



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

ADEMIR CAVALCANTE E SILVA

**PROJETO DE UM SISTEMA DE DOMÓTICA UTILIZANDO  
ARDUÍNO**

Campina Grande, Paraíba  
Maio de 2014

ADEMIR CAVALCANTE E SILVA

# PROJETO DE UM SISTEMA DE DOMÓTICA UTILIZANDO ARDUÍNO

*Trabalho de Conclusão de Curso submetido à  
Unidade Acadêmica de Engenharia Elétrica da  
Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
Domínio da Engenharia Elétrica.*

Área de Concentração: Automação Predial

Orientador:

Professor Edson Guedes da Costa, D. Sc.

Campina Grande, Paraíba  
Maio de 2014

ADEMIR CAVALCANTE E SILVA

# PROJETO DE UM SISTEMA DE DOMÓTICA UTILIZANDO ARDUÍNO

*Trabalho de Conclusão de Curso submetido à Unidade  
Acadêmica de Engenharia Elétrica da Universidade  
Federal de Campina Grande como parte dos requisitos  
necessários para a obtenção do grau de Bacharel em  
Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Automação Predial

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Professor Edson Gudes da Costa, D. Sc.**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho a minha família, em especial aos meus pais, que sempre me apoiaram na realização dos meus sonhos.

## AGRADECIMENTOS

Agradeço a Deus, em primeiro lugar, pela minha vida e por tudo de bom que Ele tem proporcionado.

Agradeço aos meus pais, Ademir e Eliane, por tudo que eu aprendi na vida, por sempre estarem me apoiando nas minhas realizações, e aos meus irmãos que sempre estiveram do meu lado.

Agradeço toda a minha família, por tudo que me proporcionaram.

Agradeço a todos os meus amigos. Aos de longas datas de João Pessoa, e também as amizades feitas na universidade e aos amigos de Campina Grande, que me fizeram sentir tão bem nesta cidade que me acolheu nestes últimos 6 anos.

Agradeço a todos os professores que fizeram este sonho ser possível. Em especial ao meu orientador Edson Guedes que contribui com sua experiência e toda a ajuda disponibilizada. Ao professor George que também foi meu orientador em projeto durante o curso e ainda ajudou na realização deste TCC.

Agradeço ao LAT (Laboratório de Alta Tensão) que disponibilizou todos os recursos utilizados para realização do trabalho.

Enfim, agradeço a todos que de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje.

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

Marthin Luther King.

## RESUMO

A domótica é uma tecnologia recente que permite a gestão de vários recursos habitacionais. O termo domótica resulta da palavra “Domus”(casa) com “Robótica”. Este mercado tem crescido bastante nos últimos anos, pois a tecnologia tem avançado nesse sentido barateando os custos dos sistemas e simplificando a vida diária das pessoas, seja no conforto, segurança, comunicação, etc. Este trabalho tem como objetivo o desenvolvimento de um sistema de automação predial utilizando Arduino e tem um propósito didático na graduação de implementar futuramente este conteúdo na disciplina de instalações elétricas. É utilizado uma maquete para representar uma área de um hotel e no modelo é implementado o sistema de controle de temperatura de uma sauna, luminosidade da sala de recepção, alarme do cofre, luzes do banheiro e consumo energético.

**Palavras-chave:** Domótica, Arduino, sensores, controle.

## LISTA DE ILUSTRAÇÕES

Figura 1: Arduíno Mega 2560. ....	14
Figura 2: Diagrama de blocos do microprocessador ATmega2560. ....	15
Figura 3: diagrama de pinagem do Mega 2560. ....	16
Figura 4: plataforma de <i>software</i> do arduino: IDE. ....	19
Figura 5: <i>Shields</i> Arduino. <i>Ethernet, WIFI, motor e Xbee</i> . ....	20
Figura 6: Alcance do sensor de presença(Fonte: datasheet ELEC Freaks).....	22
Figura 7: sensor de presença infravermelho. ....	22
Figura 8: Fotografia do LM35, sensor de temperatura. ....	23
Figura 9: Teclado matricial dividido em linhas e colunas .....	24
Figura 10: teclado matricial. ....	24
Figura 11: ERRO - O VALOR LIDO É INDETERMINADO. ....	25
Figura 12: Resistor de <i>pull-up</i> . ....	25
Figura 13: Circuito para acionamento de um relé.....	26
Figura 14: Módulo com 4 canais optoisolados. ....	26
Figura 15: Maquete sem as ligações elétricas.....	27
Figura 16: Sala de recepção.....	28
Figura 17: Cofre. ....	29
Figura 18: Fotografia do cômodo Sauna. ....	30
Figura 19:Fotografia do Cômodo Banheiro.....	31
Figura 20: área de serviços. ....	31



## LISTA DE ABREVIATURAS E SIGLAS

LAT	Laboratório de Alta Tensão
FTDI	<i>Future Technology Devices International</i>
PWM	<i>Pulse Width Modulation</i>
SPI	<i>Serial Peripheral Interface</i>
TTL	<i>Transistor-transistor logic</i>
TWI	<i>Two Wire Interface</i>
UARTs	<i>Universal Asynchronous Receiver/Transmitter</i>
UFCG	Universidade Federal de Campina Grande
USB	<i>Universal Serial Bus</i>

# SUMÁRIO

Agradecimentos.....	5
Resumo.....	7
Lista de Ilustrações.....	8
Lista de Abreviaturas e Siglas.....	9
Sumário.....	10
1 Introdução.....	11
2 Fundamentação teórica.....	13
2.1 Arduíno.....	13
2.1.1 Energia (alimentação).....	15
2.1.2 Memória.....	17
2.1.3 Entradas e saídas.....	17
2.1.4 Comunicação.....	18
2.1.5 Programação.....	18
2.1.6 Proteção de sobrecorrente USB.....	18
2.1.7 <i>Software</i> do Arduino.....	19
2.1.8 <i>Shields</i> do Arduino.....	19
2.2 Transdutores.....	20
3 Sistema de Controle.....	21
3.1 Descrição do sistema.....	21
3.2 Componentes do sistema.....	21
3.2.1 Sensores de presença infravermelho.....	21
3.2.2 Sensor de temperatura.....	22
3.2.3 Teclado matricial.....	23
3.2.4 Módulo de relés.....	25
3.2.5 Arduíno.....	26
4 Desenvolvimento e testes do sistema.....	27
4.1 Sala de recepção.....	27
4.2 Cofre.....	28
4.3 Sauna.....	29
4.4 Banheiro.....	30
4.5 Área de serviço.....	31
5 Conclusão.....	32
Bibliografia.....	33
APÊNDICE A – Código Arduino.....	34

# 1 INTRODUÇÃO

O termo domótica surgiu da junção da palavra “Domus” (casa) com “Robótica”. Esta área da automação residencial vem crescendo muito no mercado nacional devido às novas tecnologias disponíveis, como também pelo barateamento dos produtos aplicados e pela simplificação, segurança e conforto na vida das pessoas. A domótica pode ser aplicada em vários seguimentos, como: iluminação, segurança, climatização, controle multimídia, entre outros.

O aumento da criminalidade no Brasil é um dos fatores que mais contribuíram para o crescente interesse pela domótica, pois um sistema de controle de câmeras, por exemplo, pode proporcionar maior segurança ao proprietário de um estabelecimento comercial ou residência. Outro fator importante é o conforto e também a racionalização de energia.

Um ponto positivo em um projeto de domótica adequado é a modularidade, pois possibilita o aperfeiçoamento e incremento de novas funções no sistema ao decorrer do temp. Portanto, pode-se fazer um projeto de automação da iluminação, em seguida pode-se instalar um módulo de segurança com controles de acesso e câmeras.

Para implementação de um sistema de controle pode-se utilizar dispositivos de comunicação, sensores e dispositivos como microcontroladores, CLP ou algum outro microprocessador. Neste trabalho, em particular, buscar-se-á trabalhar com o Arduíno para o controle e processamento da informação.

O Arduíno é uma plataforma de hardware e software, com bibliotecas *open-source*, criado com o objetivo de permitir o desenvolvimento de controle de sistemas interativos de baixo custo. Tem sido utilizado nas mais diversas aplicações.

Este trabalho tem como objetivo, estudar e implementarna plataforma Arduíno, sensores e circuitos eletrônicos em aplicações voltada para a domótica. Em particular será implementado controle de iluminação, de presença, de temperatura e do consumo energético. O projeto em desenvolvimento tem a perspectiva de ser utilizado em um laboratório de ensino vinculado a uma futura disciplina do curso de Engenharia Elétrica da UFCG.

Para validação dos resultados foi desenvolvida uma maquete para representação de uma área de um hotel. Fez-se manualmente o esboço da maquete e em seguida esta maquete foi desenvolvida por uma empresa especializada em cortes a laser de MDF. Na maquete implementou-se sensores de presença, temperatura e luminosidade que funcionam como transdutores. No Arduino são feitas todas as leituras dos sensores para que seja feito o processamento da informação e em seguida o controle do protótipo.

## 2 FUNDAMENTAÇÃO TEÓRICA

No decorrer deste capítulo serão apresentados alguns aspectos teóricos com ênfase no Arduino Mega2560 que é utilizado no protótipo experimental. Os tópicos servirão como base para o melhor entendimento da construção do protótipo do modelo residencial e do controle do mesmo.

### 2.1 ARDUÍNO

A plataforma Arduino surgiu em 2005, na Itália, com um professor chamado Massimo Banzi, que queria ensinar eletrônica e programação de computadores a seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica. Porém, ensinar eletrônica e programação para pessoas que não são da área não é uma tarefa simples. Outra dificuldade era a inexistência de placas poderosas e baratas no mercado [4].

Banzi, amante de longa data da Eletrônica, propôs-se então a utilizar a computação física em suas aulas, empregando a eletrônica para o desenvolvimento da mencionada prototipagem de novos materiais. Porém, seu grande desafio seria o de delimitar a eletrônica como meio, e não finalidade, evitando que os alunos tivessem que se debruçar por horas no entendimento de conceitos elementares, e outros não tão elementares, de eletrônica. Dessa forma o Prof. Banzi objetivava fazer com que os alunos concentrassem seus esforços no desenvolvimento dos protótipos e no desenvolvimento de caminhos cada vez mais rápidos e poderosos para a construção de melhores soluções às suas necessidades.

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 se chama de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. Por exemplo, um uso simples do Arduino seria para acender uma luz por certo intervalo de tempo, como, 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. O

usuário poderia utilizar essa configuração para controlar uma lâmpada em um closet, por exemplo [3].

O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à internet para recuperar e enviar dados do Arduino e atuar sobre eles. Na Figura 1 se apresenta o Arduino Mega2560.



Figura 1: Arduíno Mega 2560.

O Arduino Mega 2560 é uma placa de microcontrolador baseada no Atmega2560. Ele tem 54 pinos de entradas/saídas digitais (das quais 15 podem ser usadas como saídas PWM (*Pulse Width Modulation*)), 16 entradas digitais, 4 UART(*Universal Asynchronous Receiver/Transmitter*), um oscilador cristal de 16 MHz, conexão USB(*Universal Serial Bus*), conector de alimentação e um botão de reset. Ele contém tudo que é necessário para suportar o microcontrolador. O Mega é compatível com a maioria dos *shields* projetados para o *Arduino Duemilanove* ou *Diecimila*[1]. A Figura 2 representa o diagrama de blocos da unidade de processamento e a Figura 3 apresenta toda a pinagem do microprocessador ATmega 2560.

Figure 2-1. Block Diagram

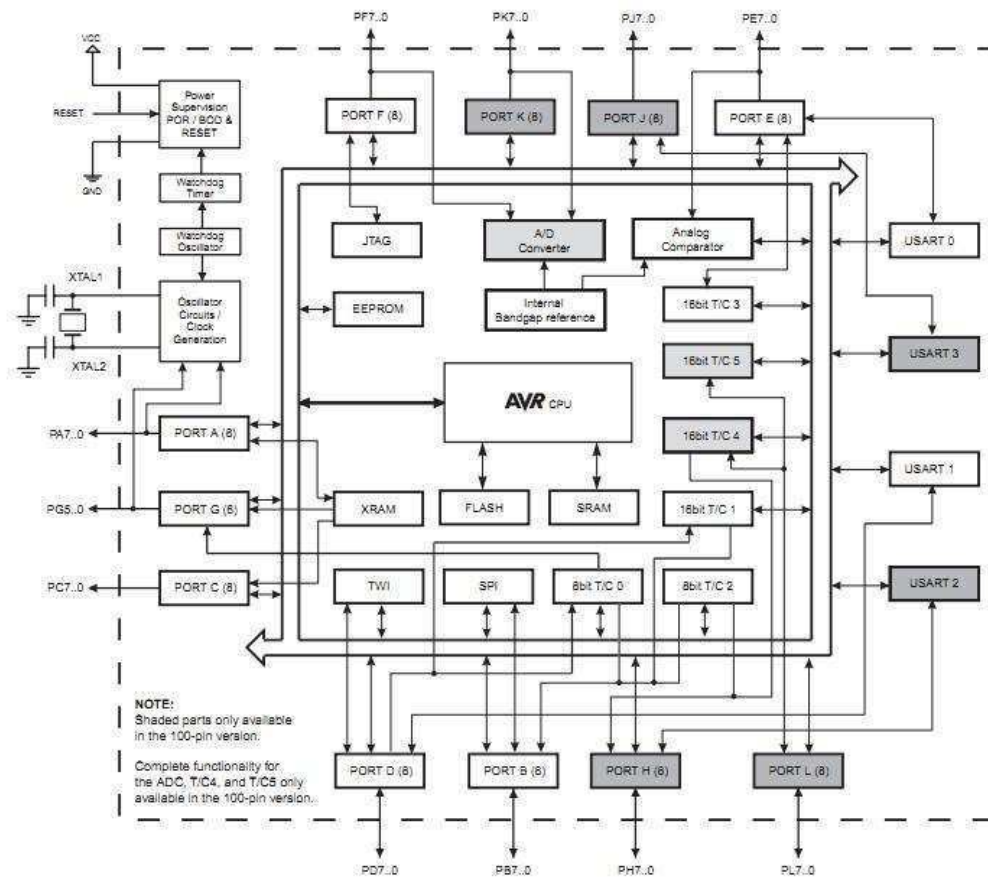


Figura 2: Diagrama de blocos do microprocessador ATmega2560.

### 2.1.1 ENERGIA (ALIMENTAÇÃO)

O Arduino Mega2560 pode ser alimentado através da conexão USB ou com uma fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente.

A placa pode operar com um fornecimento externo de 6 a 20 volts. Se fornecido com menos de 7 V, no entanto, o pino de 5 V pode fornecer menos de cinco volts e a placa pode permanecer instável. Se usar mais do que 12 V, o regulador de tensão pode superaquecer e danificar a placa. O intervalo recomendado é de 7 a 12 volts.

Os pinos de energia são os seguintes:

- VIN: tensão de entrada da placa do arduíno quando ele está utilizando uma fonte de alimentação externa.
- 5 V: Este nível de tensão pode vir a partir do VIN via o regulador da placa ou pode ser fornecida pela porta USB ou outro regulador de 5 V.

- 3,3 V: Este nível de tensão é gerado pelo FTDI(*Future Technology Devices International*) chip da placa. Máxima corrente fornecida é de 50 mA.
- GND: pinos terra.

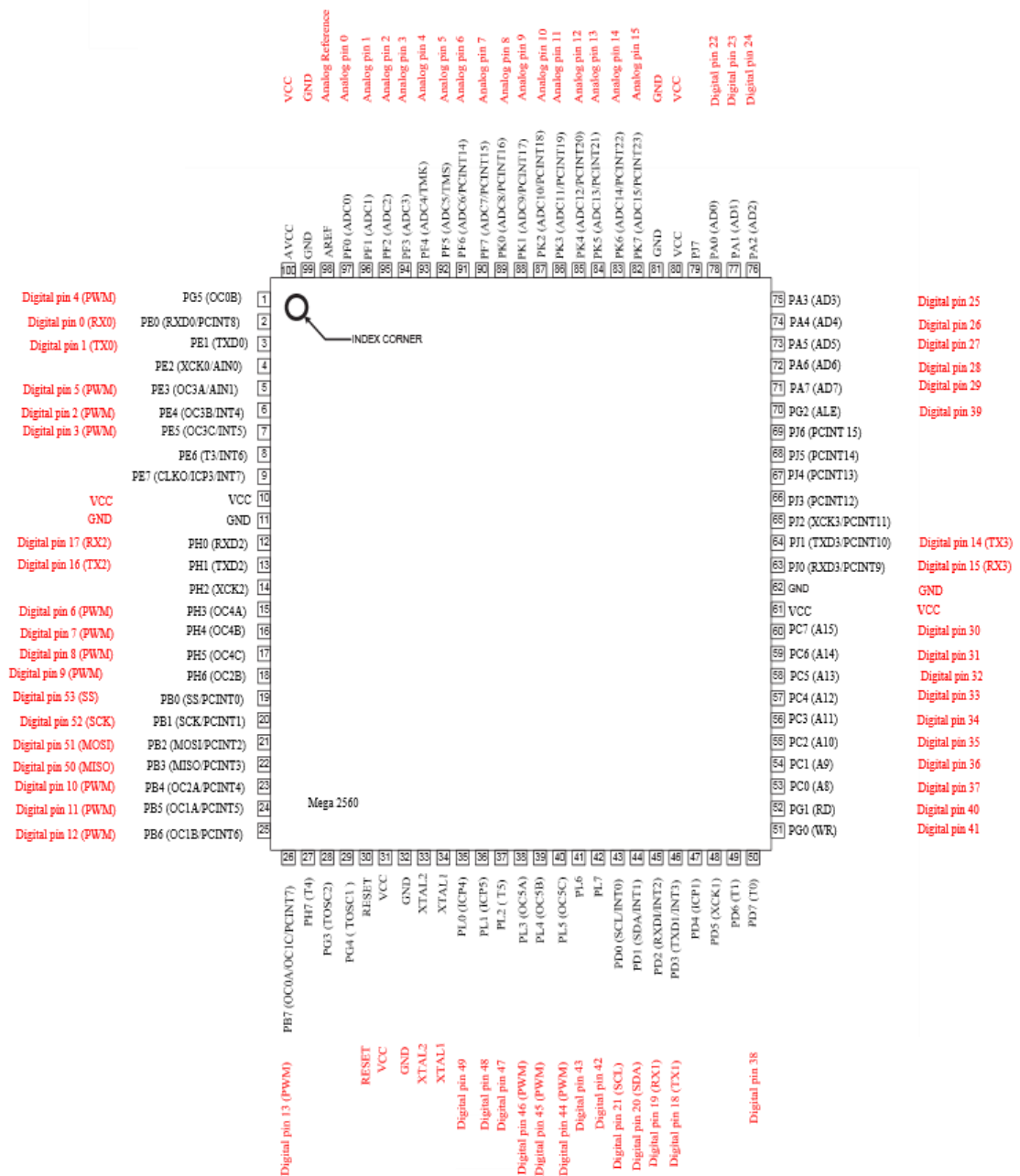


Figura 3: diagrama de pinagem do Mega 2560.



### 2.1.2 MEMÓRIA

A grande diferença entre os vários modelos de Arduino disponíveis no mercado reside na memória do microcontrolador, sendo essa uma característica determinante em seu desempenho. No microprocessador ATmega2560 há três tipos de memória: *Flash*, SRAM, EEPROM. A memória *Flash* é utilizada para o armazenamento do programa que é carregado no microcontrolador para ser executado, bem como para o armazenamento do *bootloader*. O *bootloader* é responsável pelo gerenciamento da inicialização do microchip, e desempenha algumas tarefas programadas como determinar quando reprogramar ou passar para a aplicação principal [1].

A memória SRAM (*Static Random Access Memory*) é a memória utilizada para armazenamento das variáveis de processamento. Quando o Arduino é desenergizado, as informações armazenadas nesta memória são perdidas. Diferentemente de EEPROM (*Electrically Erasable Programmable Read-Only Memory*), que é responsável pelo armazenamento de constantes ou dados de configuração, e apresenta longos ciclos de gravação e rápida leitura de dados.

O Atmega 2560 tem 256 Kb de memória flash para armazenamento de código (dos quais 8 Kb é usado para *bootloader*), 8 Kb de SRAM e 4 Kb de EEPROM (dos quais podem ser lidos e escritos com a biblioteca EEPROM).

### 2.1.3 ENTRADAS E SAÍDAS

Cada um dos 54 pinos digitais do Mega podem ser usados como entradas ou saídas, utilizando as funções *pinMode()*, *digitalWrite()* e *digitalRead()*. Eles operam a 5 volts. Cada pino pode fornecer ou receber uma corrente máxima de 40 mA e tem um resistor de *pull-up* interno (por default é desconectado) de 20-50 k $\Omega$  [1].

O ATmega2560, assim como outros microcontroladores, oferece ainda um conversor A/D. Na conversão de um sinal analógico em digital, fazem-se amostragens, e representa-se a leitura realizada através de um valor equivalente binário, de acordo com a resolução do conversor A/D. A precisão do conversor é determinada pelo número de bits que o mesmo gera para representar as grandezas analógicas. O conversor A/D do ATmega é de 10 bits, representando  $2^{10}$  níveis de tensão distintos. Assim, a resolução do conversor pode ser expressa por (1):

$$resolução = \frac{V_{ref}}{2^n} \quad (1)$$

O Mega2560 tem 16 entradas analógicas, cada uma delas fornecem 10 bits de resolução (1024 valores diferentes). Pela configuração inicial eles fazem as medições a partir do terra até 5 volts, diz-se, portanto, que o Arduino pode detectar variações de apenas 5 mV, embora seja possível mudar o limite superior de sua faixa usando o pino AREF e a função *analogReference()*.

#### 2.1.4 COMUNICAÇÃO

O Arduino Mega2560 tem várias facilidades para comunicação com um computador, outro Arduino, ou outros microcontroladores.

O *software* do Arduino inclui um monitor serial que permite que dados simples de textos sejam enviados para placa e a partir da placa. Os leds Rx e Tx da placa piscam quando os dados estão sendo transmitidos através do chip Atmega8U2/Atmega16U2 e da conexão USB do computador.

A biblioteca *Software Serial* permite comunicações serial em alguns pinos digitais do Mega2560.

O Atmega2560 também suporta comunicação TWI (*Two Wire Interface*) e SPI (*Serial Peripheral Interface*). O *software* inclui uma biblioteca *Wire* para simplificar o uso do barramento TWI.

#### 2.1.5 PROGRAMAÇÃO

O Arduino pode ser programado com o *Arduino software*. O Atmega2560 no Arduino Mega vem com um *bootloader* que permite o envio de novos códigos sem o uso de um programador de *hardware* externo. Ele se comunica utilizando o protocolo original.

#### 2.1.6 PROTEÇÃO DE SOBRECORRENTE USB

O Arduino Mega2560 tem um dispositivo que protege as portas USB de curtos e sobrecorrentes. Apesar da maioria dos computadores fornecerem sua própria proteção interna, este componente fornece uma proteção extra. Se forem aplicados mais de 500 mA na porta USB, a ligação desconecta automaticamente até que seja removido a sobrecarga ou o curto.

### 2.1.7 SOFTWARE DO ARDUINO

O ambiente de desenvolvimento Arduino contém um editor de texto para a escrita de código, uma área de mensagens, um *console* de texto, uma barra de ferramentas com botões para variadas funções, e uma série de menus. Ele se conecta ao *hardware* do Arduino, permitindo o *upload* de programas e a comunicação com eles.

Os programas escritos usando Arduino são chamados *sketches*. As *sketches* são escritas no editor de texto, e são salvas com a extensão de “*arquivo.ino*”. Elas permitem funcionalidades como recortar/colar e procurar/substituir texto. Na área de mensagem, temos retorno de ações como salvar e exportar, além da exibição de erros. O *console* mostra a saída de texto do Arduino, incluindo mensagens de erro completas e outras informações. O canto inferior direito mostra a placa atual e a portal serial. Os botões da barra de ferramentas permitem que você verifique e faça *upload* de programas, crie, abra e salve *sketches*, e abra o monitor serial. A Figura 4 apresenta a plataforma.

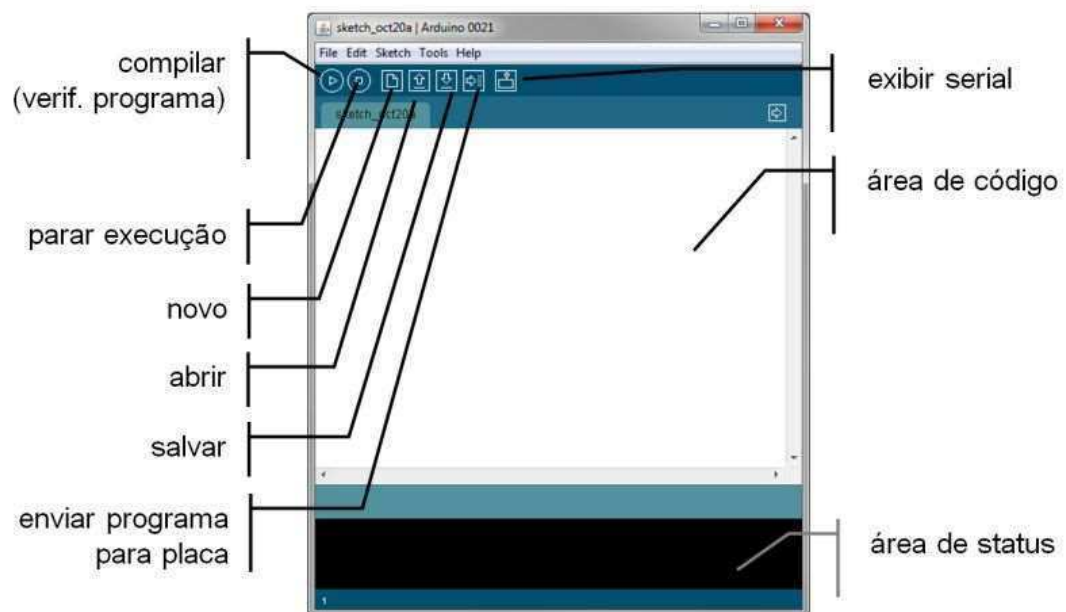


Figura 4: plataforma de *software* do arduino: IDE.

### 2.1.8 SHIELDS DO ARDUINO

Placas externas, denominadas *Shield*, podem ser acopladas ao Arduino, com a finalidade de expandir suas funcionalidades, de maneira prática, eficiente e barata. Há modelos de *Shields* para comunicação, como *Ethernet*, *Wi-Fi*, *Xbee* e *Bluetooth*, ou mesmo para controle de atuadores e sensores. Outras opções são disponíveis como: *Shield Joystick*, que contém todos os componentes necessários para se implementar um

*Joystick* com o Arduino; o *ReléShield*, diretamente controlado por um Arduino, através de suas saídas digitais com alimentação externa de 9V, capazes de controlar quatro relés, com chaveamento de potência de 90 W em CC ou 360 VA; o *Shield* celular, dotado de todos os componentes necessários para conectar o Arduino com o módulo celular SM5100B, *quad-band*, permitindo o envio de SMS e outras funcionalidades; *Shield* GPS, MP3 player, reconhecimento de voz EasyVR, MicroSD, de controle de corrente de entrada na placa, o *Power Driver Shield*, *Shield* MIDI, permitindo comunicação do Arduino com o protocolo MIDI, fazendo com que o mesmo possa controlar sintetizadores, sequenciadores e outros dispositivos musicais, além dos módulos de expansão da EEPROM, o Multiplexador de entradas e saídas, etc. A Figura 5 apresenta alguns tipos de *Shields* Arduino.



Figura 5: *Shields* Arduino. *Ethernet*, *WiFi*, *motor* e *Xbee*.

## 2.2 TRANSDUTORES

Transdutor é um componente que converte variações de uma determinada grandeza física em outra. Por exemplo, um transdutor de posição converte variações de movimento em um sinal de saída do tipo elétrico. Em geral, as grandezas físicas são convertidas em sinais elétricos adequados a aplicação em sistemas de aquisição de dados. No processo de conversão, podem ser usados sensores ativos ou passivos. Os sensores ativos são responsáveis pela detecção do evento e geração direta do sinal elétrico. Enquanto nos sensores passivos, alguns dos seus parâmetros intrínsecos variam em função de variações da grandeza medida. Esta variação paramétrica pode então ser convertida em informação mensurável através de circuitos de condicionamento apropriados [5]. Os sensores utilizados neste trabalho são apresentados mais a frente.

## 3 SISTEMA DE CONTROLE

No decorrer deste capítulo serão apresentadas a descrição do sistema proposto, componentes utilizados em cada parte do sistema, bem como a funcionalidade e explicação de cada parte do protótipo.

### 3.1 DESCRIÇÃO DO SISTEMA

Neste trabalho é proposto um projeto de automação predial. Foi construída uma maquete que representava um plano de um hotel para implementação do sistema. A escolha de representar uma área de um hotel foi tomada com base nos controles que foram escolhidos para este estudo. Objetivou-se fazer o controle de temperatura, luminosidade, presença de pessoas e consumo energético. Portanto, representou-se na maquete uma sauna para que pudesse ser feito o controle de temperatura, uma sala de recepção com controle de luminosidade, banheiro para o controle da luminosidade a partir da presença de pessoas e ainda representação do chuveiro elétrico para controle do consumo de energia a partir da faixa horária, cofre para controle do alarme a partir da presença de pessoas e área de serviços com controle do consumo energético a partir da faixa horária.

### 3.2 COMPONENTES DO SISTEMA

Para o desenvolvimento do trabalho foram utilizados vários componentes, como, sensores, relés, teclado matricial e Arduino. A seguir serão apresentados os principais componentes do sistema.

#### 3.2.1 SENSORES DE PRESENÇA INFRAVERMELHO

O sensor de presença é muito utilizado atualmente, especialmente em muitas aplicações para economia de energia: em restaurantes, prédios, e em estabelecimentos comerciais em geral são bastante utilizados. Seu funcionamento é relativamente

simples: ele emite um sinal infravermelho que é espalhado pela redoma convexa montada na parte superior, o sinal é refletido pelos objetos de volta a redoma que agrupa os feixes em um detector infravermelho. Existem ajustes de temporização e de sensibilidade, de forma que o sensor não capte qualquer variação de movimento instantânea. O sinal de saída do sensor é um sinal digital, 1 (3,3 V) para detecção e 0 (0 V) para não detecção de presença. Ele pode ser alimentado dentro da faixa de 4,5-20 V e possui um baixo consumo, pois quando está inativo utiliza uma corrente menor que 50  $\mu$ A. Possui um bom alcance como apresentado na Figura 6.

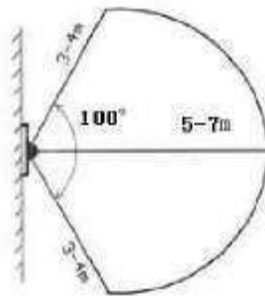


Figura 6: Alcance do sensor de presença(Fonte: datasheet ELEC Freaks)

Este tipo de sensor pode ser aplicado em produtos de segurança, brinquedos com sensores de presença de pessoas, controle de luminosidade, automação e controle industrial, etc. A Figura 7 apresenta o sensor.



Figura 7: sensor de presença infravermelho.

### 3.2.2 SENSOR DE TEMPERATURA

O sensor LM35 é um sensor de precisão, fabricado pela *National Semiconductor*, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontra no momento em que for alimentado por uma tensão de 4-20 Vdc, tendo em sua saída um sinal de 10 mV para cada grau Celsius de temperatura, sendo assim, apresenta uma boa vantagem com relação aos demais sensores de temperatura

calibrados em “Kelvin”, não necessitando nenhuma subtração de variáveis para que se obtenha uma escala de temperatura em graus Celsius.

O LM35 não necessita de qualquer calibração externa ou “trimming” para fornecer com exatidão, valores de temperatura com variações de  $\frac{1}{4}^{\circ}\text{C}$  ou até mesmo  $\frac{3}{4}^{\circ}\text{C}$  dentro da faixa de temperatura de  $-55^{\circ}\text{C}$  à  $150^{\circ}\text{C}$ . Ele sensor tem saída com baixa impedância, tensão linear e calibração inerente precisa, fazendo com que o interfaceamento de leitura seja especificamente simples, barateando todo o sistema em função disto.

Este sensor poderá ser alimentado com alimentação simples ou simétrica, dependendo do que se desejar como sinal de saída, mas independentemente disso, a saída continuará sendo de  $10\text{ mV}/^{\circ}\text{C}$ . Ele drena apenas  $60\text{ }\mu\text{A}$  para as alimentações, sendo assim seu auto-aquecimento é de aproximadamente  $0,1^{\circ}\text{C}$  ao ar livre fazendo com que seu aquecimento não interfira as medições. A Figura 8 apresenta uma fotografia do componente.



Figura 8: Fotografia do LM35, sensor de temperatura.

### 3.2.3 TECLADO MATRICIAL

Como é apresentado na Figura 9, o teclado é dividido em linhas e colunas, onde internamente há condutores que se tocam quando a tecla é pressionada, formando uma matriz 4x4 com chaves independentes [2]. O teclado matricial tem várias aplicações, para o caso deste trabalho foi utilizado para ativar ou desativar certos controles, por exemplo o cofre.

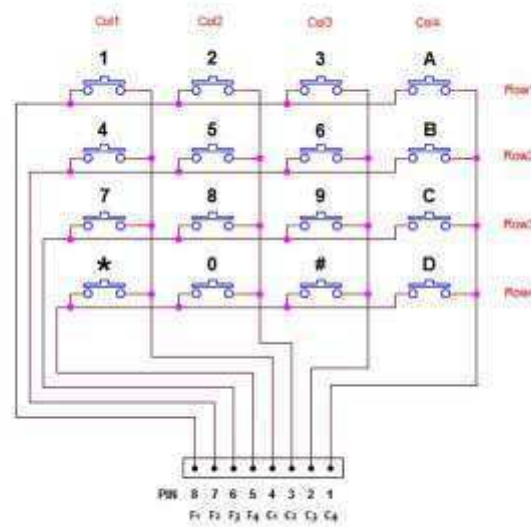


Figura 9: Teclado matricial dividido em linhas e colunas



Figura 10: teclado matricial.

O problema do teclado é que não se pode simplesmente ligar as 4 linhas e as 4 colunas do teclado ao Arduino acreditando que algo vai acontecer no fechamento de uma chave. Se fosse conectado como apresentado na Figura 11 o pino digital conectado ao teclado provavelmente ficaria flutuando, ou seja sem nível de tensão definido.



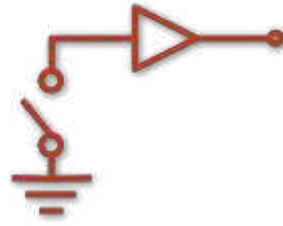


Figura 11: ERRO - O VALOR LIDO É INDETERMINADO.

Para que o nível de tensão no pino do Arduino não fique flutuando, é colocado um resistor de *pull-up* forçando uma tensão de 5 V na entrada, e o fechamento da chave força o valor para 0 V devido à ligação com o terra, como apresentado na Figura 12.

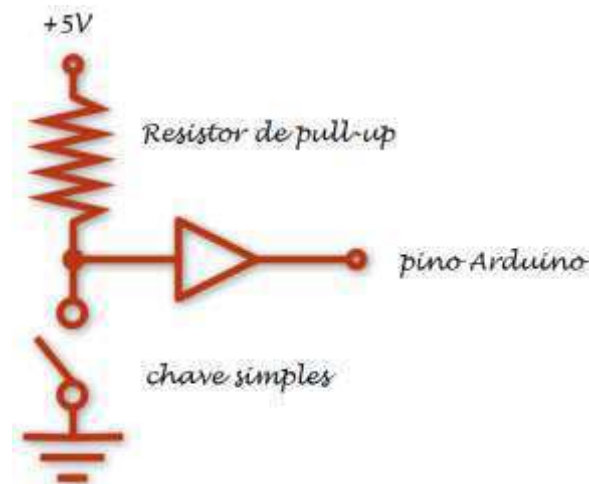


Figura 12: Resistor de *pull-up*.

No caso do Arduino, o resistor de *pull-up* pode ser adicionado via código da seguinte forma:

```
pinMode(n°, INPUT);
```

```
digitalWrite(n°, HIGH);
```

Quando um pino de entrada tem o valor *HIGH* escrito, um resistor de *pull-up* com valor de 20 k $\Omega$  é acoplado a entrada. Com isso, garante-se que o valor lido vai ser sempre 1 ou *HIGH* quando a chave estiver aberta e usa-se a chave fechada para forçar um valor 0 ou *LOW*.

### 3.2.4 MÓDULO DE RELÉS

O relé é um importante componente para controle de circuitos pois funcionará como uma chave controlada. Ele recebe um sinal de controle a nível TTL e aciona seus contatos eletromecânicos, podendo acionar uma carga de corrente alternada na tensão de

220 V e até 10 A, mantendo isolado de forma magnética o circuito de acionamento em tensão CC, do circuito da carga em tensão CA.

O circuito de acionamento de um relé é tal como apresentado na Figura 13. Para cada carga que se deseja controlar o acionamento, é necessário um novo circuito. O circuito U3 é uma proteção redundante para desacoplar opticamente o circuito de acionamento e o circuito da carga.

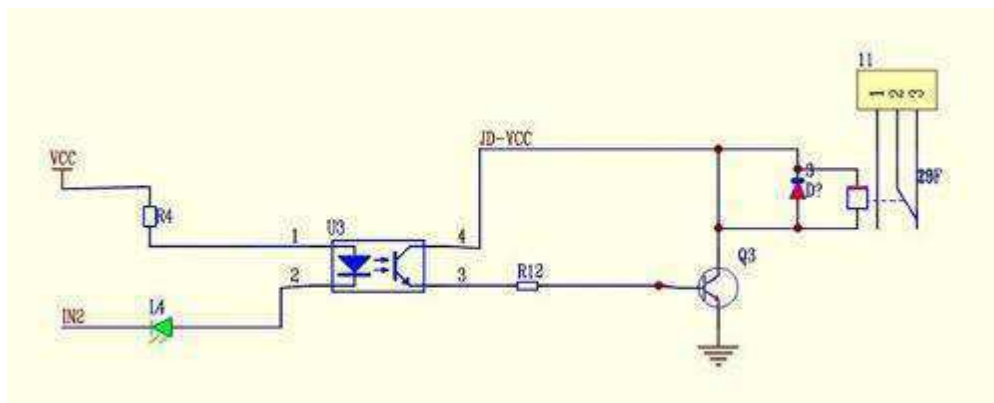


Figura 13: Circuito para acionamento de um relé.

Para este trabalho utilizou-se um módulo com 4 relés opto isolado comprado no mercado, como apresentado na Figura 14; e 2 relés utilizados separadamente.



Figura 14: Módulo com 4 canais optoisolados.

### 3.2.5 ARDUÍNO

O Arduino Mega2560 foi utilizado para fazer todo o controle e processamento dos dados de entrada e saída do sistema e ele já foi apresentado e explicado no Tópico 2.1.

## 4 DESENVOLVIMENTO E TESTES DO SISTEMA

Com o intuito de representar uma área de um hotel, fez-se o esboço para que uma empresa especializada em cortes a laser de MDF construísse o modelo. A Figura 15 apresenta a maquete desenvolvida ainda sem as ligações elétricas, fez-se a automatização de 5 cômodos, foram eles: sauna, sala de recepção, cofre, banheiro e área de serviços. Aqui se descreverá detalhadamente cada cômodo da maquete explicando também o processo de controle feito em cada área.

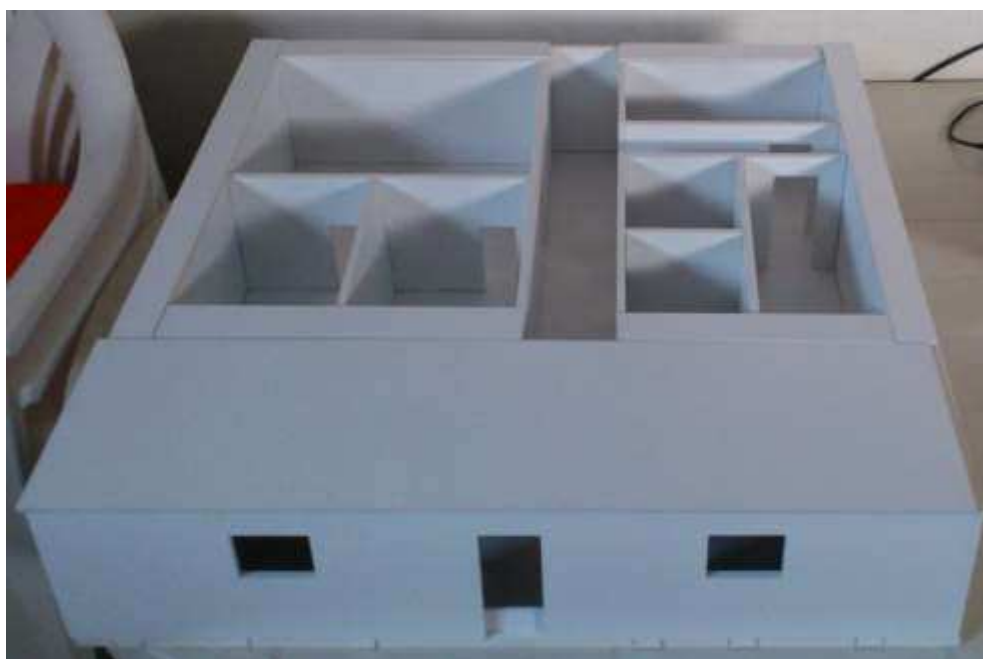


Figura 15: Maquete sem as ligações elétricas.

### 4.1 SALA DE RECEPÇÃO

Na sala de recepção foi feito o controle das luzes de acordo com a intensidade luminosa. Na Figura 16 é apresentada a sala de recepção. As luzes foram representadas pelas 6 fileiras de leds que são ligadas aos pares. Existem 4 estados de controle – muito iluminado, iluminado, escuro, muito escuro. Quando está muito iluminado todos os leds são desligados, para o estado de iluminado são ligados duas fileiras de leds, para escuro são ligados 4 fileiras de leds e para o estado de muito escuro são ligados as 6 fileiras de leds.



Figura 16: Sala de recepção.

O sensor de luminosidade utilizado foi um LDR e conectou-se a saída do sensor em uma entrada analógica do Arduino para que fosse feita a verificação da intensidade de luz. Colocou-se primeiramente um resistor de  $100\ \Omega$  em série com o LDR mas as tensões medidas estavam muito baixas, pois a resistência do LDR pode ser muito maior que  $100\ \Omega$ . Trocou-se por uma resistência de  $1\ M\Omega$  mas foi percebido o efeito contrário, pois o LDR possui resistência muito inferior a  $1\ M\Omega$ . Finalmente, colocou-se uma resistência de  $4,7\ k\Omega$  e contactou-se um bom comportamento.

As fileiras de leds são conectadas aos relés que por sua vez são controlados pelo Arduino. A alimentação dos leds são  $12V$  e do LDR são de  $5\ V$ . Os botões 4 e 5 do teclado matricial foram utilizados para ativar ou desativar o controle. Quando a tecla 4 é pressionada o Arduino recebe um sinal e faz com que o controle da sala seja ativado, caso a tecla 5 seja pressionada o controle é desativado e independentemente da intensidade luminosa os leds ficarão desligados.

## 4.2 COFRE

No cômodo do cofre foi feito o controle da presença de pessoas no recinto. O controle utilizou apenas um sensor de presença, o teclado matricial e um *buzzer*. Caso a tecla 3 seja pressionada o controle do cofre será ativado e se houver movimentos dentro da sala o sensor de presença detectará e o Arduino enviará o sinal para que o *buzzer* (alarme) seja ativado. Na Figura 17 se apresenta o cômodo.



Figura 17: Cofre.

### 4.3 SAUNA

Na sauna foi feito o controle da temperatura. Os botões 1 e 2 do teclado foram utilizados para ativar ou desativar o controle do sistema de aquecimento da sauna. Para representar o aquecimento utilizou-se 2 resistores de potência em série que quando ativados são alimentados por um fonte de 12 V, gerando um aquecimento. Utilizou-se um sensor de temperatura LM35 alimentado a 12 V. O transdutor possui um nível de tensão baixo na saída dificultando a leitura no arduíno, portanto foi utilizado um circuito amplificador não inversor com fator de multiplicação de 10x na saída do sensor, para o circuito utilizou-se o CI LM324 e resistores de 1,1 k $\Omega$  e 10 k $\Omega$ .

O Arduíno recebe a informação do sensor através de uma entrada analógica e verifica se a temperatura estará maior ou menor que 32°C. Se estiver menor enviará um informação para que o sistema de aquecimento seja ativado, caso contrário (>32°C) o sistema de aquecimento será desligado automaticamente para que haja um controle da temperatura. Os valores de temperatura podem ser ajustados pelo código do Arduino de acordo com a preferência do usuário. Na Figura 18 é mostrada a fotografia do cômodo Sauna.



Figura 18: Fotografia do cômodo Sauna.

#### 4.4 BANHEIRO

No banheiro foram feitos dois tipos de controle, o de presença de pessoas e o de economia de energia. O controle do sensor de presença é muito utilizado nos estabelecimentos comerciais (restaurante, lanchonete, lojas, etc.) e também visa a economia no consumo de energia. Quando entra uma pessoa neste cômodo, o sensor de presença detectará e enviará a informação para que o Arduino possa fazer o controle do relé que esteja conectado aos leds (iluminação).

Já o controle da economia de energia visa controlar o ligamento do chuveiro elétrico em horários de pico. Para isso foi utilizado um led representando o chuveiro elétrico. Para determinados horários do dia que o consumo de energia seja mais caro o chuveiro será desconectado da fonte fazendo com que não seja possível o seu ligamento (LED desligado). Já para os outros horários do dia é permitido o ligamento do chuveiro elétrico. O controle visa o estudo do racionamento energético em determinados horários, pois algumas concessionárias de energia já diferenciam as taxas de acordo com a faixa horária. Na Figura 19 apresenta a fotografia deste cômodo.



Figura 19: Fotografia do Cômodo Banheiro.

## 4.5 ÁREA DE SERVIÇO

No cômodo Área de Serviço é feito o controle do consumo energético de máquinas da área de serviço. Contudo, no caso deste trabalho será feito controle da máquina de lavar que é representada por um led para que seja observado a ação do sistema. O controle tem mesmo princípio de controle do chuveiro elétrico, fazendo com que para determinados horários de pico não seja possível a utilização da máquina de lavar. Na Figura 20 se representa o recinto.



Figura 20: área de serviços.

## 5 CONCLUSÃO

Este trabalho, teve como objetivo, desenvolver um sistema de controle predial utilizando microcontrolador. No trabalho fez-se o projeto dos circuitos, levantamento e compra do material utilizado, execução dos projetos e o esboço da maquete.

O microcontrolador utilizado foi o Arduino que se mostrou um ótimo dispositivo para controle e que sua utilização vem crescendo bastante ultimamente. O Arduino possui biblioteca *open-source* e apresenta vasta bibliografia de fácil acesso para estudo. Um ponto negativo deste microcontrolador utilizado é que o chip utilizado na placa é microSD, portanto não é facilmente removível da placa, dificultando sua utilização em aplicações no dia-a-dia.

Foi abordado neste trabalho de conclusão de curso vários conteúdos de diversas disciplinas, tais como: eletrônica, dispositivos eletrônicos, instalações elétricas, arquitetura de sistemas digitais. Este também visa o estudo para que no futuro seja implementado o conteúdo de automação de instalações prediais na disciplina da graduação de instalações elétricas.

Um dos pontos positivos da domótica é a modularidade que permitirá novos estudos no futuro para aperfeiçoamento deste sistema de controle. Atualmente existem vários *shields* para o Arduino que poderá ser implementado neste sistema. Uma próxima etapa para o aperfeiçoamento deste estudo seria a implementação de comunicação via WEB e *bluetooth*.



## BIBLIOGRAFIA

- [1] ARDUINO (Itália). **Arduino Mega 2560**. 2013. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em: 29 abr. 2014.
- [2] **Arduino - Teclado Matricial**. 2011. Disponível em: <[http://kemper.com.br/aulas/teclado\\_matricial/assets/fallback/index.html](http://kemper.com.br/aulas/teclado_matricial/assets/fallback/index.html)>. Acesso em: 14 abr. 2014.
- [3] McROBERTS, Michael. **Arduino Básico**. 1.ed. Tradução de Rafael Zanolli. São Paulo: Novatec, 2011, 456 p.
- [4] ROBÓTICA, Grupo. **Introdução ao Arduino**. Mato Grosso do Sul: UFMS, 2012, 25 p.
- [5] SILVEIRA, José Luiz Lopes da. **Introdução ao comando numérico**. Rio de Janeiro: Dem, 2010. p. 13-21.

# APÊNDICE A – CÓDIGO ARDUINO

```

/*****

```

```

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Laboratório de Alta Tensão
Trabalho de Conclusão de Curso

```

```

Orientador: Prof. Edson Guedes da Costa

```

```

Aluno: Ademir Cavalcante e Silva

```

```

*****/

```

```

#define sensorCofre 24      //Entrada digital
#define sensorTemperatura A1 //Entrada analógica
#define sensorLDR A4       //Entrada analógica do sensor de luminosidade
#define sensorBanheiro 22  //Entrada digital
#define rele1 28           //rele para ativar a luminosidade do banheiro
#define rele2 30           //rele para ativar fila 1 de leds da sala ---- baixo ativo
#define rele3 32           //rele para ativar fila 2 de leds da sala ---- alto ativo
#define rele4 34           //rele para ativar fila 3 de leds da sala ---- alto ativo
#define rele5 36           //rele para ativar o aquecimento da sauna
#define rele6 38           //rele para ativar o alarme do cofre
#define porta_aux_sauna 42 //Porta auxiliar para verificação do teclado
#define porta_aux_cofre 44 //Porta auxiliar para verificação do teclado
#define porta_aux_sala 46 //Porta auxiliar para verificação do teclado

#define led_chuveiro 41
#define led_servico 52

//Definição do teclado

#define linha_1 2
#define linha_2 3
#define linha_3 4
#define linha_4 5

#define coluna_1 6
#define coluna_2 7
#define coluna_3 8
#define coluna_4 9

//#define chuveiro 5

#include <Time.h>
#include <TimeAlarms.h>

//Definição de variáveis auxiliares

int val = 0;           //variavel auxiliar para o banheiro
int val2 = 0;         //variavel para leitura do sensor cofre
int aux_sala = 600;   //variavel auxiliar para a sala
int aux_sauna = 0;
int samples_sala = 0; //numero de amostras para leitura do sensor da sala
int samplesSauna = 0; //numero de amostras para leitura do sensor da sauna
int aux_sala_atual = 0;

```

```

int aux_sauna_atual = 0;
int teclado_sauna = 0;
int teclado_cofre = 0;
int teclado_economico = 0;
int flag_sauna = 0;
int porta_aux_cofre2 = 0;
int porta_aux_sauna2 = 0;
int porta_aux_sala2 = 0;

/*****Configurações de setup
inicial*****/

void setup()
{
  pinMode(led_chuveiro, OUTPUT);
  digitalWrite(led_chuveiro, LOW);
  pinMode(led_servico, OUTPUT);
  digitalWrite(led_servico, LOW);
  setTime(6,39,56,1,1,11); // set time to Saturday 8:29:00am Jan 1 2011
  // create the alarms
  Alarm.alarmRepeat(6,40,15, desligaChuveiro); // 8:30am every day
  Alarm.alarmRepeat(6,40,0,ligaChuveiro); // 5:45pm every day
  //Alarm.alarmRepeat(dowSaturday,8,30,30,WeeklyAlarm); // 8:30:30 every Saturday
  // Alarm.alarmRepeat(6,40,20,ligaChuveiro);

  pinMode(sensorBanheiro, INPUT);
  digitalWrite(sensorBanheiro, LOW);
  pinMode(sensorCofre, INPUT);
  digitalWrite(sensorCofre, LOW);
  pinMode(rele1, OUTPUT);
  digitalWrite(rele1, HIGH);
  pinMode(rele2, OUTPUT);
  digitalWrite(rele2, HIGH);
  pinMode(rele3, OUTPUT);
  digitalWrite(rele3, HIGH);
  pinMode(rele4, OUTPUT);
  digitalWrite(rele4, HIGH);
  pinMode(rele5, OUTPUT);
  digitalWrite(rele5, HIGH);
  pinMode(rele6, OUTPUT);
  digitalWrite(rele6, HIGH);

  pinMode(porta_aux_sala, OUTPUT);
  digitalWrite(porta_aux_sala, LOW);

  pinMode(porta_aux_sauna, OUTPUT);
  digitalWrite(porta_aux_sauna, LOW);

  pinMode(porta_aux_cofre, OUTPUT);
  digitalWrite(porta_aux_cofre, LOW);

  //Definicao da pinagem do teclado
  pinMode(linha_1, OUTPUT);
  pinMode(linha_2, OUTPUT);
  pinMode(linha_3, OUTPUT);
  pinMode(linha_4, OUTPUT);

  // Coloca as colunas como entradas e liga o resistor de pull-up
  pinMode ( coluna_1, INPUT );

```

```

digitalWrite ( coluna_1, HIGH );

pinMode ( coluna_2, INPUT );
digitalWrite ( coluna_2, HIGH );

pinMode ( coluna_3, INPUT );
digitalWrite ( coluna_3, HIGH );

pinMode ( coluna_4, INPUT );
digitalWrite ( coluna_4, HIGH );

}

/*****Função que se repete*****/
void loop()
{
  funcaoLerTeclado();      //Chama a função que ler o teclado
  funcaoCofre();          //Chama a função que ativa/desativa o alarme do cofre
  funcaoSensorBanheiro(); //Chama a função que Chama a função que ativa/desativa a iluminação do
banheiro
  funcaoSensorSala();     //Chama a função que controla a luminosidade da sala de recepção
  funcaoSensorSauna();    //Chama a função que ativa/desativa o sistema de aquecimento da sauna
  Alarm.delay(1000); // wait one second between clock display
}

/*****Definição da função que faz a leitura do
teclado*****/
/*
Descrição: esta função faz uma varredura em cada linha e em cada coluna do teclado matricial para
verificação de que tecla foi pressionada
*/

void funcaoLerTeclado(void){
  digitalWrite ( linha_1, LOW );
  digitalWrite ( linha_2, HIGH );
  digitalWrite ( linha_3, HIGH );
  digitalWrite ( linha_4, HIGH );

  // Testa as colunas uma por uma
  if ( digitalRead ( coluna_1) == LOW ) //tecla 1
  {
    digitalWrite(porta_aux_sauna, HIGH);
  }

  if ( digitalRead ( coluna_2) == LOW ) //tecla 2
  {
    digitalWrite(porta_aux_sauna, LOW);
    digitalWrite(rele5, HIGH);
  }

  if ( digitalRead ( coluna_3) == LOW ) //tecla 3
  {
    digitalWrite(porta_aux_cofre, HIGH);
  }

  if ( digitalRead ( coluna_4) == LOW ) //tecla A
  {
    digitalWrite(porta_aux_cofre, LOW);
    digitalWrite(rele6, HIGH);
  }
}

```

```

}

// Troca a linha em estado baixo

digitalWrite ( linha_1, HIGH );
digitalWrite ( linha_2, LOW );

// Testa as colunas uma por uma
if ( digitalRead ( coluna_1 ) == LOW ) //tecla 4
{
    digitalWrite(porta_aux_sala, HIGH);
}

if ( digitalRead ( coluna_2 ) == LOW ) //tecla 5
{
    digitalWrite(porta_aux_sala, LOW);
    digitalWrite(rele2, HIGH);
    digitalWrite(rele3, LOW);
    digitalWrite(rele4, LOW);
}

delay(100);
}

/*****Definição da função que faz a automatização do
banheiro*****/
/*
Descrição: Faz a leitura do sensor de presença e ativa ou desativa o relé da luminosidade
*/

void funcaoSensorBanheiro(void)
{
    val = digitalRead(sensorBanheiro);
    if(val == HIGH)
    {
        digitalWrite(rele1, LOW);
        delay(100);
    }
    else
    {
        digitalWrite(rele1, HIGH);
        delay(100);
    }
}

/*****Definição da função que controla o
cofre*****/
/*
Descrição: Verifica se a tecla 3 havia sido pressionada e verifica o sensor de presença detecta pessoas ou
não para ativar o alarme
*/

void funcaoCofre(void)
{
    porta_aux_cofre2 = digitalRead(porta_aux_cofre);
    if(porta_aux_cofre2 == HIGH){ //Se a tecla 3 for pressionada esta hipótese será verdadeira

```

```

val2 = digitalRead(sensorCofre);
if(val2 == HIGH)           //Se o sensor de presença detectar alguém, isto será verdadeiro
{
  digitalWrite(rele6, LOW);    //Ativa o rele6(alarme)
  delay(100);
}
else
{
  digitalWrite(rele6, HIGH);   //Desativa o rele6(alarme)
  delay(100);
}
}
else if(porta_aux_cofre == LOW){ //Se a tecla A for pressionada esta hipótese será verdadeira
  digitalWrite(rele6, HIGH);   //Desativa o Alarime
  delay(100);
}
}
}

```

```

/*****Definição da função do sensor da
sala*****/
/*

```

Descrição: Verifica se a luminosidade da sala está alta ou baixa e ativa ou desativa os leds de acordo com o sensor de luminosidade

```

*/

void funcaoSensorSala(void)
{
  porta_aux_sala2 = digitalRead(porta_aux_sala);
  if(porta_aux_sala2 == HIGH){ //Se a tecla 4 for pressionada esta hipótese será verdadeira
  if(samples_sala <= 30) //filtro para diminuir a interferência
  {
    aux_sala_atual = analogRead(sensorLDR) + aux_sala_atual;
    samples_sala = samples_sala + 1;
  }
  else{
    aux_sala = aux_sala_atual/samples_sala;
  }
  samples_sala = 0;
  aux_sala_atual = 0;

  delay(10);

  //aux_sala = analogRead(sensorLDR);
  if(820 < aux_sala && aux_sala <= 1023){ //muito iluminado
    digitalWrite(rele2, HIGH);
    digitalWrite(rele3, LOW);
    digitalWrite(rele4, LOW);
    delay(1000);
  }
  if(512 < aux_sala && aux_sala <= 820){ //iluminado
    digitalWrite(rele2, LOW);
    digitalWrite(rele3, LOW);
    digitalWrite(rele4, LOW);
    delay(1000);
  }
  else if(200 < aux_sala && aux_sala <= 512){ //escuro
    digitalWrite(rele2, LOW);
    digitalWrite(rele3, HIGH);
    digitalWrite(rele4, LOW);
  }
}

```

```

    delay(1000);
}
else if(0 <= aux_sala && aux_sala <= 200){ //muito escuro
    digitalWrite(rele2, LOW);
    digitalWrite(rele3, HIGH);
    digitalWrite(rele4, HIGH);
    delay(1000);
}
delay(200);
}

//rele 3 e 4 sao nivel alto ativo
//rele 2 eh nivel baixo ativo
else if(porta_aux_sala2 == LOW){ //Se a tecla 5 for pressionada esta hipótese será verdadeira
    digitalWrite(rele2, HIGH);
    digitalWrite(rele3, LOW);
    digitalWrite(rele4, LOW);
    delay(100);
}
}

void desligaChuveiro(){
    digitalWrite(led_servico, LOW);
    digitalWrite(led_chuveiro, LOW);
    delay(100);
}

void ligaChuveiro(){
    digitalWrite(led_servico, HIGH);
    digitalWrite(led_chuveiro, HIGH);
    delay(100);
}

/*****Definição da função que faz a automatização da
sauna*****/
/*
    Descrição: Verifica se a temperatura da sauna está abaixo ou acima de 32°C para ativar ou desativar o
    sistema de aquecimento
*/

void funcaoSensorSauna(void){ //falta testar esta funcao

    porta_aux_sauna2 = digitalRead(porta_aux_sauna);
    if(porta_aux_sauna2 == HIGH){ //Se a tecla 1 for pressionada esta hipótese será verdadeira e se a tecla
    2 for pressionada será falsa. Estas teclas funcionam como um interruptor digital liga/desliga.

    if(samplesSauna <= 30){ //Filtro
        aux_sauna_atual = analogRead(sensorTemperatura) + aux_sauna_atual;
        samplesSauna = samplesSauna + 1;
        delay(1);
    }
    aux_sauna = aux_sauna_atual/samplesSauna; //Média da leitura da temperatura
    aux_sauna_atual = 0;
    samplesSauna = 0;

    if(aux_sauna < 655){ //Temperatura Baixa (abaixo de 32°C)
        digitalWrite(rele5, LOW); //Liga o sistema de aquecimento
        delay(30000); //
    }
}

```

```
else if(aux_sauna >= 655){ //Temperatura Alta (acima de 32°C)
  digitalWrite(rele5, HIGH); //Desliga o sistema de aquecimento
}

delay(100);
}

}
```