
Software de Integração com o encoder

— Laboratório de Controle e
Servomecanismo —

0 método

0 método

- Chama uma interrupção a cada intervalo de tempo.
- Recebe contagem em barramento paralelo do contador.
- Converte em um valor decimal para facilitar tratamento.
- Usa a diferença entre as contagens para projetar a posição do robô.
- Calcula a velocidade de movimento de acordo com a variação da contagem a cada chamada de interrupção.

A chamada de interrupção

Uso da biblioteca TimerOne.h

```
#define intervalo (tempo em  $\mu$ s)
```

```
#include <TimerOne.h>
```

```
Timer1.initialize(intervalo);
```

```
Timer1.attachInterrupt(função);
```

A chamada de interrupção

```
/*  
 * Intervalo da interrupção no sistema, em microssegundos  
 */  
#define intervalo 1000000  
  
/*  
 * Biblioteca que contém interrupção por tempo  
 */  
#include <TimerOne.h>|
```

A chamada de interrupção

```
void setup()
{

/*
 * Inicialização do timer
 * O intervalo de interrupção pode ser definido aqui
 * Valor mínimo de lus, que gera uma frequência máxima de 1MHz
 */
  Timer1.initialize(intervalo);

/* Vinculação da função de interrupção com o temporizador
 * A função de interrupção é chamada sempre que o temporizador entra em overflow
 * Cuidado na configuração do período do temporizador, pois ele pode travar seu
 * programa para nunca voltar para a função principal se o período for curto e
 * a função de interrupção for complexa
 */
  Timer1.attachInterrupt(leEncoder);
```

A chamada de interrupção

```
/*  
 * Valor que vem do contador de 8 bits  
 * Intervalo de 0 a 255  
 */  
int oldContagem = 0;  
int contagem = 0;  
  
/  
void leEncoder()  
{  
  
    contagem = binToInt();           //Converte os valores que vêm do contador em um valor inteiro  
    getPosicao();                     //Obtem a posição do robô a partir da contagem do contador  
    getVelocidade();                 //Obtem a velocidade a partir da nova posição do robô  
  
}
```

Conversão da contagem

De binário paralelo para
decimal

Contador: 8 bits

Contagem em barramento paralelo

Criada função para fazer conversão

Conversão da contagem

```
#define tam 8 //Tamanho do barramento do contador

int pinos[] = {
    46, 48, 50, 52, 47, 49, 51, 53
};

void setup()
{
    for (int i = 46; i < 54; i++) {
        pinMode(pinos[i], INPUT);
    }
}
```

Conversão da contagem

```
int binToInt()
{
    int recebe;
    int somaParcial = 0;

    for (int pos = 0; pos < tam; pos++)
    {
        // Variável que recebe o valor lido no pino
        recebe = digitalRead(pinos[pos]);

        /* Calcula o shift do valor, assim evita de fazer
        * potências e fica mais eficiente, por conta dos
        * valores lidos serem só 0 ou 1
        */
        recebe = recebe << pos;

        // Coloca em uma soma parcial;
        somaParcial += recebe;
    }

    return somaParcial;
}
```

Uso da contagem

Transforma a diferença entre contagens em posição do robô

Contagem e posição iniciadas em 0

Cada giro de $0,9^\circ$ gera uma nova contagem

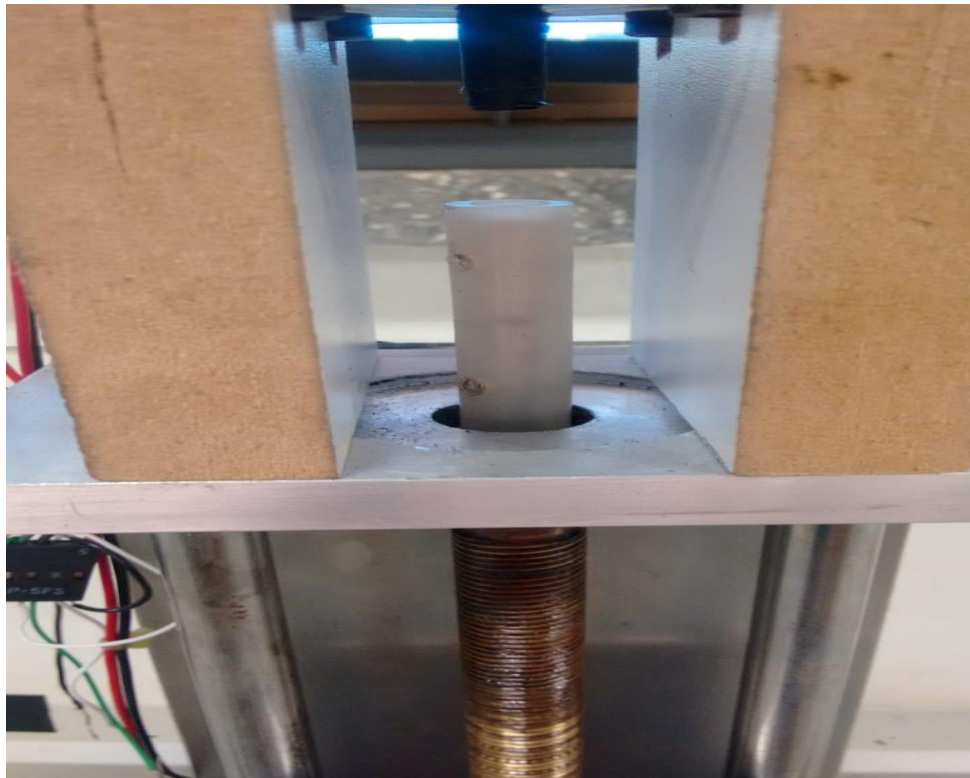
Cada 360° move o braço do robô em 4mm

Diferença entre contagens é a diferença entre posições

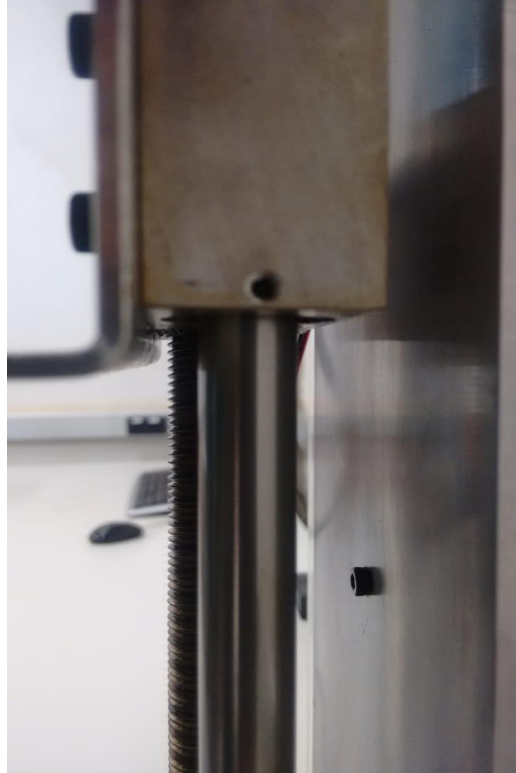
Uso da contagem



Uso da contagem



Uso da contagem



Uso da contagem

```
/*  
 * Valores das variáveis do robô  
 */  
double oldPosicao = 0;  
double newPosicao = 0;  
double velocidade = 0;
```

Uso da contagem

```
void getPosicao ()
{
    oldPosicao = newPosicao;

    int deltaContagem = 0;           //Diferença entre a última contagem e a contagem atual

    if (contagem < oldContagem)      //O contador completou um ciclo de contagem
    {
        deltaContagem = ((256 - oldContagem) + contagem); //Soma do que faltava para completar uma volta com o que passou depois da volta completa
    } else
    {
        deltaContagem = contagem - oldContagem;           //Diferença entre a última contagem e a contagem atual
    }

    oldContagem = contagem;

/*
 * Necessário as especificações do encoder e do roscamento em que o robô irá se deslocar
 * Isso irá definir sua posição vertical através do tempo
 */
    convert(deltaContagem);

/*
 * Aqui, a posição que o braço do robô vai estar já foi definida
 */
}
```


Uso da contagem

```
void convert(int deltaContagem)
{
    /*
     * Medindo no braço do robô, foi constatado que as 400 contagens do encoder equivalem a uma variação
     * vertical de (4,0 0,5)mm, ou seja, cada contagem equivale a 0,01mm de movimento vertical
     */
    newPosicao = oldPosicao + (double)deltaContagem*0.01;
}
```

Cálculo da velocidade

Estimado a partir do tempo de chamada de interrupção

Chamada da interrupção em intervalo de tempo definido

Verificação de quanto a distância variou nesse intervalo de tempo

Cálculo da velocidade

```
void getVelocidade ()
{
    /*
     * A velocidade do motor é definida pela diferença de posição no tempo entre chamadas de interrupção
     * Uma velocidade negativa indica que o motor está descendo
     * Unidade: mm/s
     */
    velocidade = (newPosicao - oldPosicao)*1000000/intervalo;
}
```

Aprimoramentos

Funcionalidades em processo
de finalização

Avaliação do tempo de execução
do programa geral

Controle de sentido de rotação
