

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E  
AUTOMAÇÃO**

---

---

**Manual da aplicação da *ufrn\_al5d.h*  
no braço robótico Lynx AL5D - v2.0**

---

---

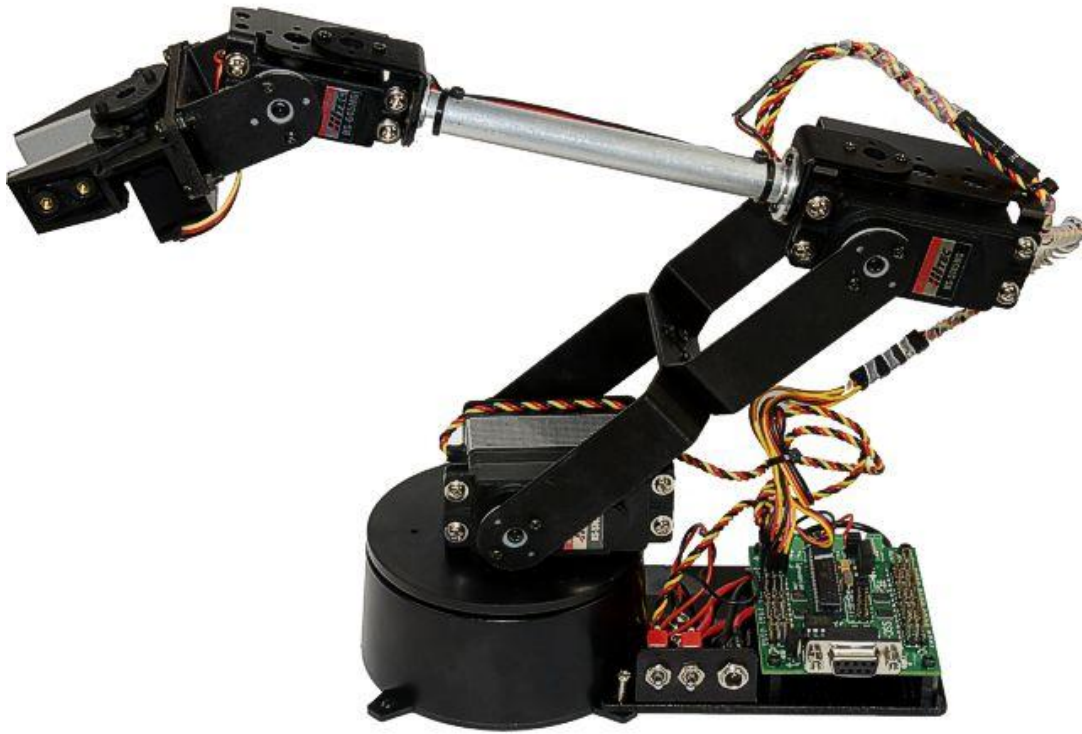
## SUMÁRIO

<b>1.0</b>	<b>INTRODUÇÃO .....</b>	<b>3</b>
<b>2.0</b>	<b>BRAÇO AL5D .....</b>	<b>4</b>
<b>3.0</b>	<b>COMUNICAÇÃO .....</b>	<b>7</b>
<b>3.1</b>	<b>Dicas iniciais .....</b>	<b>7</b>
<b>3.2</b>	<b>Programação .....</b>	<b>7</b>
<b>4.0</b>	<b>REFERÊNCIAS .....</b>	<b>10</b>

## 1.0 INTRODUÇÃO

Os braços robóticos da Lynxmotion têm como objetivo criar aplicações robóticas de forma didática e robusta. Essa empresa é uma das mais antigas fabricantes de kits robóticos didáticos, incluindo braços robóticos, robôs bípedes, quadrúpedes, hexápodes, veículos rastreados com rodas, e muito mais.

Figura 1 - Braço AL5D da Lynxmotion



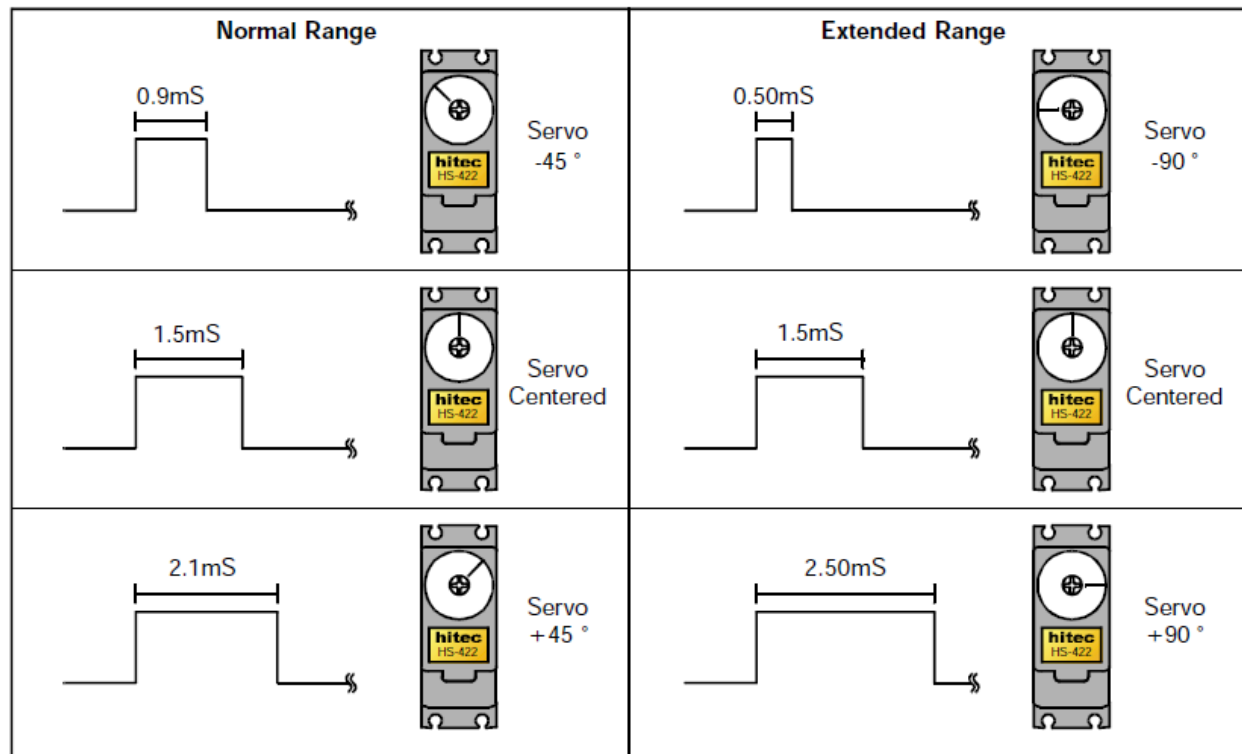
Nos capítulos a seguir, serão abordados os conceitos necessários para a realização do controle dos servo motores do braço robótico AL5D da Lynxmotion, exibido na Figura 1.

## 2.0 BRAÇO AL5D

O braço AL5D utiliza uma bateria de 9v para alimentar a placa SSC-32 e uma fonte de 6V/2A para alimentar os servos. Existem alguns softwares da empresa que proporcionam sua utilização em sistema Windows, entretanto, sugere-se a utilização da biblioteca *ufrn\_al5d.h* (disponível na seção acadêmica de <http://www.dca.ufrn.br/~engdesnes/>) para programação em C e em sistemas Linux. Contudo, é importante salientar que pode ser utilizada qualquer linguagem em qualquer sistema operacional, desde que os protocolos de comunicação para a placa SSC-32 sejam obedecidos corretamente.

Os servos do AL5D são servos de pulso proporcional, projetados para sistemas rádios-controlados (R/C), carros, barcos, aviões e etc. Eles fornecem um controle com precisão para a direção, aceleradores, lemes e etc., fazendo uso de um sinal de fácil de transmissão e recepção. O sinal é constituído por impulsos positivos que vão desde 0,9 a 2,1ms (milissegundos) de comprimento, e são repetidos 50 vezes por segundo (a cada 20 ms). Em suma, o servo posiciona o eixo de saída em relação à largura do pulso, tal como ilustrado na Figura 2 abaixo.

Figura 2 - Controle normal e estendido da posição dos servos



Em aplicações de controle de rádio, um servo motor não precisa de mais do que 90° de movimento, já que limitam manivelas que não podem mover-se mais do que 90°. Deste modo, quando pulsos forem enviados no intervalo especificado pelo fabricante (0,9 a 2,1ms), será percebido uma rotação por volta de 90 ° em amplitude de movimento.

A maioria dos servos proporcionam rotações maiores que 90° podendo atingir 180°. No entanto, é indispensável atentar para as limitações mecânicas de alguns servos, afim de evitar danificar estes.



ESTEJA ATENTO DURANTE SUA PROGRAMAÇÃO PARA NÃO  
FORÇAR AS LIMITAÇÕES MECÂNICAS DOS SERVOS  
DURANTE A PROGRAMAÇÃO

O AL5D permite o uso desta faixa extra por meio de comandos de valores absolutos enviados ao SSC-32.

- O VALOR 500 CORRESPONDE A UM PULSO DE 0,50 ms.
- O VALOR 2500 CORRESPONDE A UM PULSO DE 2,50 ms.
- UMA UNIDADE DE VARIAÇÃO PROVOCA UM PULSO DE 1μs.
- A RESOLUÇÃO DE POSICIONAMENTO É DADA EM GRAUS POR UNIDADE. PORTANTO:

$$RESOLUÇÃO = \frac{180^\circ}{(2500 - 500)} = \frac{180^\circ}{2000} = 0,09^\circ / unidade$$

- DESTE MODO:

$$POS = \frac{\theta}{RESOLUÇÃO} + 500 \rightarrow POS = \frac{\theta}{0,09} + 500$$

Lembre-se que alguns servos não podem e nem devem ser capazes de mover toda a gama de 180°. Tenha muito cuidado ao utilizar os servos, pois caso estoure os limites mecânicos nos seus cálculos certamente queimará o servo. Teste incrementalmente a extrema esquerda ou direita, buscando o ponto em que incrementos de posição já não resultem em movimentação do eixo de saída. Quando este valor for encontrado, use-o como limite na programação para evitar danos.



REALIZE TRAVAS NO SEU PROGRAMA PARA QUE NUNCA SEJA  
ENVIADO UM VALOR DE POSICIONAMENTO  
QUE POSSA DANIFICAR UM SERVO

A comunicação com os servos é realizada via comandos em forma de strings, que são enviados pela porta serial de um computador para a SSC-32 do AL5D. Essas strings devem seguir o padrão de comandos descrito no próprio manual do SSC-32, que está na pasta *doc/* no diretório do programa demo.



LEIA O MANUAL DO SSC-32 PARA ENTENDER COMO FUNCIONAM OS  
PROTOCOLOS DE COMUNICAÇÃO NECESSÁRIOS PARA  
REALIZAR A PROGRAMAÇÃO DOS SERVOS

O Braço AL5D é formado por cinco servo motores, sendo estes definidos como:

- Servo da base: HS-485HB
- Servo do ombro: HS-805BB
- Servo do cotovelo: HS-755HB
- Servo do punho: HS-645MG
- Servo da garra: HS-322HD

A seguir, será discutido a utilização da *ufrn\_al5d.h* para a comunicação com o SSC-32.

## 3.0 COMUNICAÇÃO

### 3.1 Dicas iniciais

Baixe e descompacte a versão 2.0 do *demo\_lyxn.tar.gz* em <http://www.dca.ufrn.br/~engdesnes/>. Para a facilidade e dinamização da programação, o programa demo já vem com um Makefile que limpa temporários, realiza compilação e a linkagem da *libufrn\_lynx.a* ao programa executável. Para compilar faça:

```
usuario@maquina:~/Desktop/demo_lynx$ make
```

Como somente o usuário *root* tem acesso a abertura de portas no Linux, o usuário que for executar o programas que utilizem esta biblioteca deve ter tal acesso ou isto pode ser configurado por meio dos comandos de Linux *chown* e *chmod*. A execução pode ser dada da seguinte maneira:

```
usuario@maquina:~/ Desktop/demo_lynx$ sudo ./demo
```

### 3.2 Programação

Na biblioteca *ufrn\_al5d.h* foram definidos um padrão de nomes para os servos, representando seu pino de conexão ao SSC-32, tão como o limite de posição de cada servo, como mostrado abaixo:

```
// Servo da base HS-485HB //
#define BAS_SERVO 0
#define BAS_MIN    500
#define BAS_MAX    2380

// Servo do ombro HS-805BB //
#define SHL_SERVO 1
#define SHL_MIN    1200
#define SHL_MAX    2000

// Servo do cotovelo HS-755HB//
#define ELB_SERVO 2
#define ELB_MIN    1100
#define ELB_MAX    2100

// Servo do punho HS-645MG //
#define WRI_SERVO 3
#define WRI_MIN    500
```

```
#define WRI_MAX    2500

// Servo da garra HS-322HD //
#define GRI_SERVO 4
#define GRI_MIN 1500
#define GRI_MAX 2400
```

Também na biblioteca *ufrn\_al5d.h*, foram definidos as seguintes funções importantes para a comunicação com os servos:

```
/* Abre a porta serial e retorna um file descriptor */
int abrir_porta(void);

/* Configura comunicação, banda e outros */
int configurar_porta(int);

/* Envia um char* por meio do file descriptor à porta */
/* Retorna -1 em caso de falha */
int enviar_comando(char*,int);

/* Trava o limite máximo para que não se estoure o limite físico de cada servo */
unsigned int trava(unsigned int canal, unsigned int pos);

/* Fecha a porta serial por meio do file descriptor */
void fechar_porta(int);
```

Por meio desses comandos a comunicação com os servos do AL5D pode ser estabelecida. Agora basta ler os comandos de strings do manual SSC-32. O posicionamento básico é realizado pelo envio de uma string tal como: "#<canal>P<valor\_posicao>".

#### EXEMPLO 1:

```
/* Copia para a string comando um pedido de posicionamento do servo 2 com valor 500 */
sprintf(comando,"#2P500");

/* Envia a string para a porta serial */
enviar_comando(comando,serial_fd);
```

#### EXEMPLO 2:

```
/* LEMBRE-SE: sempre "zere" o buffer na memória */
memset(comando, 0, BUFSIZE);

/* Copia para a string comando um pedido de posicionamento de todos servos com valor 1500 */
sprintf(comando," #0P1500#1P1500#2P1500#3P1500#4P1500");

/* Envia a string para a porta serial */
```



```
enviar_comando(comando,serial_fd);
```

EXEMPLO 3:

```
/* LEMBRE-SE: sempre "zere" o buffer na memória */  
memset(comando, 0, BUFSIZE);  
  
/*  
 * Copia para a string comando um pedido de posicionamento do servo do punho  
 * com um valor muito alto; que será travado no limite máximo do servo, realizada  
 * pela função trava(canal,pos) da ufrn_al5d.h.  
 */  
sprintf(comando,"%dP%d",GRI_SERVO,trava(GRI_SERVO, 429496729));  
  
/* Envia a string para a porta serial */  
enviar_comando(comando,serial_fd);
```



EMBORA O MANUAL EXPLÍCITE A INCLUSÃO DE UM COMANDO <cr>  
NO FINAL DE CADA STRING, VOCÊ NÃO DEVE FAZÊ-LO.  
A BIBLIOTECA *ufrn\_al5d.h* JÁ IMPLEMENTANO ISTO  
NO COMANDO DE ENVIO DE STRING.



LIMPE A STRING ANTES DE ENVIAR UM NOVO COMANDO



UTILIZE OS LIMITES MÁXIMOS DE POSIÇÃO DE CADA SERVO E A  
FUNÇÃO trava(canal,posição), DISPONÍVEIS NA BIBLIOTECA *ufrn\_al5d.h*



Pronto! Muita atenção e agora é só programar.

#### **4.0 REFERÊNCIAS**

[1] LYNXMOTION INC., *Manual written for firmware version SSC32-1.03XE Range is 0.50 ms to 2.50 ms*, Ver 2.0, Pekin IL 61555-0818.

[2] LYNXMOTION, Lynxmotion Inc. Disponível em: <<http://www.lynxmotion.com/>> . Acesso em 16 de Maio de 2013.