

Universidade Anhembi Morumbi

Ellen Cristina

Gabriel Jorge

Ivan Rocha

João Bispo

Tainá Correia

Projeto de Arquitetura e Organização de Computadores
Automação Residencial com Arduino

São Paulo

2017

Ellen Cristina

Gabriel Jorge

Ivan Rocha

João Bispo

Tainá Correia

Projeto de Arquitetura e Organização de Computadores
Automação Residencial com Arduíno

Trabalho apresentado à disciplina Arquitetura e
Organização de Computadores Faculdade:
Universidade Anhembi Morumbi São Paulo, São
Paulo

Professor: Ricardo Souza Jacomini

São Paulo

2017

Funcionamento do Projeto:

- Acendendo uma lâmpada com sensor de presença:

Esse sensor de presença com diversos componentes pode ser utilizado para diversas funções como: acionar um portão, acender uma lâmpada, tocar uma campainha ou acionar qualquer outro dispositivo, por meio de relés. Utilizaremos nesse projeto o eficiente módulo sensor de movimento PIR, um módulo compacto com sensor infravermelho e ajustes de sensibilidade e tempo de acionamento.

Este módulo permite o ajuste da sensibilidade do sensor, ou seja, à qual distância um objeto será detectado (3 à 7 metros) , e também o tempo de delay (tempo que a saída permanece acionada, em nível alto), de 5 a 200 segundos. Os ajustes são feitos nos 2 potenciômetros soldados à placa. Observando a foto, vemos que o da esquerda ajusta a sensibilidade, e o da direita ajusta o tempo

O sensor aceita alimentação de 4,5 à 20V, e a conexão com o Arduino utiliza apenas um pino, que apresenta o estado HIGH (alto), ao detectar um movimento, e LOW (baixo) quando não há movimentação perto do sensor. Maiores informações sobre o funcionamento do módulo podem ser obtidas no datasheet do produto, nesse link.

Para utilização com um relé, precisaremos construir um circuito de proteção para evitar danos ao Arduino. Se você utilizar um módulo relé, o circuito de proteção não será necessário, pois ele já está embutido no módulo. Aproveite e confira um post usando este sensor de presença PIR para Acionar lâmpadas com módulo relé arduino.

Neste circuito, estamos utilizando uma lâmpada ligada à rede elétrica (110 volts). Ao montar o circuito, confira com cuidado a ligação dos componentes para evitar choques.

IMPORTANTE: Alguns módulos tem a pinagem invertida, com o Vcc (5V) no lado esquerdo e o GND no lado direito, ao contrário da imagem mostrada abaixo. Confira a pinagem do seu módulo removendo a tampa superior e observando a ligação correta, evitando assim a queima do componente.

Lembre-se que dependendo do tipo de relé que você estiver utilizando, a disposição dos pinos pode variar. No circuito, **NO** corresponde ao pino **Normal Open (Aberto)**, **NC** ao **Normal Close (Fechado)**, e **C** ao **comum**. Siga a mesma ligação ao relé que você estiver usando.

O programa abaixo verifica se o pino 7 (entrada do sensor de movimento) foi acionado, ou seja, se está em estado alto (HIGH), e então aciona o pino 2 (saída para o relé), que por sua vez aciona o relé e acende a lâmpada.

Opcionalmente, você pode acrescentar um **LDR** (resistor sensível à luz) ao circuito para que a lâmpada seja acionada apenas durante a noite, evitando desperdício de energia

- Mostrando informações de temperatura no LCD 16X2 com o DHT11:

Nesse projeto o intuito é mostrar a temperatura e a umidade de cada cômodo da casa no LCD através de um código, esse projeto mostra a temperatura exata, por exemplo: 28,5° ou arredondará para 29° ou manterá em 28°.

Em um país tropical como o nosso o clima em boa parte do Brasil é quente e úmido. Logo creio que este projeto irá te ajudar a monitorar com apenas 1 sensor a temperatura e umidade de seu clima local com este [Sensor DHT11](#).

Este sensor inclui um componente medidor de umidade e um componente NTC para temperatura, ambos conectados a um controlador de 8-bits. O interessante neste componente é o protocolo usado para transferir dados entre o MCU e DHT11, pois as leituras do sensor são enviadas usando apenas um único fio de barramento.

Formato dos dados: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum = 40 bits.

O DHT11 possui 4 terminais sendo que somente 3 são usados: GND, VCC e Dados. Se desejar, pode-se adicionar um resistor pull up de 10K entre o VCC e o pino de dados.

Conecte o pino de dados do DHT11 ao pino 2 do seu Arduino Uno como mostra o código exemplo abaixo, mas você poderá alterar por outro se desejar.

Para facilitar o seu trabalho já existe uma **biblioteca** que pode ser baixada [neste link](#). Após o download descompacte o arquivo .zip e mova-o para a pasta

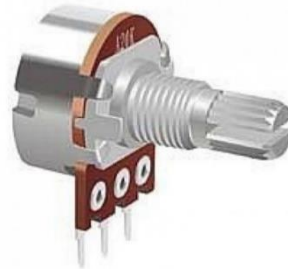
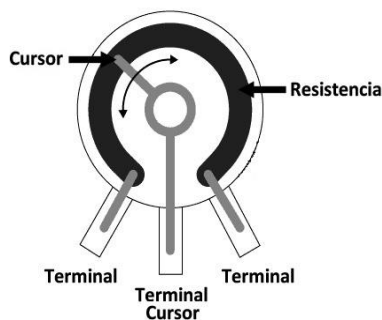
arduinofolder/libraries/ e reinicie a IDE do Arduino. Não retire o arquivo dht.cpp. e não esqueça de renomear a pasta para “**DHT**”. Talvez será necessário criar uma sub-pasta da biblioteca caso não exista.

Se você quiser mostrar estas informações em um display, veja como é fácil seguindo este post com sensor DHT11 e mostrando as informações em um [Display LCD 16x2 com Arduino](#)

Potenciômetro:

Potenciômetro é um componente eletrônico que cria uma limitação para o fluxo de corrente elétrica que passa por ele, e essa limitação pode ser ajustada manualmente, podendo ser aumentada ou diminuída. Os potenciômetros e o resistores tem essa finalidade de limitar o fluxo de corrente elétrica em um circuito, a diferença é que o potenciômetro pode ter sua resistência ajustada e o resistor comum não pode pois ele possui um valor de resistência fixo.

O potenciômetro comumente possui três terminais e um eixo giratório para ajuste da sua resistência, e normalmente são usados em controle de volumes de aparelhos de som, controle de posicionamento em controles de vídeo games, controle de brilho e contraste em telas LCD, etc.



Sensor de Temperatura e Umidade DHT11:

O Sensor de Umidade e Temperatura DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, **muito usado** para projetos com Arduino.

Especificações:

- Modelo: DHT11
- Faixa de medição de umidade: 20 a 90% UR
- Faixa de medição de temperatura: 0° a 50°C

- Alimentação: 3-5VDC (5,5VDC máximo)
- Corrente: 200uA a 500mA, em stand by de 100uA a 150 uA
- Precisão de umidade de medição: $\pm 5,0\%$ UR
- Precisão de medição de temperatura: ± 2.0 °C
- Tempo de resposta: 2s
- Dimensões: 23 x 12 x 5mm (incluindo terminais)



Sensor de Movimento/ Presença – PIR:

O Sensor de Movimento PIR DYP-ME003 consegue detectar o movimento de objetos que estejam em uma área de até 7 metros! Caso algo ou alguém se movimentar nesta área o pino de alarme é ativo.

É possível ajustar a duração do tempo de espera para estabilização do PIR através do potenciômetro amarelo em baixo do sensor bem como sua sensibilidade. A estabilização pode variar entre 5-200 seg.

Especificações:

- Modelo: DYP-ME003
- Sensor Infravermelho com controle na placa
- Sensibilidade e tempo ajustável
- Tensão de Operação: 4,5-20V
- Tensão Dados: 3,3V (Alto) - 0V (Baixo)
- Distância detectável: 3-7m (Ajustável)
- Tempo de Delay: 5-200seg (Default: 5seg)
- Tempo de Bloqueio: 2,5seg (Default)
- Trigger: (L)-Não Repetível (H)-Repetível (Default: H)
- Temperatura de Trabalho: -20 ~ +80°C
- Dimensões: 3,2 x 2,4 x 1,8cm
- Peso: 7g

Conectando:

Conecte uma fonte de 5v ao GND e VCC. O pino DADOS refere-se ao sinal de saída que será 'Alto' indicando movimento ou 'Baixo' indicando nenhuma movimentação.

Quando a saída é acionada pelo movimento detectado esta ficará em alto por um curto período de tempo, mesmo se não haja mais movimento. O tempo em alto pode ser setado variando o potenciômetro 'Time', sendo que o outro altera a sensibilidade.



Sensor de Luminosidade LDR:

O **LDR**, sigla em inglês de Light-Dependent Resistor, que significa resistor dependente de luz, nada mais é do que o que o próprio nome diz. Tipicamente, ***quanto maior a luz incidente nesse componente, menor será sua resistência.***



LDR – sensor de luminosidade

O LDR é constituído de um semicondutor de alta resistência, que ao receber uma grande quantidade de fótons oriundos da luz incidente, ele absorve elétrons que melhoram sua condutibilidade, reduzindo assim sua resistência.

Dessa forma, esse semicondutor pode assumir resistências na ordem de mega Ohm no escuro e resistência na ordem de poucas centenas quando exposto a luz.

Display LCD 16X2:

O Display LCD 16x2 Backlight Azul. São 16 colunas por 2 linhas, backlight azul e escrita branca. Possui o controlador HD44780 usado em toda indústria de LCD's como base de interface.

A interface com Arduino é muito simples, sendo basicamente 4 pinos de dados e 2 de controle.

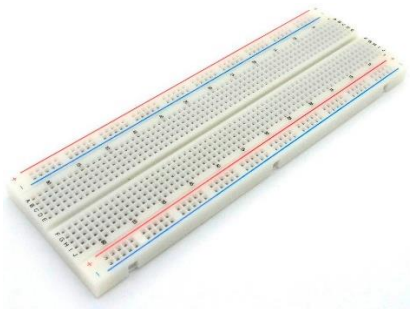
Especificações:

- Cor backlight: Azul
- Cor escrita: Branca
- Dimensão Total: 80mm X 36mm X 12mm
- Dimensão Área visível: 64.5mm X 14mm
- Dimensão Caracter: 3mm X 5.02mm
- Dimensão Ponto: 0.52mm X 0.54mm



Protoboard 830 Pontos:

A protoboard, também conhecida como breadboard, é utilizada para desenvolvimento de circuitos de teste ou protótipo.



Características:

- Marca: Minipa MP-830
- Número de Furos: 830
- Material do Corpo: ABS (resistente até 90°C)
- Material da Base: Alumínio
- Material do Contato: Bronze Fosforo
- Acabamento do Contato: Banho de Níquel
- Bitola do Fio: 0,41 ~ 0,81mm (20-29 AWG)
- Tensão Máxima: 300V RMS
- Corrente Máxima: 3A RMS
- Dimensões: 165(A) x 54(L) x 10(P)mm
- Peso: 103g

Arduino Uno R3 + Cabo USB:

O Arduino Uno R3 é a placa Arduino mais vendida e usada atualmente. Costuma ser a primeira opção de muitos, pois apresenta uma ótima quantidade de portas disponíveis e grande compatibilidade com os Shields Arduino.



. Especificações Uno R3:

- Microcontrolador: ATmega328
- Tensão de Operação: 5V
- Tensão de Entrada: 7-12V
- Portas Digitais: 14 (6 podem ser usadas como PWM)
- Portas Analógicas: 6
- Corrente Pinos I/O: 40mA
- Corrente Pinos 3,3V: 50mA
- Memória Flash: 32KB (0,5KB usado no bootloader)
- SRAM: 2KB
- EEPROM: 1KB
- Velocidade do Clock: 16MHz

Rele de 2 Canais/ Entradas:

Este Módulo Relé 5V com 2 canais é a alternativa perfeita pra quem busca um módulo compacto e de qualidade para projetos com Arduino e outros controladores. Com este módulo você consegue fazer acionamento de cargas de 200V AC, como lâmpadas, equipamentos eletrônicos, motores, ou usá-lo para fazer um isolamento entre um circuito e outro.

Especificações:

- Modelo: SRD-05VDC-SL-C
- Tensão de operação: 5VDC
- Permite controlar cargas de 220V AC
- Corrente típica de operação: 15~20mA
- 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: 5~10ms
- Dimensões: 51 x 38 x 20mm
- Peso: 30g



Código-Fonte:

// Bibliotecas Auxiliares

#include <LiquidCrystal.h> // Biblioteca para Display LCD

#include <DHT.h> // Biblioteca para Sensor DHT11(Temperatura)

// Portas Analógicas

**#define LDR A0 //define o pino analógica que está o sensor LDR
(luminosidade)**

**#define DHTPIN A5 //define o pino analógica que está o sensor DHT11
(temperatura)**

// Portas Digitais

#define rele1 7 // define a porta que está o relé 1

#define rele2 10 // define a porta que está o relé 2

#define Sensor_PIR 13 // define a porta que está o sensor PIR (presença)

**//declaração das variáveis de leitura do LDR (sensor de luz) e DHT11 (sensor de
temperatura)**

int valorLDR;

int TempEntrada;

float valorDHT11_Temp;

```
//declaração de variável para ler sensor PIR
```

```
boolean valorPIR;
```

```
double leituraSensor;
```

```
// Cria objeto dht da classe DHT
```

```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
// Cria objeto lcd da classe LiquidCrystal
```

```
LiquidCrystal lcd(12, // RS
```

```
11, // Enable (E)
```

```
5, // DB4
```

```
4, // DB5
```

```
3, // DB6
```

```
2); // DB7
```

```
void setup() {
```

```
Serial.begin(9600);
```

```
lcd.begin(16, 2);
```

```
dht.begin();
```

```
pinMode(rele1, OUTPUT);
```

```
pinMode(rele2, OUTPUT);

pinMode(Sensor_PIR, INPUT);


lcd.clear();

}

void loop() {

    TempEntrada = analogRead(DHTPIN); // Faz leitura da porta A5 e já faz
    conversão de dados para temperatura

    valorDHT11_Temp = (500*TempEntrada)/1023;

    valorLDR = analogRead(LDR); // Faz leitura da porta A0 (LDR)

    valorLDR = map(valorLDR, 0, 1023, 0, 100); // Faz mapeamento de dados
    equivalentes a 0 - 100 % de luminosidade


    // fazendo leitura do DHT11 a cada 2 segundos

    if (millis() - leituraSensor >= 2000 )

    {

        valorDHT11_Temp = dht.readTemperature();


        // Imprime na serial os 2 sensores a cada 2s.

        Serial.print("Luminosidade(LDR): "); Serial.print(valorLDR);
        Serial.println("%");

        Serial.print("Temperatura(DHT11): "); Serial.print(valorDHT11_Temp);
        Serial.println("°C");
```

```
    leituraSensor = millis();
}

// Se luminosidade atingir 50%, ligará o relé1
if (valorLDR > 50) {
    digitalWrite(rele1, HIGH);
}

    else digitalWrite(rele1, LOW);

// Se detectar movimento, liga relé2 se não, deixa apagado
if(valorPIR) {
    digitalWrite(rele2, HIGH);
}

    else digitalWrite(rele2, LOW);

// exibe a porcentagem de luz no LCD
lcd.setCursor(0, 0);
lcd.print("Luz: ");
lcd.print(valorLDR);
lcd.println("%    ");

// exibe a temperatura no LCD
lcd.setCursor(0, 1);
lcd.print("Temp: ");
lcd.print(valorDHT11_Temp);
```

```
lcd.write(B11011111); //Simbolo de graus celsius  
lcd.println("C    ");  
  
}
```

Fluxograma do Projeto:

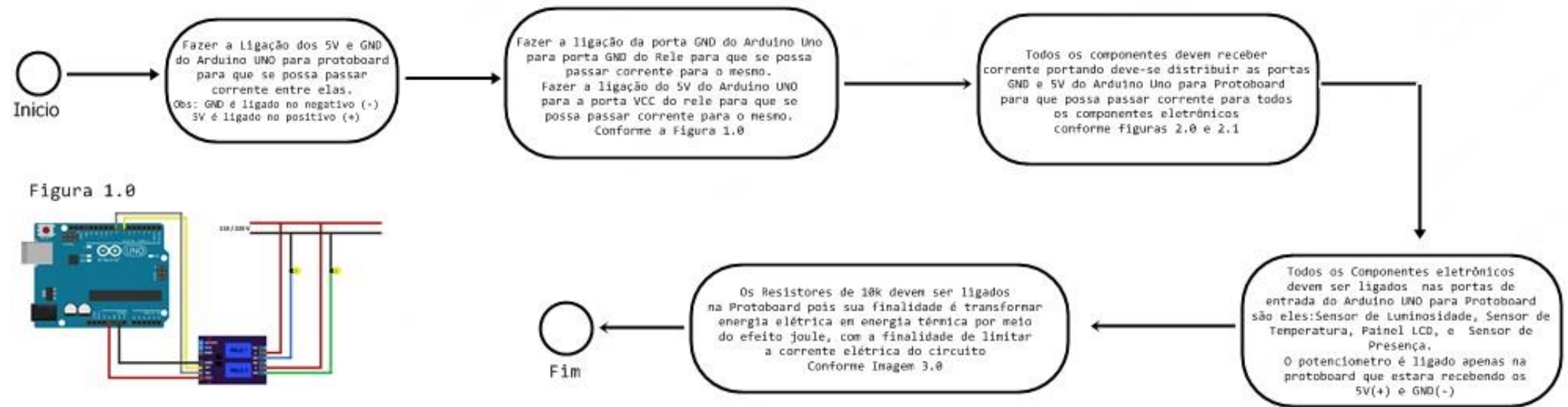


Figura 1.0

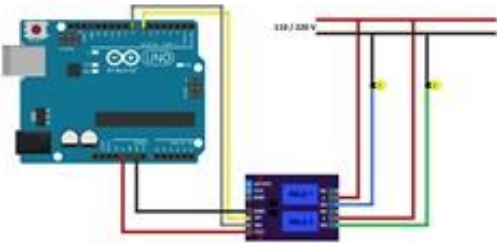


Figura 2.0

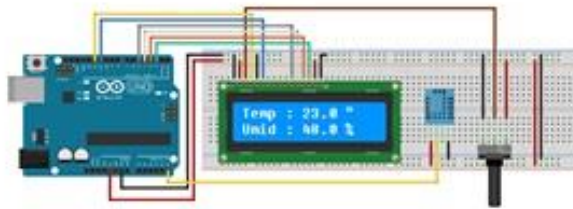


Figura 2.1

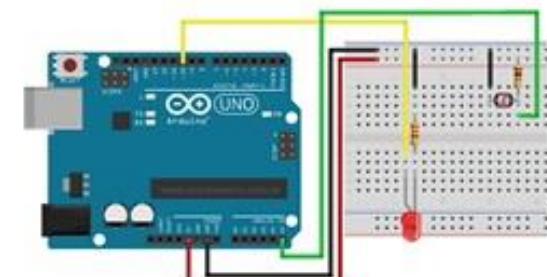
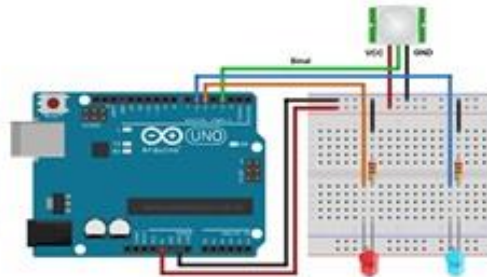


Figura 3.0