unesp UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

Alan de Matos Lemos

Automação Residencial

Guaratinguetá
2016

Alan de Matos Lemos

Automação Residencial

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação da Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador: Prof. Dr Samuel E. De Lucena

Guaratinguetá
2016

Lemos, Alan de Matos
L557a Automação residencial / Alan de Matos Lemos – Guaratinguetá, 2017.
54 f : il.
Bibliografia: f. 43

Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2017.
Orientador: Prof. Dr. Samuel Euzédice de Lucena

1. Automação residencial. 2. Circuitos elétricos. 3. Engenharia elétrica. I. Título.

CDU 681.3.068

Luciana Máximo
Bibliotecária/CRB-8 3595

Unesp

UNIVERSIDADE ESTADUAL PAULISTA

Faculdade de Engenharia do Campus de Guaratinguetá

Automação Residencial

ALAN DE MATOS LEMOS

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE **GRADUADO EM ENGENHARIA ELÉTRICA**

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Prof. Dr Leonardo Mesquita

Coordenador

BANCA EXAMINADORA:

Prof. Dr. Samuel Euzédice. De Lucena

Orientador/UNESP-FEG

Prof Dr. Teófilo Miguel de Souza

UNESP-FEG

Prof Dr. Fernando Ribeiro Filadelfo

UNESP/FEG

Dezembro 2016

ALAN DE MATOS LEMOS

NASCIMENTO	19.01.1990 - São Bernardo – SP
FILIAÇÃO	Inês de Matos Lemos Camilo de Jesus Almeida Lemos
2009/2016	Curso de Graduação em Engenharia Elétrica Universidade Estadual Paulista – "Júlio de Mesquita Filho", Campus de Guaratinguetá.

AGRADECIMENTOS

Agradeço primeiramente e especialmente à minha querida família, minha mãe, meu pai, minha irmã e meus avos, que me educaram, apoiaram e me ajudaram a superar todas as dificuldades que tive que enfrentar na vida, sem eles jamais estaria onde estou.

Amo vocês.

Gostaria de agradecer também aos meus amigos/irmãos do meu prédio (Maison) e agregados, que já são mais de 23 anos, com muitas histórias e risadas, e que com certeza será uma amizade eterna. Gostaria de agradecê-los também pela dedicada ajuda na confecção do protótipo aonde foram dias de muito trabalho.

Gostaria de agradecer à família Susexo que me acolheu e me ensinou muitas coisas, compartilhou o sofrimento de "virar a noite" estudando e também às alegrias de poder ir para festas e churrascos por Guaratinguetá, que me aguentou por bastantes anos e aguentará por muitos mais.

Gostaria de agradecer a república La Granja, Damas de Copos e 3 Pinheiros que me acolheram em Campinas em um período cheio de viagens e estresse, e ainda me recebem de braços abertos com muito carinho.

Gostaria de agradecer aos meus colegas de trabalho da Continental, esse ano aprendi muitas coisas que não se vê na faculdade e vocês tiveram a paciência e boa vontade de me ensinar do melhor jeito possível sempre com muito humor.

Gostaria de agradecer aos meus mestres da Faculdade de Engenharia de Guaratinguetá, pois ensinaram com competência e dedicação e sou muito grato por passarem toda sua experiência para nos alunos.

E por fim gostaria de agradecer meus colegas de classe, esses realmente sabem na pele o que foi estudar em uma Faculdade do nível da Unesp, passando dias e noites estudando, um colaborando com o outro, conseguimos alcançar o tão sonhado objetivo.

Um Beijo e um Abraço a todos.

RESUMO

Este trabalho de graduação apresenta uma implementação de um sistema de automação residencial em um esquema representando uma residência. Esse sistema utilizará como central de automação o Placa Arduino Uno, que é capaz de controlar diversos processos dentro de uma residência conforme o usuário deseja. Um sistema que interage com o Android do usuário. Abordaremos também questões de boas práticas, equipamentos e procedimentos necessário para essa implementação. Para esse projeto foram desenvolvidos circuitos com módulo Modulo Rele Driver 4 Canais DC 5V, com os componentes Arduino Ethernet Shield R3, Modulo Rele Driver 4 Canais DC 5V, Motor Servo (Cys model S0009 analog servo), um *cooler* de 12V, buzina sensor LDR, sensor de temperatura LM35 e um modulo I2C, sendo esse dispositivo responsável pela economia de saídas vindas do LCD.

PALAVRAS-CHAVE: Automação Residencial. Arduino UNO. Android,Ethernet.

ABSTRACT

This graduation work will present an implementation of a home automation system in a scheme representing a residence. This system uses a central automation Arduino Uno board, which is capable of controlling several processes within a residence as the user wants. A system that interacts with the user's Android. We will also explore issues of good practices, equipment and procedures necessary for its implementation.
For this project were developed circuits Module Relay Driver 4 Channels 5V DC, module with Arduino Ethernet Shield R3, components Modulo Relay Driver 4 Channel DC 5V, Servo Motor (Cys model S0009 analog servo), A 12V cooler, LDR sensor horn, LM35 temperature sensor and an I2C module, this device is responsible for saving outputs from the LCD.

KEYWORDS: Residential Automation. Arduino UNO. Android.

SUMÁRIO

1 INTRODUÇÃO

1.1 ASPECTOS GERAIS

O nosso planeta vem enfrentando sérias crises energéticas ao longo dos últimos anos, nós somos desafiados diariamente a desenvolvermos métodos para se conseguir um melhor resultado no que se diz respeito a gestão energética, a cada dia a escassez de energia e o seu alto custo vem fazendo com que os cidadãos mais comuns percebam a necessidade de se realizar uma gestão energética dentro de suas próprias casas.

A partir dos anos 80, a automação passou a ser utilizadas em residências e prédios proporcionando diversos benefícios como: segurança, conforto pessoal e economia de energia (Finder, 2011). Quando a domótica surgiu, com os primeiros edifícios na década de 80 do século passado, pretendia-se controlar a iluminação, as condições climáticas, a segurança e a interligação entre os três elementos. Atualmente, a ideia base continua a mesma (ALIEVI, 2008).

Assim, uma importante ferramenta que nos auxilia nessa economia de energia é a chamada automação residencial, que nada mais é que a utilização dos conhecimentos em automação para trazer benefícios ao seu usuário, seja em relação ao conforto, ou no que diz respeito ao gerenciamento de energia dessa residência. Nesse contexto, a eficiência energética se apresenta como uma das soluções que agregam mais benefícios, tanto ambientais, como econômicos e sociais. (PROCEL, 2012).

A automação residencial existe há muitos anos, ainda assim, esse mercado é considerado um artigo de luxo pela maioria dos consumidores, muitos problemas são encontrados no que diz respeito a automação industrial, a falta de padronização dos equipamentos talvez seja o problema que mais influência para a difusão da automação em residências.

Segundo Bortoluzzi (2013), “a década de 70 pode ser considerada o marco inicial da automação residencial, quando foram lançados nos EUA, os primeiros módulos inteligentes chamados X-10”. O protocolo X-10 foi projetado para um controle remoto de alguns dispositivos, utilizando a própria rede elétrica como canal de comunicação.

Alguns anos depois, já na década de 80, com a difusão dos computadores pessoais (PC’s), começou a se desenvolver uma ideia em que o PC poderia ser a central de automação, mas havia uma séria desvantagem, no que se diz respeito ao consumo energético, uma vez que o PC deveria ficar sempre ligado. Foi a partir desse raciocínio que se desenvolveram

dispositivos embarcados que substituíssem, através da utilização de microcontroladores e microprocessadores.

Desde esse período foram sendo acrescentados uma série de equipamentos de comunicação com a central de automação. Com a popularização da internet de banda larga, essa tecnologia passou a ser altamente explorada, possibilitando também técnicas de monitoramento não só presenciais, mas também à distância, sendo possível dessa forma controlar a residência de uma "*Web Page*" (CRUZ,2009). Ao final da década de 90, com o advento da telefonia celular e da internet, houve um despertar ainda maior dos consumidores pelo conforto, facilidades estas que, expuseram uma maior comodidade e economia de tempo em atividades cotidianas (PAIVA, 2007). Com o passar do tempo, outros meios, como o "*Bluetooth"*, padrão de comunicação desenvolvido para integração entre celulares e periféricos, também foram incorporados, como objetivo de, por exemplo, controlar lâmpadas à pequenas distâncias (SILVA, B. C. R. & CÂNDIDO, L. A. A, 2011).

1.2 OBJETIVOS GERAIS

O objetivo desse trabalho é realizar a implementação da automação residencial em um sistema que seja similar a uma residência familiar, realizando alguns processos residenciais, como por exemplo: controle de lâmpadas, alarme, controle motorizado, sensores de temperatura e ar condicionado sendo que esse controle será realizado por meio de um *Smarth phone* que possua sistema operacional do tipo Android.

1.3 OBJETIVOS ESPEFÍFICOS

Com o objetivo de melhorar o conforto dos integrantes de uma família, obter o controle a distância das funções de acender e apagar as luzes.

A segunda aplicação que será desenvolvida no projeto é o controle do portão da garagem da casa, essa aplicação irá trazer mais conforto e segurança aos moradores dessa residência, uma vez que com o controle a distância desses processos a vida desses moradores se torna mais cômoda.

Outras aplicações que serão utilizadas na casa são: o controle de temperatura interna da residência, por meio de um ar condicionado, o que contribui significativamente para o bem-estar e conforto da família.

Outro ponto importante, é o sistema de alarme que contribui de forma significativa para a segurança da casa, notificando a violação da residência para o usuário por meio do aplicativo *Inventor APP*.

1.4 ESTRUTURA DO TRABALHO

O primeiro capítulo como vimos anteriormente, trata de uma introdução básica sobre o tema e o projeto desenvolvido para a transcrição deste trabalho de graduação, nele abordamos alguns aspectos sobre a automação residencial, nesse primeiro capítulo também são apresentados os objetivos deste trabalho, o objetivo geral e o objetivo específico, dando uma ênfase à implementação do projeto em uma residência familiar. Para finalizar o primeiro capítulo abordaremos os aspectos da metodologia utilizada neste trabalho de graduação.

No segundo capítulo desse trabalho iremos abordar os conceitos teóricos do tema, nele será apresentado toda a fundamentação teórica necessária para o desenvolvimento deste trabalho de graduação, nesse capítulo também iremos abordar alguns esclarecimentos sobre o funcionamento e composição de alguns componentes que foram utilizados não implementação do projeto.

O terceiro capítulo é mostrada a implementação do projeto em uma maquete representando uma residência familiar.

E no quarto e último capítulo deste trabalho encontra-se as discussões e resultados obtidos na implementação do projeto. Nesse capítulo são apresentadas as dificuldades encontradas nessa implementação.

1.5 METODOLOGIA

Este trabalho foi dividido em duas etapas descritas abaixo:

1º Etapa: A descrição dos componentes utilizados no projeto.

2º Etapa: A montagem, o teste e a validação do protótipo, que contém a automação.

Na primeira parte desse trabalho abordaremos os componentes mais importantes para a realização do protótipo em questão. Nesse capitulo iremos descrever detalhadamente os funções e características de cada componente, também abordaremos os códigos de programação que os dispositivos necessitam para serem utilizados de uma forma correta.

Na segunda parte desse trabalho iremos abordar a descrição do desenvolvimento do projeto do protótipo propriamente dita, descrevendo os procedimentos utilizados para a construção do protótipo, com ilustrações do projeto.

2 MATERIAIS E MÉTODOS

2.1 ARDUINO UNO

A placa Arduino UNO já está em sua terceira revisão, essa placa possui apenas duas camadas e várias características interessantes de projeto. Essa placa pode ser alimentada por uma conexão USB ou por uma fonte de alimentação externa. Essa alimentação é realizada através de um conector Jack com positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites de 6V a 20V.

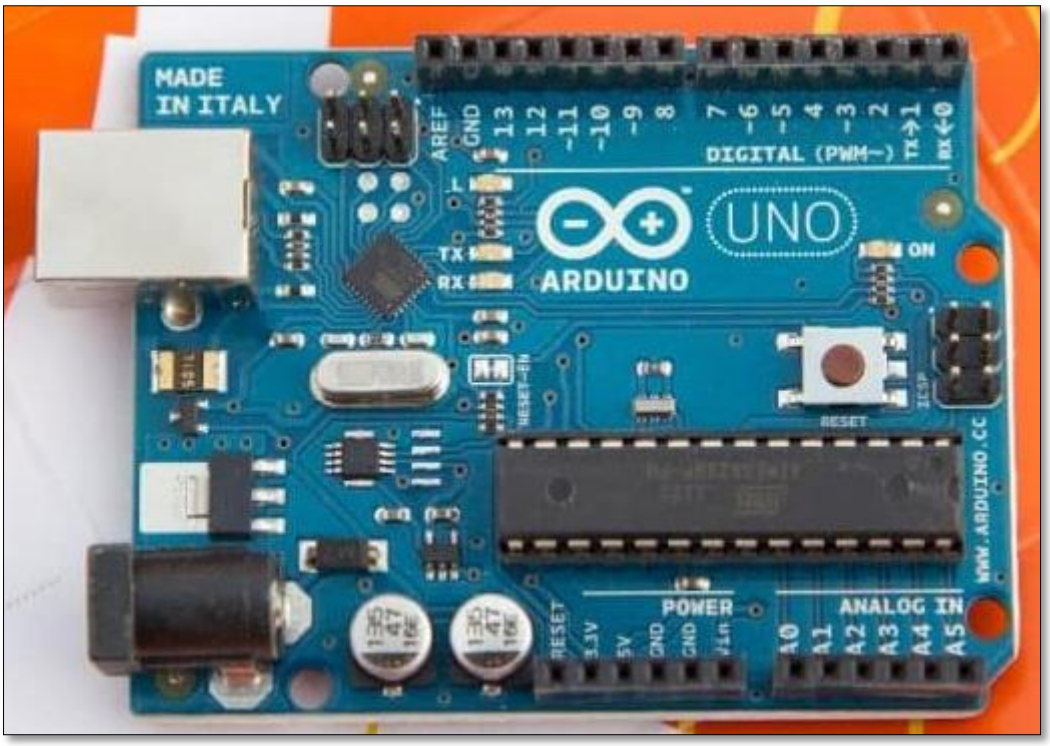
O componente principal da placa é o microcontrolador ATMEL ATMEGA17328, um dispositivo de 8 bits da família AVR com arquitetura RISC avançada e com encapsulamento DIP28. Contando com 32KB de *Flash*, 2KB de RAM e 1 KB de EEPROM. Podendo operar em até 20MHz, porém na placa Arduino UNO opera em 16 MHz.

Este microcontrolador pode operar com tensões baixas, de até 1,8V, mas nessa tensão apenas opera até 4MHz, possuindo dois modos de consumo bem baixos, *Power-down Mode* e o *Power-save Mode*, para que o sistema possa poupar energia em uma situação de espera. Possui, como periféricos uma USART que funciona a até 250kbps, uma SPI, que vai a até 5MHz, e uma I2C que pode operar até 400kHz. Conta com um comparador analógico interno ao CI e diversos *timers*, além de 6 PWMs. A corrente máxima por pino é de 40mA, mas a soma da corrente de todo o CI não pode ultrapassar 200mA. Ele possui um oscilador interno de 32kHz que pode ser utilizado, por exemplo, em situações de baixo consumo.

Segundo McRoberts (2001), o Arduino é o que chamamos de plataforma de comunicação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. Assim, o Arduino é aplicado em muitos projetos no mundo da eletrônica, sendo possível o controle de uma série de dispositivos eletrônico.

A figura 1 abaixo, nos mostra o esquema da placa Arduino UNO.

Figura 1 - Placa Arduino Uno



Fonte Embarcados (2016)

É importante lembrar que a placa Arduino não possui a facilidade de "*debugar"* "*breakpoints"*, consultar variáveis ou mesmo parar o "*firmware"* em tempo real para conferir endereços de memória ou variáveis. Ainda sobre a placa Arduino, ao término desse trabalho encontra-se o anexo contendo circuito de comunicação serial da placa Arduino UNO.

O Arduino foi desenvolvido por uma empresa italiana em meados do ano de 2005, criado pelo professor Massimo Banzi, afim de ensinar os seus alunos sobre eletrônica e programação de dispositivos. O professor, com auxílio de David Cuartielles, decidiu criar sua própria placa, com a ajuda de um aluno do professor Massimo, David Mellis, que ficou responsável por criar a linguagem da programação do Arduino (BOEIRA,2013).

Uma característica que nos chama bastante atenção nesse componente eletrônico é o fato do Arduino fornece uma certa liberdade ao usuário. O *software* e o *hardware* são completamente livres, de modo que qualquer pessoa possa ter acesso. Inclusive qualquer usuário pode desenvolver seu próprio Arduino.

Um outro ponto a favor da placa Arduino é o fato da existência de vários "*Shields"* que permitem aumentar a capacidade do seu sistema ou ressaltar uma aplicação desejada.

2.2 ETHERNET SHIELD W5100

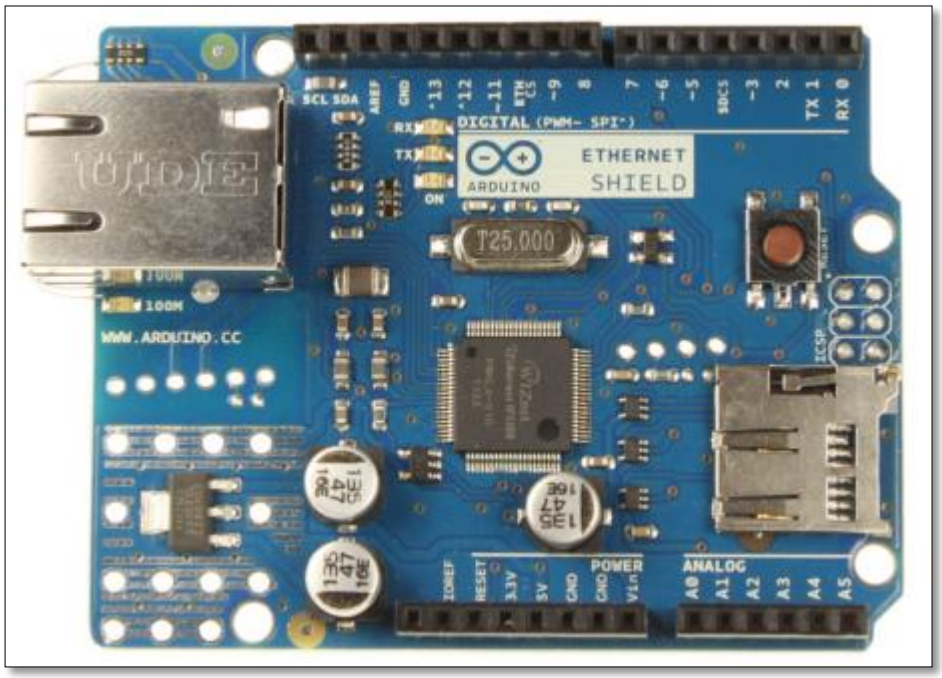
O Arduino Ethernet "*Shield"* é responsável por conectar seu Arduino à "*internet"*.

Basta ligar este módulo em sua placa Arduino, conectá-lo à sua rede com um cabo de rede RJ45. Com o chip Ethernet Wiznet W5100. O Wiznet W5100 fornece uma rede (IP). (ARDUINO-2, 2013)

O módulo pode suportar até quatro conexões de soquete simultâneos. Usando a biblioteca Ethernet para escrever esboços que se conectam à internet usando o *shield*. O *shiel*d *ethernet* conecta a uma placa Arduino usando longos cabeçalhos fio-*wrap* que se estendem através do *shield*. Isso mantém a aparência de pino intacto e permite que outro *shield* para serem pilhados em cima.

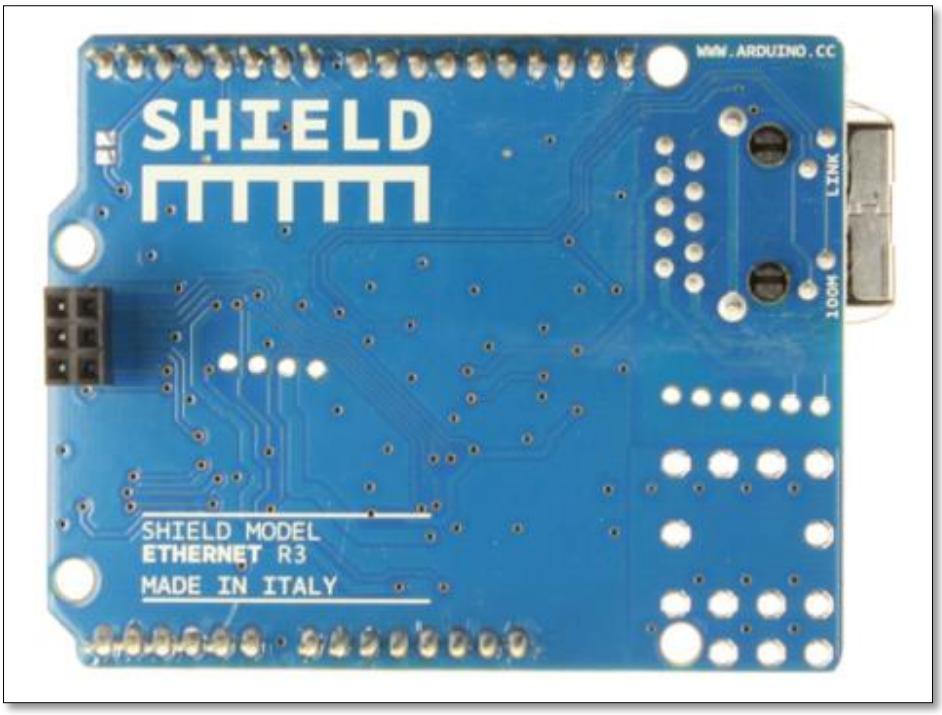
Nas imagens 2 e 3, que seguem a baixo temos o esquema frontal e da parte de trás da placa Arduino Ethernet Shield R3.

Figura 2 - Arduino Ethernet Shield R3 Front.



Fonte: Arduino (2016)

Figura 3 - Arduino Ethernet Shield R3 Back



Fonte: Arduino (2016)

Há um "*slot"* para cartão micro-SD integrado, que pode ser usado para armazenar arquivos para servir através da rede. É compatível com todas as placas Arduino / Genuino. O leitor de cartão micro SD "*on-board"* é acessível através da Biblioteca SD. Ao trabalhar com esta biblioteca, SS é no Pino 4. A revisão original do *shield* continha um "*slot"* para cartão SD de tamanho completo; isso não é suportado.
O escudo também inclui um controlador de reposição, para garantir que o módulo W5100 Ethernet está devidamente repor em "*power-up"*.

Para se montar o sistema Ethernet "*Shield"* e o Arduino UNO, basta encaixar os terminais e ligar o cabo de rede do roteador na entrada RJ45 do "*Shield"*, logo abaixo temos uma ilustração do sistema ja montado.

Na figura 4 abaixo, podemos ver como essa montagem do sistema fica ao final do encaixe entre a placa e o "*shield"*.

Figura 4 - Arduino UNO e Ethernet Shield conectados

Fonte: Arduino e Cia (2016)

2.3 RELE DRIVER 4 CANAIS DC 5V

O módulo Relé 5V com 4 canais é uma ótima opção para quem precisa fazer vários acionamentos com a facilidade de ter tudo em apenas uma só placa, de forma confiável, compacta e robusta. Com este módulo Relé 5V é possível acionar cargas de 220V AC, como lâmpadas, equipamentos eletrônicos e motores com o auxílio de um microcontrolador Arduino, PIC, ARM. Tudo isso sem a necessidade de montar um circuito com transistores, relés, conectores, *leds* e diodos.

As especificações do componente são:

-Modelo: SRD-05VDC-SL-C

-Tensão de operação: 5VDC

-Permite controlar cargas de 220V AC

- Corrente típica de operação: 15 à 20mA

- LED indicador de status

- Pinagem: Normal Aberto, Normal Fechado e Comum

- Tensão de saída: (30 VDC a 10A) ou (250VAC a 10A)

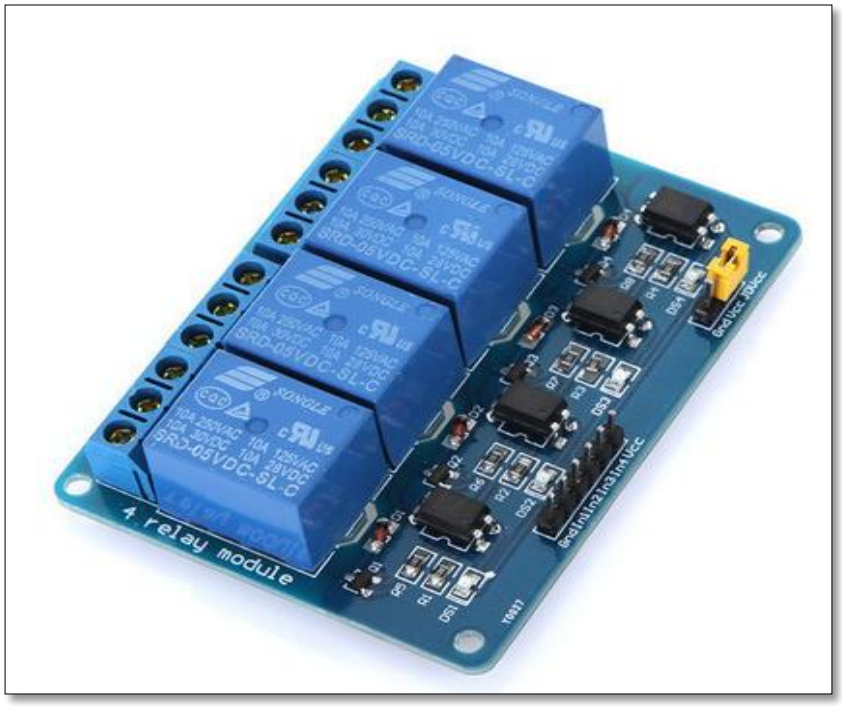
- Furos de 3mm para fixação nas extremidades da placa

- Tempo de resposta: 5 à 10ms

- Dimensões: 135 x 52 x 20mm

A figura 5 abaixo nos mostra o esquema do Módulo Rele Drive de 4 canais, ainda encontramos o seu "*Datasheet"* em anexo ao final desse trabalho.

Figura 5 - Modulo Rele Driver 4 Canais DC 5V.



Fonte: Arduino e Cia (2016)

2.4 MOTOR SERVO (CYS MODEL S009 ANALOG SERVO)

O motor servo é uma máquina, eletromecânica, que apresenta um movimento que é proporcional a um comando ele recebe um sinal de controle, que verifica a posição inicial para o controle do seu movimento, se deslocando para a posição desejada com a sua velocidade sendo monitorada por um dispositivo denominado taco ou sensor de efeito Hall. O motor servo que será utilizado em nosso projeto é o Motor Servo (Cys model S0009 analog servo), o mesmo ilustrado na figura 6, logo abaixo.

Figura 6 - Motor Servo (Cys model S0009 analog servo)

Fonte: Arduino e Cia (2016)

2.5 MÓDULO I2C

O modulo I2C é muito utilizado para projetos que envolvem telas de LCDs, podendo também estar presente em projetos que envolvam o microprocessador Arduino ou outros microprocessadores. O modulo I2C foi desenvolvido pela Philips (atualmente NXP). Sempre visando conectar diversos dispositivos utilizando apenas as duas linhas de dados, a (SDA e a SCL) *Serial Data* e *Serial Clock*.

De forma análoga aos dispositivos mencionados anteriormente o módulo I2C, requer uma programação básica para sua utilização. Na programação abaixo temos como exemplo como fazer com que a temperatura captada pelo sensor LM35 apareça em uma tela de LCD.

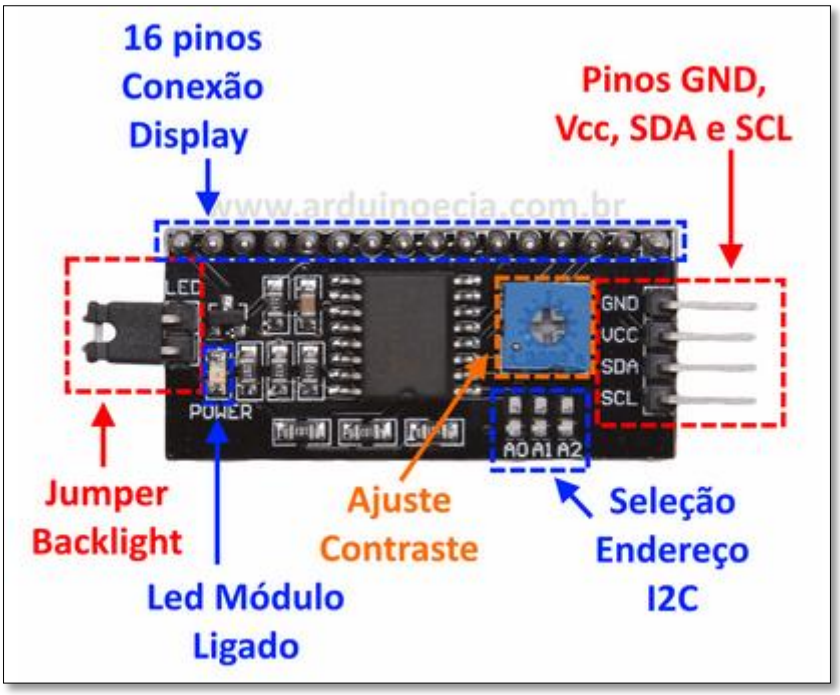
Figura 7 - Programação para leitura de temperatura no LM35 com o I2C.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int LM35 = 0;
float temp = 0;
int LM35_ADC = 0;

byte custom[8] = {
  0b00111,          // Caractere customizado
  0b00101,
  0b00111,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};

LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  Serial.begin(9600);
  analogReference(INTERNAL);  // Muda a referência de 5V para 1,1V
  lcd.init();
  lcd.backlight();
  lcd.createChar(5, custom); // Cria nosso caractere definindo-o como o byte 5

}

void loop()
{
  LM35_ADC = analogRead(LM35);   // Lê o valor no pino A0
  temp = LM35_ADC * 0.1075268817;    //Transforma o nº analógico para ºC
  lcd.print("  Temp = ");
  lcd.print(temp);
  lcd.print(" C");
  lcd.home(); // Seta o cursor para o inicio caracter 0, na linha 0
  lcd.print("->");
  lcd.setCursor(14,0); // Seta o cursor para o caracter 14, na linha 0
  lcd.write(5); // Imprime o byte 5(nosso caracter custom)
  delay(1000);
  lcd.clear(); //Limpa a tela do LCD
}
```

Fonte: Arduino (2016)

Na figura 8 abaixo temos um esquema da estrutura do módulo I2C, onde pode-se perceber que esse módulo possui 4 pinos, sendo que dois são para alimentação, e os outros dois para interface I2C (SDA e SCL), o potenciômetro da placa é utilizado para o ajuste de contraste do display, e o *jumper* na lateral permite que a luz de fundo seja controlada pelo programa ou fique apagada.

Figura 8 - Estrutura do módulo I2C



Fonte: Arduino e Cia (2016)

2.6 SENSOR DE TEMPERATURA LM35

O sensor de temperatura LM35 é um sensor de precisão em graus centígrados e esse componente possui uma voltagem de saída analógica, com uma faixa de medição de -55ºC a 150ºC, sendo que a precisão é de 0,5ºC. Sendo a tensão de saída de 10mV / ºC. A saída desse componente pode ser conectada a saída de diversos microcontroladores.

Esse sensor tem um funcionamento muito básico, a cada grau Celsius representa 10mV na saída do componente.

O LM35 pode ser utilizado em qualquer dispositivo de medição de temperatura, podendo ser um circuito simples ou até mesmo um microprocessador.

O microprocessador Arduino e o Sensor de Temperatura LM35 trabalham excepcionalmente bem e facilmente manipulável, bastando ligar o pino 1 do LM 35 n conector de +5V da placa do Arduino, o pino 2 do LM35 se conecta a porta A0, e o pino 3 que é o negativo vai conectado ao GND do Arduino.

Após essa instalação básica, devemos mandar o código de programação para o Arduino e o mesmo vai medir a temperatura em tempo real.

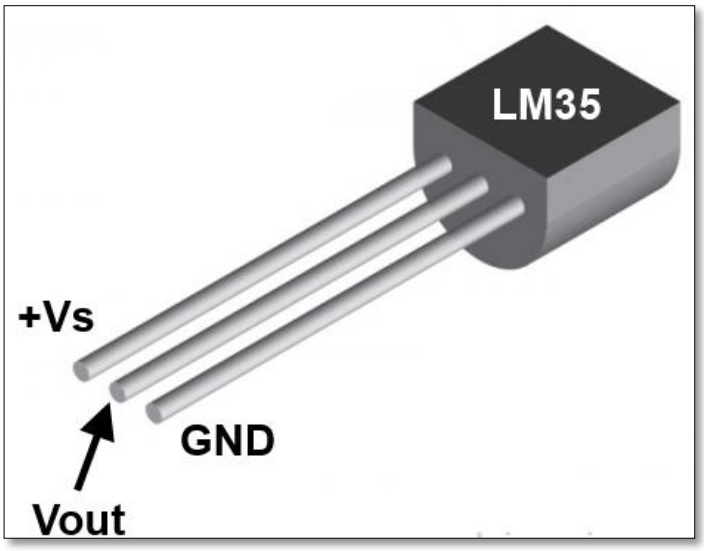
Figura 9 - Código para o sensor LM35 ler a temperatura em uma placa Arduino

```
float temp;
int tempPin = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  temp = analogRead(tempPin);
  temp = temp * 0.48828125;
  Serial.print("Temperatura agora = ");
  Serial.print(temp);
  Serial.print("°C");
  Serial.println();
  delay(1000);
}
```

Fonte: Arduino e Cia (2016)

O sensor de temperatura LM35 pode ser considerado um sensor de precisão, fabricado pela *National Semiconduct*, na figura 10 abaixo temos uma ilustração do sensor.

Figura 10 - Sensor de temperatura LM35.



Fonte: ARDUINO E CIA (2016)

2.7 SENSOR DE LUMINOSIDADE LDR

O sensor de luminosidade LDR, é um sensor sensível à luz que conforme é alterado a intensidade luminosa, sua resistência varia, conforme é incidido sobre ele luz, sua resistência diminui para alguns Ohms, e quando a intensidade luminosa sobre ele diminui, a sua resistência aumenta para alguns mega Ohms.

Com essa característica esse sensor de luminosidade é muito utilizado em projetos de automação residencial, utilizado para detecção da luminosidade ambiente e para a tomada decisão de acender ou apagar uma lâmpada.

Com uma instalação simples como a do Sensor de Temperatura LM35, bastando conecta-lo a placa Arduino e com um simples código de programação, o sensor de luminosidade LDR já entra em funcionamento.

Figura 11 - Programação para utilização do Sensor de Luminosidade LDR

```
/*
 * Entrada analógica
 * le o valor do sensor de luminosidade e envia valor para o computador
 */

int sensorPin = 0;    // selecione o pino de entrada ao potenciômetro
int val;

void setup() {
 Serial.begin(9600);
}

void loop() {
 val = analogRead(sensorPin);    // ler o valor do potenciômetro
 Serial.println(val);            //emvia valor para o pc
 delay(1000);                    //aguarda 1 segundo
}
```

Fonte: ARDUINO E CIA (2016)

Na figura 12 abaixo, temos a imagem do sensor de luminosidade LDR.

Figura 12 - Sensor de Luminosidade LDR

Fonte: Arduino (2016)

É importante mencionarmos que um LDR é um feixe no infra-vermelho de entrada que converte a luz em valores de resistência, ele é constituído de sulfeto de cádmio.

3 DESENVOLVIMENTO DO PROJETO

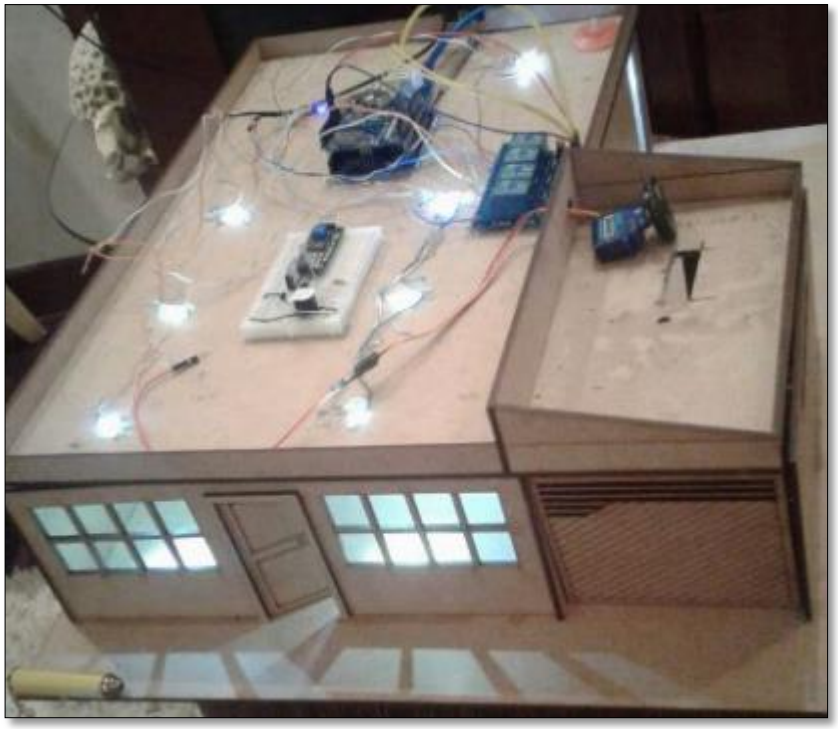
O projeto foi desenvolvido com relés de impulso que são controlados por um microprocessador Arduino, os relés são parte da estrutura de automação do protótipo, facilitando a instalação do módulo microcontrolador. O Arduino é o responsável pela centralização e comandos de cargas de tensão.

O Arduino Ethernet Shield, quando conectado com o Arduino, fez a conexão com a internet para que o usuário através de um celular com o aplicativo consiga ter acesso ao controle do protótipo remotamente.

Primeiramente foi configurado o módulo Ethernet com exemplos da biblioteca do Arduino, aonde se configura o IP e portas do roteador.

A partir desse passo verifica-se o funcionamento do módulo através de uma página criada na internet, logo após foi conectado o módulo relé e realizado o teste, como ilustrado na figura 13 na próxima página.

Figura 13 - Teste das cargas iniciais

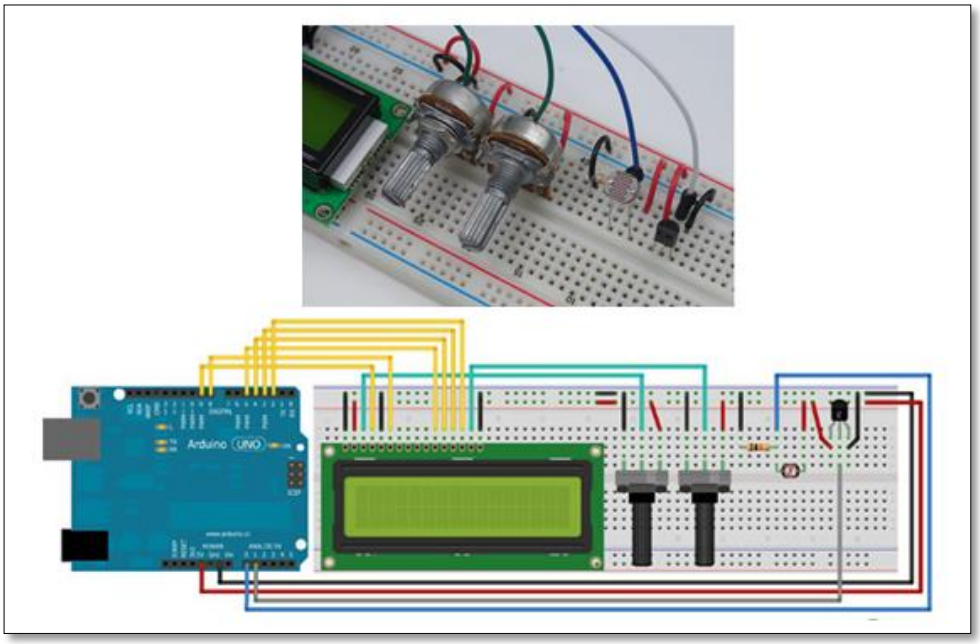


Fonte: Autor

A partir desse passo, foram testados os módulos de temperatura (LM35) e de luminosidade (LDR). Sendo configurados os módulos de temperatura e de luminosidade por meio da IDE do Arduino.

A figura 14 abaixo nos mostra o circuito conectado do sensor de temperatura.

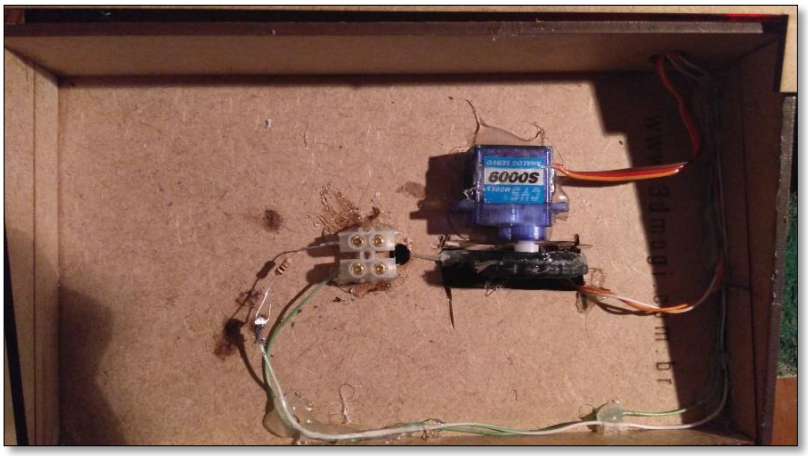
Figura 14 - Circuito do sensor de temperatura



Fonte**:** ARDUINO E CIA (2016)

Após a constatação do funcionamento dos módulos citados anteriormente, implementamos no protótipo a configuração do servo motor, que será responsável pela abertura e fechamento do portão do protótipo.

Figura 15 - Implementação do servo motor



Fonte: Autor

A figura abaixo nos ilustra o portão do protótipo fechado.

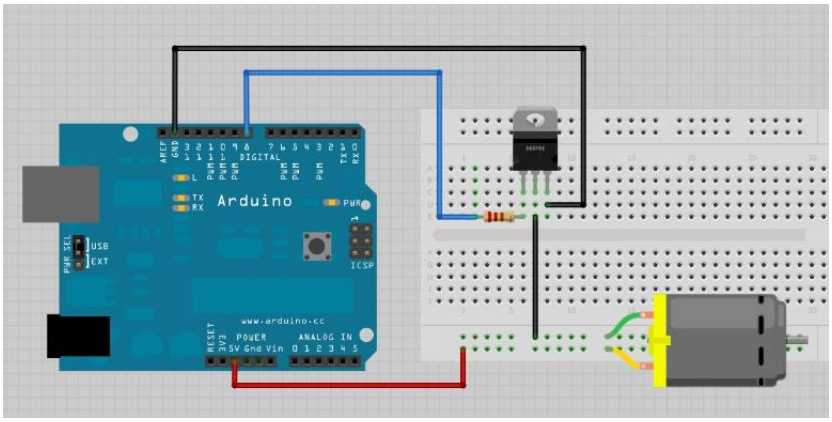
Figura 16 - Portão do protótipo

Fonte: Autor

Com a ajuda das bibliotecas do Arduino, é possível realizar a configuração do LCD e também foram realizados testes nessa implementação.

Como o Arduino não aguenta altas correntes, foi necessário a implementação de um circuito utilizando um transistor de potência (TIP 122) para o acionamento de um cooler de 12V simulando um ar condicionado.

Figura 17 - Transistor de potência



Fonte: ARDUINO E CIA (2016)

Com o intuito de desenvolver um sistema de alarme, também foi conectado uma buzina de 5V que será acionada todas as vezes em que o sensor de luminosidade atingir um valor pré-determinado pelo usuário.

Na implementação de todos os módulos juntos percebe-se que faltavam portas digitais, para todos os módulos. Para solução implementamos um módulo I2C que é responsável por

reduzir o número de portas consumidos pelo LCD, com isso é possível reduzir a utilização de 10 digitais para 2 portas analógicas. Assim, pode-se ligar todos os módulos simultaneamente.

A figura abaixo nos mostra o display do projeto em funcionamento.

Figura 18 - Display de LCD

Fonte: Autor

Um importante passo que deve ser mencionado nesse trabalho é a integração das bibliotecas e funcionalidades de cada módulo em um só programa, sendo necessário um aprendizado aparte em linguagem HTML para a realização do desenvolvimento do código *on line*.

Com todos os circuitos funcionando perfeitamente foi iniciado a fase de construção da maquete no material MDF, como podemos observar na figura abaixo.

Figura 19 - Protótipo do projeto

Fonte: Autor

Após o término da construção da maquete, iniciamos as instalações dos LED's, soldando resistências junto aos LED's e os distribuindo de acordo com os cômodos da casa e assim, realizando novos testes com uma fonte externa de 12 V, conectando os LED's junto ao relé (NA).

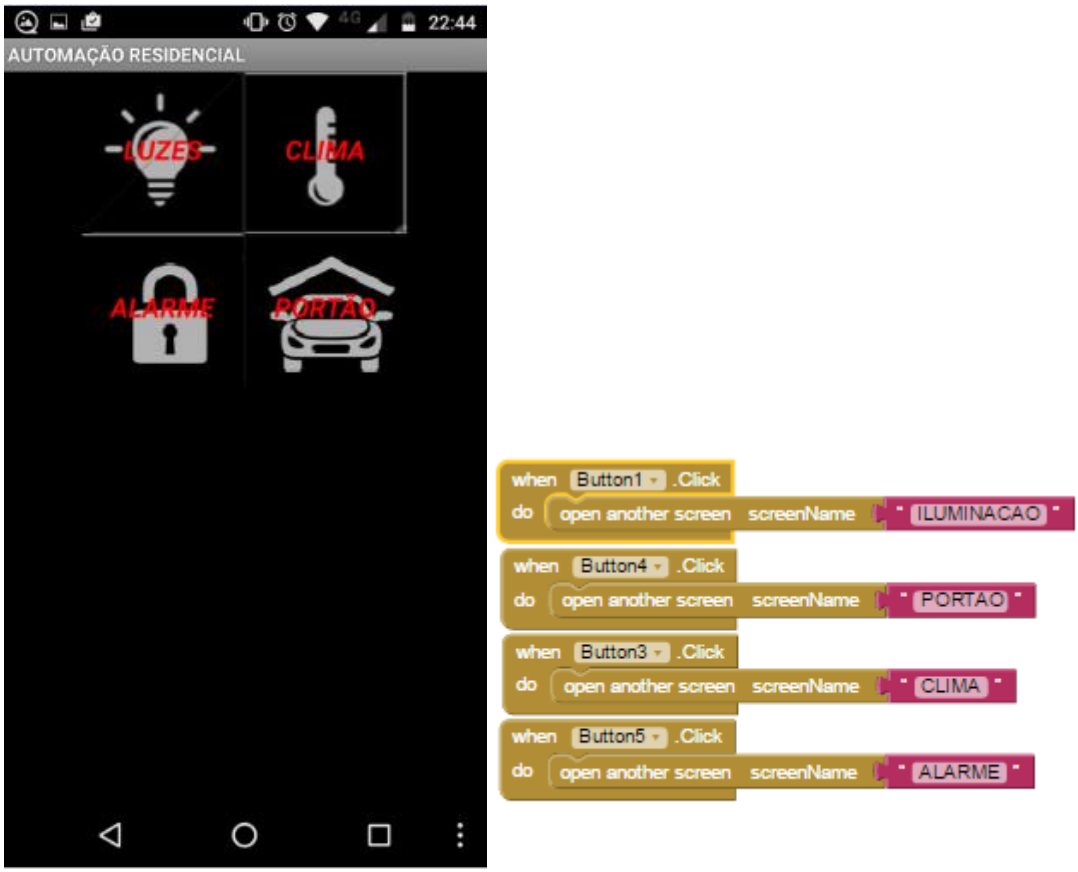
Com o funcionamento da iluminação, partimos para a implementação do sistema de acionamento do portão por meio de um servo motor. Ajustando na configuração o ângulo de abertura e fechamento do mesmo.

Foi instalado também um sensor de luminosidade (LDR), estrategicamente posicionado na entrada da casa, afim de quando o feche de luz do *laser* for interrompido um alarme sonoro será imediatamente acionado.

Para o controle das funcionalidades do projeto nesse protótipo, foi proposto um aplicativo o *MIT INVENTOR* 2, uma plataforma do *Google*. Foi escolhido esse tipo de controle, aja vista a facilidade criada para o usuário utilizar as ferramentas implementadas no projeto, de uma forma clara e simples e podendo ser acessadas e controladas remotamente por um aparelho móvel.

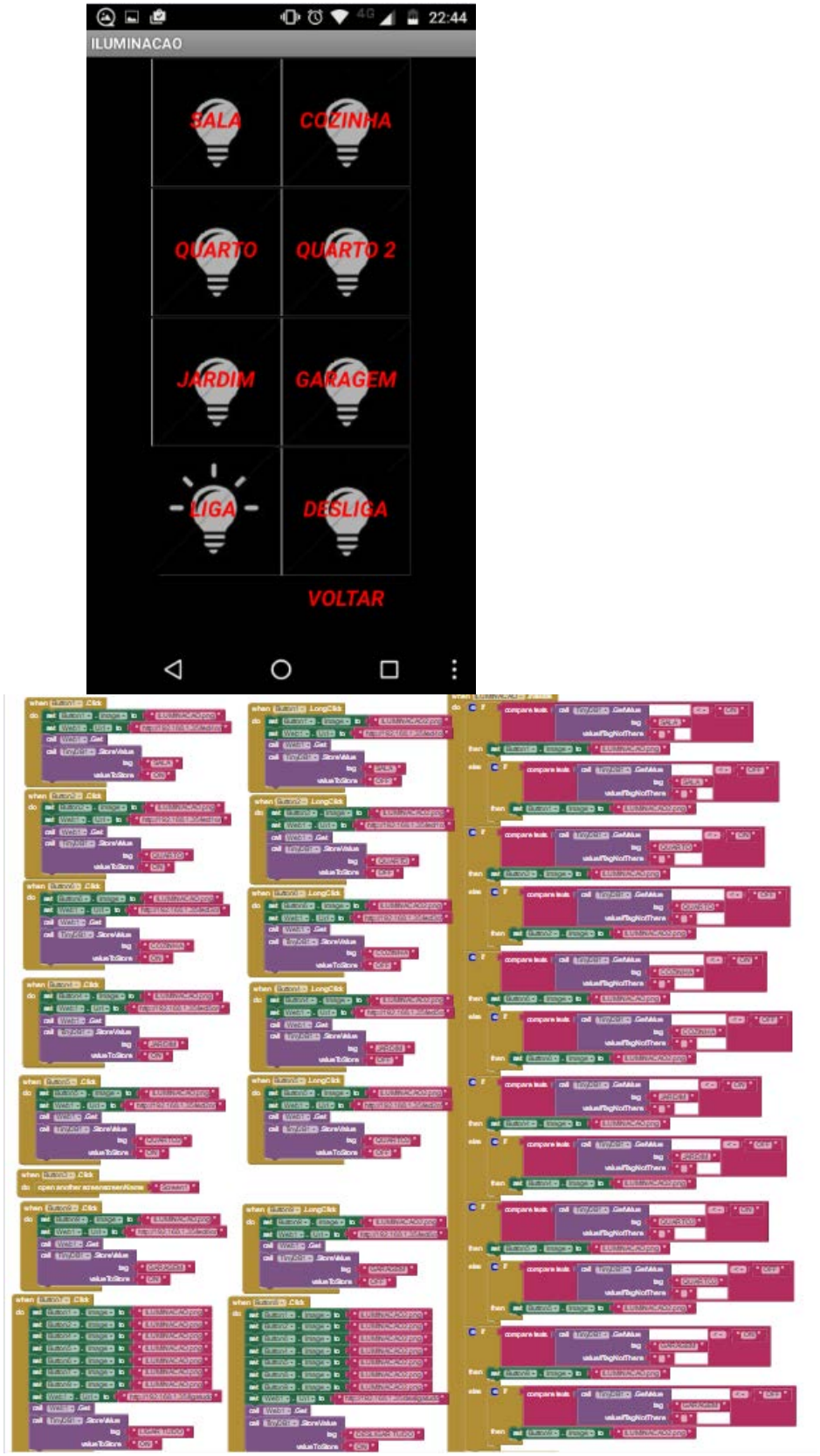
As imagens a seguir nos ilustram como o *Layout* desse aplicativo ficou ao término do seu desenvolvimento.

Figura 20 - *Layout e programação menu principal* do aplicativo para Android



Fonte: Autor

Figura 21 - *Layout e programação tela iluminação* do aplicativo para controle de iluminação da residência

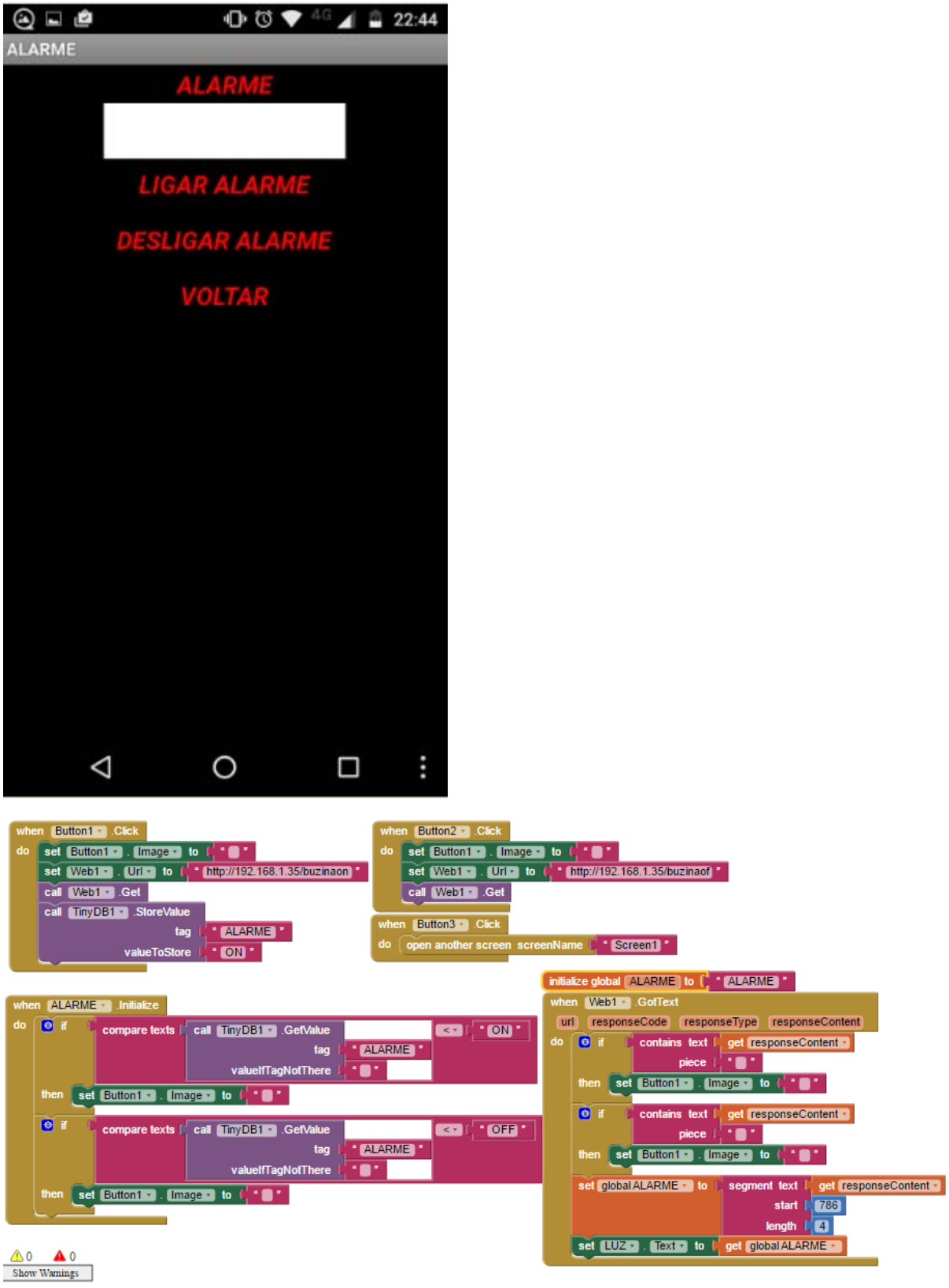


Fonte: Autor

Figura 22 - *Layout e programação, tela clima* do aplicativo para controle de temperatura na residência

Fonte: Autor

Figura 23 - *Layout e programação, tela alarme* do aplicativo para controle do alarma da residência.



Fonte: Autor

Figura 24 - *Layout* do aplicativo para controle do portão

Fonte: Autor

4 CONCLUSÃO

Desde o início deste trabalho de graduação foi proposto a realização de um protótipo que simulasse um projeto de instalação em automação residencial, com os sensores sendo controlados por um aplicativo em um celular com plataforma Android.

Foram encontradas dificuldades na configuração do modulo de Ethernet, tivemos que utilizar um programa para encontrar o IP que o Arduino estava utilizando, que para um acesso remoto tivemos que configurar a porta NAT do roteador, afim de direcioná-lo ao endereço do Arduino. Outra dificuldade enfrentada foi quanto ao número de portas digitais disponíveis, pois após testes falhos com os dispositivos em certas portas descobrimos que alguns módulos utilizavam as mesmas portas, os módulos como o Ethernet *Shield* entraram em conflito com as portas 10, 11, 12 e 13, pois essas portas estavam sendo utilizadas pelo módulo para comunicação com a internet. Para resolver esse problema, utilizamos o módulo I2C, que é responsável pela redução de portas utilizadas pelo LCD reduzindo-as para apenas duas portas analógicas, como já mencionado ao longo deste trabalho.

Outro problema enfrentado no desenvolvimento desse projeto foi na calibração no sensor de temperatura, aonde o mesmo possui pouca sensibilidade e grande variação de tensão, o tornando assim, impreciso.

É interessante ressaltar que esse projeto está aberto a outras melhorias e desenvolvimento de mais funcionalidades, pois a automação residencial oferece muitas oportunidades além das abordadas nesse trabalho.

É importante mencionar também que existe a possibilidade do usuário configurar o projeto de automação.

Um fato interessante que pudemos constatar nesse trabalho de graduação é que apesar das dificuldades encontradas no desenvolvimento desse projeto, pode-se observar que é possível construir alguns sistemas de automação residencial efetivos e com um custo-benefício mais baixo aos encontrados no mercado atual. Isso nos mostra o quanto a automação residencial deve estar mais aparente e com um acesso mais fácil para seus usuários, tornando-se um segmento com características promissoras no mercado brasileiro.

Esse trabalho descreveu alguns conceitos fundamentais e extremamente necessários para o desenvolvimento de um projeto de automação residencial.

REFERÊNCIAS

ALIEVE, C. A. **Automação residencial com utilização de controlador lógico programável.** Trabalho de conclusão de curso. 2008. 20 f. (Graduação em Ciência da Computação) – Centro Universitário Feevale, Novo Hamburgo, 2008.

ARDUINO E CIA. **Arduino UNO e ethernet shield**. Disponível em: <http://www.arduinoecia.com.br/2013/06/ethernet-shield-wiznet-w5100-parte-1.html>. Acesso em: 19 set. 2016.

ARDUINO. **Arduino ethernet shield**. Disponível em: <http://arduino.cc/en/Main/ArduinoEthernetShield>. Acesso em: 19 set. 2016.

BOEIRA, M. **O que é arduino?** 2013. Disponível em: <http://blog.marceloboeira.com/arduino/o-que-e/>. Acesso em: 19 set. 2016.

BORTOLUZZI, M**. Histórico da automação residencial**, 2013. Disponível em <http://sraengenharia.blogspot.co.br/2013/historico-da-automaçaoresidencial_10.html>. Acesso em: 06 set. de 2013.

CRUZ, R. P. **Desenvolvimento de um sistema de supervisão via web aplicado à automação residencial.** 2009. 43 f. Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2009.

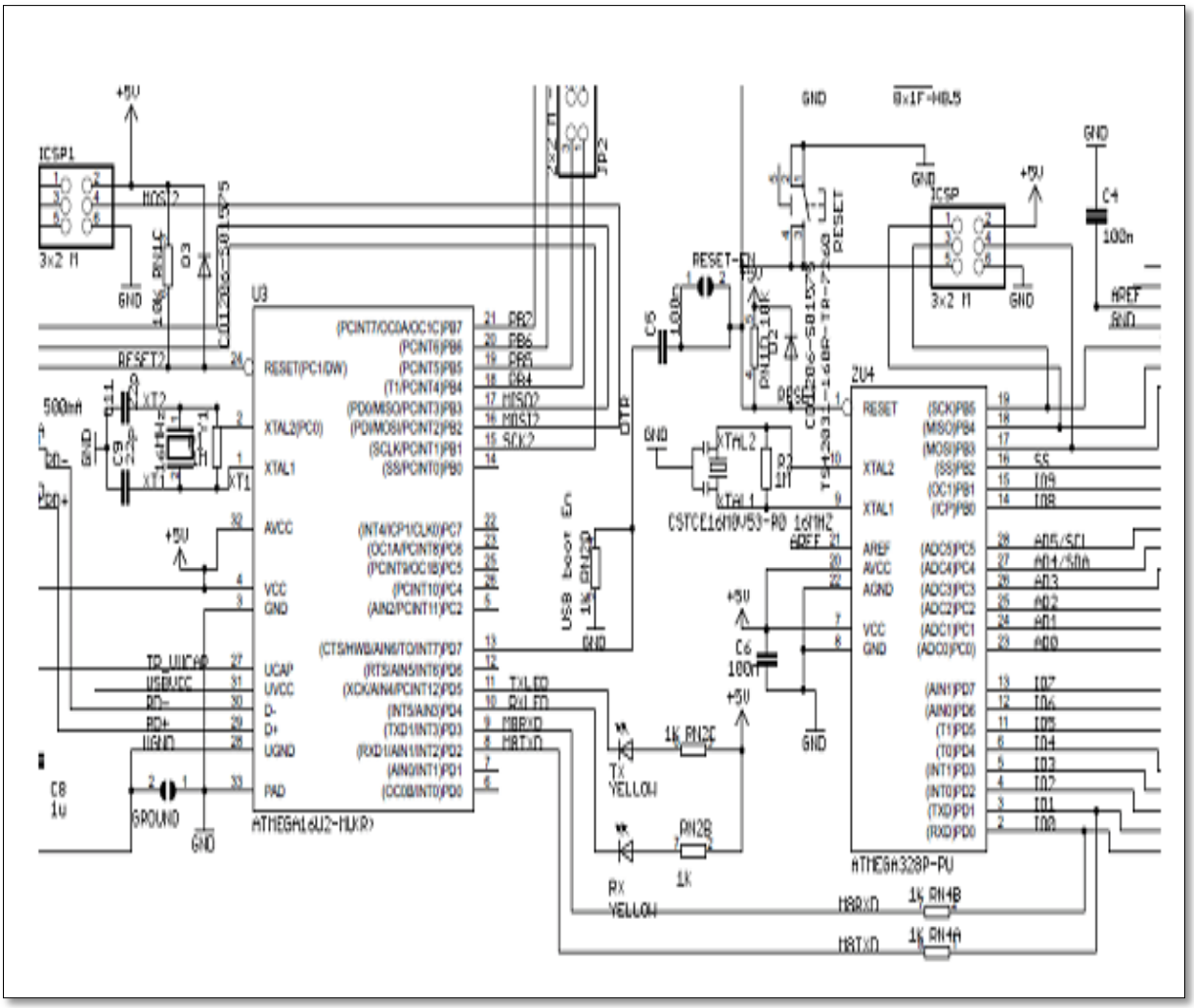
PAIVA, L. S**. Metodologia para a implantação de automação residencial.** 2007. 83f. Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) – Universidade Federal de Ouro Preto, Ouro Preto, 2007. Disponível em: <http://www.em.ufop.br/cecau/monografias/2007/LEONARDO%20%20PAIVA.pdf>. Acesso em: 13 set. 2016.

PROCEL – **Programa Nacional de conservação de energia elétrica. Dicas de economia de energia para um mundo melhor. Ministério de Minas e Energia**, 2013. Disponível em: <http://www.eletrobras.com/elb/main.asp?TeamID=%7B6751E537-0EC0-4B83-BE03-82831A153042%7D >. Acesso em: 02 set. 2016.

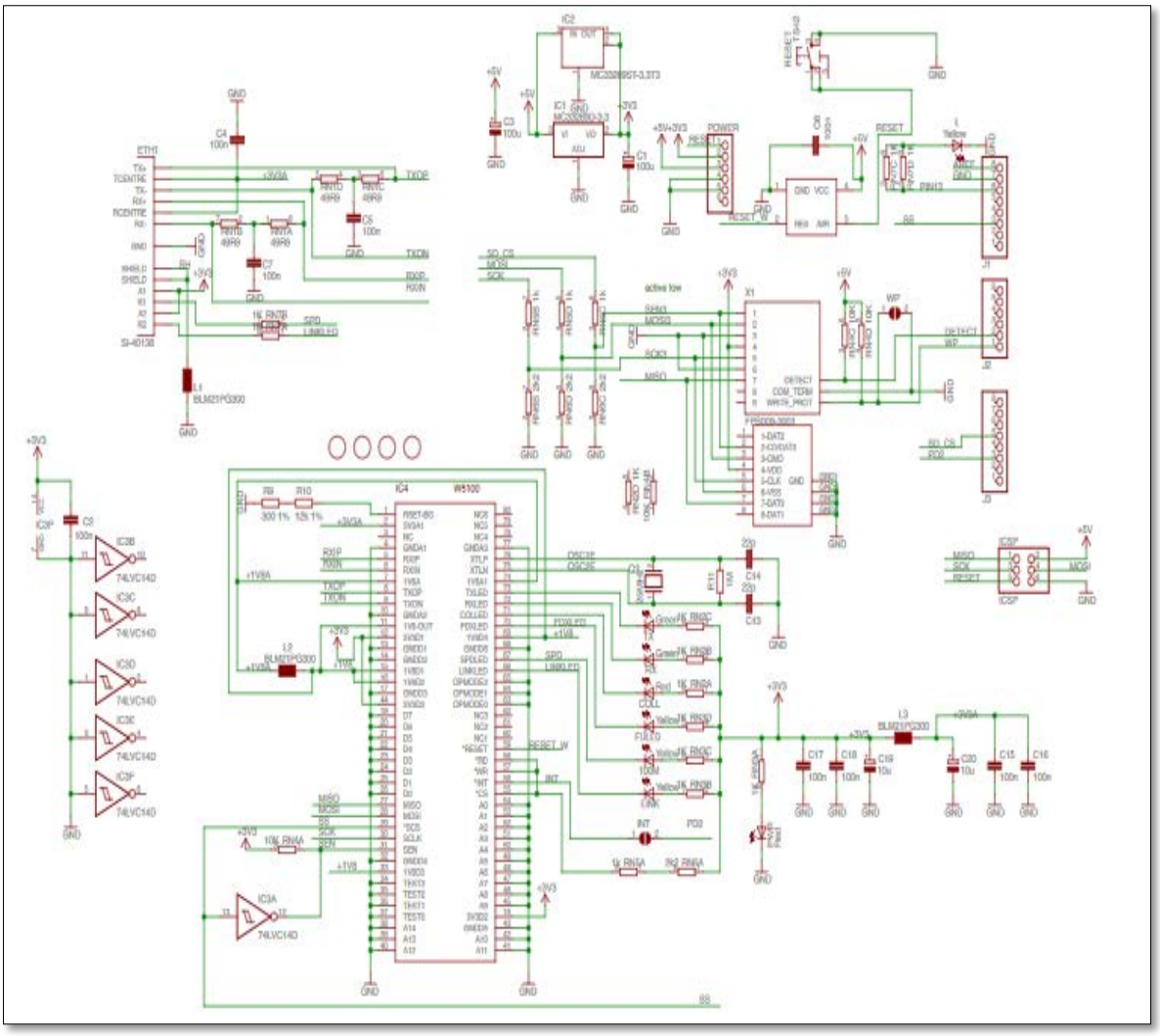
PULCINELLI, M. **Microcontrolador–TM4C123G LaunchPad (Cortex-M4).** 2014. Disponível em: < http://blog.marcio-pulcinelli.com/2014/02/04/microcontrolador-tm4c123g-launchpad-cortex-m4/>. Acesso em: 19 set. 2016.

SILVA, B. C. R; CÂNDIDO, L. A. A. **Sistema de controle residencial baseado na plataforma arduino**. 2011. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Ensino Superior Objetivo, Goiânia, 2011. Disponível em: <http://pt.scribd.com/doc/80802432/Sistema-de-Controle-Residencial-Baseado-Na-Plataforma-Arduino>. Acesso em: 02 set. 2016.

ANEXOS A : Circuito de comunicação serial da placa Arduino UNO



ANEXOS B : Arduino ETHERNET – Shield V5



ANEXOS C : Data Sheet Rele Driver 4 canais DC 5V

SONGLE RELAY

松乐继电器 SONGLE RELAY	RELAY ISO9002	SRD

1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
  (Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

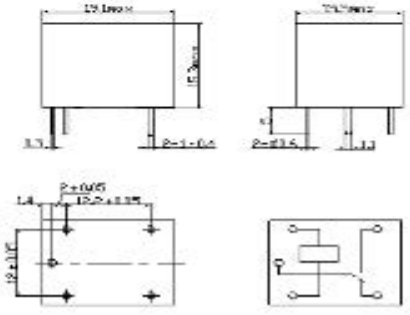
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03、05、06、09、12、24、48VDC	S:Sealed type	L:0.36W	A:1 form A
				B:1 form B
		F:Flux free type	D:0.45W	C:1 form C

4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

5. DIMENSION(uni:mm) DRILLING(uni:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) ±10%	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

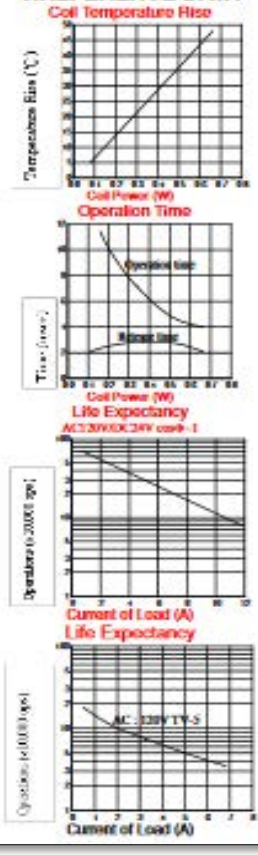
7. CONTACT RATING

Item \ Type	SRD	
	FORM C	FORM A
Contact Capacity Resistive Load (cosΦ=1)	7A 28VDC 10A 125VAC 7A 240VAC	10A 28VDC 10A 240VAC
Inductive Load (cosΦ=0.4 L/R=7msec)	3A 120VAC 3A 28VDC	5A 120VAC 5A 28VDC
Max. Allowable Voltage	250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force	800VAC/240W	1200VA/300W
Contact Material	AgCdO	AgCdO

8. PERFORMANCE (at initial value)

Item \ Type	SRD
Contact Resistance	100mΩ Max.
Operation Time	10msec Max.
Release Time	5msec Max.
Dielectric Strength Between coil & contact Between contacts	 1500VAC 50/60HZ (1 minute) 1000VAC 50/60HZ (1 minute)
Insulation Resistance	100 MΩ Min. (500VDC)
Max. ON/OFF Switching Mechanically Electrically	 300 operation/min 30 operation/min
Ambient Temperature	-25°C to +70°C
Operating Humidity	45 to 85% RH
Vibration Endurance Error Operation	 10 to 55Hz Double Amplitude 1.5mm 10 to 55Hz Double Amplitude 1.5mm
Shock Endurance Error Operation	 100G Min. 10G Min.
Life Expectancy Mechanically Electrically	 10^7 operations. Min. (no load) 10^5 operations. Min. (at rated coil voltage)
Weight	abt. 10grs.

9.REFERENCE DATA



ANEXOS D : Programação do IDE do Arduino

```
#include <SPI.h>
#include <String.h>
#include <Ethernet.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>



LiquidCrystal_I2C lcd(0x27,16,2);   // Criando um LCD de 16x2 no
endereço 0x27
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x9B, 0x36 }; // Endereço Mac
byte ip[] = { 192, 168, 1, 35 }; // Endereço de Ip da sua Rede
EthernetServer server(80); // Porta de serviço

int motor = 9;  //Pino do Transistor(Cooler)
int led1 = 5;   // Led 1 ligado na saida digital 5
int led2 = 4;   // Led 2 ligado na saida digital 4
int led3 = 3;   // Led 3 ligado na saida digital 3
int led4 = 2;   // Led 4 ligado na saida digital 2
int led5 = 16;  // Led 5 ligado na saida digital 16
int led6 = 17;  // Led 6 ligado na saida digital 17
int buzina = 6; // Buzina ligado na saida digital 6
String readString = String(30); // string para buscar dados de
endereço
boolean statusLed = false;// Variável para o status do led
boolean statusServo = true;// Variável para o status do SERVO
boolean statusMotor = false;
boolean statusBuzina = false;
float temperatura = A0;
int ldrPin = A1; //LDR no pino analígico 1
int ldrValor = 0; //Valor lido do LDR
Servo myservo; //Criando objeto Servo




void setup(){



Ethernet.begin(mac, ip); // Inicia o Ethernet
pinMode(motor, OUTPUT);  //motor como saída
pinMode(led1, OUTPUT);   // Define o pino como saída
pinMode(led2, OUTPUT);   // Define o pino como saída
pinMode(led3, OUTPUT);   // Define o pino como saída
pinMode(led4, OUTPUT);   // Define o pino como saída
pinMode(led5, OUTPUT);   // Define o pino como saída
pinMode(led6, OUTPUT);   // Define o pino como saída
pinMode(buzina, OUTPUT); // Define o pino como saída
Serial.begin(9600);      // Inicia a comunicação Serial
lcd.init();              // Inicializando o LCD
myservo.attach(7);       // Servo ligado no pino 7
myservo.write(10);       //Define ângulo inicial
```

```
}

void loop(){

EthernetClient client = server.available();// Criar uma conexão de
cliente

ldrValor = analogRead(ldrPin); //O valor lido será entre 0 e 1023

if (ldrValor<= 800)

{
digitalWrite (buzina,HIGH);//se o valor lido for maior que 800, liga
A BUZINA
delay(1000);
digitalWrite(buzina,LOW);
lcd.clear();  //limpa o display do LCD.
lcd.print("ALARME LIGADO");  //imprime a string no display do LCD.
delay(2000);
}

else digitalWrite (buzina,LOW);// senão, desliga alarme
Serial.println(ldrValor); //imprime o valor lido do LDR no monitor
serial
delay(100);
delay(100);


int valorLido = analogRead(0);
temperatura = (valorLido * 0.022975);




//Exibindo valor da leitura do sensor de temperatura no display LCD.
lcd.clear();  //limpa o display do LCD.
lcd.print("Temp: ");  //imprime a string no display do LCD.
lcd.print(temperatura);
lcd.write(B11011111); //Simbolo de graus celsius
lcd.print("C");

//Exibindo valor da leitura do sensor de luz no display LCD.
lcd.setCursor(0,1);  //posiciona o cursor na coluna 0 linha 1 do LCD.
lcd.print("Luz: ");  //imprime a string no display do LCD.
lcd.print(ldrValor);

delay(200); //aguarda 2 segundos

if (client) {
while (client.connected())
{
if (client.available())
{
char c = client.read();
// ler caractere por caractere vindo do HTTP
if (readString.length() < 30)
{
// armazena os caracteres para string
readString += (c);
Serial.println(readString);
```

```
}


//se o pedido HTTP terminou
if (c == '\n')
{
// vamos verificar se o LED deve ser ligado
// Se a string possui o texto L=Ligar
if(readString.indexOf("led1on")>=0)
{
// O Led vai ser ligado
digitalWrite(led1, HIGH);
statusLed = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led1of")>=0)
{
// O Led vai ser desligado
digitalWrite(led1, LOW);
statusLed = false;
}



if(readString.indexOf("led2on")>=0)
{
// O Led2 vai ser ligado
digitalWrite(led2, HIGH);
statusLed = true;

}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led2of")>=0)
{
// O Led vai ser desligado
digitalWrite(led2, LOW);
statusLed = false;

}




if(readString.indexOf("led3on")>=0)
{
// O Led vai ser ligado
digitalWrite(led3, HIGH);
statusLed = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led3of")>=0)
{
// O Led vai ser desligado
digitalWrite(led3, LOW);
statusLed = false;
}
```

```
if(readString.indexOf("led4on")>=0)
{
// O Led vai ser ligado
digitalWrite(led4, HIGH);
statusLed = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led4of")>=0)
{

// O Led vai ser desligado
digitalWrite(led4, LOW);
statusLed = false;
}



if(readString.indexOf("led5on")>=0)
{
// O Led vai ser ligado
digitalWrite(led5, HIGH);
statusLed = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led5of")>=0)
{
// O Led vai ser desligado
digitalWrite(led5, LOW);
statusLed = false;
}



if(readString.indexOf("led6on")>=0)
{
// O Led vai ser ligado
digitalWrite(led6, HIGH);
statusLed = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("led6of")>=0)
{
// O Led vai ser desligado
digitalWrite(led6, LOW);
statusLed = false;
}
```

```
if(readString.indexOf("ligatudo")>=0)
{
// O Led vai ser ligado
digitalWrite(led1, HIGH);
statusLed = true;
// O Led vai ser ligado
digitalWrite(led2, HIGH);
statusLed = true;
// O Led vai ser ligado
digitalWrite(led3, HIGH);
statusLed = true;
// O Led vai ser ligado
digitalWrite(led4, HIGH);
statusLed = true;
// O Led vai ser ligado
digitalWrite(led5, HIGH);
statusLed = true;
// O Led vai ser ligado
digitalWrite(led6, HIGH);
statusLed = true;

}



// Se a string possui o texto L=Desligar
if(readString.indexOf("desligatudo")>=0)
{
// O Led vai ser desligado
digitalWrite(led1, LOW);
statusLed = false;
// O Led vai ser desligado
digitalWrite(led2, LOW);
statusLed = false;
// O Led vai ser desligado
digitalWrite(led3, LOW);
statusLed = false;
// O Led vai ser desligado
digitalWrite(led4, LOW);
statusLed = false;
// O Led vai ser desligado
digitalWrite(led5, LOW);
statusLed = false;
// O Led vai ser desligado
digitalWrite(led6, LOW);
statusLed = false;

}
```

```
if(readString.indexOf("motoron")>=0)
{
// O Led vai ser ligado
digitalWrite(motor, HIGH);
statusMotor = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("motorof")>=0)
{
// O Led vai ser desligado
digitalWrite(motor, LOW);
statusMotor = false;
}
```

```
if(readString.indexOf("buzinaon")>=0)
{
// O Led vai ser ligado
digitalWrite(buzina, HIGH);
statusBuzina = true;
}
// Se a string possui o texto L=Desligar
if(readString.indexOf("buzinaof")>=0)
{
// O Led vai ser desligado
digitalWrite(buzina, LOW);
delay(10000);
statusBuzina = false;
}
```

```
// dados HTML de saída começando com cabeçalho padrão
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();

client.println("<HTML>");
client.println("<HEAD>");
client.println("<TITLE>AUTOMACAO RESIDENCIAL</TITLE>");
client.println("</HEAD>");


client.println("<H1>AUTOMACAO RESIDENCIAL</H1>");

client.println("<a href=\"/led1on\"\">LED ON</a>");
client.print("<P>");
client.println("<a href=\"/led1of\"\">LED OFF</a>");
client.print("<P>");
```

```
client.println("<a href=\"/led2on\"\">LED2 ON</a>");
client.print("<P>");
client.println("<a href=\"/led2of\"\">LED2 OFF</a>");
client.print("<P>");


client.println("<a href=\"/led3on\"\">LED3 ON</a>");
client.print("<P>");
client.println("<a href=\"/led3of\"\">LED3 OFF</a>");
client.print("<P>");


client.println("<a href=\"/led4on\"\">LED4 ON</a>");
client.print("<P>");
client.println("<a href=\"/led4of\"\">LED4 OFF</a>");
client.print("<P>");


client.println("<a href=\"/led5on\"\">LED5 ON</a>");
client.print("<P>");
client.println("<a href=\"/led5of\"\">LED5 OFF</a>");
client.print("<P>");



client.println("<a href=\"/led6on\"\">LED6 ON</a>");
client.print("<P>");
client.println("<a href=\"/led6of\"\">LED6 OFF</a>");
client.print("<P>");


client.println("<a href=\"/ligatudo\"\">LIGA TUDO</a>");
client.print("<P>");
client.println("<a href=\"/desligatudo\"\">DESLIGA TUDO</a>");
client.print("<P>");

client.println("<a href=\"/mostrara\"\">MOSTRA RA</a>");
client.print("<P>");

client.println("<a href=\"/motoron\"\">MOTOR ON</a>");
client.print("<P>");
client.println("<a href=\"/motorof\"\">MOTOR OFF</a>");
client.print("<P>");


client.println("<a href=\"/buzinaon\"\">BUZINA ON</a>");
client.print("<P>");
client.println("<a href=\"/buzinaof\"\">BUZINA OFF</a>");
client.print("<P>");


// separa as linhas
client.print("<P>");

client.print("Temperatura: ");
client.print(temperatura);

client.print("<P>");
```

```
client.print("LUZ: ");
client.print(ldrValor);

client.print("<P>");


client.println("<a href=\"/?on\"\">FECHA</a>");
client.print("<P>");
client.println("<a href=\"/?off\"\">ABRE</a>");




delay(1);
// parar cliente
client.stop();



if(readString.indexOf("?on") >0)//checks for on
{
myservo.write(10);

Serial.println("Led On");
}
if(readString.indexOf("?off") >0)//checks for off
{
myservo.write(125);

Serial.println("Led Off");
}
//limpa string para a próxima leitura
readString="";

}
}
}
}
}
```