

Global Solution 2024 AICSS - Grupo 24

Integrantes:

Ciro Vioto

Enyd Bentivoglio

Marco Antonio Franzoi

Rodrigo Mazuco

Introdução

O programa apresentado é uma solução integrada para automação de iluminação baseada em sensores. Ele utiliza um conjunto de dispositivos eletrônicos, incluindo um sensor de luminosidade (LDR), um sensor ultrassônico (HC-SR04), e um sensor de movimento (PIR), para controlar de forma inteligente duas lâmpadas: uma externa e outra interna.

A lâmpada **externa** é projetada para operar em dois níveis de intensidade, dependendo das condições ambientais:

- **Baixa intensidade** (40%) quando o ambiente está escuro, mas sem detecção de objetos próximos.
- **Alta intensidade** (100%) ao detectar a presença de um objeto a menos de 15 cm de distância, indicando aproximação, também em ambiente escuro.

Já a lâmpada **interna** é controlada com base na presença de movimento detectada pelo sensor PIR, mas apenas em condições de baixa luminosidade, para economizar energia durante o dia.

Descrição

1. Sensor LDR (Luminosidade):

- Mede o nível de luz ambiente.
- As lâmpadas interna e externa são ativadas apenas quando a luminosidade está abaixo de um limite configurado (10 lux).

2. Sensor Ultrassônico HC-SR04:

- Mede a distância de objetos próximos.
- Ajusta a intensidade da lâmpada externa para 100% quando detecta objetos a menos de 15 cm.

3. Sensor PIR (Movimento):

- Detecta movimento humano no ambiente interno.
- Liga a lâmpada interna em ambientes escuros ao identificar presença.

4. Controle Inteligente de LEDs:

- As lâmpadas internas e externas são desligadas automaticamente quando as condições para ativação não são atendidas.

Essa solução combina eficiência energética e conforto, tornando-a ideal para residências ou espaços externos onde a automação inteligente de iluminação é desejável. O uso modular de sensores e funções no código facilita ajustes e personalizações futuras, garantindo flexibilidade na adaptação para diferentes necessidades.

Códigos fonte:

Setch.ino em C++

```
#define ldrPin 34    // Pino ADC conectado ao LDR

#define trigPin 5    // Pino GPIO conectado ao Trig do HC-SR04

#define echoPin 18    // Pino GPIO conectado ao Echo do HC-SR04

#define pinLED 23     // Pino GPIO conectado ao resistor de LED

#define pinLEDLight 22    // Pino GPIO conectado ao resistor de LED (lampada)

#define pinLEDLightHome 21 // Pino GPIO conectado ao resistor de LED (interno)

#define pinPIR 17

const float GAMMA = 0.7;

const float RL10 = 50;

void setup() {

    Serial.begin(115200); // Inicia a comunicação serial

    Serial.println("Inicializando...");

    // Configuração do HC-SR04

    pinMode(trigPin, OUTPUT); // Define o pino Trig como saída

    pinMode(echoPin, INPUT); // Define o pino Echo como entrada

    Serial.println("HC-SR04 configurado");
```

```
// Configuração do PIR
```

```
pinMode(pinPIR, INPUT); //Define o pino como entrada
```

```
Serial.println("PIR configurado");
```

```
// Configuração dos LEDs
```

```
pinMode(pinLED, OUTPUT); // Define o pino do LED como saída
```

```
pinMode(pinLEDLight, OUTPUT); // Define o pino de lâmpada externa
```

```
pinMode(pinLEDLightHome, OUTPUT); // Define o pino de lâmpada
```

```
Serial.println("LED configurado");
```

```
}
```

```
float executeMeasurement(int minValue, int maxValue) {
```

```
    return minValue + (rand() % (maxValue - minValue + 1)) + (rand() % 100) / 100.0;
```

```
}
```

```
void loop() {
```

```
    Serial.println("Loop iniciado");
```

```
    // Leitura do HC-SR04
```

```
    long duration;
```

```
    int distance;
```

```
// Envia pulso de 10 microsegundos para o trig
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Calcula o tempo de duração do pulso de retorno
duration = pulseIn(echoPin, HIGH);

//distance = 15;
distance = executeMeasurement(0, 200);

// Exibe a distância no monitor serial
Serial.print("Distância medida: ");
Serial.print(distance);
Serial.println(" cm");

bool isDistance = (distance < 100);

// Acende o LEDLight se o objeto estiver a uma distância menor que 20 cm
if (isDistance == true) {
    Serial.println("Objeto se aproximando! Lampada externa ligada!");
    digitalWrite(pinLED, HIGH);
} else {
    Serial.println("Nenhum objeto próximo. Lampada externa desligada.");
}
```

```

    digitalWrite(pinLED, LOW);
}

// Leitura do LDR

int ldrValue = analogRead(ldrPin);

float voltage = ldrValue / 1024. * 5;

float resistance = 2000 * voltage / (1 - voltage / 5);

float lux = pow(RL10 * 1e3 * pow(10, GAMMA) / resistance, (1 / GAMMA));

// Simulação de valores aleatórios

lux = executeMeasurement(0, 100.000);

Serial.print("Nível de Luz ( lux ): ");

Serial.println(lux); // Exibe o valor lido no monitor serial

if(lux < 10) {

    float percentualLight = 40.00;

    if(isDistance == true) {

        percentualLight = 100.00;

    }

    // Executar ação para ligar a lâmpada externa

```

```
digitalWrite(pinLEDLight, HIGH);
```

```
Serial.print("Potência da lâmpada externa: ");
```

```
Serial.print(percentualLight);
```

```
Serial.println(" %");
```

```
}
```

```
else {
```

```
digitalWrite(pinLEDLight, LOW);
```

```
}
```

```
// Leitura do PIR
```

```
int pirState = digitalRead(pinPIR);
```

```
if (lux < 10) { // Está escuro
```

```
    if (pirState == HIGH) { // Corpo detectado
```

```
        digitalWrite(pinLEDLightHome, HIGH); // Liga o LED
```

```
        Serial.println("Movimento detectado! Lampada interna ligada.");
```

```
    } else {
```

```
        digitalWrite(pinLEDLightHome, LOW); // Desliga o LED
```

```
        Serial.println("Sem movimento. Lampada interna desligada.");
```

```
    }
```

```
} else {
```

```
    // Está claro, desliga o LED independentemente do movimento
```

```
digitalWrite(pinLEDLightHome, LOW);

Serial.println("Está claro. Lampada interna desligada.");

}
```

```
delay(1500);

}
```

Diagram.json

```
{

  "version": 1,

  "author": "Anonymous maker",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 48, "left": 43.24, "attrs": {} },

    { "type": "wokwi-led", "id": "led1", "top": -90, "left": -169, "attrs": { "color": "red" } },

    {

      "type": "wokwi-resistor",

      "id": "r1",

      "top": -53.65,

      "left": -96,

      "attrs": { "value": "100" }

    },

    {

      "type": "wokwi-photoresistor-sensor",

      "id": "ldr1",
```

```
"top": 108.8,

"left": -172,

"attrs": {}

},

{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -113.7, "left": 82.3, "attrs": {} },

{

  "type": "wokwi-led",

  "id": "led2",

  "top": 34.8,

  "left": 330.2,

  "attrs": { "color": "white" }

},

{

  "type": "wokwi-resistor",

  "id": "r2",

  "top": 99.95,

  "left": 249.6,

  "attrs": { "value": "100" }

},

{

  "type": "wokwi-led",

  "id": "led3",

  "top": 121.2,

  "left": 339.8,

  "attrs": { "color": "yellow" }

},
```



```

{
  "type": "wokwi-resistor",
  "id": "r3",
  "top": 167.15,
  "left": 259.2,
  "attrs": { "value": "100" }
},
{ "type": "wokwi-pir-motion-sensor", "id": "pir1", "top": 244, "left": 223.02, "attrs": {} }
],
"connections": [
  [ "esp:TX", "$serialMonitor:RX", "", [] ],
  [ "esp:RX", "$serialMonitor:TX", "", [] ],
  [ "ldr1:AO", "esp:34", "green", [ "h38.4", "v-29.5" ] ],
  [ "ldr1:GND", "esp:GND.1", "black", [ "h38.4", "v66.8" ] ],
  [ "led1:A", "r1:1", "red", [ "v0" ] ],
  [ "led1:C", "esp:GND.2", "black", [ "v86.4", "h298" ] ],
  [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],
  [ "ldr1:VCC", "esp:3V3", "red", [ "h28.8", "v-48" ] ],
  [ "ultrasonic1:VCC", "esp:5V", "red", [ "v0" ] ],
  [ "ultrasonic1:ECHO", "esp:18", "gold", [ "v0" ] ],
  [ "ultrasonic1:TRIG", "esp:5", "green", [ "v0" ] ],
  [ "r1:2", "esp:23", "red", [ "v48", "h181.2" ] ],
  [ "led2:A", "r2:2", "red", [ "v0" ] ],
  [ "led2:C", "esp:GND.2", "black", [ "v0" ] ],
  [ "r2:1", "esp:22", "red", [ "v0" ] ],
  [ "led3:A", "r3:2", "red", [ "v0" ] ],

```

```
[ "r3:1", "esp:21", "red", [ "v0" ] ],

[ "led3:C", "esp:GND.2", "black", [ "v0" ] ],

[ "pir1:GND", "esp:GND.1", "black", [ "v0" ] ],

[ "pir1:VCC", "esp:5V", "red", [ "v0" ] ],

[ "pir1:OUT", "esp:17", "green", [ "v0" ] ]

],

"dependencies": {}

}
```

Conclusão e discussões

Por tratar-se de uma solução modular, é flexível e de fácil integração, proporcionando conforto e segurança. A possibilidade de ter a variação de intensidade, abandonando a necessidade de utilização em 100% para a iluminação do ambiente externo, o consumo de energia, quando comparado a soluções que não oferecem tal programação, é otimizado. Contudo, os sensores de movimento não distinguem entre pessoas e animais, proporcionando assim um obstáculo para uma melhor eficiência energética.

Projetos mais robustos, com a inclusão de cameras e células de energia solar poderiam refinar a presente solução, entregando melhorias ao meio ambiente e a economia do consumidor.

LINK WOKWI: <https://wokwi.com/projects/414803136406705153>

LINK YOUTUBE: <https://youtu.be/ppN4fz2IAOE>



