



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**

**Faculdade de Engenharia Elétrica**

**Engenharia de Controle e Automação**

Lucas Gonçalves e Silva – 11811EAU016  
Ítalo Marangoni de Souza– 11811EAU014

**Sistema de posicionamento de painéis fotovoltaicos  
utilizando arduino**

**Uberlândia**

**2018**

Lucas Gonçalves e Silva – 11811EAU016

Ítalo Marangoni de Souza– 11811EAU014

## **Sistema de posicionamento de painéis fotovoltaicos utilizando arduino**

Trabalho apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para obtenção dos créditos na disciplina Introdução à Engenharia de Controle e Automação do Curso de Eng. de Controle e Automação.

Área: Ciclo básico

Professor: Josué Silva de Moraes

# Uberlândia

2018

## Resumo

Este trabalho tem por objetivo colocar em pauta o estudo e a construção de um protótipo que visa melhorar o desempenho da captação de energia solar pelos painéis fotovoltaicos através de um microcontrolador e um sistema de posicionamento. O projeto traz uma forma de energia renovável em crescente desenvolvimento que demanda de altos investimentos em sua aplicação, também busca reduzir seu custo de aplicação e tornar mais acessível à população. O sistema é formado por um hardware que controla a posição do painel solar de acordo com dados coletados através de sensores de luminosidade, e por um software embarcado, construído para o único propósito da sua aplicação. Neste trabalho foi utilizado uma estrutura mecânica controlada por um circuito eletrônico, através do software os movimentos da estrutura tendem corrigir o posicionamento e a inclinação das placas solares, permitindo maior incidência dos raios solares sobre o painel.

*Palavras-chave:* energia solar, microcontrolador, painel fotovoltaico.

## **Abstract**

The objective of this work is to study and construct a prototype that aims to improve the performance of solar energy capture by photovoltaic panels through a microcontroller and a positioning system. The project brings a form of renewable energy in growing development that demands high investments in its application, also seeks to reduce its cost of application and make it more accessible to the population. The system consists of hardware that controls the position of the solar panel according to data collected through light sensors, and embedded software, built for the sole purpose of its application. In this work, a mechanical structure controlled by an electronic circuit was used. Through the software, the movements of the structure tended to correct the positioning and inclination of the solar panels, allowing a greater incidence of solar rays on the panel.

*Keywords:* solar energy, microcontroller , photovoltaic panel.

## Sumário

<b>1- Introdução.....</b>	<b>Pág 01</b>
<b>2- Materiais.....</b>	<b>Pág 02</b>
2.1. Arduino UNO.....	Pág 02
2.2. IDE Arduino.....	Pág 03
2.3. Protoboard.....	Pág 04
2.4. Sensor LDR.....	Pág 04
2.5. Servo Motor.....	Pág 05
2.6. Estrutura.....	Pág 05
<b>3- Métodos.....</b>	<b>Pág 06</b>
3.1. Desenvolvimento do Circuito Eletrônico.....	Pág 06
3.2. Desenvolvimento do Software.....	Pág 06
<b>4- Resultados da Implementação.....</b>	<b>Pág 07</b>
<b>5- Conclusões.....</b>	<b>Pág 07</b>
<b>6- Referências.....</b>	<b>Pág 09</b>
<b>Anexo A (Código fonte do programa para o Arduino) .....</b>	<b>Pág 10</b>

## 1. Introdução

Fontes de energia são fundamentais para o funcionamento da sociedade. Em função de alterações climáticas, existe um debate amplo relacionado à diversificação da matriz energética e à adoção de fontes de energia renováveis.

A procura por novas fontes renováveis de energia surge como alternativa importante para superar dois problemas atuais: a escassez de fontes não renováveis de energia, principalmente do petróleo, e a poluição ambiental causada por essas fontes (combustíveis fósseis).

O Brasil possui expressivo potencial para geração de energia elétrica a partir de fonte solar, contando com níveis de irradiação solar superiores aos de países onde o investimento em projetos para aproveitamento de energia solar são maiores.

Apesar dos altos níveis de irradiação solar no território brasileiro, o uso da fonte para geração de energia elétrica não apresenta a mesma relevância que possui em outros países, nem o mesmo desenvolvimento de outras fontes renováveis, como eólica e biomassa, que já representam, respectivamente, 6,7% e 9,4% da capacidade de geração instalada no Brasil, contra apenas 0,05% da fonte solar.

Considerando o compromisso de ampliar a participação de fontes renováveis na matriz energética, principalmente fonte solar, o presente estudo visa apresentar uma alternativa viável e que demanda de investimentos menores para melhorar o rendimento da captação de energia solar, de forma automatizada.

## 2. Materiais

Nesta seção serão apresentados conceitos, definições, técnicas e equipamentos que serviram de base no desenvolvimento deste protótipo.

### 2.1. Arduino UNO

O Arduino é uma plataforma open-source e possui seu próprio ambiente de desenvolvimento baseado na linguagem C, deixando a programação bem intuitiva para iniciantes. Existem várias versões da placa Arduino como o Arduino Uno, Arduino Due, Arduino Diecimila, Arduino Leonardo, Arduino Mega 2560, entre outros. Optou-se pela placa Arduino UNO, devido ao seu custo benefício.

A placa pode ser alimentada pela conexão USB ou por uma fonte de alimentação externa feita através do conector Jack com positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites 6V. a 20V., porém se alimentada com uma tensão abaixo de 7V., a tensão de funcionamento da placa, que no Arduino Uno é 5V, pode ficar instável e quando alimentada com tensão acima de 12V, o regulador de tensão da placa pode sobreaquecer e danificar a placa. Dessa forma, é recomendado para tensões de fonte externa valores de 7V. a 12V.

O componente principal da placa Arduino UNO é o microcontrolador [ATMEL ATMEGA328](https://www.makerlab-electronics.com/my_uploads/2017/07/Arduino-UNO-R3-CH340G-02.jpg), um dispositivo de 8 bits da família AVR com arquitetura RISC avançada e com encapsulamento DIP28. Ele conta com 32 KB de Flash (mas 512 Bytes são utilizados pro bootloader), 2 KB de RAM e 1 KB de EEPROM. Pode operar a até 20 MHz, porém na placa Arduino UNO opera em 16 MHz, valor do cristal externo que está conectado aos pinos 9 e 10 do microcontrolador.



Figura 1- Arduino Uno. Disponível em<[https://www.makerlab-electronics.com/my\\_uploads/2017/07/Arduino-UNO-R3-CH340G-02.jpg](https://www.makerlab-electronics.com/my_uploads/2017/07/Arduino-UNO-R3-CH340G-02.jpg)>

## 2.2. IDE Arduino

A programação para o Arduino é feita utilizando a linguagem Wiring, que é derivada da linguagem C/C++ e estruturada da mesma forma que a linguagem de código aberto Processing, que é voltada para criações visuais e interativas. A linguagem Wiring permite a escrita de programas para controlar aparelhos conectados ao Arduino e assim criar todo o tipo de objetos interativos, correspondendo à experiência do usuário através do mundo físico. [MCROBERTS, 2011].

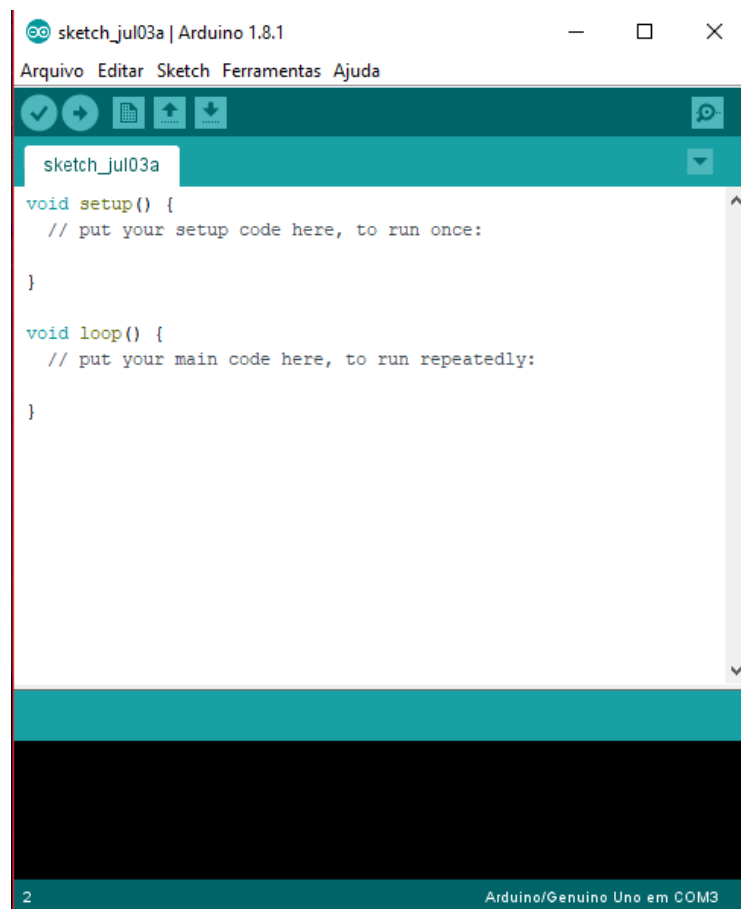


Figura 2- IDE Arduino. Fonte: Autoria própria.

Como pode ser observado, o ambiente de desenvolvimento para Arduino é bem simples, possuindo apenas funcionalidades básicas como compilação, depuração, edição e upload (carregamento) do programa à memória do Arduino.



### 2.3. Protoboard

A protoboard é uma base reutilizável (por não requerer solda) para construção de protótipos eletrônicos, utilizada para criação de protótipos temporários. A utilização de uma protoboard em projetos Arduino torna possível a construção de circuitos mais complexos. A ligação de circuitos e componentes eletrônicos é feita por jumpers (pequenos fios) que interligam as vias da protoboard. Normalmente uma protoboard possui quatro matrizes, mas este número pode variar.

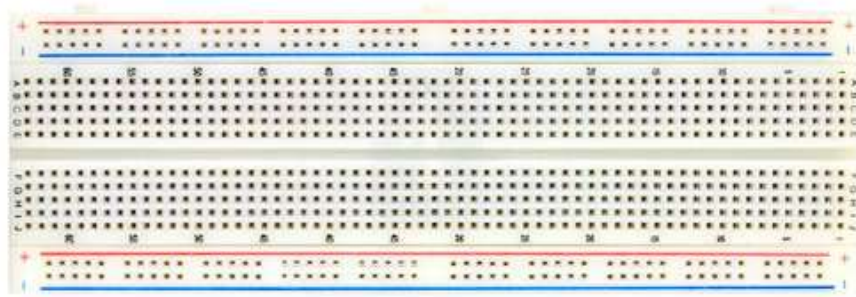


Figura 3- Protoboard. Fonte: Autoria própria.

### 2.4. Sensor LDR

É um sensor que varia sua resistência conforme a intensidade de luz. Quanto mais luz, menor sua resistência. Porém o LDR não fornece uma leitura da quantidade de LUX em um ambiente, somente consegue fornecer a informação se está claro ou escuro. Como o LDR varia sua resistência de acordo com a intensidade da luz, iremos utilizar uma porta analógica do Arduino para ler esta variação. No protótipo serão utilizados quatro destes componentes.

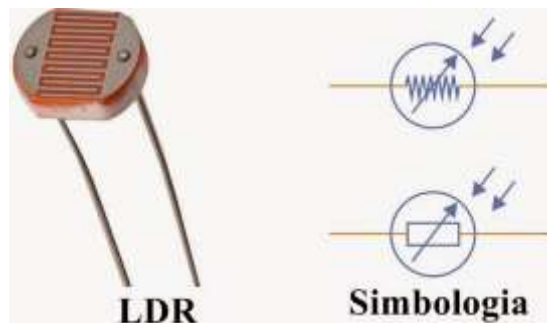


Figura 4- Sensor LDR e Simbologia. Disponível em< [http://3.bp.blogspot.com/-qPpT1\\_2RQM/VHO6wnsaLpI/AAAAAAAAAUA/PWcabl86Phg/s1600/LDR-light-dependent-resistor.jpg](http://3.bp.blogspot.com/-qPpT1_2RQM/VHO6wnsaLpI/AAAAAAAAAUA/PWcabl86Phg/s1600/LDR-light-dependent-resistor.jpg)>

## 2.5. Servo Motor

Ao contrário do motor que gira indefinidamente quando alimentado, o servo motor possui seu movimento restrito à cerca de 180° graus, em contrapartida apresenta alta precisão no posicionamento. Para o controle de posicionamento do painel solar é necessário o uso do servo motor pois suas características conseguem atingir um controle preciso de posição e torque constante. No protótipo serão utilizados dois destes componentes.



Figura 5- Micro Servo Motor 9g. Fonte: Autoria própria.

## 2.6. Estrutura

A estrutura do protótipo foi feita em M.D.F cortado por uma máquina a Laser para atingir uma maior precisão e um melhor acabamento. Além das peças em M.D.F foi necessário o uso de alguns parafusos e porcas para a fixação dos componentes.

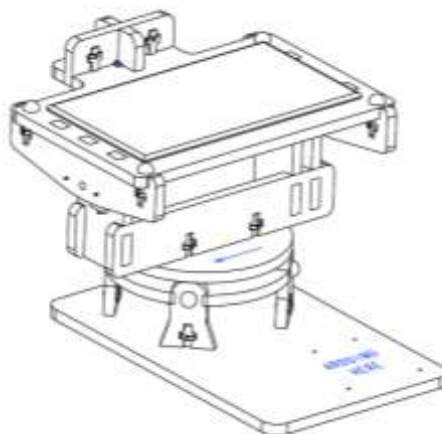


Figura 6- Estrutura Vetorizada. Disponível em<<https://cdn.instructables.com/FQX/ZA2Q/I6QD3JUUFQXZA2QI6QD3JUULARGE.jpg>>

Todo o desenho técnico da estrutura foi encontrado em uma plataforma digital aberta chamada “GitHub”. Muito utilizada por desenvolvedores de projetos e protótipos, desde os mais simples até os mais complexos.

### 3. Métodos

Todos os materiais mencionados no item anterior se unem através de um circuito, o Arduino recebe um software confeccionado na IDE e após a conexão dos componentes a placa, o hardware envia e recebe as informações necessárias para o funcionamento do protótipo.

#### 3.1. Desenvolvimento do Circuito Eletrônico

Inicialmente foi necessário configurar o Arduino para funcionamento do sensor LDR, Servo Motor e o programa para gerenciar suas funções. Após, foi realizada a conexão da placa Arduino aos sensores e aos servos, sendo necessária a utilização de quatro resistores de 10K Ohms ligados um em cada sensor LDR. Os resistores são necessários por permitirem que a resistência elétrica permaneça constante independentemente da tensão ou corrente elétrica que circular pelo dispositivo.

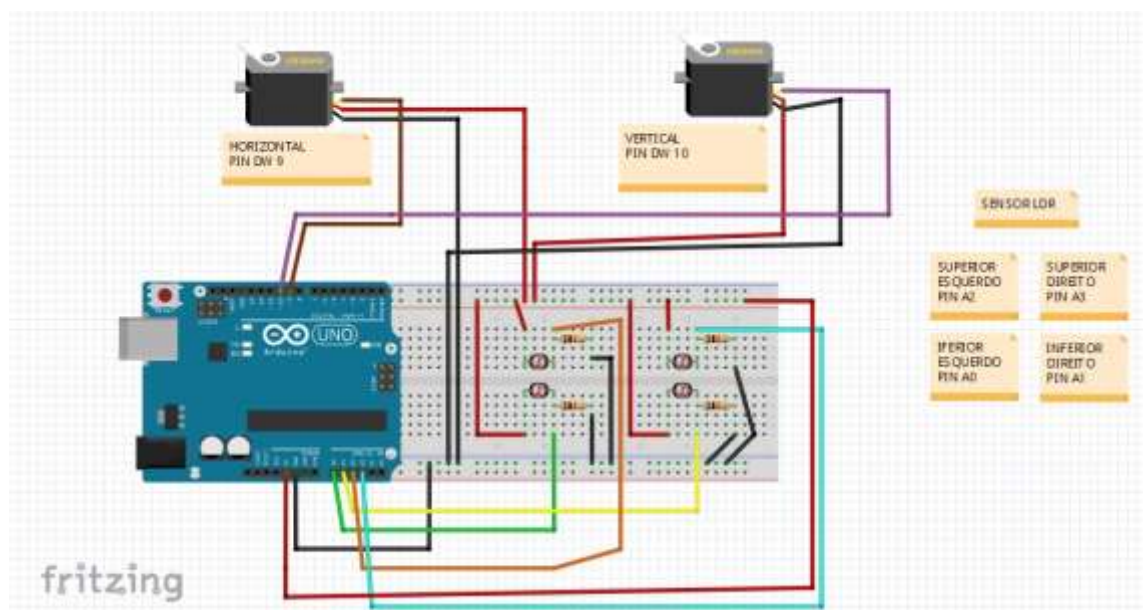


Figura 7- Esquema do Circuito Eletrônico. Fonte: Autoria própria.

A “Figura 7” representa o esquema do circuito eletrônico feito por um programa chamado “Fritzing”. O **Fritzing** é um software em ambiente gráfico que facilita o aprendizado e as primeiras montagens com Arduino, ele mostra de modo virtual como seria uma montagem física em protoboard, bem como o circuito elétrico do projeto em questão.

### **3.2. Desenvolvimento do Software**

Inicialmente foi definido um objetivo, o qual era fazer um software que, através das informações coletadas pelos sensores, buscasse o melhor ângulo de rotação dos servos motores e definisse uma posição favorável para melhor a captação de raios solares pelas células fotovoltaicas.

De acordo com os dados coletados pelos Sensores LDR, o programa faz uma média dos valores e analisa qual dos sensores estão nesta média, como cada sensor está relacionado com um movimento, horizontal ou vertical, o software entende qual Servo Motor deve rotacionar e o quanto ele deve girar. Por final foi adicionado um potenciômetro para calibrar a sensibilidade (tolerância) dos Sensores LDR.

## **4. Resultados de Implementação**

O objetivo principal do módulo foi concluído, a placa conseguiu-se posicionar onde havia maior incidência de luz, conseqüentemente maior captação de raios solares pela célula fotovoltaica. Foi realizado o teste com uma luz de LED, observando os resultados a rotação dos servos motores foi calibrada de acordo com os valores recebidos pelos Sensores LDR. Quanto a estrutura do protótipo foi encontrado um problema em relação ao peso que os servos motores colocados no sistema suportavam e é facilmente resolvido trocando os Servos de engrenagens de plástico, por Servos com engrenagens de metal que suportariam maior peso. Porém para uma real aplicação, estudos mais complexos devem ser feitos a fim de buscar uma maior resistência de acordo com o peso do sistema completo e alterações climáticas no ambiente.

## 5. Conclusões

Com este trabalho pode-se verificar o quanto a plataforma Arduino é um equipamento acessível, de fácil manuseio e eficiente para a realização de inúmeros projetos conseguindo alcançar os objetivos de forma fácil e adequada a situação, como as propostas neste projeto, que visa uma forma de tornar mais acessível uma fonte de energia limpa e renovável.

A utilização de sensores que verificam a incidência de luz (LDR), permite que a posição adequada das placas solares seja ajustada automaticamente. Além da facilidade dada do uso do servo motor, uma vez que não há a necessidade de ir ao local externo onde se encontra o sistema para poder posiciona-lo na posição favorável.

Desse modo, este artigo mostra que, com implantações fáceis e de baixo custo é possível melhorar o rendimento de captação de energia solar pelas estações e residências. Tal projeto é acessível e as próprias pessoas podem adequar seu sistema de captação de energia com base nesse estudo, a fim de trazer maior vantagens para a população e meio ambiente.

Como proposta de melhorias futuras, propõe-se adicionar novas funcionalidades no protótipo, tais como, instalar uma célula fotovoltaica, para colher resultados mais precisos e alimentar o próprio sistema. Além da utilização de um display para informar a voltagem recebida e eficiência da célula fotovoltaica.

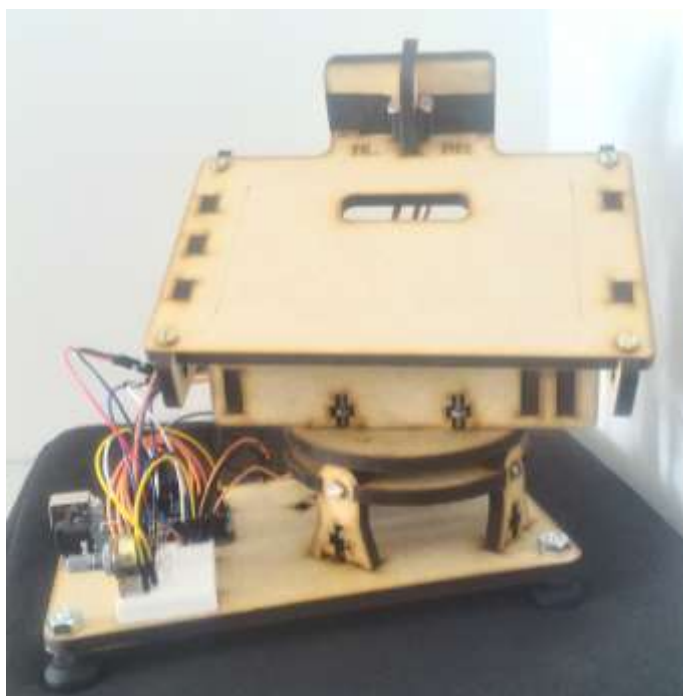


Figura 8- Protótipo finalizado. Fonte: Autoria própria.

## 6. Referências

ARDUINO. Arduino. Disponível em: <http://arduino.cc/> , acessado em Junho, 2018.

COMETTA, E. Energia Solar: Utilização e empregos práticos. São Paulo: Hemus, 2004.

DUTRA, R. M. *et al.* Manual de engenharia para sistemas fotovoltaicos. Rio de Janeiro: CRESESB, 2008.

EMBARCADOS. Sistema Embarcado. Disponível em: <https://www.embarcados.com.br/sistema-embarcado/> , acessado em Julho, 2018.

INSTRUCTABLES. Arduino 2-Axis Solar tracker. Disponível em: <http://www.instructables.com/id/Arduino-2-axis-servo-solar-tracker/>, acessado em Junho, 2018.

INSTRUCTABLES. ARDUINO SUN TRACKER TURRET. Disponível em: <http://www.instructables.com/id/Arduino-Sun-Tracker-Turret/>, acessado em Junho, 2018.

MCROBERTS, M. *Arduino Básico*. São Paulo: Novatec, 2011.  
Volume Único, pp. 18-31.

ROBOCORE. Sensor LRD Arduino. Disponível em: <https://www.robocore.net/tutoriais/sensor-ldr-arduino.html> ,acessado em Julho, 2018

## Anexo A (Código fonte do programa para o Arduino)

**Arquivo: *programa.ino***

```
#include <Servo.h> // inclui a biblioteca do servo

// 180 MÁXIMO horizontal
Servo horizontal; // servo na horizontal
int servoh = 180; // 90; // mantém o servo na horizontal

int servohLimitHigh = 180;
int servohLimitLow = 65;

// 65 graus MÁXIMO
Servo vertical; // vertical servo
int servov = 45; // 90; // mantém o servo na vertical

int servovLimitHigh = 80;
int servovLimitLow = 15;

// conexões pin do LDR
// nome = analogpin;
int ldrlt = 0; //LDR em cima / esquerda - INFERIOR ESQUERDA
int ldrrt = 1; //LDR em cima / direita - INFERIOR DIREITA
int ldrlb = 2; //LDR em baixo / esquerda - SUPERIOR ESQUERDA
int ldrrb = 3; //LDR em baixo / direita - SUPERIOR DIREITA

void setup()
{
  Serial.begin(9600);
  // conexões do servo
  // nome.attach(pin);
  horizontal.attach(9);
  vertical.attach(10);
  horizontal.write(180);
  vertical.write(45);
  delay(3000);
}

void loop()
{
  int lt = analogRead(ldrlt); //LDR em cima / esquerda
  int rt = analogRead(ldrrt); //LDR em cima / direita
  int lb = analogRead(ldrlb); //LDR em baixo / esquerda
  int rb = analogRead(ldrrb); //LDR em baixo / direita

  int tol = analogRead(5)/4; // lê o valor do potenciômetro para calibrar a
  sensibilidade dos sensores
  int dtime = 10; // define o tempo que o programa demora para fazer outra leitura

  int avt = (lt + rt) / 2; // valor médio de cima
  int avb = (lb + rb) / 2; // valor médio de baixo
  int avl = (lt + lb) / 2; // valor médio da esquerda
  int avr = (rt + rb) / 2; // valor médio da direita

  int dvert = avt - avb; // checa a diferença do de cima com o de baixo
  int dhoriz = avl - avr; // checa a diferença do da esquerda com o da direita
```

```

Serial.print(avt);
Serial.print(" ");
Serial.print(avd);
Serial.print(" ");
Serial.print(avl);
Serial.print(" ");
Serial.print(avr);
Serial.print(" ");
Serial.print(dtime);
Serial.print(" ");
Serial.print(tol);
Serial.println(" ");
if (-1*tol > dvert || dvert > tol) // verifica se a diferença está na tolerância
senão muda o ângulo vertical
{
  if (avt > avd)
  {
    servov = ++servov;
    if (servov > servovLimitHigh)
    {
      servov = servovLimitHigh;
    }
  }
  else if (avt < avd)
  {
    servov = --servov;
    if (servov < servovLimitLow)
    {
      servov = servovLimitLow;
    }
  }
  vertical.write(servov);
}

if (-1*tol > dhoriz || dhoriz > tol) // verifica se a diferença está na tolerância
senão muda o ângulo horizontal
{
  if (avl > avr)
  {
    servoh = --servoh;
    if (servoh < servohLimitLow)
    {
      servoh = servohLimitLow;
    }
  }
  else if (avl < avr)
  {
    servoh = ++servoh;
    if (servoh > servohLimitHigh)
    {
      servoh = servohLimitHigh;
    }
  }
  else if (avl = avr)
  {
    // nada
  }
  horizontal.write(servoh);
}
delay(dtime);
}

```