

## PONTO DE CONTROLE 2 - ELETRÔNICA EMBARCADA

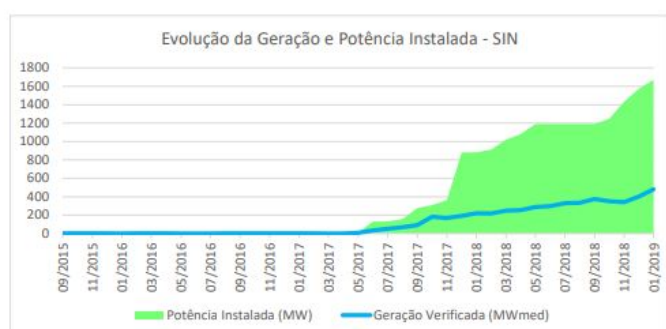
Clara Fonseca da Justa, Hallana Rayssa Alves da Silva

Programa de Graduação em Engenharia Eletrônica, Faculdade Gama  
Universidade de Brasília  
Gama, DF, Brasil  
email: clarajusta31@gmail.com, hallanarayssa@gmail.com

### 1. JUSTIFICATIVA

O Brasil é um país localizado em uma região inter - tropical e possui grande potencial para aproveitamento de energia solar durante todo o ano. Apesar disso, a matriz energética brasileira ainda apresenta a energia hidráulica como principal fonte da geração de eletricidade. Entretanto, nos últimos anos há um intenso esforço para que a geração solar fotovoltaica ocupe um espaço maior na matriz energética.[1]

Dados do Operador Nacional do Sistema Elétrico - ONS, demonstram a evolução da potência instalada e geração de usinas solares fotovoltaicas, desde setembro de 2015 até janeiro de 2019.[2]



**Fig. 1.** Evolução da geração e potência instalada de usinas solares fotovoltaicas.

Esses dados são referentes à geração de usinas solares fotovoltaicas do Sistema Interligado Nacional - SIN, e abrangem apenas as usinas em operação comercial. De acordo com as informações oferecidas pelo ONS e ao analisar a figura 1 é possível verificar que em setembro de 2015, quando as primeiras usinas entraram em operação, a potência instalada foi de 10 MW. Já em março de 2019 a potência instalada foi de 1800 MW. Através dessas informações é possível verificar que houve um crescimento significativo no uso dessa energia limpa.

Além das usinas em operação comercial, há uma outra modalidade de geração de energia solar: os sistemas sola-

res fotovoltaicos de microgeração e minigeração em Unidades Consumidoras - UC (residências, comércios, indústrias, edifícios públicos e na zona rural). Nesta modalidade as unidades que adotam o uso de placa solar, são compensadas pela energia injetada na rede em relação a energia consumida.[3]

Relatórios sobre Unidades Consumidoras com geração distribuída levantados pela CEB, traz informações compiladas e mapas sobre o uso de sistemas fotovoltaicos e fonte solar na região Centro-Oeste. Geração distribuída é o termo utilizado para referenciar a energia elétrica que é gerada próxima ou no local de consumo. Na figura 2 demonstra-se a quantidade de geração distribuída (1.233), o número de unidades consumidoras que utilizam esta energia (1.312) e a potência instalada (16.959,61 kW). Já na figura 3 é possível notar que a adoção desse tipo de sistema vem em uma crescente desde de 2015.

REGIAO	QTD GD	UCs REC CRÉDITOS	POT INSTALADA (kW)
Centro	1.233	1.312	16.959,61
Oeste			
Total	1.233	1.312	16.959,61

**Fig. 2.** Geração de energia solar em Unidades Consumidoras na região centro - oeste.

ANO	QTD GD	UCs REC CRÉDITOS	POT INSTALADA (kW)
2019	497	537	6.734,90
2018	403	424	5.813,08
2017	210	222	3.274,97
2016	87	92	853,35
2015	36	37	283,31
Total	1.233	1.312	16.959,61

**Fig. 3.** Evolução da geração de energia solar em Unidades Consumidoras na região centro - oeste.

Os benefícios desse sistema são muitos, vão desde ao baixo impacto ambiental até a redução de perdas por transmissão e distribuição da energia. Porém, o custo para a

instalação de um sistema fotovoltaico é elevado. O preço médio para a instalação de placas solares para um consumo de 2kWp é de aproximadamente 15.000 reais.[4]

Na tabela abaixo (figura 4) é possível estimar o investimento necessário para a instalação de um sistema fotovoltaico dependendo do tamanho da residência.

Tamanho da Residência	Modelo do Sistema	Preço Médio
Casa pequena, com 2 pessoas	Sistema de 1,32Kwp	R\$ 10.673,36
Casa média, com 3 ou 4 pessoas	Sistema de 2,64Kwp	R\$ 17.570,00
Casa média, com 4 pessoas	Sistema de 3,3Kwp	R\$ 20.320,00
Casa grande, com 4 ou 5 pessoas	Sistema de 4,62Kwp	R\$ 25.695,00
Casa grande, com 5 pessoas	Sistema de 6,6Kwp	R\$ 32.410,00
Mansão, com mais de 5 pessoas	Sistemas de até 10,56Kwp	R\$ 52.240,00

**Fig. 4.** Preço da energia solar fotovoltaica residencial.

Por esse motivo, quanto maior a eficiência das placas solares, mais rápido será o retorno financeiro da instalação. Para a otimização da captação solar, propõem-se a implementação de um dispositivo de baixo custo, um seguidor solar.

## 2. OBJETIVOS

Implementar um sistema de seguidor solar de baixo custo e eficaz em comparação aos que existem.[5]

### 2.1. Objetivos Específicos

- Movimentar dois servos motores para controlar os eixos de rotação em bases de placas fotovoltaicas usando o microcontrolador MSP430 ;
- Monitorar com eficácia dados de sensores de luminosidade e sensores de irradiação solar de forma com que a placa guie-se através do local com maior incidência dos raios solares ;

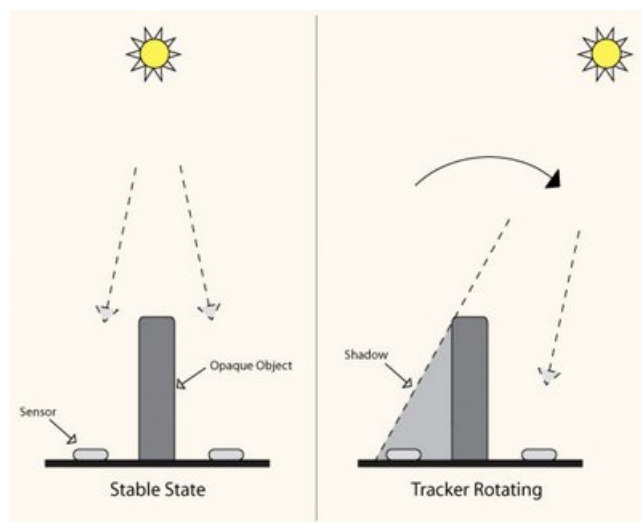
## 3. REQUISITOS

- 1) Microcontrolador (MSP430)
- 2) Converter energia solar em elétrica
- 3) Estrutura capaz de atender os requisitos
- 4) Movimentar placa solar nos eixos x e y
- 5) Medir incidência solar sobre a placa
- 6) Manter a placa no local com maior incidência solar
- 7) Abordar todo o conteúdo de Eletrônica Embarcada
- 8) Baixo custo
- 9) Desenvolvimento completo no período de 3 meses

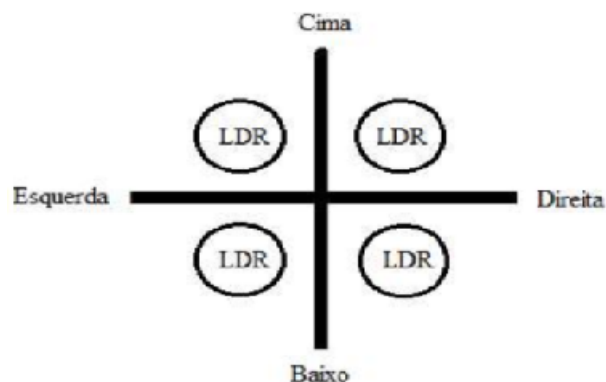
Para desenvolver o projeto será usado um microcontrolador MSP430, que tem como função movimentar dois servos motores, que serão posicionados no eixo “x” e no eixo “y”. Esses servos motores irão se movimentar a partir dos dados coletados por 4 sensores de luminosidade. Os LDR’s, serão analisados em pares, gerando dois conjuntos, o primeiro responsável pelo movimento azimutal e o segundo pelo movimento de declinação.

Os LDR’s de cada conjunto serão separados um do outro por meio de um perfil “T”, como mostra a figura 5. Sendo o microcontrolador acionado quando existe uma diferença entre as impedâncias dos LDR’s.

O primeiro conjunto de LDR’s compara a intensidade luminosa entre o lado direito e o lado esquerdo. O segundo conjunto compara os valores de um referencial em cima e outro abaixo, como mostra a figura 6.



**Fig. 5.** Conjunto de sensores separados por meio de perfil T.



**Fig. 6.** Referencial de orientação utilizado

## 4. DESENVOLVIMENTO

Para o primeiro protótipo funcional utilizou-se o software Energia IDE, pois através dele é possível utilizar bibliotecas e funções em linguagem C de forma que as mesmas sejam executadas na MSP430. Com o intuito de validar o projeto, foi usado o Energia IDE junto do Arduino Uno. Esta etapa é apenas para a validação do projeto, de forma que será necessário o desenvolvimento dos códigos em assembly no code composer posteriormente.

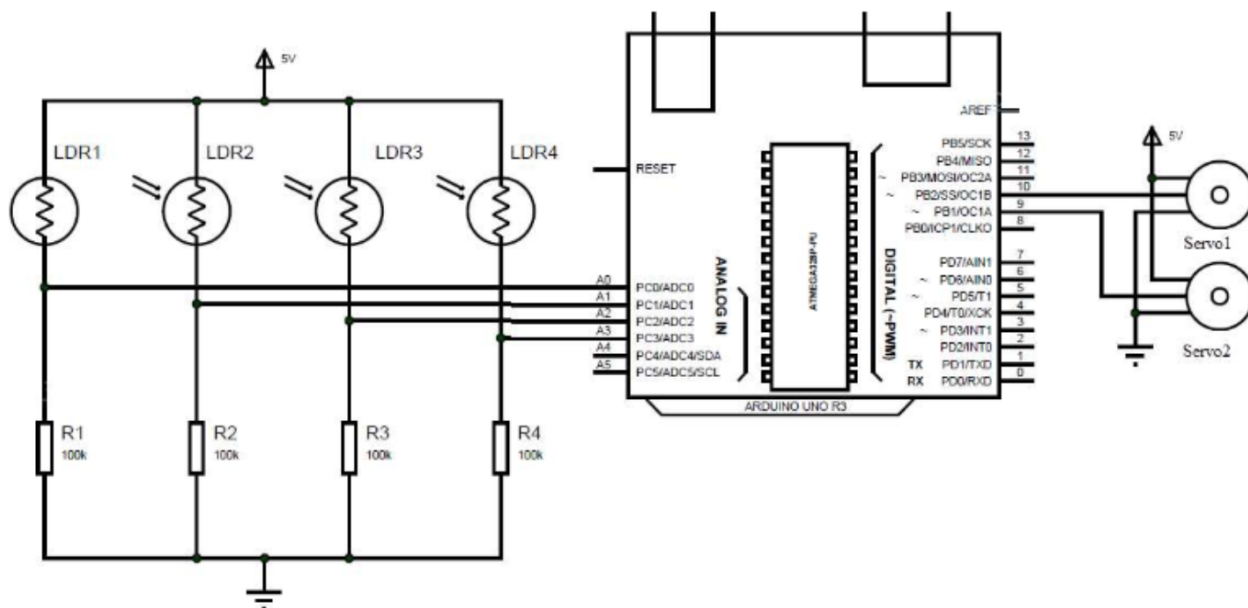


Fig. 7. Esquematico - Arduino Uno

### 4.1. Sensores

Para cumprir os requisitos do projeto descritos anteriormente foi necessário um levantamento dos sensores capazes de adquirir os dados de forma mais eficiente para as tomadas de decisão.

#### 4.1.1. LDR

O sensor LDR, também conhecido como foto-resistor, apresenta uma resistência que varia de acordo com a incidência solar.



Fig. 8. LDR

Através da curva característica do componente, é possível observar que sua resistência cai a medida em que a intensidade da luz aumenta.

Características do LDR modelo GL5528:

- Diâmetro: 5mm.
- Tensão máxima: 150VDC
- Potência máxima: 100mW.
- Espectro: 540nm
- Comprimento com terminais: 32mm.
- Resistência no escuro: 1 M (Lux 0)
- Resistência na luz: 10-20 K (Lux 10)

Cada LDR funciona como um divisor de tensão. A saída deste divisor de tensão é atribuída a um pino do microcontrolador. Desta forma utiliza-se um registrador para guardar o valor coletado de cada um dos sensores de luminosidade.

Os LDR's são separados em dois conjuntos, como explicado anteriormente, sendo que o microcontrolador é acionado quando existe uma diferença entre as impedâncias dos LDR's. O MSP430 envia um pulso para a movimentação dos servos quando nota essa diferença e reposiciona o sistema até que a discrepância entre as impedâncias seja mínima.

#### 4.1.2. Servo Motor

Os servomotores, também conhecidos como servos, são componentes elaborados com o uso de engrenagens, capazes de gerar movimento em sentido horário e anti horário. É um motor em que podemos controlar a sua posição angular com o uso de um sinal PWM.



**Fig. 9.** Servo Motor.

Características do servo modelo: Servo 9g Tower Pro SG90:

- Tensão de Operação: 3 a 7,2V.
- Faixa de Rotação: 180°
- Modulação: PWM.

- Velocidade (4.8V): 0.12 s / 60°
- Torque(4.8V): 1,5kg/cm.
- Dimensões: 22x12x19mm.
- Peso: 9g;
- Tamanho do cabo: 24cm.

## 4.2. Funcionamento dos códigos

### 4.2.1. Código Servomotor

Para testar o servomotor na estrutura, utilizou-se a biblioteca Servo.h. Esta é uma biblioteca comumente utilizada em projetos desenvolvidos no arduino, é necessário apenas instanciar o servomotor a um pino e chamar as funções informando o ângulo de movimento. O código pode ser visualizado no anexo.

### 4.2.2. Código LDR

A leitura da entrada analógica é feita com a função analogRead, que recebe como parâmetro o pino analógico a ser lido e retorna o valor digital que representa a tensão no pino. Em resumo, nosso programa lerá qual é o valor do sinal em A0 com o auxílio do comando analogRead(), que retornará um valor entre 0 a 1023, e o comparará com um valor de referência que em nosso caso é 800. Tendo em vista que, quanto mais escuro, maior será o valor de A0, caso A0 seja maior que o valor de referência o programa liga o LED conectado ao pino 7. Do contrário, ele apaga o LED. O programa também imprime o valor de A0 para que possamos verificar a faixa de valores e até mesmo calibrar nosso sensor. O código pode ser visualizado no anexo.

### 4.2.3. Código Geral

Para a funcionalidade total do sistema utilizou-se o código mostrado nas figuras 12,13,14 e 15. São criadas duas variáveis para controle dos servomotores e quatro variáveis do tipo inteiro para armazenar os valores lidos pelos sensores de luminosidade.

O loop da linha 35 utiliza os valores armazenados pelos LDR's de acordo com a incidência para fazer comparações entre os dois LDR's de cada conjunto. No primeiro conjunto, é comparada a incidência entre os lados direito e esquerdo (linha 45). Enquanto o segundo conjunto compara os valores de incidência entre a parte de cima e a parte de baixo (linha 59).

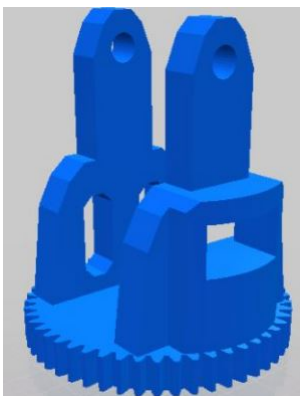
Ainda dentro do loop o servo R1 é controlado a partir dos dados do primeiro conjunto e o servo R2 controlado a partir dos dados do segundo conjunto. Após analisar a variação de valores entre os conjuntos dos sensores de luminosidade, os motores são movimentados até que essa discrepância seja mínima. O código pode ser visualizado no anexo.

### 4.3. Estrutura

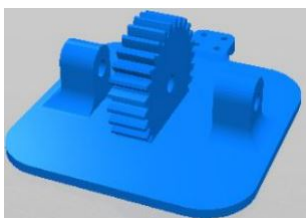
Com o intuito de montar uma estrutura capaz de atender os requisitos estabelecidos e nas características do projeto, foi utilizada uma estrutura elaborada na impressora 3D.

A estrutura foi projetada para que a movimentação necessária à otimização da captação solar fosse possível e de fácil encaixe dos servomotores. A mesma possui um suporte para a placa solar e engrenagens para a rotação dos dois eixos do motor. Para a alocação dos LDR's construiu-se uma peça em formato de T, como descrito anteriormente.

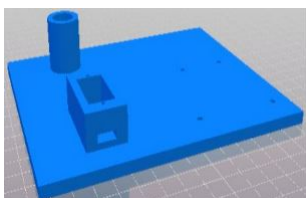
- Imagens da estrutura.



**Fig. 10.** Estrutura part.1



**Fig. 11.** Estrutura part.2



**Fig. 12.** Estrutura part.3



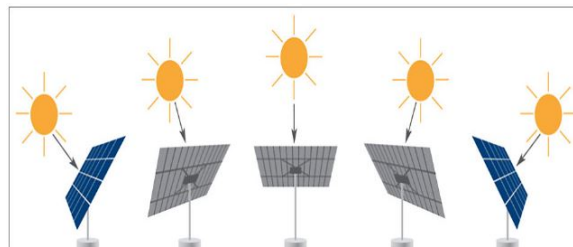
**Fig. 13.** Estrutura part.4



**Fig. 14.** Estrutura part.5

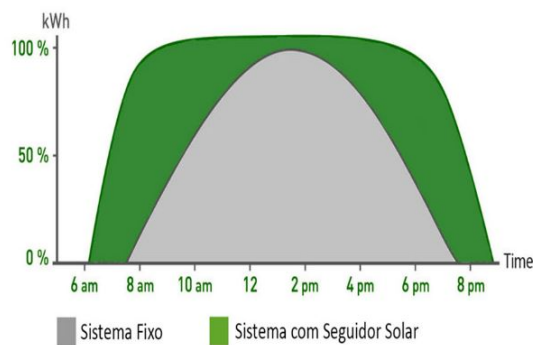
## 5. REVISÃO BIBLIOGRÁFICA

Um seguidor solar é um sistema microcontrolado que altera a posição de um painel solar em função da irradiação do sol. A energia solar fotovoltaica vem sendo amplamente utilizada e é considerada uma fonte de energia limpa. Devido aos benefícios e eficiência dessa tecnologia, uma relatório da Agência Internacional de Energia (AIE) constatou que até 2060 é possível que um terço da energia do mundo seja gerada por tecnologia fotovoltaica.



**Fig. 15.** Funcionamento do seguidor solar - Tracker.

Uma célula solar ou uma célula fotovoltaica é um dispositivo que converte a luz solar em energia elétrica. Essas células apresentam eficiência na conversão da energia de 16 por cento, dependendo do material utilizado na fabricação de até 28 por cento.



**Fig. 16.** Comparação entre a energia gerada por um sistema fixo e outro com seguidor solar.

Esses sistemas descritos, quando possuem seguidores solares, também chamados de tracker, melhoram o rendimento ao captar a energia considerando o movimento do sol como um fator real e fazendo com que as placas solares sigam na direção em que a incidência solar for maior. O aumento é em torno de 30 por cento na produção de energia quando comparados a sistemas fixos.

## 6. REFERENCIAS

- [1] S. L. d. A. R. R. Enio Bueno Pereira, Fernando Ramos Martins, "Atlas brasileiro de energia solar."
- [2] O. N. do Sistema Elétrico ONS, "Boletim mensal de geração solar fotovoltaica."
- [3] CEB, "Relatórios sobre unidades consumidoras com geração distribuída - informações compiladas e mapas."
- [4] "Portal solar. seguidor solar - tracker: Vantagens e desvantagens. disponível em: <https://www.portalsolar.com.br/blog-solar/painel-solar/seguidor-solar—tracker-vantagens-e-desvantagens-parte-1.html>."
- [5] J. D. B. de Araújo, "Protótipo de rastreador solar de um eixo baseado em microcontrolador," 2015.



## Anexos

### Leitura motor

```
include <Servo.h>

#define SERVO 6 // Porta Digital 6 PWM

Servo s; // Variável Servo
int pos; // Posição Servo

void setup() {
    // put your setup code here, to run once:
    s.attach(SERVO);
    Serial.begin(9600);
    s.write(0); // Inicia motor posição zero
}

void loop() {
    // put your main code here, to run repeatedly:

    for(pos = 0; pos < 90; pos++)
    {
        s.write(pos);
        delay(15);
    }
    delay(1000);
    for(pos = 90; pos >= 0; pos--)
    {
        s.write(pos);
        delay(15);
    }
}
```

## Leitura LDR

```
//Validar aquisicao de dados do LDR

int led = 6; //Led conectado a porta 6
int ldr_porta = 0; //LDR conectado a porta analogica 8
int ldr_leitura = 0; //Valor lido do LDR

void setup() {
  pinMode(led,OUTPUT); //define a porta 6 como saída
  Serial.begin(9600); //Inicia a comunicação serial
}

void loop() {
  ///leitura do valor do LDR
  ldr_leitura = analogRead(ldr_leitura);

  //se o valor lido for maior que 800, o led liga
  if (ldr_leitura>= 800) digitalWrite(led,HIGH);
  // caso contrario, apaga o led
  else digitalWrite(led,LOW);

  //imprime o valor lido do LDR no monitor serial a cada 100ms
  Serial.println(ldr_leitura);
  delay(100);
}
```



## Código Geral

```
#include <Servo.h> //Biblioteca de servos

Servo servo1; // servo responsavel pelo movimento azimutal
Servo servo2; // servo responsavel pelo movimento de declinacao

//4 LDR's (lembrando que a comparacao entre eles
//e feita em par para determinar a incidencia solar):
int ldr_E = A2; //LDR da esquerda
int ldr_D = A3; //LDR da direita
int ldr_C = A0; //LDR de cima
int ldr_B = A1; //LDR da baixo

//Posicao inicial dos servos
int posicao_servo1 = 90; // a posição do servo é determinada de acordo com o
//seu posicionamento na estrutura e com o ângulo
//de movimento necessário da haste do servo
int posicao_servo2 = 135;

//Variaveis para guardar valores
//analogicos dos pinos

int val_E;
int val_D;
int val_C;
int val_B;

//tempo de delay dos servos
int delay_movimento= 100;
```

```

//tempo de delay dos servos
int delay_movimento= 100;

void setup()
{
    // Determinar pino dos servos 1 e 2
    servo1.attach(10);
    servo2.attach(9);
}

void loop()
{
    // leitura analogica do LDR de acordo
    //com a incidencia solar
    val_E =analogRead(ldr_E);
    val_D =analogRead(ldr_D);
    val_C =analogRead(ldr_C);
    val_B =analogRead(ldr_B);

    //comparacao dos ldr's para estabilidade
    if (val_E - val_D > 2 || val_E - val_D < -2)
    {
        if (val_E > val_D && posicao_servo1 > 0)
        {
            posicao_servo1--;
        }
        if (val_D > val_E && posicao_servo1 < 180)
        {
            posicao_servo1++;
        }
    }
}

```

```
//comparacao dos ldr's para estabilidade
if (val_C - val_B > 2 || val_C - val_B < -2)
{
    if (val_C > val_B && posicao_servo2 > 90)
    {
        posicao_servo2--;
    }
    if (val_B > val_C && posicao_servo2 < 180)
    {
        posicao_servo2++;
    }
}

//posicao de acordo com o valor calculado:
servo1.write(posicao_servo1);
servo2.write(posicao_servo2);

//Delay para o servo se mover ate a posicao
delay(delay_movimento);
}
```