



ETEC JORGE STREET

TRABALHO DE CONCLUSÃO DO CURSO TÉCNICO EM MECATRÔNICA

ILLUMIWATT

Gabriel da Silva Ribeiro

**Professor(es) Orientador(es):
Cristina Moura Ramos
Vera Lúcia Guimarães Silveira Beneti**

**São Caetano do Sul / SP
2021**

ILLUMIWATT

Trabalho de Conclusão de Curso
apresentado como pré-requisito para
obtenção do Diploma de Técnico em
Mecatrônica.

**São Caetano do Sul / SP
2021**

1. AGRADECIMENTOS

Com grande honra estamos finalizando essa etapa do curso em um meio pandêmico que ocupou grande parte das nossas aulas, impedindo-nos de prestigiar de perto a parte prática, mas é com muita felicidade que lhes apresento o meu projeto. Obrigado.

À minha família que me ajudou emocionalmente e moralmente. Obrigado.

Aos professores(as): Cristina Moura Ramos, Vera Lúcia Guimarães Silveira Beneti e Willian Hesley Marton que me orientaram muito para que esse projeto fosse construído. Obrigado.

Aos colegas de classe que com os quais pude compartilhar conhecimento e risadas. Obrigado.

Centro Paula Souza, Muito Obrigado.

2. DEDICATÓRIA

[Dedico totalmente esse trabalho a minha mãe, Eliane, que sempre me apoiou muito, realizando o sonho dela de eu ser técnico em Mecatrônica e futuramente um Engenheiro Elétrico. Obrigado.].

3. RESUMO

O presente Trabalho de Conclusão de Curso trata-se de um meio em que qualquer pessoa possa monitorar a iluminação, temperatura e umidade de sua residência remotamente pelo Telegram. A ideia do projeto veio para facilitar a vida de indivíduos com facilidade em ligar/desligar lâmpadas de sua casa de onde você estiver com uma conexão de internet estável, além de, saber a temperatura e umidade local. Concomitantemente, poder chegar em casa à noite e não haver a necessidade de acender as luzes externas. Essa automatização residencial usa tecnologia para automatizar tarefas presentes em sua casa. Tarefas que antes dependiam somente do morador. O conceito é fazer com que todos os sistemas eletrônicos funcionem como um só, sob nosso total controle trazendo praticidade, segurança, economia e conforto.

Palavras-chave: Telegram, Monitoramento, Automação residencial.

4. ABSTRACT

My Course Completion Paper is a way in which anyone can monitor the lighting, temperature and humidity of their home remotely via Telegram. The idea of the project came to make life easier, with ease in turning on/off light bulbs in your house wherever you are with a stable internet connection, in addition to knowing the local temperature and humidity. At the same time, being able to get home at night and do not having to turn on the external lights. This home automation uses technology to automate tasks present in your home. Tasks that previously depended only on the resident. The concept is to make all electronic systems work as one, under our total control, bringing practicality, safety, economy and comfort.

Keywords: Telegram, Monitoring, Home automation.

SUMÁRIO

1.	Agradecimentos	3
2.	Dedicatória	4
3.	Resumo	5
4.	Abstract	6
5.	Lista de figuras	8
6.	Lista de tabelas	9
7.	Introdução	10
8.	Tema e delimitação	10
9.	Objetivo – geral e específico	10
10.	Justificativa	10
11.	Metodologia	11
12.	Fundamentação teórica	12
13.	Planejamento do projeto	13
13.1.	Diagrama de blocos	13
13.2.	Previsão de custos	13
13.3.	Componentes	14
13.4.	Fluxograma do processo	23
13.5.	Cronograma geral	23
14.	Desenvolvimento do projeto	24
14.1.	Logo	24
14.2.	Programação em C	24
14.3.	Circuito eletrônico	31
15.	Resultados obtidos	33
16.	Conclusão	35
17.	Referências	36

5. LISTA DE FIGURAS

Figura 1 - Diagrama de blocos	13
Figura 2 - ESP32.....	14
Figura 3 - ESP32 Pinout.....	16
Figura 4 - DHT11.....	16
Figura 5 - LDR 5mm.....	17
Figura 6 - Jumpers Macho-Macho.....	18
Figura 7 - Resistores 330Ω	19
Figura 8 – LEDs	19
Figura 9 - Matriz de contato.....	20
Figura 10 - Módulo relé 1 canal 5V/10 ^a	21
Figura 11 - Lâmpada 7W.....	22
Figura 12 - Fluxograma do processo.....	23
Figura 13 – Logo	24
Figura 14 - Circuito eletrônico	31
Figura 15 - Maquete para o protótipo	33
Figura 16 - Protótipo na matriz de contato	33
Figura 17 - Interface do projeto pelo Telegram - Parte I.....	34
Figura 18 - Interface do projeto pelo Telegram - Parte II.....	34

6. LISTA DE TABELAS

Tabela 1 - Custos do protótipo	13
Tabela 2 - Custos do projeto real	14
Tabela 3 - Demais custos.....	14
Tabela 4 - Cronograma geral	23

7. Introdução

O atual projeto nomeado de ILLUMIWATT trata-se de um monitoramento de iluminação, temperatura e umidade de uma casa a partir de um microcontrolador e ChatBot via Telegram.

8. Tema e delimitação

O tema escolhido para o projeto foi a automação residencial, com foco em ser acessível para qualquer indivíduo que queira aprimorar seu local de conforto com tecnologia. A automação residencial proporciona uma comodidade que antes só estava presente na imaginação e nos livros de ficção científica. Além de tudo, ela é facilmente adaptada aos dispositivos domésticos, o que faz dela uma tecnologia flexível. Podemos fazer diversos tipos de monitoramento a partir do microcontrolador, como: monitorar câmeras, portões, iluminação, temperatura, umidade etc. Fica ao seu critério decidir quais cômodos e quais aparelhos serão automatizados.

9. Objetivo – geral e específico

O objetivo é facilitar a vida do cidadão, modernizando a casa onde reside, pois, daqui a alguns anos está evidente que a automação residencial será comum no dia a dia de cada um por causa do constante crescimento da tecnologia. Por exemplo, você sai cedo de casa para ir trabalhar e deixa a luz acesa pela manhã para que a noite você possa enxergar o caminho até a porta, mas, com o ILLUMIWATT as luzes acenderiam sozinhas após escurecer automaticamente ou se você acionasse de forma manual remotamente por um comando no Telegram.

10. Justificativa

No projeto atual, através de um conselho de um professor da Etec Jorge Street, fez-me pensar bastante nesta questão de controlar os cômodos ou

aparelhos domésticos remotamente, e assim, possibilitando com que qualquer pessoa do mundo poderia fazer o mesmo.

11. Metodologia

Em meu projeto foram realizadas pesquisas de campo em condomínios, chácaras e casas localizadas em Mauá e Ribeirão Pires, além de pesquisas em sites estrangeiros sobre automatização e programação. Juntamente com a orientação dos professores que foi primordial para o desenvolvimento do projeto para melhor funcionalidade.

12. Fundamentação Teórica

Inicialmente, o sistema funcionará a partir do componente principal, o nosso microcontrolador (ESP32), que irá receber todas as informações e, assim podendo executar os comandos programados.

Ele irá receber as leituras de luz do Fotorresistor (LDR), temperatura e umidade do sensor (DHT11). Posteriormente, inserindo o comando respectivo a sua função no Telegram, o Bot irá executá-lo, fazendo com que você saiba por mensagem de texto a temperatura e umidade, monitorar as lâmpadas da sua casa (No caso do protótipo foi utilizado LEDs, com cada um do mesmo representando um cômodo) ou saber se estão ligadas/deligadas remotamente. Ainda assim, tendo um sistema automático na parte das luzes externas que acenderão quando ficar a noite e podendo acioná-las/desacioná-las manualmente.

13. Planejamento do Projeto

13.1. Diagrama de Blocos

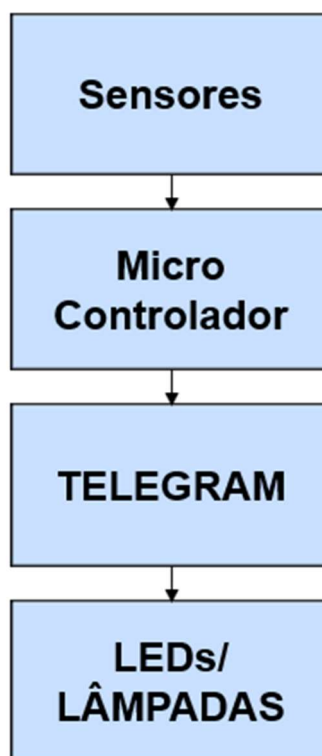


Figura 1 - Diagrama de blocos

13.2. Previsão de custos

Custos do Protótipo		
Componentes	Previsto	Comprado
ESP32	R\$ 30,00	R\$ 33,75
DHT11	R\$ 8,00	R\$ 10,60
LDR (5 mm)	R\$ 1,00	R\$ 1,50
Resistor (330 Ω)	R\$ 0,20	R\$ 0,10
Led	R\$ 0,50	R\$ 0,50
Total	R\$ 39,70	R\$ 46,45

Tabela 1 - Custos do protótipo

Custos do Projeto Real		
Componentes	Previsto	Comprado
ESP32	R\$ 30,00	R\$ 33,75
DHT11	R\$ 8,00	R\$ 10,60
LDR (5 mm)	R\$ 1,00	R\$ 1,50
Lâmpada (9W)	R\$ 13,00	R\$ -
Cabos (2,5 mm ²)	R\$ 2,70	R\$ -
Módulo Relê 1 Canal 5V	R\$ 12,60	R\$ -
Total	R\$ 67,30	R\$ 45,85

Tabela 2 - Custos do projeto real

Outros	Comprado
Matriz de Contato	R\$ 18,00
Tubo termo retrátil	R\$ 0,25
Jumpers macho-macho	R\$ 0,40

Tabela 3 - Demais custos

13.3. Componentes



Figura 2 - ESP32

Esp32 Esp-32 Esp-wroom-32

- Baseado no SoC (System on Chip) ESP32-D0WDQ6 (Datasheet ESP-32)
- Módulo controlador ESP-WROOM-32 (Datasheet ESP-WROOM-32)

- Microprocessador dual core Tensilica Xtensa 32-bit LX6
- Clock ajustável de 80MHz até 240MHz
- Desempenho de até 600 DMIPS
- Possui ROM de 448KB
- Possui SRAM de 520KB
- Possui RTC Slow SRAM de 8KB
- Possui RTC Fast SRAM de 8KB
- Possui memória flash externa de 32Mb (4 megabytes)
- Opera na faixa de 2.2V – 3.6VDC
- Pode ser alimentado com 5VDC através do conector micro USB
- Opera em nível lógico 3.3V (não tolerante a 5V)
- Opera com corrente típica de 80mA
- Corrente máxima por pino é de 12mA (recomenda-se usar 6mA)
- Possui Interfaces de GPIO / Sensores capacitivos / ADC / DAC / LNA pré amplificado / CAN
- Possui 36 GPIOs
- GPIOs com função PWM / I2C e SPI
- Possui ADC (conversor analógico digital) de 18 canais com resolução de 12 bits
- Possui 2 DAC (conversor digital analógico) com resolução de 8 bits
- Possui suporte a redes WiFi padrão 802.11 b/g/n
- O WiFi opera na faixa de 2.4 a 2.5GHz
- WiFi possui opções de segurança WPA / WPA2 / WPA2-Enterprise / WPS
- WiFi possui opções de criptografia AES / RSA / ECC / SHA
- Opera nos modos Station / SoftAP / SoftAP + Station/ P2P
- Possui Antena integrada
- Possui Bluetooth v4.2 BR / EDR e BLE (Bluetooth Low Energy)
- Opera em temperaturas na faixa de -40° a 85°C
- Programável via USB, host ou WiFi (OTA / Over The Air)
- Compatível com a IDE do Arduino
- Compatível com módulos e sensores utilizados no Arduino

ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs

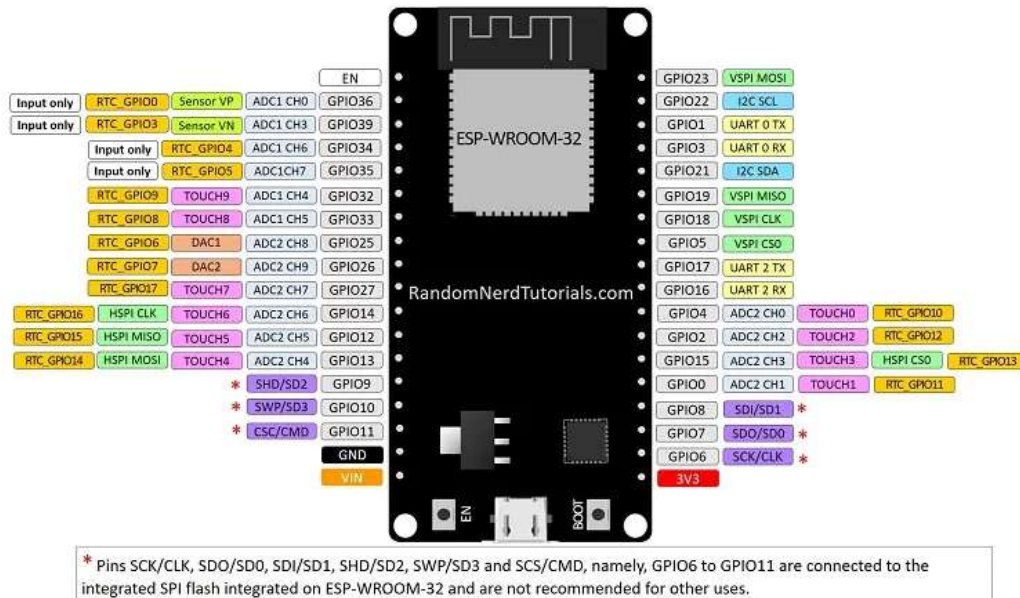


Figura 3 - ESP32 Pinout



Figura 4 - DHT11

Modulo / Sensor DHT11 Temperatura e Umidade

Arduino - PIC

O DHT11 é um sensor de temperatura e umidade de saída de sinal digital garantindo alta confiabilidade e estabilidade a longo prazo.

O sensor de temperatura é um termistor do tipo NTC e o sensor de Umidade é do tipo HR202, o circuito interno faz a leitura dos sensores e se comunica a um microcontrolador através de um sinal serial de uma via.

Especificação:

- Faixa de medição de Umidade: 20% ~ 90% RH
- Erro de medição de umidade : ± 5 % RH
- Temperatura faixa de medição : 0 ~ 60°
- Erro de medição de temperatura: $\pm 2^\circ$
- Tensão de funcionamento : 5 V
- Corrente: 200 a 500 ma
- Tamanho: 28 x 12 x 8mm

Itens Inclusos:

- Modulo DHT11
- Cabos

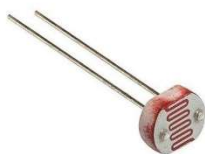


Figura 5 - LDR 5mm

- Sensor de luminosidade LDR 5mm (fotoresistor)



Figura 6 - Jumpers Macho-Macho

Jumpers Macho-Macho

Os Cabos Jumpers Macho-macho são utilizados na prototipagem de componentes eletrônicos e são peças indispensáveis na sua bancada de projetos. O uso destes cabos é ideal para efetuar as conexões entre componentes eletrônicos, seja com Arduino, NodeMCU ESP8266 ou outros microcontroladores. As pontas dos cabos são isoladas por um material não condutivo, garantido segurança na hora de efetuar as ligações. Os cabos são enviados lado a lado (grudados), porém podem ser destacados para uso individual.

Especificações:

- Conjunto com 40 jumpers
- Cores diversas
- Secção do fio condutor: 24 AWG
- Comprimento do jumper: 20cm
- Largura do conector: 2,54mm
- Destacáveis

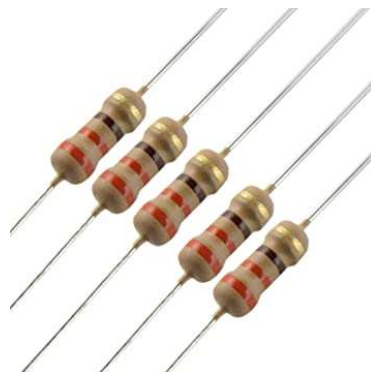


Figura 7 - Resistores 330Ω

Resistores são dispositivos que compõem circuitos elétricos diversos, a sua finalidade básica é a conversão de energia elétrica em energia térmica (Efeito Joule). Outra função dos resistores é a possibilidade de alterar a diferença de potencial em determinada parte do circuito, isso ocorre por conta da diminuição da corrente elétrica devido à presença do equipamento.



Figura 8 – LEDs

Um diodo emissor de luz (LED) é uma fonte de luz semicondutora que emite luz quando a corrente flui através dele. Os elétrons no semicondutor se recombinam com orifícios de elétrons , liberando energia na forma de fótons . A cor da luz (correspondente à energia dos fótons) é determinada pela energia necessária para que os elétrons cruzem o espaço de banda do semicondutor.

Especificações:

- Potência: 60mW
- Corrente Direta: 20mA
- Tensão Direta: 3,0-3,2V
- Comprimento de Onda (nm) ou Temperatura de Cor (K): 460-465nm
- Fluxo Luminoso ou Intensidade Luminosa: 6000-8000mcd
- Tensão Reversa [V]: ≤ 5
- Corrente Reversa [μ A]: ≤ 10
- Resistência à ESD [V]: 1000 Temperatura de Armazenamento [$^{\circ}$ C]: 40 a 90
- Temperatura de Operação [$^{\circ}$ C]: -40 a 85
- Vida Útil Estimada [h]: 100.000

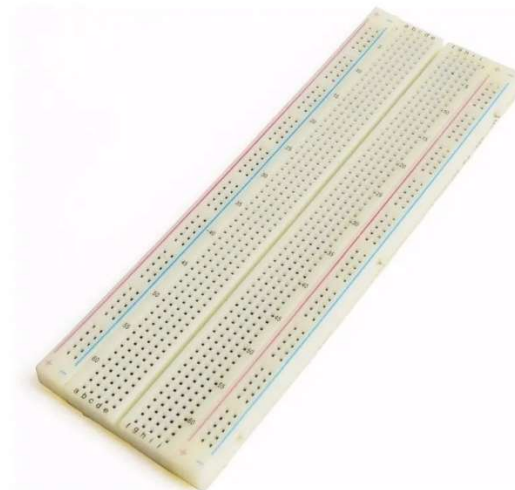


Figura 9 - Matriz de contato

Uma matriz de contato, ou popularmente conhecida como protoboard é uma placa com conexões condutoras para montagem de circuitos elétricos experimentais. A grande vantagem da placa de ensaio na montagem de circuitos eletrônicos é a facilidade de inserção de componentes, uma vez que não necessita soldagem. As placas variam de 800 furos até 6000 furos, tendo conexões verticais e horizontais.

Especificações:

- Furos: 830
- Faixa de Temperatura: -20 a 80 $^{\circ}$ C

- Para terminais e condutores de 0,3 a 0,8 mm (20 a 29 AWG)
- Resistência de Isolamento: 100M Ω min.
- Tensão Máxima: 500V AC por minuto
- Dimensões: 165 mm x 57mm x 10mm

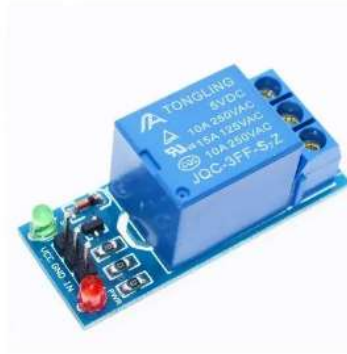


Figura 10 - Módulo relé 1 canal 5V/10ª

Módulo Relé 1 Canal 5v Automação Arduino Pic Raspberry Pic

Pode ser acionado diretamente por microcontroladores como Arduino, AVR, PIC, ARM, TTL. Ótimo para automações que necessitam controlar cargas maiores AC/DC, pois o módulo já possui optoacoplador. Atende profissionais e hobistas.

Especificações:

- Tensão de operação: 5VDC (VCC e GND);
- Corrente de operação por canal: 15~20mA;
- Contatos: NA, 1 NF e o Comum;
- Relé: 30 VDC a 10A ou 250VAC a 10A;
- 4 Furos de 3mm para fixação;
- Leds indicadores;
- Dimensões: 51mm x 38mm x 20mm.

Itens Inclusos:

1 Módulo Relé 1 Canal 5v Arduino, Pic, Raspberry Pic.



Figura 11 - Lâmpada 7W

Características:

- Iluminação de qualidade, deixando qualquer ambiente aconchegante e moderno
- Longa vida útil, chegando em até 50.000 horas
- Baixo consumo de energia, representando até 80% de economia em sua conta
- Instalação simples, garantindo total praticidade para seu dia-a-dia
- Descarte não prejudicial ao meio ambiente, devido não emitir raio ultravioleta nem infravermelho

Informações técnicas:

Produto: Lâmpada led bulbo

Potência: 7w

Material: Plástico e acrílico

Soquete: E27

Temperatura de cor: Branco Frio (6500k)

Vida útil estimada de até 50.000 horas

Luminosidade: Aproximadamente 950 lúmens

Tensão: AC 110/220v (Bivolt)

Frequência: 50/60 Hz

IRC: 70%

Certificado: CE

Itens Inclusos:

- Lâmpada Led 7W Bulbo Branco Frio Bivolt E27

13.4. Fluxograma do Processo

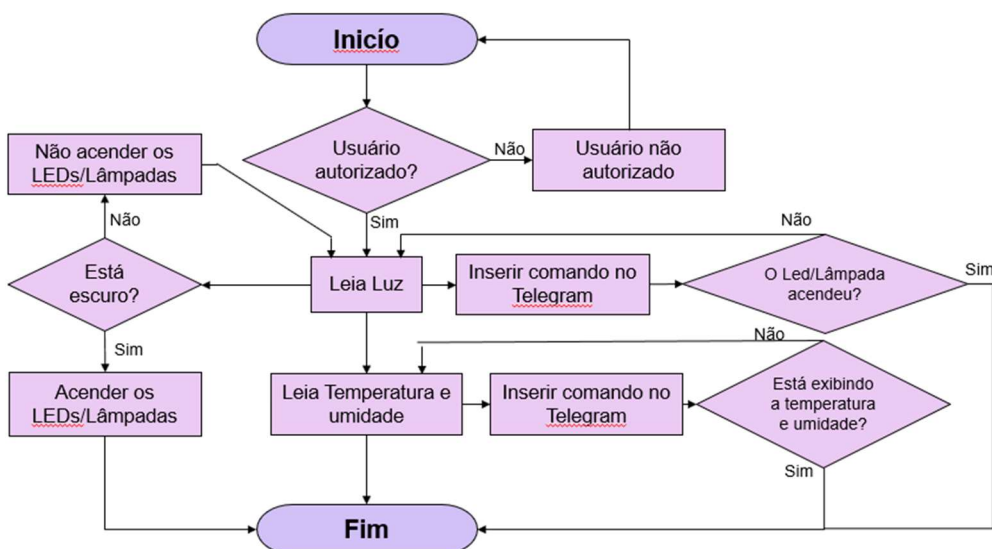


Figura 12 - Fluxograma do processo

13.5. Cronograma Geral

P - Programado R - Realizado

Atividades	Cronograma		
	Elaboração Projeto TCC		
	Outubro	Novembro	Dezembro
Escolha do Orientador	P		
	R		
1ª Reunião com Orientador para apresentação das propostas	P		
	R		
Planejamento	P		
	R		
Organização	P		
	R		
Pesquisa	P		
	R		
Compras	P		
	R		
Monografia	P		
	R		
Programação	P		
	R		
Produção do Protótipo	P		
	R		
Revisão	P		
	R		
Conclusão	P		
	R		

PROJETO: ILLUMIWATT
 TURMA: 4AT - MECATRÔNICA À TARDE
 INTEGRANTES: GABRIEL DA SILVA RIBEIRO

Tabela 4 - Cronograma geral

14. Desenvolvimento do Projeto

14.1. Logo



Figura 13 – Logo

Brevemente, explicando a origem da logo, *Illumi* vem de *illuminate* (em português, iluminação) e *Watt* é a medida de potência (Por exemplo uma lâmpada comum de LED de 9W). Logo em seguida temos o Slogan “think about the ideia”(em português, “pense sobre a ideia”) que surgiu dos Cartoons e desenhos animados, que quando a personagem em determinada situação tinha uma ideia, sua respectiva representação era por uma lâmpada acima de sua cabeça.

14.2. Programação em C

```
#ifndef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <DHT.h>
#include <Adafruit_Sensor.h>
// I removed some personal information from the schedule for security reasons.
//Retirei algumas informações pessoais da programação, por motivos de segurança.
```



```

// Place your network credentials
const char* ssid = "Replace your SSID here";
const char* password = "Replace your password here";

// Initialize Telegram BOT
#define BOTtoken "Place your Bot Token here" // your Bot Token (Get from
Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
#define CHAT_ID "Place your ID or Group ID here"

#ifdef ESP8266
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
#endif

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int ledPin1 = 15; // Bedroom lamp
bool ledState1 = LOW;

const int ledPin2 = 5; //Bathroom lamp
bool ledState2 = LOW;

const int ledPin3 = 18; //Living room lamp
bool ledState3 = LOW;

const int ledPin4 = 19; //Hall lamp
bool ledState4 = LOW;

const int ledPin5 = 21; //Kitchen lamp
bool ledState5 = LOW;

const int ledPin6 = 22; //Outside lamp
bool ledState6 = LOW;

#define DHTPIN 4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

String getReadings() {
  float temperature, humidity;
  temperature = dht.readTemperature();

```

```

humidity = dht.readHumidity();
String message = "Temperature: " + String(temperature) + " °C \n";
message += "Humidity: " + String (humidity) + " % \n";
return message;
}

String getTemperature() {
float temperature;
temperature = dht.readTemperature();
String message = "Temperature: " + String(temperature) + " °C \n";
return message;
}

String getHumidity() {
float humidity;
humidity = dht.readHumidity();
String message = "Humidity: " + String (humidity) + " % \n";
return message;
}

// Handle what happens when you receive new messages
void handleNewMessages(int numNewMessages) {
Serial.println("handleNewMessages");
Serial.println(String(numNewMessages));

for (int i = 0; i < numNewMessages; i++) {
// Chat id of the requester
String chat_id = String(bot.messages[i].chat_id);
if (chat_id != CHAT_ID) {
bot.sendMessage(chat_id, "Unauthorized user", "");
continue;
}

// Print the received message
String text = bot.messages[i].text;
Serial.println(text);

String from_name = bot.messages[i].from_name;

if (text == "/start") {
String welcome = "Welcome, " + from_name + ".\n";
welcome += "Use the following commands to get current readings and control your
lamps.\n\n";
welcome += "/bedrm_on to turn bedroom lamp ON \n";
welcome += "/bedrm_off to turn bedroom lamp OFF \n\n";
welcome += "/bathrm_on to turn bathroom lamp ON\n";
welcome += "/bathrm_off to turn bathroom lamp OFF\n\n";
welcome += "/livrm_on to turn living room lamp ON\n";
welcome += "/livrm_off to turn living room lamp OFF\n\n";
welcome += "/hall_on to turn hall lamp ON\n";
}
}

```

```

welcome += "/hall_off to turn hall lamp OFF\n\n";
welcome += "/kit_on to turn kitchen lamp ON\n";
welcome += "/kit_off to turn kitchen lamp OFF\n\n";
welcome += "/outl_on to turn outside lamp ON\n";
welcome += "/outl_off to turn outside lamp OFF\n\n";
welcome += "/state to request all current lamps state \n\n";
welcome += "/dht11 to request current temperature and humidity\n";
welcome += "/temp to request current temperature\n";
welcome += "/humi to request current humidity\n";
bot.sendMessage(chat_id, welcome, "");
}

if (text == "/bedrm_on") {
  bot.sendMessage(chat_id, "Bedroom lamp state set to ON", "");
  ledState1 = HIGH;
  digitalWrite(ledPin1, ledState1);
}

if (text == "/bedrm_off") {
  bot.sendMessage(chat_id, "Bedroom lamp state set to OFF", "");
  ledState1 = LOW;
  digitalWrite(ledPin1, ledState1);
}

if (text == "/bathrm_on") {
  bot.sendMessage(chat_id, "Bathroom lamp state set to ON", "");
  ledState2 = HIGH;
  digitalWrite(ledPin2, ledState2);
}

if (text == "/bathrm_off") {
  bot.sendMessage(chat_id, "Bathroom lamp state set to OFF", "");
  ledState2 = LOW;
  digitalWrite(ledPin2, ledState2);
}

if (text == "/livrm_on") {
  bot.sendMessage(chat_id, "Living Room lamp state set to ON", "");
  ledState3 = HIGH;
  digitalWrite(ledPin3, ledState3);
}

if (text == "/livrm_off") {
  bot.sendMessage(chat_id, "Living Room lamp state set to OFF", "");
  ledState3 = LOW;
  digitalWrite(ledPin3, ledState3);
}

if (text == "/hall_on") {
  bot.sendMessage(chat_id, "Hall lamp state set to ON", "");
}

```

```
    ledState4 = HIGH;
    digitalWrite(ledPin4, ledState4);
}

if (text == "/hall_off") {
    bot.sendMessage(chat_id, "Hall lamp state set to OFF", "");
    ledState4 = LOW;
    digitalWrite(ledPin4, ledState4);
}

if (text == "/kit_on") {
    bot.sendMessage(chat_id, "Kitchen lamp state set to ON", "");
    ledState5 = HIGH;
    digitalWrite(ledPin5, ledState5);
}

if (text == "/kit_off") {
    bot.sendMessage(chat_id, "Kitchen lamp state set to OFF", "");
    ledState5 = LOW;
    digitalWrite(ledPin5, ledState5);
}

if (text == "/outl_on") {
    bot.sendMessage(chat_id, "Outside lamp state set to ON", "");
    ledState6 = HIGH;
    digitalWrite(ledPin6, ledState6);
}

if (text == "/outl_off") {
    bot.sendMessage(chat_id, "Outside lamp state set to OFF", "");
    ledState6 = LOW;
    digitalWrite(ledPin6, ledState6);
}

if (text == "/dht11") {
    String dht11 = getReadings();
    bot.sendMessage(chat_id, dht11, "");
}

if (text == "/temp") {
    String temp = getTemperature();
    bot.sendMessage(chat_id, temp, "");
}

if (text == "/humi") {
    String humi = getHumidity();
    bot.sendMessage(chat_id, humi, "");
}

if (text == "/state") {
```

```

if (digitalRead(ledPin1)) {
  bot.sendMessage(chat_id, "Bedroom lamp is ON", "");
}
else {
  bot.sendMessage(chat_id, "Bedroom lamp is OFF", "");
}

if (digitalRead(ledPin2)) {
  bot.sendMessage(chat_id, "Bathroom lamp is ON", "");
}
else {
  bot.sendMessage(chat_id, "Bathroom lamp is OFF", "");
}

if (digitalRead(ledPin3)) {
  bot.sendMessage(chat_id, "Living Room lmap is ON", "");
}
else {
  bot.sendMessage(chat_id, "Living Room lamp is OFF", "");
}

if (digitalRead(ledPin4)) {
  bot.sendMessage(chat_id, "Hall lamp is ON", "");
}
else {
  bot.sendMessage (chat_id, "Hall lamp is OFF", "");
}

if (digitalRead(ledPin5)) {
  bot.sendMessage(chat_id, "Kitchen lamp is ON", "");
}
else {
  bot.sendMessage(chat_id, "Kitchen lamp is OFF", "");
}

if (digitalRead(ledPin6)) {
  bot.sendMessage (chat_id, "Outside lamp is ON", "");
}
else {
  bot.sendMessage (chat_id, "Outside lamp is OFF", "");
}
}
}
}
}
}
}
}
}

void setup() {
  Serial.begin(115200);

  dht.begin();

```

```

#ifdef ESP8266
  configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP
  client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
#endif

  pinMode(ledPin1, OUTPUT);
  digitalWrite(ledPin1, ledState1);

  pinMode(ledPin2, OUTPUT);
  digitalWrite(ledPin2, ledState2);

  pinMode(ledPin3, OUTPUT);
  digitalWrite(ledPin3, ledState3);

  pinMode(ledPin4, OUTPUT);
  digitalWrite(ledPin4, ledState4);

  pinMode(ledPin5, OUTPUT);
  digitalWrite(ledPin5, ledState5);

  pinMode(ledPin6, OUTPUT);
  digitalWrite(ledPin6, ledState6);
  delay(1000);

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
#ifdef ESP32
  client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org
#endif
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());

  bot.sendMessage(CHAT_ID, "Bot Started", "");
}

void loop() {
  delay (2000);

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
}

```

```

}

int light = analogRead(LDR);
if(light < 150)
{
  digitalWrite(ledPin6, HIGH);
}
else
{
  digitalWrite(ledPin6, LOW);
}
delay(100);

if (millis() > lastTimeBotRan + botRequestDelay) {
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

  while (numNewMessages) {
    Serial.println("got response");
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  }
  lastTimeBotRan = millis();
}
}
}

```

14.3. Circuito Eletrônico

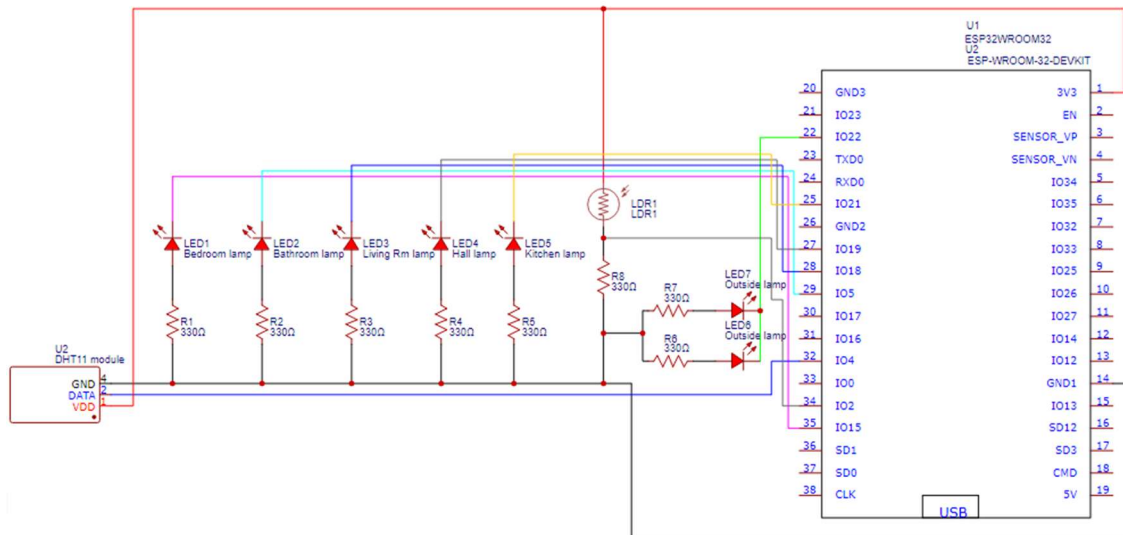


Figura 14 - Circuito eletrônico

Este circuito foi feito baseado no protótipo, mas em um meio realista, colocaríamos o relé, tiraríamos os resistores e trocaríamos os LEDs por

lâmpadas fazendo a ligação com cabos de 2,5mm² que iria funcionar perfeitamente.

15. Resultados Obtidos

Podemos obter resultados relativamente bons pelo curto tempo de trabalho que foi exercido (42 dias) após a mudança repentina do projeto, com apenas um problema em relação a parte automatizada com o LDR que funcionou por um momento e depois não voltou mais, porém ainda é funcional manualmente.

Simulando um cômodo de uma casa (quarto) na maquete obtemos o resultado a seguir:



Figura 15 - Maquete para o protótipo

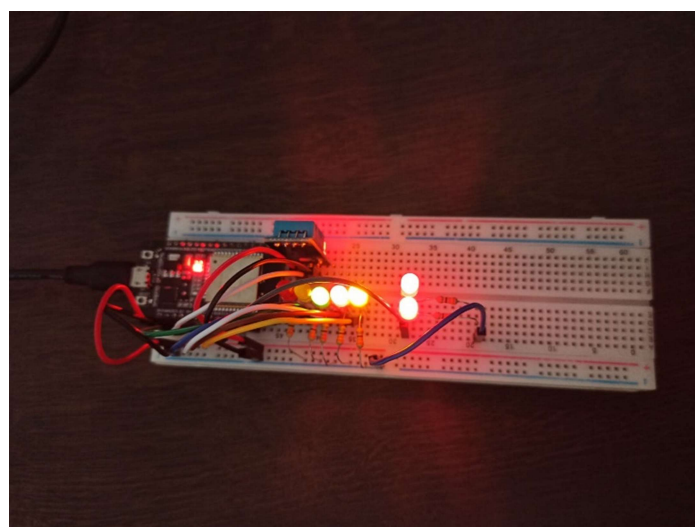


Figura 16 - Protótipo na matriz de contato

Circuito do projeto na matriz de contato funcionando

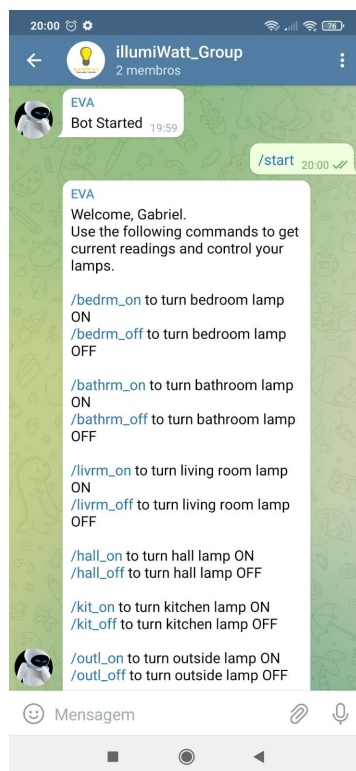


Figura 17 - Interface do projeto pelo Telegram - Parte I

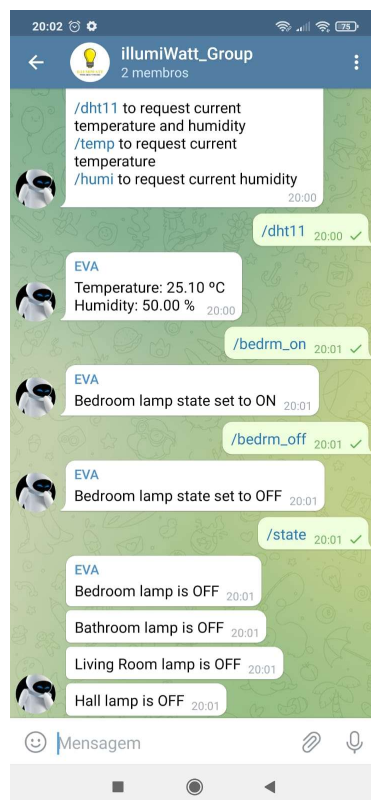


Figura 18 - Interface do projeto pelo Telegram - Parte II

Essa é a interface pelo Telegram na qual o usuário interage e inicialmente digitando `/start`, o Bot informará todos os comandos possíveis.

16. Conclusão

Esse projeto é voltado para a área de automação residencial, sendo que a área desenvolvida foi na criação de um sistema de monitoração, onde o mesmo denominado ILLUMIWATT atua para a facilitação, acessibilidade e praticidade do indivíduo. Na implementação do projeto foi utilizado o ESP32 com linguagem de programação em C, responsável por receber as leituras dos sensores e após a intervenção do usuário executar os respectivos comandos selecionados. O sistema se mostrou muito satisfatório em seu funcionamento sendo muito versátil e útil, podendo ser utilizado em redes residenciais e empresariais.

17. Referências

- ArduinoLivre.* (s.d.). Fonte: <https://www.youtube.com/watch?v=btBfTapYx1o>
- Infortrônica Para Zumbis.* (s.d.). Fonte: <https://www.youtube.com/watch?v=t103WbEr2Kc>
- Lobo da Robótica.* (s.d.). Fonte: <https://www.youtube.com/watch?v=GpYtwlu9KoY>
- Mendes, A. (s.d.). *Antonio Mendes.* Fonte: <https://www.youtube.com/watch?v=D-FX4VSoZCA>
- Santos, R. (s.d.). *Rui Santos.* Fonte: <https://www.youtube.com/watch?v=tDdL5urWvH4>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/esp32-door-status-telegram/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/telegram-control-esp32-esp8266-nodemcu-outputs/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/telegram-group-esp32-esp8266/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/telegram-request-esp32-esp8266-nodemcu-sensor-readings/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/complete-guide-for-dht11dht22-humidity-and-temperature-sensor-with-arduino/>
- Santos, R., & Santos, S. (2013 - 2021). *Random Nerd Tutorials.* Fonte: <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>
- WR Kits.* (s.d.). Fonte: <https://www.youtube.com/watch?v=onVxqTngED8>