO, <<http://arduino.cc/>>).

2.5 Domótica

A domótica utiliza muitos dos elementos que são utilizados na automação industrial como sensores, microcontroladores, temporizadores, e até mesmo pequenos CLP's (controladores lógicos programáveis) para automatizar processos e tarefas na automação residencial. (BOLZANI, 2004).

Automação residencial é uma evolução para o conceito de Domótica Inteligente, no qual se entende que dados obtidos dos sensores da casa serão avaliados de modo a adaptar as regras de automação do ambiente ao comportamento dos habitantes. (Tonidandel, 2004).

As pessoas estão mudando de comportamento constantemente e o que passa a ser regra ou rotina hoje, amanhã pode ser automatizado. A forma com que as atividades diárias e rotineiras acabam por constituir um padrão, contribui para que possamos cumprir com tais atividades de forma automática. Para tanto, os sistemas têm desenvolvido-se e chegarão a tal maneira de aprender e se adaptar a isto, que terão capacidade de autoajustarem-se.

2.6 Aplicações da Domótica - Automação Residencial

Hoje em dia, com os recursos tecnológicos disponíveis para a área de automação, o Arduino deixou muito mais simples e barata a automação residencial, que cresce a cada dia. As simples tarefas de casa como irrigação, abertura de cortinas, luminosidade, segurança, abertura de portões, dentre outras podem ser praticamente realizadas pela domótica.

Aproveitando a revolução da tecnologia e aplicativos que atendem a maioria dos telefones celulares e tablets, conseguimos, de forma prática, o benefício da automação. Outro fator de grande peso é a disponibilidade da internet, alavancando o avanço da domótica na grande maioria das residências, tornando-se possível tal tecnologia.

A seguir, são apresentadas algumas aplicações práticas da domótica.

2.7. Iluminação

Podem ser utilizados temporizadores e sensores de presença que controlam o acionamento de lâmpadas, ou sensores de luminosidade, que analisam a proporção luminosa do ambiente e determinam se acionam ou não a lâmpada, dentre muitas outras possibilidades.

2.8. Segurança

Em relação à segurança o assunto é bem amplo, pode-se usar uma infinidade de sensores como de pressão, temperatura, fumaça, para alertar sobre início de incêndios, vazamentos de gás, enchentes, etc, e em conjunto com um controlador, CLP (Central Lógica Programável) ou relé inteligente para alguma ação como desligar a energia, acionar dispositivos; alguns, inclusive, já permitem comunicação via linha telefônica , GSM, Internet, ou Intranet.

Ainda dentro da aplicação da domótica, pode ser feito controle de acesso à residência, utilizando sensores de leitura biométrica, RFID para aberturas de portas, Sensores Infra-vermelho para a presença, câmeras de segurança, e outros níveis de acesso. As aplicações de domótica em segurança residencial são inúmeras.

2.9. Climatização

Podem ser utilizados sensores de temperatura, presença e até mesmo de umidade. São sensores que enviam sinal a um controlador que liga ou desliga os equipamentos de ar-condicionado, aquecedores, e umidificadores, conforme o valor referencial definido pelo usuário, definindo e programando horários para o acionamento desses aparelhos.

3. PROGRAMAÇÃO DO ARDUINO

O Arduino vem pré-gravado com um *bootloader* que permite enviar novos programas sem o uso de uma interface de programação externa. A comunicação pode ser serial ou USB para a gravação de novos programas; nesse caso, é utilizado o protocolo original STK500 para fazer o upload.

O software de programação para controlar o Hardware Arduino está disponível para download no site www.arduino.cc para os sistemas operacionais Windows, Linux e Mac OS. A programação do Arduino tem como base a Linguagem C.

A interface IDE é simples e intuitiva, e caso tenha algum erro na linguagem da programação o próprio software recusa o upload.

A figura a seguir demonstra o ambiente gráfico Arduino IDE.

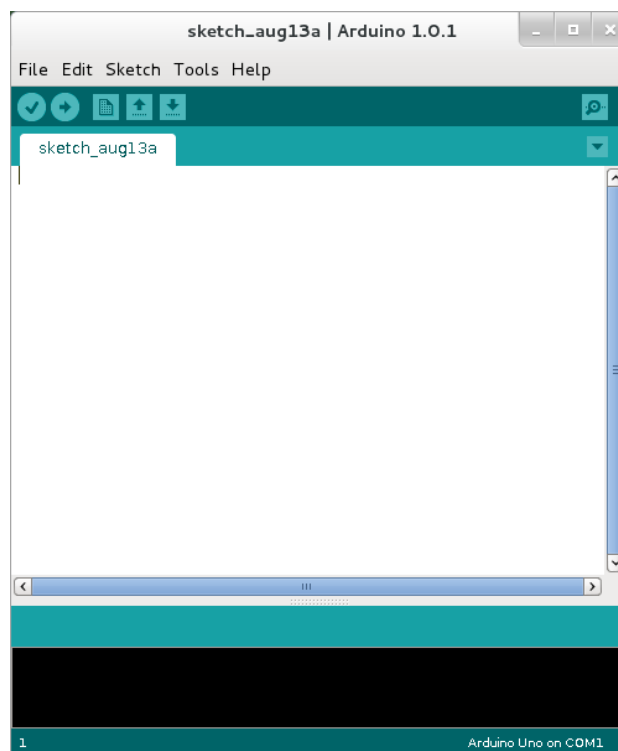


Figura 5 - Software Arduino IDE (Ambiente Gráfico).

O software Arduino é composto pelo toolbar, onde se acessa o menu de arquivos, edição, esboço, ferramentas e ajuda.

A figura a seguir apresenta as descrições dos botões de acesso rápido da interface IDE.

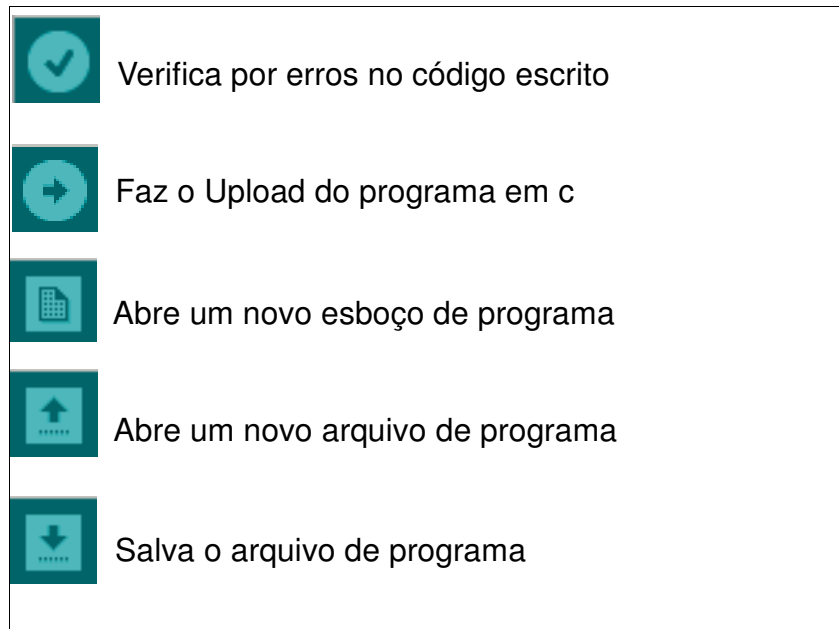


Figura 6 - Botões de acesso rápido.

3.1 Biblioteca SPI

Esta biblioteca permite comunicar com dispositivos de SPI, com o Arduino como o dispositivo master. Os dispositivos SPI - Interface Periférica Serial – constituem-se em um protocolo de dados serial síncrono usado por microcontroladores para a comunicação com um ou mais dispositivos periféricos rapidamente em curtas distâncias. Eles também podem ser usados para comunicação entre dois microcontroladores. Com uma ligação SPI há sempre um dispositivo mestre (geralmente um microcontrolador), que controla os dispositivos periféricos. Normalmente existem três linhas comuns a todos os dispositivos:

- MISO (Master In Slave Out) - A linha de Slave para o envio de dados para o mestre;
- MOSI (Master Out Slave In) - A linha do mestre para o envio de dados para os periféricos;
- SCK (Serial Clock) - Os pulsos de clock, que sincronizam a transmissão de dados gerados pelo mestre.

No início do programa a ser gravado no Arduino é mencionada essa biblioteca, pois a sua utilização proporciona fácil acesso a um dispositivo SPI específico (como a Ethernet).

A tabela a seguir demonstra o domínio e as descrições dos protocolos.

Domínio	Descrição
AF_INET	IPv4 baseado nos protocolos de Internet. TCP e UDP são protocolos comuns dessa família de protocolos.
AF_INET6	IPv6 baseado nos protocolos de Internet. TCP e UDP são protocolos comuns dessa família de protocolos. Suporta adicionado no <i>PHP 5.0.0</i> .
AF_UNIX	Família de protocolos de comunicação local. Alta eficiência e baixo sobrecarga fazem deste uma forte forma de IPC (Processos internos de comunicação).

Tabela 5 – Famílias disponíveis Endereços/Protocolos

3.2 Biblioteca Ethernet

Com o Arduino Ethernet Shield, essa biblioteca permite a conexão com uma placa Arduino para se conectar à internet. Ela pode servir como um servidor e aceitar conexões de entrada ou um cliente fazendo as saídas. A biblioteca suporta até quatro conexões simultâneas (entrada ou saída ou uma combinação). Arduino se comunica com o shield usando o barramento SPI. Esta biblioteca é usada nos pinos digitais 11, 12 e 13, no Uno o pino 10 é usado como SS.

Ao acoplar o Ethernet Shield ao Arduino, é possível monitorar através da rede o estado de sensores, chaves e outros dispositivos a partir do browser do computador ou celular.

A interface Ethernet Shield é responsável pela comunicação com a rede, tornando possível o acesso aos comandos elétricos do projeto.

A figura a seguir apresenta os pinos de conexão para os comandos elétricos.

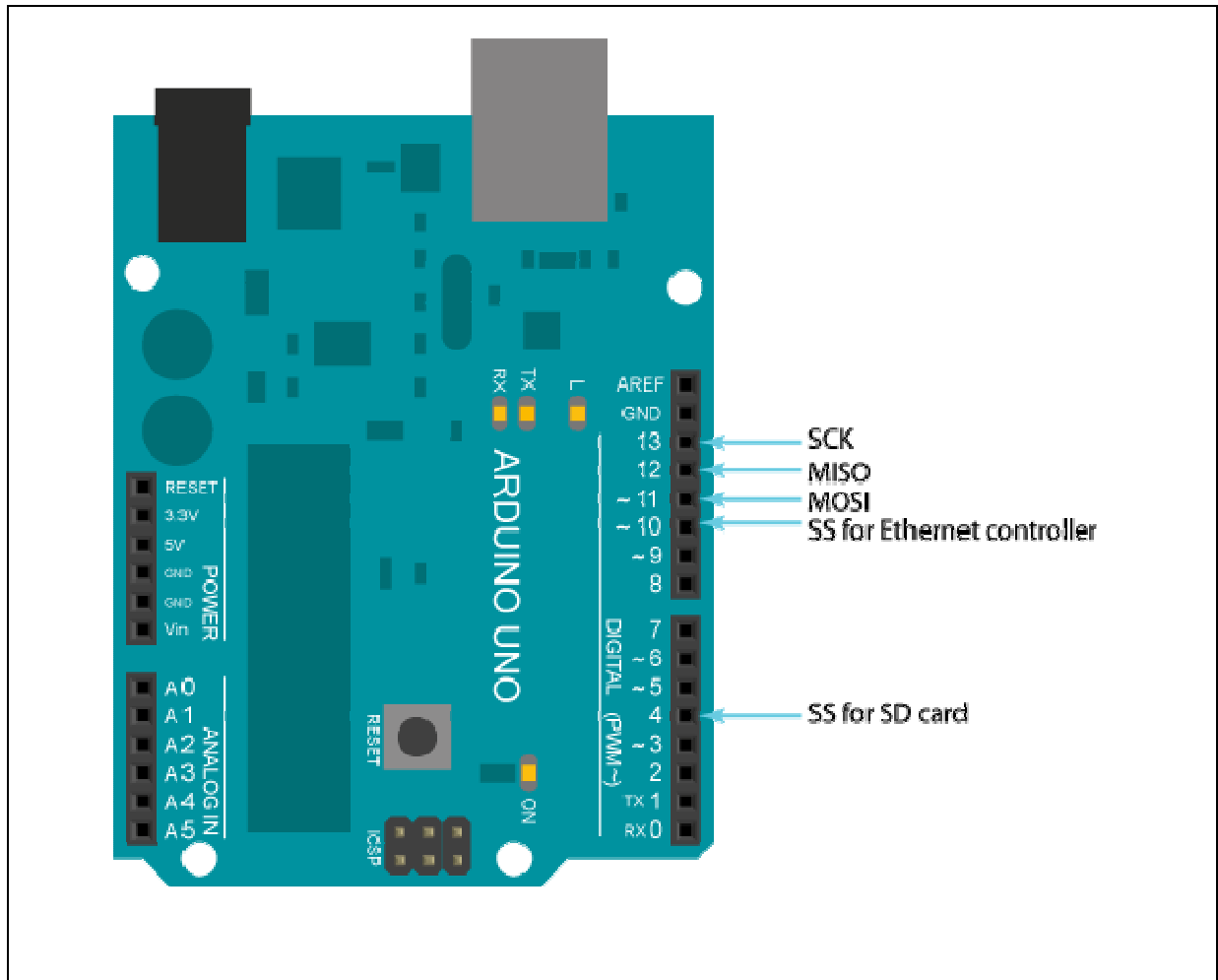


Figura 7 - Ethernet Shield Arduino (<http://arduino.cc/en/reference/ethernet>)

3.3 Conexão WebClient

Com o software IDE Arduino e a utilização de um cabo USB conectado ao Arduino, é feito o upload do código para a comunicação da interface de relés, que será usado para controlar o projeto. Primeiramente temos que definir em qual classe de IP irá trabalhar o Shield.

A tabela demonstra a biblioteca referente aos tipos de socktes:

```
#include <SPI.h>
#include<Ethernet.h>
//Configurações do Ethernet Shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,3, 10 }; // IP do Ethernet Shield
byte gateway[] = { 192,168,3,1}; // IP do Servidor Web
byte subnet[] = { 255, 255, 255, 0 };
```

Tabela 6 – Biblioteca Ethernet

byte mac= O endereço MAC (Media Access Control) para o dispositivo (array de 6 bytes). Este é o endereço de hardware do Shield.

byte ip= O endereço IP do dispositivo Shield (array de 4 bytes)

byte gateway= O endereço IP do gateway da rede (array de 4 bytes). Que no caso está apontando para a máquina onde está hospedado o servidor WEB.

byte subnet= O padrão é 255.255.255.0

A tabela a seguir demonstra os protocolos comuns de conexão ethernet

Nome	Descrição
icmp	O protocolo de controle de mensagens internet "The Internet Control Message Protocol" é usado primariamente por gateways e hosts para reportar erros no diagrama de dados de comunicação. O comando "ping" (presente em muitos modernos sistemas operacionais) é um exemplo aplicado de ICMP.
udp	O protocolo de diagrama de dados de usuários "User Datagram Protocol" é de conexão baixa, inseguro, protocolo com comprimento fixo. Dois destes aspectos, UDP requer uma quantidade mínima de sobrecarga de protocolo.
tcp	O protocolo de controle de transmissão é uma conexão, seguro, fluxo orientado, e protocolo de ida e volta. TCP garante que todos os pacotes de dados irão ser recebidos na ordem que eles são enviados. Se algum pacote está perdido de alguma maneira durante a comunicação, TCP irá automaticamente retransmitir o pacote até que o host de destino reconheça aquele pacote. Por razões de segurança e performance, o TCP implementa nele próprio decisões apropriadas de limite de octetos do diagrama de dados abaixo da camada de comunicação. Portanto, aplicações TCP devem permitir para possibilidades de registro de transmissões.

Tabela 7 – Protocolos comuns

3.4 Servidor WEB

Para o servidor WEB é necessário apenas o arquivo index.php com as instruções de socket para o hardware Arduino. É inicializada uma conexão usando o recurso `socket_create`; essa função cria e retorna o socket que também define o ponto final da comunicação, essa ligação de rede é constituída de duas bases, uma desempenhando o papel de servidor e a outra realizando o papel de cliente. O parâmetro de domínio `AF_INET` especifica a família do protocolo a ser utilizado pelo socket. `AF_INET` são protocolos baseados no Internet IPv4; TCP e UDP são comuns dessa família de protocolos. O segundo parâmetro, `SOCK_STREAM`, fornece o gerenciamento de fluxo (stream) e a conexão de dados sequencial, segura, full-duplex. Um mecanismo de transmissão de dados out-of-band pode ser suportado. O protocolo TCP `SOL_TCP` é baseado neste tipo de atividade.

A tabela a seguir apresenta o socket de conexão entre o Arduino e o servidor WEB

```
<?php

$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

// Se conecta ao IP e Porta:

socket_connect($sock,"192.168.3.10", 80);
```

Tabela 8 - com o Arduino e o servidor WEB

A tabela a seguir apresenta os tipos de socket utilizados no hardware Arduino

Tipo	Descrição
SOCK_STREAM	Fornecer sequencial, seguro, e em ambos os sentidos, conexões baseadas em "byte streams". Dados "out-of-band" do mecanismo de transmissão devem ser suportados. O protocolo TCP é baseado neste tipo de socket.
SOCK_DGRAM	Suporta diagrama de dados (baixa conexão, mensagens inconfiáveis de um comprimento máximo fixo). O protocolo UDP protocol é baseado neste tipo de socket.
SOCK_SEQPACKET	Fornecer um sequencial, seguro, e em duas direções de tipos de conexões para transmissão de dados endereçados para o diagrama de dados de comprimento máximo fixo; um consumidor é requerido para ler um pacote inteiro com cada chamada de leitura.
SOCK_RAW	Fornecer um protocolo de rede de acesso rápido. Este tipo especial de socket pode ser usado manualmente para construir algum tipo de protocolo. Um uso comum para esse tipo de socket é desempenhar requisições ICMP (como ping, traceroute, etc).
SOCK_RDM	Fornecer uma camada seguro que não garante ordenação. Isso é comumente não implementado no seu sistema operacional.

Tabela 9 - Tipos de Socket

3.5 Programação e conceitos

A ação proposta para a automação é a de controlar a iluminação e a abertura do portão eletrônico através de qualquer dispositivo móvel ou aparelho que possua internet e um browser para acessar a página para o controle dos botões. Com o Arduino conectado à internet, disposto de uma interface Ethernet Shield, recebendo as informações do servidor WEB através da página, torna-se possível o controle de dispositivos elétricos como lâmpadas, ventiladores ou abertura de fechaduras elétricas através de um circuito de relés com potência suficiente para o trabalho que lhe foi definido.

A figura a seguir apresenta parte da programação do servidor Web em linguagem HTML e PHP:

```

index.php x
1 <html>
2 <head></head>
3 <body>
4 <?php
5 $sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
6 // Se conecta ao IP e Porta:
7 socket_connect($sock,"192.168.3.10", 80);
8 // Executa a ação correspondente ao botão apertado.
9 if(isset($_POST['bits'])) {
10     $msg = $_POST['bits'];
11     if(isset($_POST['Quintal'])) { if($msg[0]=='1') { $msg[0]='0'; } else { $msg[0]='1'; }}
12     if(isset($_POST['Quarto1'])) { if($msg[1]=='1') { $msg[1]='0'; } else { $msg[1]='1'; }}
13     if(isset($_POST['Quarto2'])) { if($msg[2]=='1') { $msg[2]='0'; } else { $msg[2]='1'; }}
14     if(isset($_POST['Sala_de_TV'])) { if($msg[3]=='1') { $msg[3]='0'; } else { $msg[3]='1'; }}
15     if(isset($_POST['Ventilador'])) { if($msg[4]=='1') { $msg[4]='0'; } else { $msg[4]='1'; }}
16     if(isset($_POST['Portao'])) { $msg = 'B#'; }
17     socket_write($sock,$msg,strlen($msg));
18 }
19 socket_write($sock,'R#',2); //Requisita o status do sistema.
20 // Define a a imagem dos botões.
21 $status = socket_read($sock,8);
22 if (($status[5]=='L')&&($status[6]=='#')) {
23     $arduino[1] = $status[0];
24     $arduino[2] = $status[1];
25     $arduino[3] = $status[2];
26     $arduino[4] = $status[3];
27     $arduino[5] = $status[4];
28
29     $aparelho[1] = "Quintal";
30     $aparelho[2] = "Quarto1";
31     $aparelho[3] = "Quarto2";
32     $aparelho[4] = "Sala de TV";
33     $aparelho[5] = "Ventilador";
34 }?>
35 <html>
36 <header>
37 </header>
38 <body style="background-color:#000000">
39 <form method="post" action="indexbotao.php">
40 <input type="hidden" name="bits" value="<?php echo $status; ?>">
41 <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
42 <table width="100%" height="50">
43 <tr width="100%" >
44 <td width="100%" style="color: #FFFFFF"><p align="center" style="text-align:center">
45 <h3>Automa&cedil&atilde Residencial</h3></p>
46

```

Figura 8 - 1ª Parte da programação Web

Primeiramente é criada uma conexão com a variável \$sock em:

```
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
```

Ele fornece sequencial, seguro e, em ambos os sentidos, conexões baseadas em "byte streams". Dados fora da banda do mecanismo de transmissão são suportados. O protocolo TCP é baseado neste tipo de socket.

Na tabela a seguir é colocado os parâmetros para verificar se a variável definida foi iniciada, e é usado o comando `isset`, que executa a ação correspondente ao botão acionado.

```

if(isset($_POST['bits'])) {
    $msg = $_POST['bits'];
    if(isset($_POST['Quintal'    ])){ if($msg[0]=='1') { $msg[0]='0'; } else { $msg[0]='1'; }}
    if(isset($_POST['Quarto1'   ])){ if($msg[1]=='1') { $msg[1]='0'; } else { $msg[1]='1'; }}
    if(isset($_POST['Quarto2'   ])){ if($msg[2]=='1') { $msg[2]='0'; } else { $msg[2]='1'; }}
    if(isset($_POST['Sala_de_TV' ]){ if($msg[3]=='1') { $msg[3]='0'; } else { $msg[3]='1'; }}
    if(isset($_POST['Ventilador' ]){ if($msg[4]=='1') { $msg[4]='0'; } else { $msg[4]='1'; }}
    if(isset($_POST['Portao'    ]){ $msg = 'B#'; }
    socket_write($sock,$msg,strlen($msg));
}

```

Tabela 10 – Verificação de variáveis

A função **socket_write()** se escreve em um socket de buffer. Se esse comprimento é maior que o comprimento do buffer, isso será silenciosamente truncado para o comprimento desse buffer.

Na figura da segunda parte é utilizada a linguagem HTML para serem visualizados pelo browser os comandos pré definidos.

```

index.php
44 <td width="100%" style="color: #FFFFFF"><p align="center" style="text-align:center">
45 <h3>Automa&ccedil&atildeo Residencial</h3></p>
46
47 </td>
48 </tr>
49 </table>
50 <?php
51 for ( $i=1; $i <= count($aparelho); $i++) {
52     ?>
53     <table with="100%" height="68">
54     <tr width="100%" >
55     <td width="100%" style="color: #FFFFFF"><h2><b><?php echo $aparelho[$i]; ?></b></h2></td>
56     <td width="100%" >
57     <?php if ( $arduino[$i] == 0 ) { ?>
58     <input type="image" src="ButonOn.png" width="100" height="40" name="<?php echo $aparelho[$i]; ?>" value="1">
59     <?php } else { ?>
60     <input type="image" src="ButonOff.png" width="100" height="40" name="<?php echo $aparelho[$i]; ?>" value="0">
61
62     <?php } ?>
63     </td>
64     </tr>
65     </table>
66     <?php } ?>
67     <table with="100%" height="68">
68     <tr>
69     <td width="100%" style="color: #FFFFFF"><h2><b>Port&atildeo</b></h2></td>
70     <td width="100%" >
71     <input type="image" src="portao.png" width="100" height="40" name="Portao" value="1">
72     </td>
73     </tr>
74     </table>
75     <table with="100%" height="50">
76     <tr width="100%" >
77     <td width="100%" style="color: #FFFFFF"><p align="center" style="text-align:center">
78     <?php echo date("d/m/Y H:i:s"); ?></p>
79     </td>
80     </tr>
81     </table>
82     </form>
83     </body>
84     </html>
85
86     <?php
87     }
88     // Caso ele não receba o status corretamente, avisa erro.
89     else { echo "Interface não encontrada." }

```

Figura 9 – 2º Parte da programação Web

Na figura a seguir é demonstrado a 1ª parte da programação em C..

```

index.php x Domotica.c x
1 #include <SPI.h>
2 #include <Ethernet.h>
3 //Configurações do Ethernet Shield
4 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
5 byte ip[] = { 192,168,3, 10 }; // IP do Ethernet Shield
6 byte gateway[] = { 192,168,3,1}; // IP do Servidor web
7 byte subnet[] = { 255,255, 255, 0 };
8 char Luz[8] = "0000L#"; // String que representa o estado dos dispositivos
9 EthernetServer server(80); // O servidor responde porta 80
10 char msg[8] = "0000L#"; // String onde é guardada as msg's recebidas
11
12 void setup() {
13   Ethernet.begin(mac, ip, gateway, subnet);
14   Serial.begin(9600);
15   server.begin();
16   pinMode(A0,OUTPUT);
17   pinMode(A1,OUTPUT);
18   pinMode(A2,OUTPUT);
19   pinMode(A3,OUTPUT);
20   pinMode(A4,OUTPUT);
21   pinMode(A5,OUTPUT);
22 }
23 void loop() {
24   EthernetClient client = server.available();
25   if (client) { // Quando o client receber um caracter
26     Serial.println(msg); // Guarda o caracter na string 'msg'
27     msg[0]=msg[1]; msg[1]=msg[2]; msg[2]=msg[3]; msg[3]=msg[4]; msg[4]=msg[5]; msg[5]=msg[6];
28     msg[6] = client.read();
29     if (msg[6]!='#') {
30       switch(msg[5]) {
31         case 'R':
32           client.write(Luz);
33           break;
34         case 'B':
35           digitalWrite(A5,HIGH);
36           delay(1000);
37           digitalWrite(A5,LOW);
38           break;
39         case 'L':
40           Luz[0]=msg[0];
41           Luz[1]=msg[1];
42           Luz[2]=msg[2];
43           Luz[3]=msg[3];
44           Luz[4]=msg[4];
45           if (Luz[0]=='1') digitalWrite(A0,HIGH); else digitalWrite(A0,LOW);
46           if (Luz[1]=='1') digitalWrite(A1,HIGH); else digitalWrite(A1,LOW);

```

Figura 10 - 1ªParte programação em C

Na figura a seguir é colocado a 2ª parte da programação em C

```

index.php x Domotica.c x
11
12 void setup() {
13   Ethernet.begin(mac, ip, gateway, subnet);
14   Serial.begin(9600);
15   server.begin();
16   pinMode(A0,OUTPUT);
17   pinMode(A1,OUTPUT);
18   pinMode(A2,OUTPUT);
19   pinMode(A3,OUTPUT);
20   pinMode(A4,OUTPUT);
21   pinMode(A5,OUTPUT);
22 }
23 void loop() {
24   EthernetClient client = server.available();
25   if (client) { // Quando o client receber um caracter
26     Serial.println(msg); // Guarda o caracter na string 'msg'
27     msg[0]=msg[1]; msg[1]=msg[2]; msg[2]=msg[3]; msg[3]=msg[4]; msg[4]=msg[5]; msg[5]=msg[6];
28     msg[6] = client.read();
29     if (msg[6]!='#') {
30       switch(msg[5]) {
31         case 'R':
32           client.write(Luz);
33           break;
34         case 'B':
35           digitalWrite(A5,HIGH);
36           delay(1000);
37           digitalWrite(A5,LOW);
38           break;
39         case 'L':
40           Luz[0]=msg[0];
41           Luz[1]=msg[1];
42           Luz[2]=msg[2];
43           Luz[3]=msg[3];
44           Luz[4]=msg[4];
45           if (Luz[0]=='1') digitalWrite(A0,HIGH); else digitalWrite(A0,LOW);
46           if (Luz[1]=='1') digitalWrite(A1,HIGH); else digitalWrite(A1,LOW);
47           if (Luz[2]=='1') digitalWrite(A2,HIGH); else digitalWrite(A2,LOW);
48           if (Luz[3]=='1') digitalWrite(A3,HIGH); else digitalWrite(A3,LOW);
49           if (Luz[4]=='1') digitalWrite(A4,HIGH); else digitalWrite(A4,LOW);
50           break;
51       }
52     }
53   }
54 }
55

```

Figura 11 - 2ªParte da programação em C

A figura a seguir é a representação do ciclo de varredura do sistema de domótica em relação as portas de I/O.

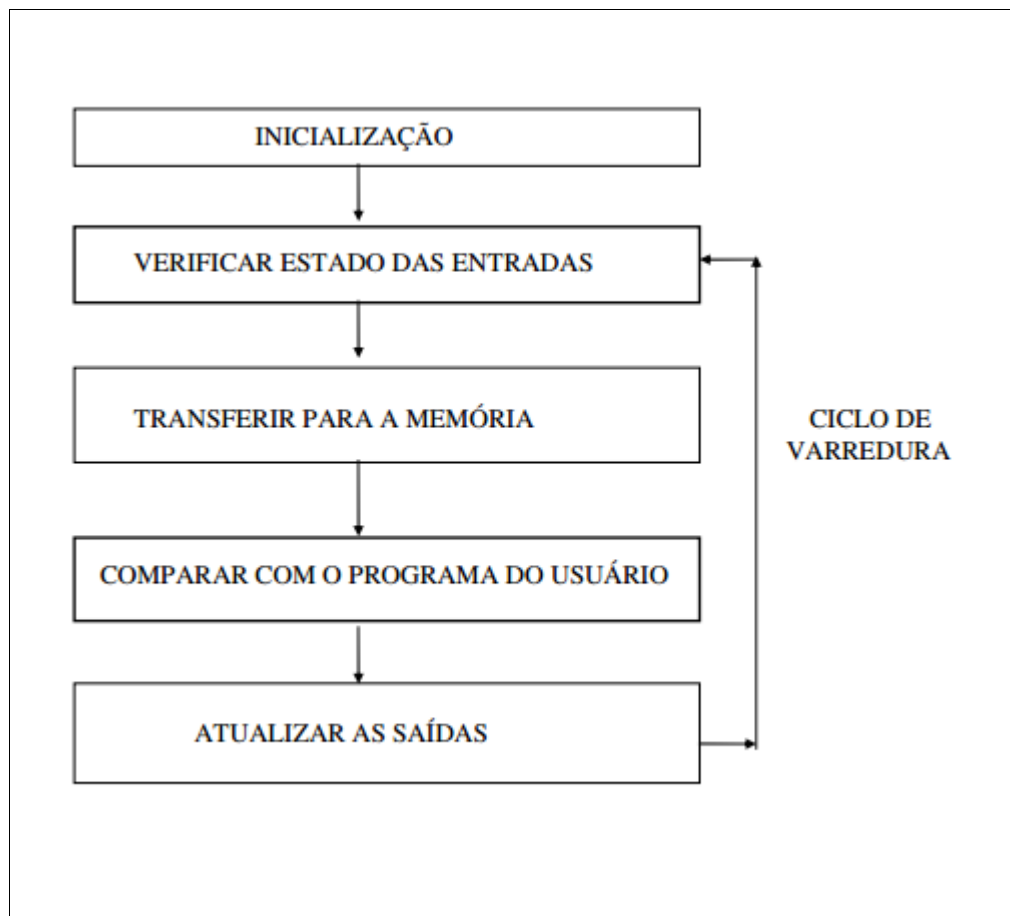


Figura 12 - Fluxo do sistema de Domótica

A comunicação da interface do Arduino com a página Web é realizada por um programa desenvolvido em C, tendo em vista que a sintaxe da linguagem é baseada na biblioteca Wiring (baseado em C e C++). A figura 12 mostra o diagrama de integração do Arduino com o dispositivo de chaveamento com relés.

4. Desenvolvimento da Maquete

A maquete foi desenvolvida como instrumento de simulação, tornando possível a visualização do sistema de domótica.,.

Com ela as operações do sistema em funcionamento, mecânico e elétrico, nos dá a descrição analítica do caso. O objetivo deste documento é a representação tridimensional, que se encarrega de facilitar o entendimento direto e didático.

A implementação do sistema utilizando a maquete permite que a parte elétrica seja testada em ambiente seguro, não correndo o risco de fazer a instalação em uma residência, sem ter feito os testes iniciais para a aplicação da domótica.

A figura a seguir demonstra a vista parcial da maquete.

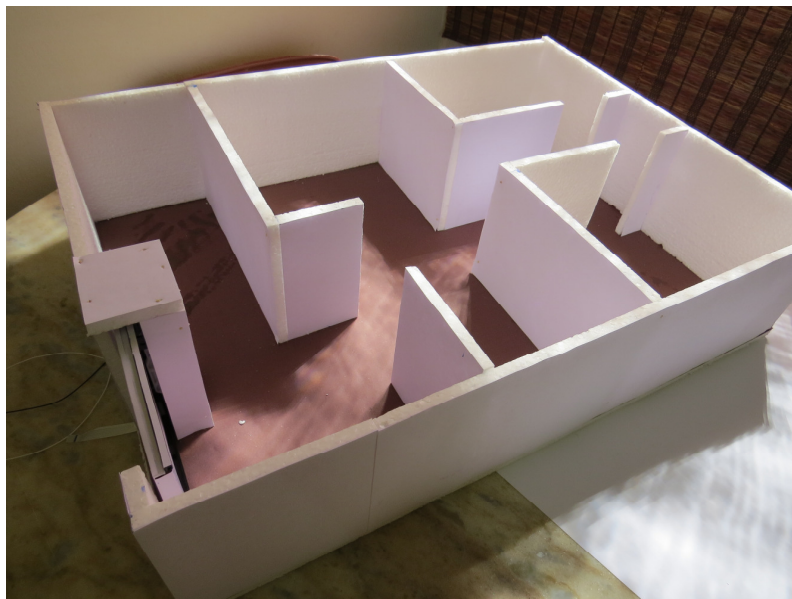


Figura 13 – Vista parcial da maquete.

A Maquete foi construída em isopor e cartolina; é dotada de 'Leds' para simular as lâmpadas, uma ventoinha para um ventilador e um acionamento de CD-ROM para a simulação de um portão eletrônico.

A figura representa a simulação do acionamento do portão de uma residência.



Figura 14 - Vista do acionamento do Portão

A figura demonstra parte do circuito utilizado para o acionamento simulado de um portão residencial.

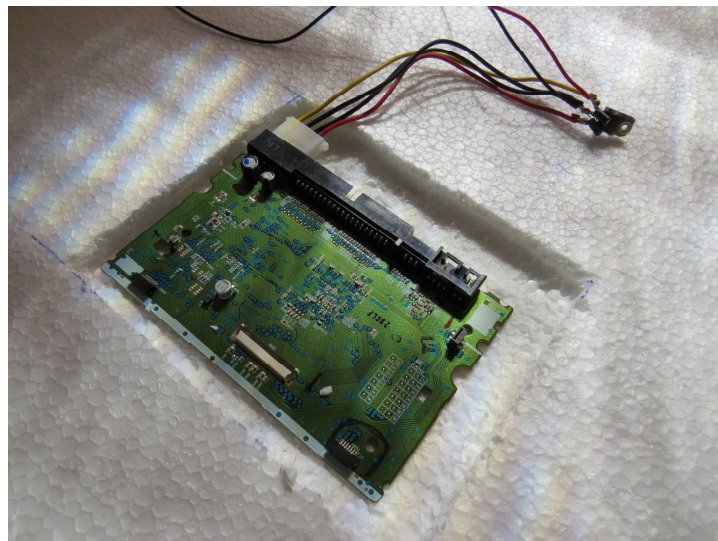


Figura 15 - Parte do acionamento CD-ROM

O circuito eletrônico e mecânico do cd-rom foi desmontado e acondicionado na parte inferior da maquete para ter a impressão real na simulação de acionamento de portão.

Disposição das interfaces de relés de acionamento, o Arduino embarcado a Ethernet Shield, a fonte auxiliar de 12V e 5V e um rádio para a conexão com a internet.

Na figura a seguir demonstra as interfaces utilizadas para a domótica.

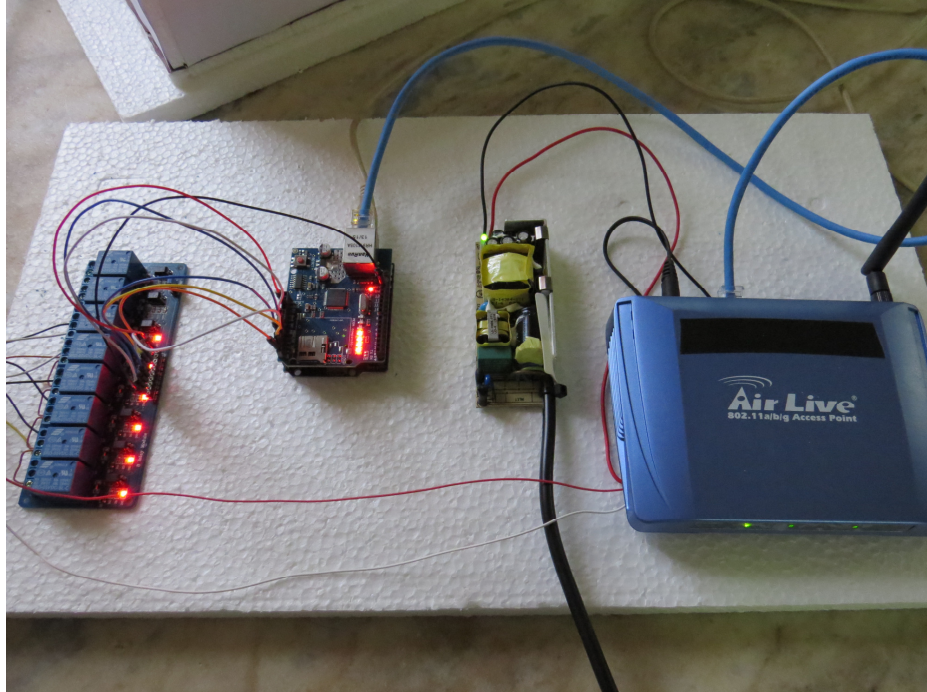


Figura 16 - Disposição das interfaces

Na figura é demonstrado a tela para utilização do sistema pelo browser do celular.



Figura 17 - Tela do sistema da Domótica

Na figura apresentada é dada as interfaces Ethernet Shield e Arduino Uno em funcionamento com o sistema.

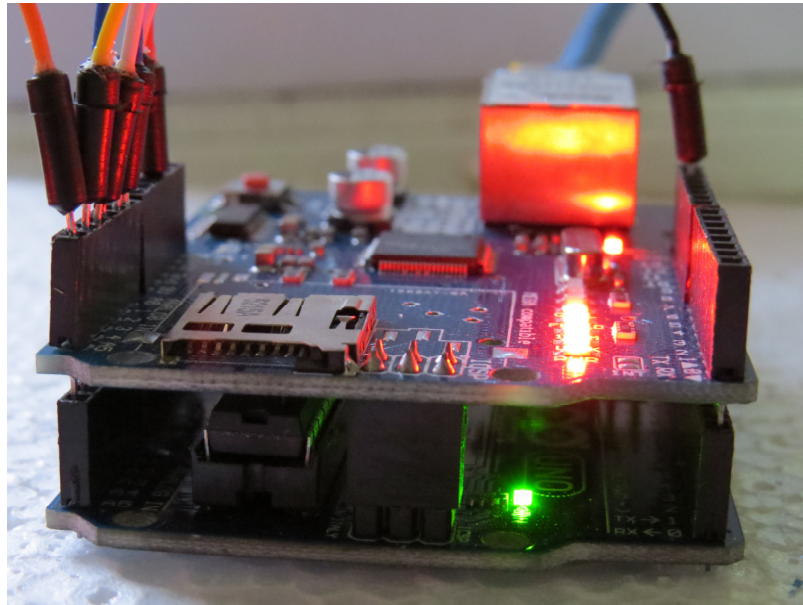


Figura 18 - Ethernet Shield e Arduino Uno

A figura demonstra o acionamento do portão



Figura 19 – Funcionamento do portão

A figura demonstra o sistema de domótica em funcionamento.



Figura 20 - Maquete acionada pelo celular

5 CONCLUSÃO

Considerando que esse trabalho aparentemente fácil, o sistema proposto mostra-se viável e simplificado, o que se torna atrativo e adaptativo ao alcance de todos. A implementação da automação residencial nesses moldes não gera custos em reformas e pode ser implantado em qualquer residência independente da forma que é construída.

Além das exemplificações, outros projetos podem ser implementados com esta tecnologia, proporcionando mais conhecimento e gerando novas idéias para novos projetos.

Através desse sistema fica a possibilidade da contribuição social para facilitar o dia a dia de pessoas portadoras de deficiência.

Por fim, esse trabalho servirá de guia para que outros alunos criem motivação na busca pelo conhecimento do arduino e interesse pelos projetos em domótica.

6. REFERÊNCIAS BIBLIOGRÁFICAS

1. ALIEVI, C. CENTRO UNIVERSITÁRIO FEEVALE. Instituto de Ciências Exatas e Tecnológicas. Automação residencial com utilização de controlador lógico programável, 2008. 84p, il. Monografia (Graduação).
2. ARDUINO. Disponível em: <<http://arduino.cc/>>.
3. BOLZANI, C.A.M. Residências Inteligentes. Editora Livraria da Física, São Paulo. 2004
4. TECNOLOGIA ASSISTIVA <<http://www.assistiva.com.br/tassistiva.html>>.
5. Tonidandel, F., Takiuchi, M., Melo, E. Domótica Inteligente: Automação baseada em comportamento. Congresso Brasileiro de Automática. (2004)