Software Embarcado

Tópicos Especiais em Redes de Telecomunicações

Apresentação

Alisson Cavalcante e Silva

Acadêmica

Mestrado Engenharia Eletrônica

Linha de Pesquisa: Rede de Computadores e Sistemas Distribuídos

Profissional

Marinha do Brasil

Analista de Segurança da Informação Digital

Objetivo

Apresentar:

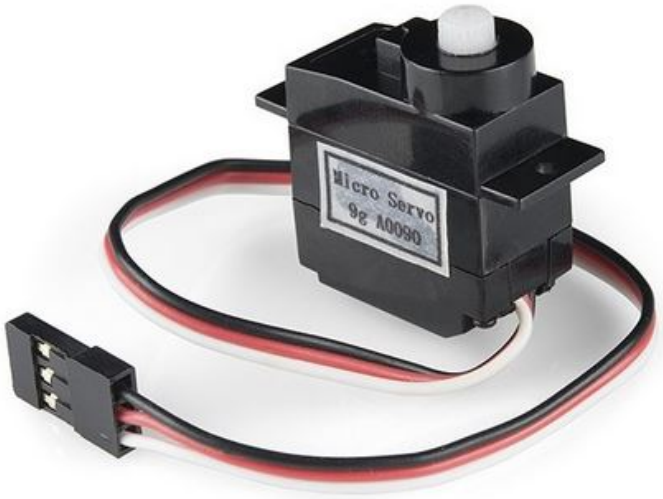
- **Dispositivo Servo Motor**
- **Sua Aplicação**
- **Layout Circuito PULL UP**
- **Solução Debounce**
- **Código Fonte proposto para Servo Motor**

- **Referências**

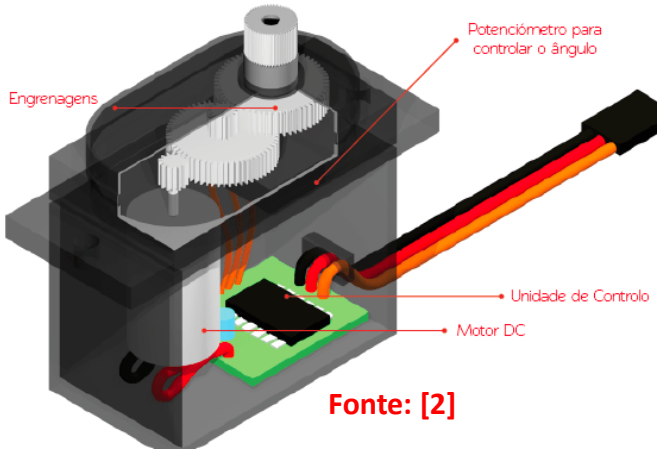
Servo Motor

Descrição:

- Equipamento eletromecânico
- Modelo: A0090 – peso : 9g
- Datasheet: sparkfun eletronics
- Diferente dos motores de “CC”
- Atuador rotativo
    - Posição controlada: 180°
    - Velocidade controlada
- Alimentação por 3vias:
    - Fio vermelho: 4.8 - 6.0v;
    - Fio preto: GND; e
    - Fio branco: Sinal de Controle



Fonte: [1]



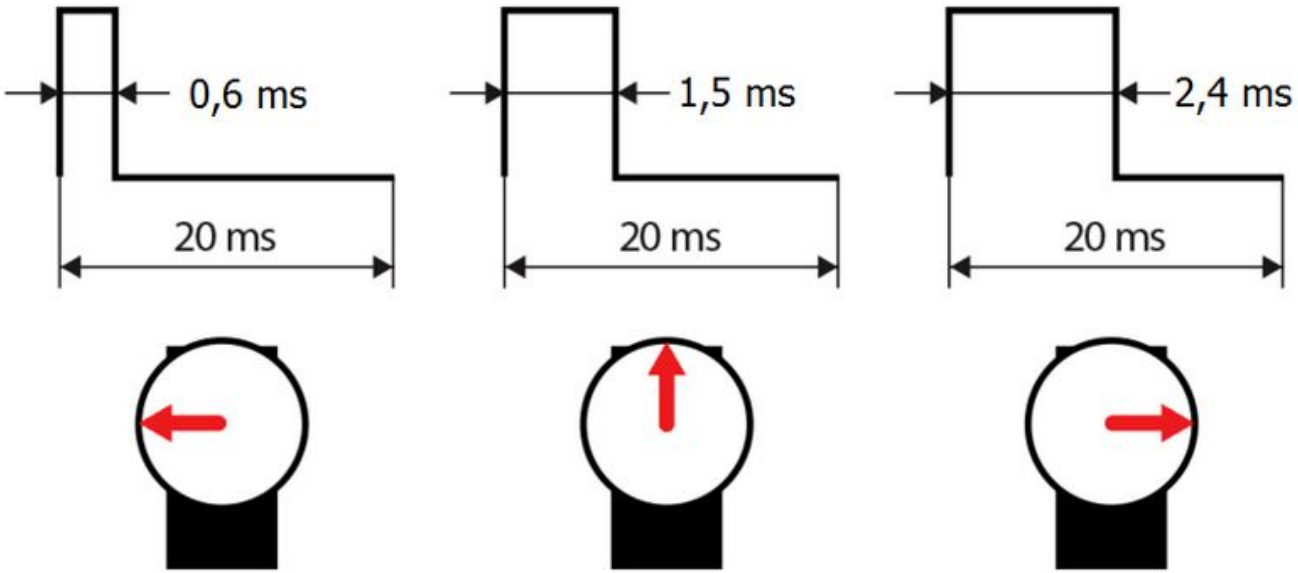
Fonte: [2]

Servo Motor

$$f = \frac{1}{T}$$

f = 50Hz => Período = 20ms

- Pulso de 0,6ms = 0°
- Pulso de 1,5ms = 90°
- Pulso de 2,4ms = 180°

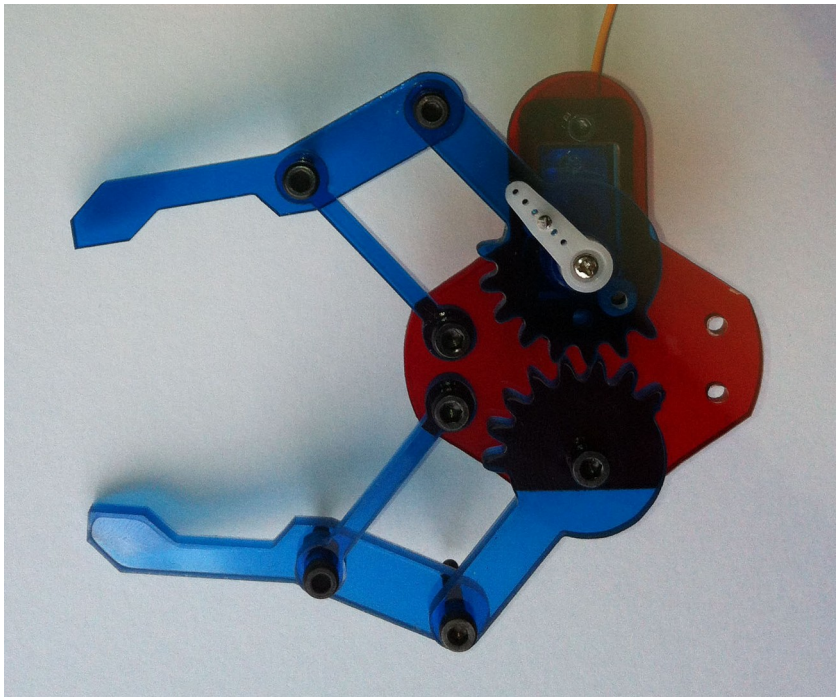


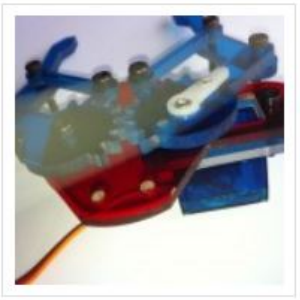
Fonte: [3]

Servo Motor

Um Exemplo de Aplicação:

- Robótica
  - Garra Robótica Servo-Controlada



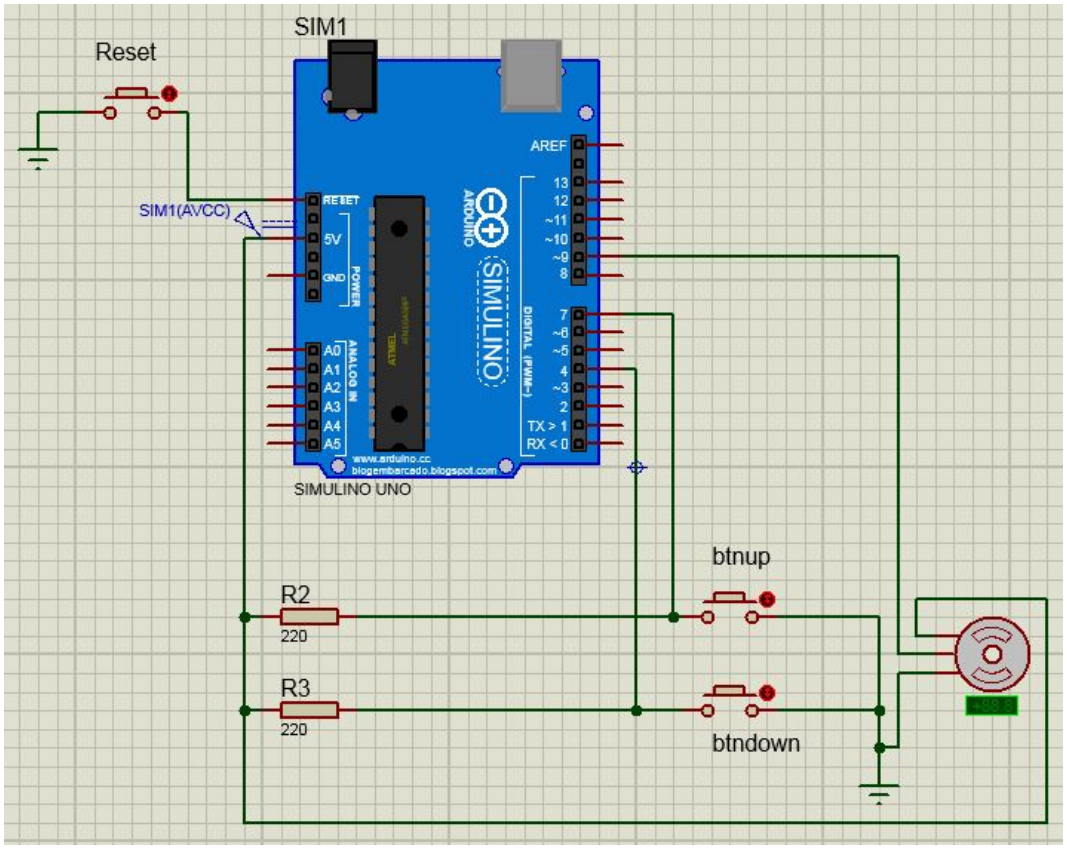


Fonte: [4]

Circuito PULL UP

Layout Circuito PULL UP

- Layout Proteus 8.5
- Alimentado com +5V
  - pino 7 – pushbuttonUP
  - pino 4 – pushbuttonDN
  - servo motor
- GND
  - servo motor
- Controle
  - pino 9 – servo motor

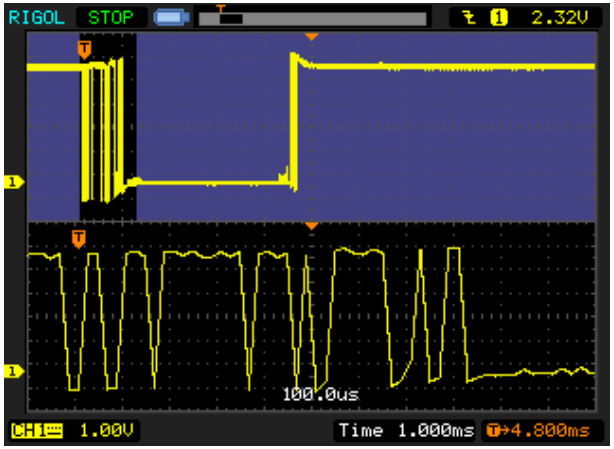


Fonte: desenhado com proteus 8.5

Bounce

O Problema:

- Comum em chaves mecânicas
- Contato ferro com ferro
- Demora na estabilidade do pressionamento
- Leituras repetidas até atingir a estabilidade



Fonte: [5]

Fonte: [6]

Bounce

Solução:

- Realizar o Debounce por meio de atraso
  - Via Software
    - implementação de código :
      - 1ª – Uso do delay()
        - gera uma parada na execução do programa
      - 2ª – Uso da millis()
        - mais trabalhosa para programar
  - Via hardware
    - Confecção de circuito eletrônicos mais elaborados
      - Uso de um capacitor para carga de descarga a uma

Qual solução devo utilizar?

Millis()

Função Millis()

- Retorno de contagem de tempo em milissegundos
- Contagem desde que Arduino começou a executar o programa
- Overflow após ≅ 50 dias (49d 17h 2m 47s e 295ms)
- Utilizar tipo “unsigned long”
  - Tamanho extendido = 4bytes (32bits)
  - Não guarda valor negativo : 0 a 4,294,967,295 (2^32 – 1)

Aplicação:

- Controle de tempo quando não se pode utilizar a função “delay()”

Código Fonte

```
// carrega biblioteca que permite controlar o servo motor
#include <Servo.h>

// bloco de declaracao de constantes que indicam quais pinos do circuito do arduino serao utilizados,
//alem de definir a limitacao da rotacao do servo motor
#define pinoServo 9 // pino utilizado para enviar dados ao servo motor
#define pinobuttonUP 7 // pino de controle que recebe sinal quando o pushbutton e pressionado.
#define pinobuttonDN 4 // pino de controle que recebe sinal quando o pushbutton e pressionado.
#define inicio 40
#define fim 180

// cria o objeto servo
Servo myservo;

// bloco de variaveis
int pos = inicio; // inicia a variavel de posicao do servo motor em 40 graus
int readingUP = HIGH;
int buttonStateUP = HIGH;
int buttonLastStateUP = HIGH;
int readingDN = HIGH;
int buttonStateDN = HIGH;
int buttonLastStateDN = HIGH;

// Para variaveis de tempo foi utilizado unsigned long, pois aumenta o valor positivo que a variavel pode armazenar
unsigned long lastDebounceTime = 0;  // Ultimo registro do tempo do debounce
unsigned long debounceDelay = 50;  // tempo de debounce 50ms (0,5s). Se o tempo desta variavel for maior que o
                                   // tempo que se leva pressionando o pushbutton a condicao nunca sera satisfeita.

void setup() {
  // ativa o uso da ferramenta serial monitor. Esta ferramenta ajuda no debug do codigo fonte.
  Serial.begin(9600);
  myservo.attach(pinoServo);  // atribui o pino do arduino ao objeto do servo motor - pino para controle
  pinMode(pinobuttonUP, INPUT); // atribui o pino do arduino ao pushbuttonUP
  pinMode(pinobuttonDN, INPUT); // atribui o pino do arduino ao pushbuttonDN
  myservo.write(inicio); // direciona o servo motor para posicao inicial
}
```

Código Fonte

```
void loop() {

  // *** Bloco 1 *** - estado inicial, botao em repouso, variaveis recebem sinal HIGH(1). Quando o botao e
  //pressionado a variavel recebe LOW (0).
  readingUP = digitalRead(pinobuttonUP);
  readingDN = digitalRead(pinobuttonDN);

  // *** Bloco 2 *** - executado se ocorrer mudanca no valor do pushbutton em relacao ao ultimo estado registrado
  //na execucao do loop. Ou seja, se o botao for pressionado. Ou se o interruptor mudou, devido a ruído ou pressão:
  if ((readingUP != buttonLastStateUP)||(readingDN != buttonLastStateDN))
  {
    lastDebounceTime = millis();
  }

  // *** bloco 3 *** - realiza o debounce, controle do bounce. Permite a execução do bloco se a subtracao entre
  // o tempo atual e o ultimo tempo de debounce registrado for maior que o tempo configurado para o debounce
  // delay. Se o tempo da variavel debounceDelay for maior que o tempo que se leva pressionando o pushbutton
  // a condicao nunca sera satisfeita.
  if ((millis() - lastDebounceTime) > debounceDelay)
  {
    // *** bloco 4 *** - executado se houver mudanca no estado do botao pushbutton. Ou seja, se o estado
    // atual do botao pushbutton for diferente do estado registrado por buttonState na ultima vez que foi
    // satisfeita a condicao.
    if (readingUP != buttonStateUP)
    {
      buttonStateUP = readingUP;

      // bloco 5 - Se o botao pushbuttonUP for pressionado (HIGH) a condicao e atendida na segunda rodada
      // apos pressionamento do botao. ou seja quando o botao retornar ao estao HIGH (1).
      if(buttonStateUP == LOW)
      {
        // quando a variavel for incrementada e atingir o valor 150 o servo motor retornara
        //para a posicao de grau 30
        if (pos < fim)
        {
          pos += 20; // atribui inicialmente 20 graus e vai somando de 20 em 20
          myservo.write(pos); // envia comando para o servo motor girar
```

Código Fonte

```
        }
      }
    }
    if (readingDN != buttonStateDN)
    {
      buttonStateDN = readingDN;

      // *** bloco 5 *** - Se o botao pushbuttonDN for pressionado (HIGH) a condicao e atendida na segunda
      // rodada apos pressionamento do botao. ou seja quando o botao retornar ao estao HIGH (1).
      if(buttonStateDN == LOW)
      {
        // quando a variavel for incrementada e atingir o valor 150 o servo motor retornara para a
        // posicao de grau 30
        if (pos > inicio)
        {
          pos -= 20; // atribui inicialmente 20 graus e vai somando de 20 em 20
          myservo.write(pos); // envia comando para o servo motor girar
          delay(15); // espera 15 milessimos de segundo para que o servo possa procurar a posição
        }
      }
    }
  }
  Serial.print("pos: ");
  Serial.println(pos);
  // *** bloco 6 *** - registra o ultimo estado do botao pushbuttonUP em buttonLastStateUP e pushbuttonDN
  //em buttonLastStateDN
  buttonLastStateUP = readingUP;
  buttonLastStateDN = readingDN;
}
```

Referências

Referências:

- [1] Servo Motor – https://www.sparkfun.com/products/9065, https://cdn.sparkfun.com/datasheets/Robotics/Small%20Servo%20-%20ROB-09065.pdf
- [2] Servo Motor - https://www.arduinoportugal.pt/controlando-um-servomotor-arduino/
- [3] Frequência - https://www.filipeflop.com/blog/video-controle-de-servo-motor-sem-biblioteca/
- [4] Garra Robótica - http://www.projetoespecial.com.br/?p=136
- [5] Bounce - https://portal.vidadesilicio.com.br/leitura-de-botoes-e-o-bounce/
- [6] Bounce - https://www.hackeduca.com.br/bounce-e-debounce-no-arduino-e-no-scratch/

Dúvidas