

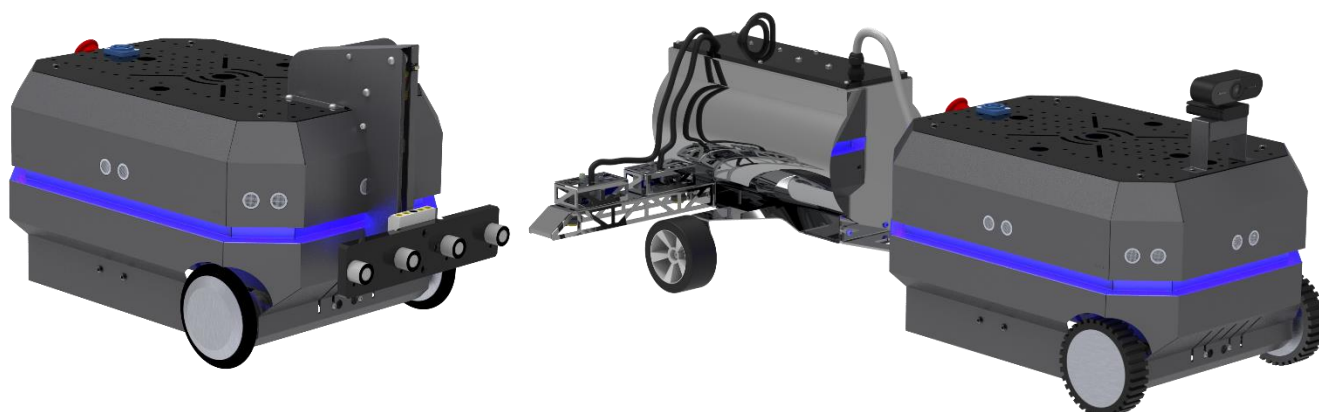
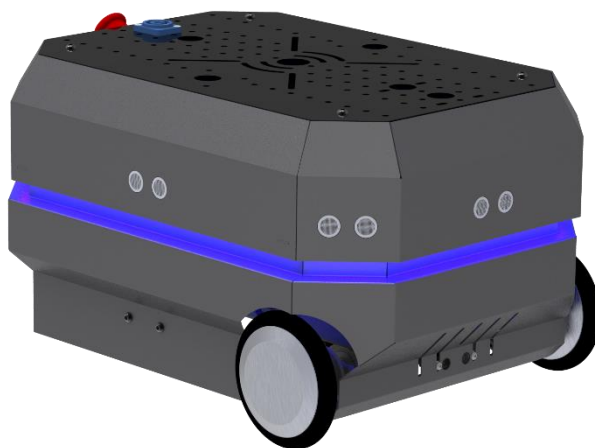
# SOBOT



INOVANDO NA FORMAÇÃO POR **COMPETÊNCIAS**

1ª Edição - 2024

# GUIA DE REFERÊNCIA COMANDOS SOBOT



1ª Edição - 2024

Av. Angélica, 2.627, Térreo  
São Paulo, SP, 01227-200, BRASIL  
+55 (11) 4304-0786 / +55 (11) 3237-2222

201 S. Biscayne Blvd, Suite 1200  
Miami, FL, 33131, EUA  
+1 (786) 475-6385

[contato@solistecnologia.com.br](mailto:contato@solistecnologia.com.br)  
[www.solistecnologia.com.br](http://www.solistecnologia.com.br)

# Sumário

1.	Introdução .....	4
2.	Descrição dos comandos .....	6
3.	Lista de comandos .....	7
4.	Estrutura de uso dos comandos .....	8
5.	Comandos e Instruções .....	8
5.1.	Conjunto de comandos utilizados para parametrização das rodas. ....	9
5.2.	Conjunto de comandos utilizados para controle dos motores de passo. ....	10
5.2.1.	Comandos utilizados para movimentação no modo comandos fixos .....	11
5.2.2.	Conjunto de comandos utilizados para movimentação no modo contínuo .....	15
5.2.3.	Comandos utilizados para interromper a movimentação do SoBot .....	18
5.2.4.	Comando para verificar status de movimentação das rodas .....	20
5.2.5.	Comando para modo aprendizado de movimentos .....	21
5.2.6.	Comandos para calibrar Ganho Proporcional na movimentação .....	23
5.2.7.	Comandos de controle dos motores no modo Pulsos por Segundo .....	27
5.3.	Comando para leitura dos sensores SONAR .....	29
5.4.	Comando para leitura do sensor Infravermelho de distância .....	30
5.5.	Comando para leitura dos sensores de linha .....	32
5.6.	Comando para leitura das entradas digitais .....	33
5.7.	Conjunto de comandos para controle das saídas digitais .....	34
5.8.	Comando para leitura das entradas analógicas .....	35
5.9.	Conjunto de comandos para controle das saídas analógicas .....	36
5.10.	Conjunto de comandos para controle da fita de LED .....	37
5.11.	Conjunto de comandos para controle do Buzzer .....	39
5.12.	Comando para leitura do sensor acelerômetro .....	39
5.13.	Conjunto de comandos para comunicação serial .....	40
5.14.	Conjunto de comandos para controlar o modulo elevador .....	42
5.15.	Comando para tempo de atraso "Delay" .....	44
5.16.	Comandos para obter retorno de ações executadas .....	45
6.	Código de erros .....	49
6.1.	ERRO 0 .....	49
6.2.	ERRO 1 .....	49
6.3.	ERRO 2 .....	52

# Índice de Tabelas

Tabela 1 – Lista de comandos .....	7
Tabela 2 – Conjunto de comandos utilizados para parametrização das rodas.....	10
Tabela 3 – Conjunto de comandos utilizados para controle dos motores de passo.....	13
Tabela 4 – Conjunto de comandos utilizados para movimentação no modo contínuo.....	17
Tabela 5 – Conjunto de comandos utilizados para interromper a movimentação do SoBot.....	19
Tabela 6 – Comando para verificar status das rodas .....	20
Tabela 7 – Comando para modo aprendizado de movimentos.....	22
Tabela 8 – Tabela de comandos utilizados para parametrização do ganho proporcional.....	24
Tabela 9 – Comando para leitura dos sensores SONAR.....	29
Tabela 10 – Exemplo de retorno do comando para leitura dos sensores SONAR.....	30
Tabela 11 – Comando para leitura do sensor infravermelho de distância.....	31
Tabela 12 – Exemplo de retorno do sensor infravermelho de distância.....	31
Tabela 13 – Comando para leitura dos sensores de linha.....	32
Tabela 14 – Exemplo de retorno do comando para leitura dos sensores de linha.....	32
Tabela 15 – Comando para leitura das entradas digitais.....	33
Tabela 16 – Exemplo de retorno do comando para leitura das entradas digitais.....	33
Tabela 17 – Conjunto de comandos para controle das saídas digitais.....	34
Tabela 18 – Comando para leitura das entradas analógicas.....	35
Tabela 19 – Exemplo de retorno do comando para leitura das entradas analógicas.....	36
Tabela 20 – Conjunto de comandos para controle das saídas analógicas.....	37
Tabela 21 – Conjunto de comandos para controle da fita LED.....	38
Tabela 22 – Conjunto de comandos para controle do Buzzer.....	39
Tabela 23 – Comando para leitura do sensor acelerômetro.....	40
Tabela 24 – Conjunto de comandos para comunicação serial.....	41
Tabela 25 – Conjunto de comandos para controlar o módulo elevador.....	44
Tabela 26 – Comando para tempo de atraso “Delay”.....	44
Tabela 27 – Comando para obter retorno de ações executadas.....	45

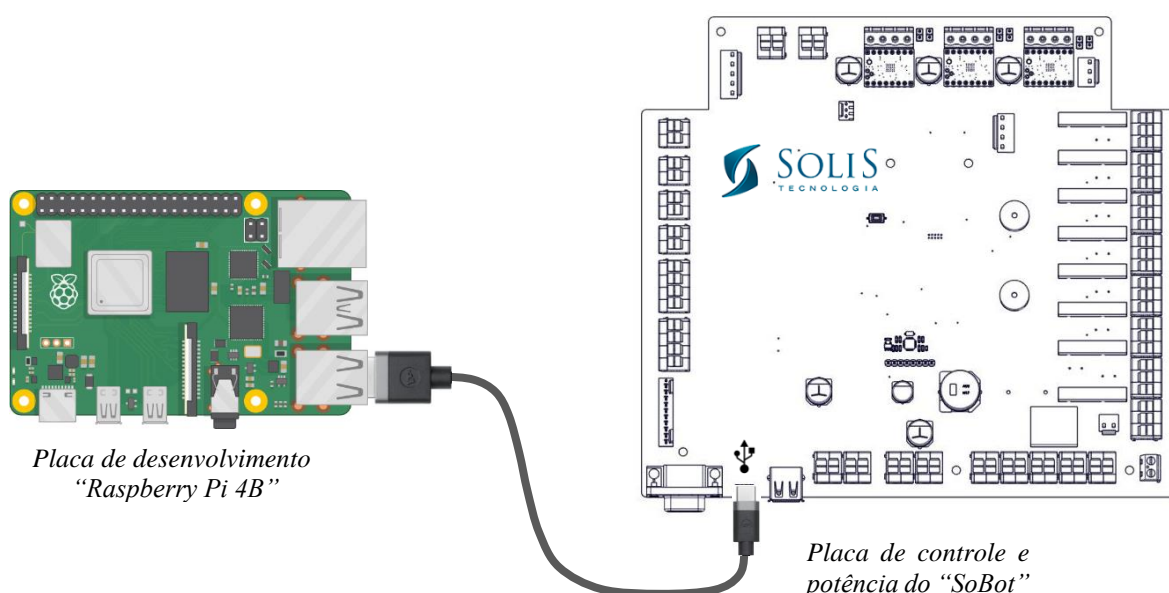
# 1. Introdução

O Guia de Referência de comandos do SoBot foi desenvolvido para auxiliar na programação de comandos disponíveis para a Placa de Controle e Potência do SoBot.



- Pré-requisito para plena absorção deste conteúdo é a leitura anterior da “Apostila SoBot” em sua versão mais recente.

Com os comandos citados neste guia, será possível desenvolver aplicações práticas de algoritmos do nível mais simples ao mais complexo, utilizando diversas placas de desenvolvimento, tais como: *Arduino, Raspberry, NVidia Jetson, Intel Galileo*, entre outras.



## Comandos e Instruções de Controle:

Os comandos deste guia estão agrupados de acordo as funções pré-definidas disponíveis para controle do SoBot, sendo:

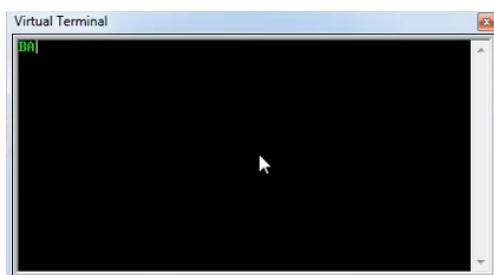
1. Conjunto de comandos de parametrização das rodas;
2. Conjunto de comandos para movimentar o SoBot em todas as direções com opções de modo de curvas:
  - a. Movimentação retilínea para frente ou para trás;
  - b. Curva sobre o próprio eixo;
  - c. Curva diferencial (circular) entre as rodas;
3. Comandos para realizar leitura de sensores de distância:
  - a. Sensores ultrassônicos;

- b. Sensor infravermelho;
  - c. Sensor acelerômetro.
4. Comandos para realizar leitura de faixa com sensores fotoelétricos;
  5. Comandos para controlar entradas e saídas do tipo digital e analógicas;
  6. Comandos para controlar sinalizadores sonoros (buzzer de sinal contínuo);
  7. Comandos para controlar sinalizadores luminosos (fita LED RGB);
  8. Comandos para enviar e receber dados de forma serial para dispositivos externos;
  9. Comandos para controlar o módulo elevador quando acoplado ao SoBot;
  10. Comandos para inserir tempos de atraso entre comandos de controle;
  11. Comandos para habilitar envio de dados de retorno, para sinalizar a execução de uma ação e identificar códigos de erros para possíveis falhas em comandos de controle.

### Comunicação e Parametrizações:

A Placa de Controle e de Potência do SoBot, recebe os comandos de forma serial por meio da comunicação USB.

Os exemplos utilizados neste guia são descritos seguindo padrões da linguagem de programação **Python** para envio de dados por meio da comunicação serial, demonstrando como devem ser escritos os comandos, utilizando como referência a placa de desenvolvimento **Raspberry Pi-4**;



Os retornos dos comandos são exibidos em uma janela exemplo de um terminal virtual, para demonstrar como as mensagens com dados de retorno, também serão recebidas pela placa de desenvolvimento Raspberry.





### 3. Lista de comandos

ITEM	CÓDIGO	DESCRIÇÃO	ITEM	CÓDIGO	DESCRIÇÃO
1	AI	<i>Input Analog</i>	29	ME	<i>Movement Enable</i>
2	AO	<i>Output Analog Pwm</i>	30	MF	<i>Movement Forward</i>
3	AT	<i>Acceleration Time</i>	31	ML	<i>Movement Left</i>
4	BC	<i>Break Command</i>	32	MP	<i>Movement Pause</i>
5	BL	<i>Blue</i>	33	MR	<i>Movement Righth</i>
6	BZ	<i>Buzzer</i>	34	MS	<i>Movement Status</i>
7	CA	<i>Curve Angle</i>	35	MT	<i>Motor</i>
8	CR	<i>Command Return</i>	36	PG	<i>Proportional Gain</i>
9	D	<i>Distance</i>	37	PS	<i>Pulse Speed</i>
10	DC	<i>Duty-Cycle</i>	38	R	<i>Right</i>
11	DF	<i>Differential Curve</i>	39	RD	<i>Red</i>
12	DI	<i>Input Digital</i>	40	RI	<i>Radius Inner</i>
13	DL	<i>Delay</i>	41	RS	<i>Read Serial</i>
14	DN	<i>Down</i>	42	SA	<i>Sensor Accelerometer</i>
15	DO	<i>Output Digital</i>	43	SC	<i>Serial Communication</i>
16	DT	<i>Deceleration Time</i>	44	SD	<i>Send Serial</i>
17	DW	<i>Distance Wheel</i>	45	SI	<i>Sensor Infrared</i>
18	E	<i>Enable</i>	46	SL	<i>Sensor Line</i>
19	EL	<i>Elevator</i>	47	SO	<i>Straight On</i>
20	F	<i>Frequency (Hz)</i>	48	SS	<i>Sensor Sonar</i>
21	GR	<i>Green</i>	49	ST	<i>Stop</i>
22	KC	<i>Kill Command</i>	50	TM	<i>Timer</i>
23	L	<i>Left</i>	51	TP	<i>Total Pulses</i>
24	LM	<i>Learn Moviments</i>	52	UP	<i>Up</i>
25	LT	<i>Led Tape</i>	53	V	<i>Velocity</i>
26	MB	<i>Movement Back</i>	54	WD	<i>Wheel Diameter</i>
27	MC	<i>Movement Continuous</i>	55	WP	<i>Wheel Parameters</i>
28	MD	<i>Movement Differential</i>			

Tabela 1 – Lista de comandos

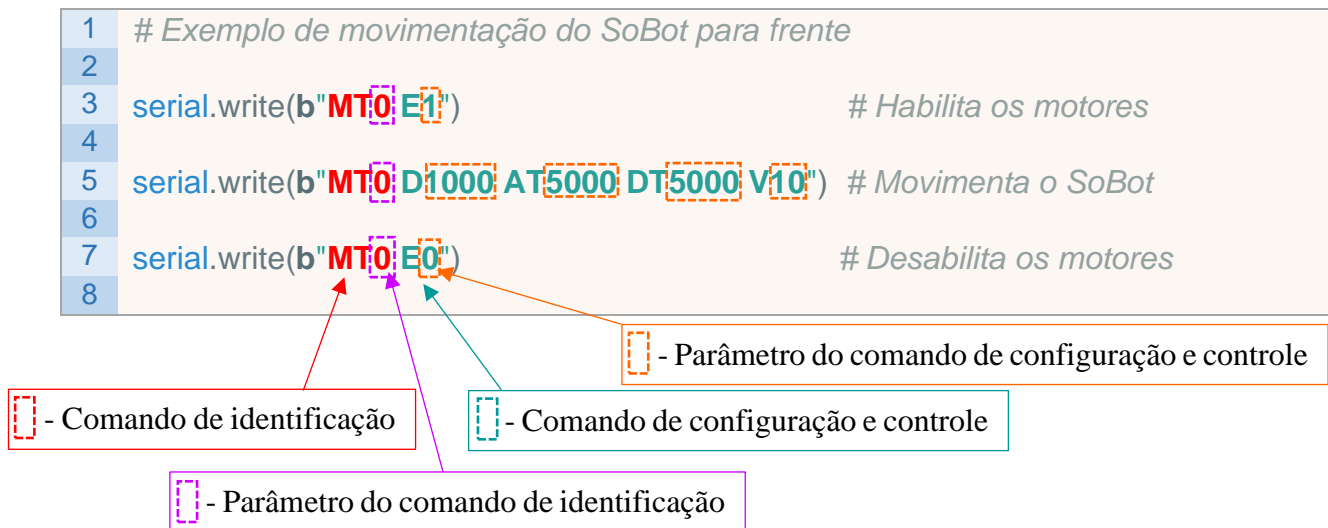


## 4. Estrutura de uso dos comandos

Os comandos são formados por conjuntos de caracteres compostos por caracteres iniciais de identificação que pode ou não conter parâmetros e comandos adicionais de configuração e controle.

Entre os parâmetros de um comando é necessário a utilização do caractere espaço para separação dos mesmos e formar a “string” do comando completo.

Segue abaixo um exemplo com explicação das características de construção de um comando de controle do SoBot:



**NOTA:** Utilize os comandos conforme os exemplos citados no capítulo 5 para evitar erros de sintaxe.

## 5. Comandos e Instruções

Os comandos do SoBot são utilizados em conjunto de caracteres ou apenas caracteres de identificação para funções específicas.

Os tópicos a seguir descrevem os comandos com especificações dos seus parâmetros e exemplos de utilização dos mesmos.

## 5.1. CONJUNTO DE COMANDOS UTILIZADOS PARA PARAMETRIZAÇÃO DAS RODAS.

A construção das partes físicas que compõem o SoBot pode sofrer uma pequena variação de tolerância em suas medidas e para corrigir essas tolerâncias, podem ser utilizados os comandos de parametrização que configuram o diâmetro de cada roda e a distância entre os pontos de contato das rodas junto ao solo.

Seguem abaixo a descrição dos comandos com os seus parâmetros e exemplo de utilização destes comandos.

**NOTA:** Os comandos de parametrização das rodas, são recebidos e armazenados para serem executados sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
WP	<b>WHEEL PARAMETER</b> Comando de identificação utilizado para parametrizar as medidas das rodas e distância entre as mesmas.		Sem parâmetro.
MT	<b>MOTOR</b> Comando de controle utilizado para indicar qual motor irá receber os comandos de parametrização a seguir.	1	Utilizado para parametrizar a roda conectada na saída MOTOR 01. (Roda esquerda)
		2	Utilizado para parametrizar a roda conectada na saída MOTOR 02. (Roda direita)
WD	<b>WHEEL DIAMETER</b> Comando de controle utilizado para configurar o diâmetro da roda. Unidade de medida utilizada no parâmetro é em milímetros com duas casas depois da virgula para maior precisão.	90,00 até 110,00	Valor do diâmetro da roda em milímetros.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.  Caso necessite de maior precisão, utilize o caractere virgula para trabalhar nas duas casas decimais.

(continua)

(conclusão)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
DW	<b>DISTANCE WHEEL</b> Comando de controle utilizado para configurar a distância entre os pontos centrais de contato das rodas ao solo. Unidade de medida utilizada no parâmetro é em milímetros com duas casas depois da vírgula para maior precisão.	240,00 até 280,00	Valor de distância entre as rodas em milímetros.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.  Caso necessite de maior precisão, utilize o caractere vírgula para trabalhar nas duas casas decimais.

Tabela 2 – Conjunto de comandos utilizados para parametrização das rodas.

## Exemplo de uso:

### 5.1.1. Parametrizando rodas:

- Considerando o valor medido da roda do motor 1 com diâmetro de 99,60 milímetros, o valor medido da roda do motor 2 com diâmetro de 100,32 milímetros e a distância entre as rodas de 260,35 milímetros:

1	# Configurações dos parâmetros das rodas:		
2	<code>serial.write(b"WP MT1 WD99,6")</code>	# Configura roda MT1 em 99,6mm	
3	<code>serial.write(b"WP MT2 WD100,32")</code>	# Configura roda MT2 em 100,32mm	
4	<code>serial.write(b"WP DW260,35")</code>	# Configura distância entre rodas em 260,35mm	

**NOTA:** O comando de identificação “WP” tem que ser o primeiro para que o SoBot compreenda que este é um comando de parametrização das rodas.

## 5.2. CONJUNTO DE COMANDOS UTILIZADOS PARA CONTROLE DOS MOTORES DE PASSO.

Neste tópico, são descritos os comandos para controlar a movimentação do SoBot em todas as direções. O sistema de controle oferece duas opções de modos de curva, além da possibilidade de controle por pulsos nas saídas 01 e 02 dos drives, que são utilizados para motores de passo. Adicionalmente, é detalhado o uso da saída 03 para o controle individual de um motor de passo, que pode ser aplicado opcionalmente a dispositivos como elevadores, pallets, garras, entre outros.

Para o controle dos motores nas saídas 01 e 02, é possível utilizar o modo de comandos fixos e/ou o modo de comandos contínuos.

Para os comandos no modo fixo, o usuário pode determinar a direção, a distância, o tempo de aceleração, desaceleração e a velocidade que o SoBot irá se movimentar.

Para comandos no modo contínuo, a configuração dos parâmetros de movimentação e o deslocamento é realizado através de comandos simples de direção.

**Os comandos do modo contínuo têm prioridade de execução**, sendo assim:

- ✓ Ao habilitar e iniciar a movimentação no modo contínuo, se um comando no modo fixo for recebido, o SoBot só irá executá-lo assim que o comando contínuo for pausado;
- ✓ Ao iniciar a movimentação com o comando fixo e em seguida enviar um comando no modo contínuo, o SoBot realizará a desaceleração e em seguida executa o comando contínuo sem executar por completo o comando fixo já iniciado, tome cuidado ao utilizar desta forma para evitar problemas mecânicos e/ou elétricos!

### 5.2.1. Comandos utilizados para movimentação no modo comandos fixos

**NOTA:** Os comandos de controle dos motores de passo no modo fixo, são recebidos e armazenados para serem executados sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
MT	<b>MOTOR</b> Comando de identificação utilizado para indicar qual motor irá receber os comandos de controle a seguir.	0	Utilizado para controle do motor 1 e 2 de forma simultânea para controle das rodas.
		3	Utilizado de forma exclusiva para controlar o motor 3
AT	<b>ACCELERATION TIME</b> Comando de configuração utilizado para controlar o tempo de aceleração do motor.  Unidade de medida utilizado no parâmetro é milissegundos.	0 até 60.000	Valor de tempo em milissegundos.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

(continua)

(continuação)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
D	<b>DISTANCE</b> Comando de configuração para indicar a distância de movimentação do SoBot no modo rodas.  Unidade de medida utilizada no parâmetro é em milímetros ou graus se for definido movimentação com curva e para movimentos no sentido inverso é necessário inserir o sinal de negativo antes do valor do parâmetro.	-	Símbolo negativo utilizado antes do valor de distância indica movimento para trás (reverso). Ausência do símbolo negativo antes do valor de distância indica movimento para frente.
		0 até 65.000,00	Valor da distância do percurso em milímetros.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
		0 até 400,00	Se definido curva para direita ou esquerda, utilizar valores de 0 até 380 para indicar angulo da curva.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo e o valor máximo em graus é maior que uma volta completa de 360° devido a fatores de acurácia.
DF	<b>DIFFERENTIAL CURVE</b> Comando de configuração utilizado para o SoBot realizar curva no modo diferencial. <i>Caso não utilizar este comando, o SoBot realizara curva sobre o próprio eixo.</i>		Sem parâmetro.
DT	<b>DECELERATION TIME</b> Comando de configuração utilizado para controlar o tempo de desaceleração do motor.  Unidade de medida utilizado no parâmetro é em milissegundos.	0 até 60.000	Valor de tempo em milissegundos.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
E	<b>ENABLE</b> Comando de controle utilizado para habilitar a movimentação dos motores. Utilize 0 no parâmetro para desabilitar o motor e 1 para habilitar a movimentação do motor.	0	Utilizado para desabilitar o motor.
		1	Utilizado para habilitar o motor.
L	<b>LEFT</b> Comando de controle utilizado para indicar movimentação em curva no próprio eixo para esquerda.		Sem parâmetro.

(continua)

(conclusão)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
PS	<b>PULSE SPEED</b> Comando de configuração utilizado para indicar a velocidade de pulsos por segundo para controle do motor 3. Resolução fixa em 800 pulsos por revolução.	0 até 2.000	Valor de pulsos por segundos.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
R	<b>RIGHT</b> Comando de controle utilizado para indicar movimentação em curva no próprio eixo para direita.		Sem parâmetro.
RI	<b>RADIUS INNER</b> Comando de configuração utilizado no parâmetro de raio interno para curva no modo diferencial.  Unidade de medida utilizado no parâmetro é em milímetros.	10 até 9.999	Valor do raio interno em milímetros.  <b>NOTA:</b> Se parâmetro maior ou menor ao valor do range permitido, o mesmo será configurado com o limite do range permitido.
TP	<b>TOTAL PULSES</b> Comando de configuração utilizado para indicar o total de pulsos para movimentar o motor 3.	-	Símbolo negativo utilizado antes do valor de distância indica movimento reverso.
		0 até 500.000	Valor total de pulsos do percurso.
V	<b>VELOCITY</b> Comando de configuração utilizado para indicar a velocidade de cruzeiro do SoBot.  Unidade de medida utilizado no parâmetro é em centímetro por segundo.  <b>Velocidade de Cruzeiro:</b> É a velocidade constante que o robô navega após a rampa de aceleração e antes da rampa de desaceleração. 	0 até 25	Valor da velocidade em centímetros por segundo.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

Tabela 3 – Conjunto de comandos utilizados para controle dos motores de passo.



## Exemplos de uso:

### 5.2.1.1. Movimentação para Frente:

- Habilita os motores, movimenta o SoBot 1 metro para frente com tempo de aceleração e desaceleração em 5 segundos e velocidade cruzeiro de 10cm/s e em seguida desabilita os motores:

1	# Movimentação do SoBot para frente	
2	<code>serial.write(b"MT0 E1")</code>	# Habilita os motores
3	<code>serial.write(b"MT0 D1000 AT5000 DT5000 V10")</code>	# Movimenta o SoBot
4	<code>serial.write(b"MT0 E0")</code>	# Desabilita os motores

### 5.2.1.2. Movimentação para Trás:

- Habilita os motores, movimenta o SoBot 1 metro para trás com tempo de aceleração e desaceleração em 5 segundos a velocidade cruzeiro de 10cm/s e em seguida desabilita os motores:

1	# Movimentação do SoBot para trás	
2	<code>serial.write(b"MT0 E1")</code>	# Habilita os motores
3	<code>serial.write(b"MT0 D-1000 AT5000 DT5000 V10")</code>	# Movimenta o SoBot
4	<code>serial.write(b"MT0 E0")</code>	# Desabilita os motores

### 5.2.1.3. Curva no próprio eixo a Direita:

- Habilita os motores, movimenta o SoBot 90° para direita com curva no próprio eixo, tempo de aceleração e desaceleração em 0,5 segundos a velocidade cruzeiro de 10cm/s e em seguida desabilita os motores:

1	# Movimentação do SoBot para direita com curva no próprio eixo	
2	<code>serial.write(b"MT0 E1")</code>	# Habilita os motores
3	<code>serial.write(b"MT0 D90 R AT500 DT500 V10")</code>	# Movimenta o SoBot
4	<code>serial.write(b"MT0 E0")</code>	# Desabilita os motores

### 5.2.1.4. Curva no próprio eixo a Esquerda:

- Habilita os motores, movimenta o SoBot 90° para esquerda com curva diferencial, raio interno de 100 milímetros a velocidade cruzeiro de 5cm/s e em seguida desabilita os motores:

1	# Movimentação do SoBot para esquerda com curva diferencial	
2	<code>serial.write(b"MT0 E1")</code>	# Habilita os motores
3	<code>serial.write(b"MT0 D90 DF L RI100 V5")</code>	# Movimenta o SoBot
4	<code>serial.write(b"MT0 E0")</code>	# Desabilita os motores

#### 5.2.1.5. Movimentar Motor 3 por pulsos:

- Habilita o motor 3 e inicia a movimentação com tempo de aceleração e desaceleração em 2 segundos em um total de 8000 pulsos com velocidade de 800 pulsos por segundo e em seguida desabilita o motor 3:

1	# Movimentação do motor 3 por controle de pulsos	
2	serial.write(b"MT3 E1")	# Habilita os motores
3	serial.write(b"MT3 AT2000 DT2000 TP8000 PS800")	# Movimenta o motor 3
4	serial.write(b"MT3 E0")	# Desabilita os motores

**NOTA:** O comando de identificação “MT” tem que ser o primeiro para que o SoBot entenda que é um comando de controle dos motores.

#### 5.2.2. Conjunto de comandos utilizados para movimentação no modo contínuo

Comandos de movimentação contínua, são usados para manter o SoBot se movimentando de forma indefinida até que seja enviado um novo comando capaz de alterar sua movimentação.

Nesse tipo de controle é necessário configurar a aceleração, desaceleração e velocidade de cruzeiro que o SoBot irá ter para os comandos no modo contínuo, caso não realize estas configurações, os parâmetros padrão de tempo de aceleração e desaceleração serão de 100ms e a velocidade de cruzeiro será de 2cm/s.

Também é possível configurar a forma de realizar curva sendo com diferencial ou sobre o próprio eixo.

Para curva diferencial é necessário configurar o raio da curva da roda interna e a velocidade, caso não realize a configuração, o parâmetro padrão do raio interno é de 50mm e a velocidade de 2cm/s.

Uma opção para utilização do modo contínuo pode ser quando for utilizar um controle remoto (Controle de vídeo game, Joystick ou Teclado) para controlar a direção do SoBot ou se a movimentação for orientada por sensores (ultrassônico sonar ou Infra Red de linha) mantendo a movimentação continua até encontrar um obstáculo.

**NOTA:** Os comandos de movimentação modo contínuo, são recebidos e executados imediatamente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
<b>MB</b>	<b>MOVEMENT BACK</b> Comando de controle utilizado para movimentar o SoBot para trás.		Sem parâmetro.
<b>MC</b>	<b>MOVEMENT CONTINUOUS</b> Comando de controle utilizado para indicar as configurações de tempo de aceleração e desaceleração e velocidade do modo de movimentação contínuo do SoBot.		Sem parâmetro.
<b>MD</b>	<b>MOVEMENT DIFFERENCIAL</b> Comando de controle utilizado para habilitar a movimentação das curvas com movimento diferencial. Utilize 0 no parâmetro para desabilitar e 1 para habilitar a movimentação com curva diferencial.	0	Utilizado para desabilitar a movimentação com curva diferencial.
		1	Utilizado para habilitar a movimentação com curva diferencial.
<b>ME</b>	<b>MOVEMENT ENABLE</b> Comando de controle utilizado para habilitar a movimentação dos motores no modo contínuo. Utilize 0 no parâmetro para desabilitar o motor e 1 para habilitar a movimentação do motor.	0	Utilizado para desabilitar a movimentação do motor no modo contínuo.
		1	Utilizado para habilitar a movimentação do motor no modo contínuo.
<b>MF</b>	<b>MOVEMENT FORWARD</b> Comando de controle utilizado para movimentar o SoBot para frente.		Sem parâmetro.
<b>ML</b>	<b>MOVEMENT LEFT</b> Comando de controle utilizado para movimentar o SoBot para esquerda.		Sem parâmetro.
<b>MP</b>	<b>MOVEMENT PAUSE</b> Comando de controle utilizado para pausar (parar) a movimentação do SoBot, até que um novo comando seja enviado.		Sem parâmetro.
<b>MR</b>	<b>MOVEMENT RIGTH</b> Comando de controle utilizado para movimentar o SoBot para direita.		Sem parâmetro.
<b>AT</b>	<b>ACCELERATION TIME</b> Comando de configuração utilizado para controlar o tempo de aceleração do motor. Unidade de medida utilizado no parâmetro é em milissegundos.	0 até 60.000	Valor de tempo em milissegundos. <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

(continua)

(conclusão)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
DT	<b>DECELERATION TIME</b> Comando de configuração utilizado para controlar o tempo de desaceleração do motor. Unidade de medida utilizado no parâmetro é em milissegundos.	0 até 60.000	Valor de tempo em milissegundos.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
RI	<b>RADIUS INNER</b> Comando de configuração utilizado no parâmetro de raio interno para curva no modo diferencial. Unidade de medida utilizado no parâmetro é em milímetros.	10 até 9.999	Valor do raio interno em milímetros.  <b>NOTA:</b> Se parâmetro maior ou menor ao valor do range permitido, o mesmo será configurado com o limite do range permitido.
V	<b>VELOCITY</b> Comando de configuração utilizado para indicar a velocidade cruzeiro do SoBot. Unidade de medida utilizado no parâmetro é em centímetro por segundo.	0 até 25	Valor da velocidade em centímetros por segundo.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

Tabela 4 – Conjunto de comandos utilizados para movimentação no modo contínuo.

### Exemplos de uso:

**NOTA:** O comando de identificação MT0 deve ser utilizado em conjunto aos comandos de movimentação contínua para o SoBot entender que o próximo comando será de movimentação das rodas.

#### 5.2.2.1. Modo Contínuo:

- Comandos para habilitar e desabilitar o funcionamento das rodas no modo contínuo:

1	# Movimentação do SoBot no modo contínuo		
2	serial.write(b"MT0 ME1")	# Habilita os motores das rodas	
3	serial.write(b"MT0 ME0")	# Desabilita os motores das rodas	

#### 5.2.2.2. Curva no Próprio Eixo em Modo Contínuo:

- Comando para configurar o tempo de aceleração, desaceleração e velocidade do funcionamento das rodas no modo contínuo com curva no próprio eixo:

1	# Comandos de configuração do modo contínuo com curva no próprio eixo		
2	serial.write(b"MT0 MC MD0 AT100 DT100 V5")	# Configuração do modo contínuo	

### 5.2.2.3. Curva Diferencial em Modo Contínuo:

- Para executar movimentação contínua com curva diferencial, basta indicar o parâmetro 1 no comando "MD" e indicar o valor de raio interno da curva junto aos comandos de configuração:

1	# Comandos de configuração do modo contínuo com curva diferencial
2	serial.write(b"MT0 MC MD1 RI100 AT100 DT100 V5") # Configuração do modo contínuo

### 5.2.2.4. Movimentação em Modo Contínuo:

- Comandos de movimentação do SoBot no modo contínuo:

1	# Comandos de movimentação do SoBot no modo contínuo
2	serial.write(b"MT0 MF") # Movimenta o SoBot para frente
3	
4	serial.write(b"MT0 MB") # Movimenta o SoBot para trás
5	
6	serial.write(b"MT0 ML") # Movimenta o SoBot para esquerda
7	
8	serial.write(b"MT0 ML-") # Movimenta o SoBot para esquerda sentido trás
9	
10	serial.write(b"MT0 MR") # Movimenta o SoBot para direita
11	
12	serial.write(b"MT0 MR-") # Movimenta o SoBot para direita sentido trás
13	
14	serial.write(b"MT0 MP") # Pausa a movimentação do SoBot

## 5.2.3. Comandos utilizados para interromper a movimentação do SoBot

Para interromper a movimentação dos motores é possível utilizar dois métodos de controle. Um deles é utilizando o comando Break onde ao receber o comando, a movimentação irá desacelerar conforme tempo de desaceleração configurado anteriormente até parar completamente. A outra forma de interromper a movimentação é utilizando o comando Kill onde ao receber o comando, a movimentação é interrompida imediatamente.

**Atenção: o comando Kill ou Break com tempo de desaceleração relativamente baixo pode causar danos mecânicos e elétricos irreversíveis aos motores e/ou aos seus drivers de controle devido as características físicas da inércia em relação a velocidade e ao peso do SoBot.**

**NOTA:** Os comandos para interromper a movimentação do SoBot, são recebidos e executados imediatamente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
BC	<b>BREAK COMMAND</b> Comando de controle utilizado para interromper a movimentação com desaceleração e automaticamente limpar a fila de comandos de ação já recebidos.		Sem parâmetro.
KC	<b>KILL COMMAND</b> Comando de controle utilizado para interromper a movimentação imediatamente e limpar a fila de comandos de ação já recebidos.  <b>NOTA:</b> Utilize este comando somente em casos de extrema urgência, pois pode causar danos mecânicos e elétricos irreversíveis ao sistema devido as características físicas da inércia.		Sem parâmetro.

Tabela 5 – Conjunto de comandos utilizados para interromper a movimentação do SoBot.

### Exemplos de uso:

**NOTA:** Se o motor estiver no modo contínuo, o comando Break tem a mesma função de um comando pause, já no modo de comandos fixos, ele interrompe o movimento, faz a desaceleração e limpa as filas de comandos recebidos.

#### 5.2.3.1. Parar com Desaceleração:

- Comando para interromper a movimentação com desaceleração dos motores das rodas:

1	# Para a movimentação com desaceleração
2	<code>serial.write(b"MT0 BC")</code> # Comando de parada no modo contínuo

- Comando para interromper a movimentação com desaceleração do motor 3:

1	# Para a movimentação com desaceleração
2	<code>serial.write(b"MT3 BC")</code> # Comando de parada no modo contínuo

#### 5.2.3.2. Parada Forçada - Kill:

- Comando para interromper a movimentação imediatamente.

1	# Para a movimentação imediatamente
2	<code>serial.write(b"KC")</code> # Comando de parada de emergência

**NOTA:** Comando KILL é semelhante ao comando break limpando a fila de comandos de ação porem não faz a desaceleração parando o movimento do motor imediatamente.



#### 5.2.4. Comando para verificar status de movimentação das rodas

O comando de Status de movimentação tem a finalidade de informar a distância percorrida na última movimentação e funciona tanto no modo de comandos fixos quanto no modo de comandos contínuos.

**NOTA:** O comando para verificar status de movimentação das rodas, é recebido e armazenado para ser executado sequencialmente.

COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
MS	<b>MOVEMENTS STATUS</b> Comando de controle utilizado para obter o status de movimentação com a distância percorrida.		Sem parâmetro.

Tabela 6 – Comando para verificar status das rodas

#### Exemplos de uso:

##### 5.2.4.1. Status de Movimentação:

- Comando para solicitar o status de movimentação realizada.

1	# Comando de status de movimentação		
2	serial.write(b"MT0 MS")		

##### 5.2.4.2. Feedback da Movimentação:

- Exemplos de uso de movimentação com solicitação de status da movimentação:

1	# Comandos de movimentação do SoBot com status de movimentação		
2	serial.write(b"MT0 E1")		# Habilita os motores
3	serial.write(b"MT0 D500 AT2000 DT2000 V8")		# Movimenta o SoBot para frente
4	serial.write(b"MT0 MS")		# Solicita status de movimentação
5	serial.write(b"MT0 D-500 AT2000 DT2000 V8")		# Movimenta o SoBot para trás
6	serial.write(b"MT0 MS")		# Solicita status de movimentação
7	serial.write(b"MT0 D90 R AT500 DT500 V6")		# Movimenta o SoBot para direita
8	serial.write(b"MT0 MS")		# Solicita status de movimentação
9	serial.write(b"MT0 D90 L AT500 DT500 V6")		# Movimenta o SoBot para esquerda
10	serial.write(b"MT0 MS")		# Solicita status de movimentação
11	serial.write(b"MT0 E0")		# Desabilita os motores

Retorno dos comandos de status enviados no exemplo anterior:

Terminal Serial	
MT0 MS D+00500,00	
MT0 MS D-00500,00	
MT0 MS D+R0090,00	
MT0 MS D+R0090,00	

“MT0 MS D+00500,00” - Indica movimentação frente com distância de 500mm;

“MT0 MS D-00500,00” - Indica movimentação trás com distância de 500mm;

“MT0 MS D+R0090,00” - Indica movimentação de curva para direita de 90°;

“MT0 MS D+L0090,00” - Indica movimentação de curva para esquerda de 90°.

### 5.2.5. Comando para modo aprendizado de movimentos

O comando de aprendizado é utilizado em conjunto aos comandos de movimentação contínua.

Quando habilitado, ao enviar comando de movimentação para frente o SoBot só aceitara comandos de frente e traz para incrementar ou decrementar a contagem de distância percorrida obtendo a distância resultante de movimentação e funciona da mesma forma de comandos para esquerda ou direita para informar o valor resultante em graus da curva realizada.

Para realizar um novo movimento, é necessário enviar o comando de Status (“MT0 MS”) para que o SoBot entenda que já informou a distância percorrida e está disponível para aprender um novo movimento.

**NOTA:** O comando para habilitar o modo aprendizado do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
LM	<b>LEARN MOVEMENTS</b> Comando de controle utilizado para habilitar o modo de aprendizado de movimentação. Utilize 0 no parâmetro para desabilitar e 1 para habilitar o modo aprendizado.	0	Utilizado para desabilitar o modo aprendizado.
		1	Utilizado para habilitar o modo aprendizado.

Tabela 7 – Comando para modo aprendizado de movimentos.

## Exemplos de uso:

**NOTA:** O comando de aprendizado é utilizado em conjunto aos comandos de movimentação contínua.

### 5.2.5.1. Teach-in – Modo de Aprendizagem:

- Comandos para habilitar e desabilitar o modo de aprendizado:

1	# Comandos do modo aprendizado	
2	<code>serial.write(b"MT0 MC LM1")</code>	# Habilita o modo de aprendizado
3	<code>serial.write(b"MT0 MC LM0")</code>	# Desabilita o modo de aprendizado

- Exemplos de uso com aprendizado habilitado:

1	# Comandos de movimentação do SoBot no modo contínuo com aprendizado de	
2	# movimentos	
3	<code>serial.write(b"MT0 ME1")</code>	# Habilita os motores
4	<code>serial.write(b"MT0 MC LM1")</code>	# Habilita modo aprendizado
5	<code>serial.write(b"MT0 MC AT100 DT100 V5")</code>	# Configuração do modo contínuo
6	<code>serial.write(b"MT0 MF")</code>	# Movimenta o SoBot para frente
7	<code>sleep(2)</code>	# Aguarda 2 segundos
8	<code>serial.write(b"MT0 MB")</code>	# Movimenta o SoBot para trás
9	<code>sleep(1)</code>	# Aguarda 1 segundos
10	<code>serial.write(b"MT0 MP")</code>	# Pausa a movimentação do SoBot
11	<code>serial.write(b"MT0 MS")</code>	# Solicita status de movimentação
12	<code>serial.write(b"MT0 ME0")</code>	# Desabilita os motores
13	<code>serial.write(b"MT0 MC LM0")</code>	# Desabilita modo aprendizado

Retorno do comando de status enviado no exemplo anterior:

Terminal Serial
MT0 MS D+00050,00

“MT0 MS D+00050,00” - Indica movimentação frente com distância de 50mm

- Neste exemplo, o SoBot configura a movimentação no modo contínuo habilitando o aprendizado de comandos e se movimenta 100 milímetros para frente e retorna 50 milímetros para trás, sendo 2 segundos para frente e 1 segundo para trás na velocidade de 5cm/s e ao solicitar o status da movimentação a resultante foi de 50 milímetros devido modo aprendizado estar habilitado.

### 5.2.6. Comandos para calibrar Ganho Proporcional na movimentação

Os comandos de “Ganho Proporcional” são utilizados para ajustar/calibrar o erro proporcional de movimentação que são causados por variações de produção mecânica.

A calibração/ganho pode variar dependendo do tipo de piso, peso do robô, ou até mesmo de inclinação da pista.

A calibração deve ser feita em movimentação em linha reta, curva no próprio eixo e também curva diferencial.

O percentual de ganho proporcional pode ser configurado com valores positivos ou negativos.

**NOTA:** Os comandos de calibração do ganho proporcional, são recebidos e armazenados para serem executados sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
PG	<b>PROPORTIONAL GAIN</b> Comando de identificação utilizado para configurar o ganho proporcional para perdas mecânicas de movimentação.		Sem parâmetro.
CA	<b>CURVE ANGLE</b> Comando de configuração utilizado para indicar a porcentagem de ganho para perdas de movimentação em curva. Unidade de medida utilizado no parâmetro é em porcentagem.	-	Símbolo negativo utilizado antes do valor de porcentagem indica ganho negativo.
		0 até 99,99	Valor de ganho em porcentagem
DF	<b>DIFFERENTIAL CURVE</b> Comando de configuração utilizado para indicar curva no modo diferencial.	-	Símbolo negativo utilizado antes do valor de porcentagem indica ganho negativo.
		0 até 99,99	Valor de ganho em porcentagem
RI	<b>RADIUS INNER</b> Comando de configuração utilizado para indicar porcentagem do raio interno da curva no modo diferencial. Unidade de medida utilizado no parâmetro é em porcentagem.	-	Símbolo negativo utilizado antes do valor de porcentagem indica ganho negativo.
		0 até 99,99	Valor de ganho em porcentagem

(continua)

(conclusão)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SO	<b>STRAIGHT ON</b> Comando de controle utilizado para indicar porcentagem de distância na movimentação em linha reta. Unidade de medida utilizado no parâmetro é em porcentagem.	-	Símbolo negativo utilizado antes do valor de porcentagem indica ganho negativo.
		0 até 99,99	Valor de ganho em porcentagem

Tabela 8 – Tabela de comandos utilizados para parametrização do ganho proporcional.

#### 5.2.6.1. Ajuste do ganho proporcional em linha reta

O ajuste de ganho em linha reta deve ser feito realizando a movimentação de 1 metro para frente.

Para iniciar o procedimento, posicionar o SoBot marcando o ponto de partida e em seguida enviar o comando de movimentação para frente.

*- Caso não complete a movimentação de 1 metro, posicionar o SoBot novamente no ponto de partida e ajustar o comando repetindo o procedimento até realizar a movimentação cravada de 1 metro!*

**NOTA:** Utilizar os parâmetros de tempo de aceleração, desaceleração e velocidade utilizado na movimentação padrão do SoBot.

*- Para calcular o valor de porcentagem do ganho proporcional em linha reta, utilize a seguinte formula:*

$$PG\_SO = \frac{\text{Valor para 1 metro} - 1000}{10}$$

#### Exemplo de uso:

Para que o SoBot percorra a distância de 1 metro, foi necessário enviar um comando de 1013,5 milímetros, sendo assim, o valor de porcentagem do ganho proporcional ficou:

$$PG\_SO = \frac{1013,5 - 1000}{10} = 1,35\%$$

```
1 # Comando de ganho proporcional em linha reta
2 serial.write(b"PG SO1,35")
```

### 5.2.6.2. Ajuste do ganho proporcional em curva no próprio eixo

O ajuste de ganho de curva no próprio eixo deve ser feito realizando a movimentação de 360° para qualquer uma das direções.

Para iniciar o procedimento, posicionar o SoBot marcando o ponto de partida e em seguida enviar o comando de movimentação de curva no próprio eixo.

*- Caso não complete a movimentação de 360°, posicionar o SoBot novamente no ponto de partida e ajustar o comando repetindo o procedimento até que o SoBot realize a movimentação de 360°.*

**NOTA:** Utilizar os parâmetros de tempo de aceleração, desaceleração e velocidade utilizado na movimentação padrão do SoBot.

*- Para encontrar o valor de porcentagem do ganho proporcional de curva no próprio eixo, utilize a seguinte formula:*

$$PG\_CA = \frac{(\text{Valor para } 360^\circ - 360) * 100}{360}$$

#### Exemplo de uso:

Para que o SoBot faça uma curva de 360°, foi necessário enviar um comando de 370,34°, sendo assim, o valor de porcentagem do ganho proporcional ficou:

$$PG\_CA = \frac{(370,34 - 360) * 100}{360} = 2,87\%$$

1	# Comando de ganho proporcional em curva no próprio eixo
2	serial.write(b"PG CA2,87")

### 5.2.6.3. Ajuste do ganho proporcional em curva diferencial

O ajuste de ganho de curva diferencial deve ser feito realizando a movimentação de 360° para qualquer uma das direções.

Para iniciar o procedimento, posicionar o SoBot marcando o ponto de partida e em seguida enviar o comando de movimentação de curva diferencial. Caso não complete a movimentação de 360° com a distância de raio interno de 100 milímetros, posicionar o



SoBot novamente no ponto de partida e ajustar o comando repetindo o procedimento até que o SoBot realize a movimentação de 360° com raio interno de 100 milímetros.

**NOTA:** Utilizar os parâmetros de tempo de aceleração, desaceleração e velocidade utilizado na movimentação padrão do SoBot.

Para encontrar o valor de porcentagem do ganho proporcional em graus da curva diferencial, utilize a seguinte formula:

$$PG\_DF = \frac{(\text{Valor para } 360^\circ - 360) * 100}{360}$$

Para encontrar o valor de porcentagem do ganho proporcional do raio interno da curva diferencial, utilize a seguinte formula:

$$PG\_RI = \text{Valor para } 100\text{mm} - 100$$

### Exemplo de uso:

Para que o SoBot faça uma curva de 360°, foi necessário enviar um comando de 372,68° e para que o raio interno fosse 100mm, foi necessário enviar um comando de 95mm, sendo assim, o valor de porcentagem do ganho proporcional para os dois parâmetros ficou:

$$PG\_DF = \frac{(372,68 - 360) * 100}{360} = 3,52\%$$

$$PG\_RI = 95 - 100 = -5\%$$

1	# Comando de ganho proporcional em linha reta
2	serial.write(b"PG DF3,52")
3	serial.write(b"PG RI-5")

Com todos os parâmetros em mãos, é possível realizar a configuração dos mesmos com apenas uma linha de comando separando com caractere de “espaço” cada comando conforme segue exemplo:

1	# Comando para configurar todos os parâmetros em uma linha
2	serial.write(b"PG SO1,35 CA2,87 DF3,52 RI-5")

**NOTA:** O comando de identificação “PG” tem que ser o primeiro para que o SoBot entenda que é um comando de parametrização do ganho proporcional.

### 5.2.7. Comandos de controle dos motores no modo Pulsos por Segundo

**NOTA:** Os comandos de movimentação modo Pulsos por Segundo, são recebidos e executados imediatamente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
MT	<b>MOTOR</b> Comando de identificação utilizado para indicar qual motor irá receber os comandos de controle a seguir.	1	Utilizado de forma exclusiva para controlar o motor 1 no modo controle por pulsos.
		2	Utilizado de forma exclusiva para controlar o motor 2 no modo controle por pulsos.
E	<b>ENABLE</b> Comando de controle utilizado para habilitar a movimentação dos motores. Utilize 0 no parâmetro para desabilitar o motor e 1 para habilitar a movimentação do motor.	0	Utilizado para desabilitar o motor.
		1	Utilizado para habilitar o motor.
PS	<b>PULSE SPEED</b> Comando de configuração utilizado para indicar a quantidade de pulsos por segundo que será aplicado ao driver de controle da saída do motor de passo.	-	Símbolo - negativo utilizado antes do valor de distância indica movimento reverso.
		0 até 2.000	Valor de pulsos por segundos. <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

#### Exemplos de uso:

##### 5.2.7.1. Controle individual dos motores:

- Habilita o motor 1, configura o controle de pulsos para ser aplicado 200 pulsos por segundo durante 2 segundos e depois desliga o motor 1 e na sequência habilita o motor 2, configura o controle de pulsos para ser aplicado 300 pulsos por segundo durante 2 segundos e depois desliga o motor 2.

1	# Movimentação dos motores individuais por pulsos	
2	serial.write(b"MT1 E1")	# Habilita o motor 1
3	serial.write(b"MT1 E1 PS200")	# Movimenta o motor 1 a 200 pulsos por segundo
4	sleep(2)	# Aguarda 2 segundos
5	serial.write(b"MT1 E0")	# Desabilita o motor 1
6		
7	serial.write(b"MT2 E1")	# Habilita o motor 2
8	serial.write(b"MT2 E1 PS300")	# Movimenta o motor 2 a 300 pulsos por segundo
9	sleep(2)	# Aguarda 2 segundos
10	serial.write(b"MT2 E0")	# Desabilita o motor 2

### 5.2.7.2. Controle simultâneo dos motores:

- Habilita os motores, configura o controle de pulsos para ser aplicado 500 pulsos por segundo durante 5 segundos para frente e depois desliga os motores e na sequência movimenta os motores para trás utilizando o controle de pulsos por segundo.

1	# Movimentação dos motores simultâneos por pulsos para frente	
2	serial.write(b"MT1 E1 MT2 E1")	# Habilita o motor 1 e o motor 2
3	serial.write(b"MT1 E1 PS500 MT2 E1 PS500")	# Movimenta os motores a 500 pulsos por
4		# segundo para frente
5	sleep(5)	# Aguarda 5 segundos
6	serial.write(b"MT1 E0 MT2 E0")	# Desabilita os motores
7	sleep(2)	# Aguarda 2 segundos
8	serial.write(b"MT1 E1 MT2 E1")	# Habilita o motor 1 e o motor 2
9	serial.write(b"MT1 E1 PS-500 MT2 E1 PS-500")	# Movimenta os motores a 500 pulsos por
10		# segundo para trás
11	sleep(5)	# Aguarda 5 segundos
12	serial.write(b"MT1 E0 MT2 E0")	# Desabilita os motores

### 5.2.7.3. Realizando curvas com controle dos pulsos

- Habilita os motores, configura o controle de pulsos para executar a movimentação com curva no mesmo eixo e em seguida configura o controle de pulsos para realizar curva diferencial.

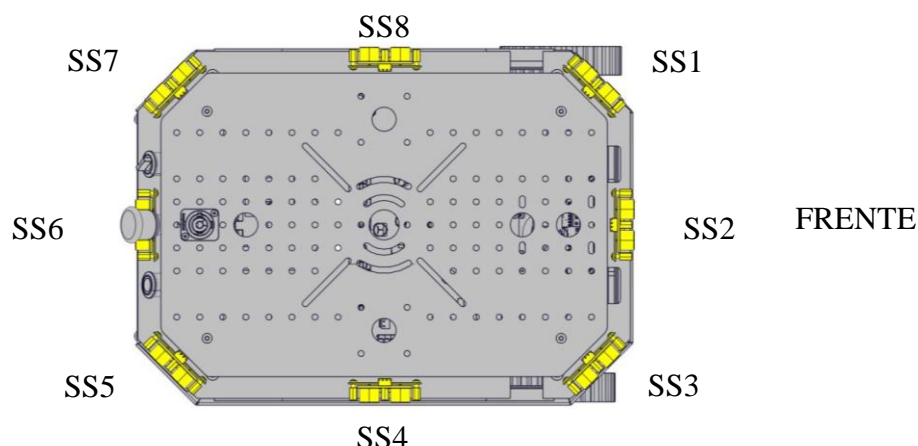
1	# Movimentação dos motores simultâneos por pulsos para frente	
2	serial.write(b"MT1 E1 MT2 E1")	# Habilita o motor 1 e o motor 2
3	serial.write(b"MT1 E1 PS-500 MT2 E1 PS500")	# Movimenta os motores a 500 pulsos por
4		# segundo movendo o SoBot para
5		# esquerda com curva no mesmo eixo
6	sleep(5)	# Aguarda 5 segundos
7	serial.write(b"MT1 E0 MT2 E0")	# Desabilita os motores
8	sleep(2)	# Aguarda 2 segundos
9	serial.write(b"MT1 E1 MT2 E1")	# Habilita o motor 1 e o motor 2
10	serial.write(b"MT1 E1 PS500 MT2 E1 PS200")	# Movimenta os motores para realizar
11		# curva diferencial para direita
12	sleep(5)	# Aguarda 5 segundos
13	serial.write(b"MT1 E0 MT2 E0")	# Desabilita os motores

**NOTA:** O comando de identificação “MT” tem que ser o primeiro para que o SoBot entenda que é um comando de controle dos motores.

### 5.3. COMANDO PARA LEITURA DOS SENSORES SONAR

O SoBot possui 8 sensores ultrassônicos de distância fixados ao redor de sua carenagem capaz de detectar objetos em um range de 4 metros.

A seguir imagem com indicações do posicionamento dos sensores:



**NOTA:** O comando para realizar a leitura dos sensores sonar do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SS	<b>SENSOR SONAR</b> Comando de identificação utilizado para realizar a leitura dos sensores ultrassônicos de distância.	0	Utilizado para obter a distância medida de todos os sensores sonar ao mesmo tempo
		1	Utilizado para obter a distância medida do sensor sonar 1
		2	Utilizado para obter a distância medida do sensor sonar 2
		3	Utilizado para obter a distância medida do sensor sonar 3
		4	Utilizado para obter a distância medida do sensor sonar 4
		5	Utilizado para obter a distância medida do sensor sonar 5
		6	Utilizado para obter a distância medida do sensor sonar 6
		7	Utilizado para obter a distância medida do sensor sonar 7
		8	Utilizado para obter a distância medida do sensor sonar 8

Tabela 9 – Comando para leitura dos sensores SONAR.

Este comando tem como retorno valores medidos em milímetros, sendo seu range de medição de 0 até 4000mm e também retorna indicação de falha de medição ou sensor desconectado.

RETORNO	DESCRIÇÃO
"SS1 0150"	Status de retorno indicando que o sensor 1 está a 150mm de distância de um obstáculo
"SS1 FFFF"	Status de retorno com letras "F" indica que a medição de distância ultrapassou os 4000mm
"SS1 NULL"	Status de retorno com a palavra "NULL" indica falha de medição ou sensor desconectado

Tabela 10 – Exemplo de retorno do comando para leitura dos sensores SONAR.

### Exemplo de uso:

- Solicitação de leitura de todos os sensores ao mesmo tempo.

1	# Solicitação de status de todos os sensores sonar
2	serial.write(b"SS0")

Retorno da solicitação de leitura dos sensores sonares enviado no exemplo anterior:

Terminal Serial
SS1 0150 SS2 0210 SS3 0182 SS4 1140 SS5 1352 SS6 2540 SS7 1364 SS8 1193

A resposta do status dos sensores inicia pelo sensor 1 e segue sucessivamente até o sensor 8, onde o sensor 1 está a 150mm de distância de algum obstáculo, seguindo até o sensor 8 que está a 1193mm de distância de algum obstáculo.

## 5.4. COMANDO PARA LEITURA DO SENSOR INFRAVERMELHO DE DISTÂNCIA

O SoBot possui um sensor infravermelho de distância localizado na parte frontal que auxilia na detecção ou manipulação de objetos.

**NOTA:** O comando para realizar a leitura do sensor infravermelho de distância do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SI	<b>SENSOR INFRARED</b> Comando de identificação utilizado para realizar a leitura do sensor infravermelho de distância.		Sem parâmetro.

Tabela 11 – Comando para leitura do sensor infravermelho de distância.

Este comando tem como retorno o valor medido em centímetros sendo seu range de medição de 20cm até 100cm.

RETORNO	DESCRIÇÃO
"SI 050"	Status de retorno indicando que o sensor está a 50cm de distância de um obstáculo
"SI FFF"	Status de retorno com letras "F" indica que a medição de distância ultrapassou os 150cm

Tabela 12 – Exemplo de retorno do sensor infravermelho de distância.

### Exemplo de uso:

- Solicitação de leitura do sensor infravermelho

1	# Solicitação de status do sensor infravermelho
2	<code>serial.write(b"SI")</code>

Retorno da solicitação de leitura do sensor infravermelho enviado no exemplo anterior:

Terminal Serial
SI 050

A resposta do status do sensor infravermelho indica que tem um obstáculo a 50cm de distância.



## 5.5. COMANDO PARA LEITURA DOS SENSORES DE LINHA

O SoBot possui sensores fotoelétricos, posicionados na parte de baixo da carenagem frontal a função de identificar o contraste de superfícies (preto e branco).

Desta forma quando um sensor não detecta o reflexo, indica que está sobre uma superfície de cor preta e com o conjunto de sensores, ele pode realizar tarefas como seguir faixas de cor preta ou branca para realizar trajetos ou tomadas de decisão.

**NOTA:** O comando para realizar a leitura da PCI Sensores de Linha do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SL	<b>SENSOR LINE</b> Comando de identificação utilizado para realizar a leitura dos sensores fotoelétricos para detecção de faixa.		Sem parâmetro.

Tabela 13 – Comando para leitura dos sensores de linha.

Este comando tem como retorno o valor de 0 ou 1 para indicar ou não a detecção de faixa na cor preta.

RETORNO	DESCRIÇÃO
"SL1 0 SL2 0 SL3 0"	Status de retorno 0 indica que o sensor está sobre uma superfície reflexiva e retorno 1 para superfície não reflexiva (preto)

Tabela 14 – Exemplo de retorno do comando para leitura dos sensores de linha.

### Exemplo de uso:

- Solicitação de leitura dos sensores de linha:

1	# Solicitação de status de todos os sensores de linha
2	serial.write(b"SL")

Retorno da solicitação de leitura do sensor infravermelho enviado no exemplo anterior:

Terminal Serial
SL1 0 SL2 1 SL3 0

A resposta da leitura dos sensores de linha indica que o sensor SL1 e SL3 estão sobre superfícies reflexivas e o sensor SL2 está sobre uma superfície de cor preta.

## 5.6. COMANDO PARA LEITURA DAS ENTRADAS DIGITAIS

A Placa de Controle e Potência do SoBot, possui 8 entradas do tipo digital que podem ser requisitadas individualmente ou de uma única vez para a leitura das 8 entradas ao mesmo tempo.

**NOTA:** O comando para realizar a leitura das entradas digitais do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
DI	<b>INPUT DIGITAL</b> Comando de identificação utilizado para realizar a leitura das entradas digitais.	0	Utilizado para obter o status de todas as entradas digitais ao mesmo tempo
		1	Utilizado para obter o status da entrada digital 1
		2	Utilizado para obter o status da entrada digital 2
		3	Utilizado para obter o status da entrada digital 3
		4	Utilizado para obter o status da entrada digital 4
		5	Utilizado para obter o status da entrada digital 5
		6	Utilizado para obter o status da entrada digital 6
		7	Utilizado para obter o status da entrada digital 7
		8	Utilizado para obter o status da entrada digital 8

Tabela 15 – Comando para leitura das entradas digitais.

Este comando tem como retorno o valor de 0 ou 1 para indicar se a entrada está em nível lógico baixo ou alto.

RETORNO	DESCRIÇÃO
"DI1 0"	Status de retorno 0 indica que a entrada digital está em nível lógico 0 e retorno 1 para leitura de nível lógico alto

Tabela 16 – Exemplo de retorno do comando para leitura das entradas digitais.

### Exemplo de uso:

- Solicitação de leitura de todas as entradas digitais:

1	# Solicitação de status de todas as entradas digitais
2	serial.write(b"DIO")

Retorno da solicitação de leitura das entradas digitais enviado no exemplo anterior:

Terminal Serial															
DI1	1	DI2	1	DI3	0	DI4	0	DI5	0	DI6	0	DI7	0	DI8	0

A resposta da leitura de todas as entradas digitais indica que a entrada DI1 e DI2 estão em nível lógico 1 e as demais entradas estão em nível lógico 0.

## 5.7. CONJUNTO DE COMANDOS PARA CONTROLE DAS SAÍDAS DIGITAIS

A Placa de Controle e Potência do SoBot, possui 8 saídas digitais de controle individual que podem ser configuradas com tempo para desligar automaticamente.

**NOTA:** Os comandos para controlar as saídas digitais do SoBot, são recebidos e armazenados para serem executado sequencialmente, com exceção quando utilizado tempo para desligar, o comando é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
DO	<b>OUTPUT DIGITAL</b> Comando de identificação utilizado para controlar as saídas digitais.  O controle das saídas digitais deve ser feito de forma individual.	1	Utilizado para controlar a saída digital 1
		2	Utilizado para controlar a saída digital 2
		3	Utilizado para controlar a saída digital 3
		4	Utilizado para controlar a saída digital 4
		5	Utilizado para controlar a saída digital 5
		6	Utilizado para controlar a saída digital 6
		7	Utilizado para controlar a saída digital 7
		8	Utilizado para controlar a saída digital 8
E	<b>ENABLE</b> Comando utilizado para habilitar a saída digital.  Utilize 0 no parâmetro para desabilitar e 1 para habilitar a saída.	0	Utilizado para desabilitar a saída digital.
		1	Utilizado para habilitar a saída digital.
TM	<b>TIMER</b> Comando utilizado para desligar a saída com tempo pré-definido.  Unidade de medida utilizado no parâmetro é em milissegundos.	0 até 60.000	Valor do tempo para desligar a saída com unidade de medida em milissegundos.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

Tabela 17 – Conjunto de comandos para controle das saídas digitais.

### Exemplo de uso:

- Comandos para controlar a saída 01 e saída 02, sendo que para a saída 02 o comando com temporizador:

1	# Comandos de controle das saídas digitais
2	serial.write(b"DO1 E1") # Habilita a saída digital 01
3	serial.write(b"DO2 E1 TM2000") # Habilita a saída digital 02 com tempo de
4	# desligar em 2 segundos
5	sleep(1) # Aguarda 1 segundos
6	serial.write(b"DO1 E0") # Desabilita a saída digital 01

**NOTA:** O comando de identificação “DO” tem que ser o primeiro para que o SoBot entenda que é um comando de controle da saída digital.

Neste exemplo a saída digital 01 e 02 são acionadas praticamente ao mesmo tempo e a saída 01 é desligada em 1 segundo e a saída 02 em 2 segundos após o seu acionamento.

## 5.8. COMANDO PARA LEITURA DAS ENTRADAS ANALÓGICAS

A Placa de Controle e Potência do SoBot, possui 6 entradas do tipo analógicas que podem ser requisitadas individualmente para realizar a leitura de tensão e corrente conforme descrição de cada entrada.

**NOTA:** O comando para realizar a leitura das entradas analógicas do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
AI	<b>INPUT ANALOG</b> Comando de identificação utilizado para realizar a leitura das entradas analógicas. As entradas analógicas 1 e 2 realizam a leitura de tensão no range de 0 a 5 volts, as entradas 3 e 4 realizam a leitura de tensão no range de 0 a 10 volts e as entradas analógicas 5 e 6 realizam a leitura de corrente no range de 4 a 20 miliamperes.	1	Comando para leitura da entrada analógica 1 de 0 a 5Volts
		2	Comando para leitura da entrada analógica 2 de 0 a 5Volts
		3	Comando para leitura da entrada analógica 3 de 0 a 10Volts
		4	Comando para leitura da entrada analógica 4 de 0 a 10Volts
		5	Comando para leitura da entrada analógica 5 de 4 a 20miliAmperes
		6	Comando para leitura da entrada analógica 6 de 4 a 20miliAmperes

Tabela 18 – Comando para leitura das entradas analógicas.

Unidade de medida utilizada no retorno da entrada 1, 2, 3 e 4 é em milivolts e das entradas 5 e 6 é em microamperes.

RETORNO	DESCRIÇÃO
"AI1 1550"	Status de retorno indicando que a entrada analógica 1 está medindo 1550 milivolts
"AI3 08100"	Status de retorno indicando que a entrada analógica 3 está medindo 8100 milivolts
"AI5 009800"	Status de retorno indicando que a entrada analógica 5 está medindo 9800 microamperes

Tabela 19 – Exemplo de retorno do comando para leitura das entradas analógicas.

### Exemplo de uso:

- Comandos para ler as entradas analógicas AI1, AI3 e AI5:

```
1 # Solicitação de status das entradas analógicas
2 serial.write(b"AI1") # Solicita status da entrada analógica 01
3 serial.write(b"AI3") # Solicita status da entrada analógica 03
4 serial.write(b"AI5") # Solicita status da entrada analógica 05
```

Retorno das solicitações de leitura das entradas analógicas enviadas no exemplo anterior:

```
Terminal Serial

AI1 1550
AI3 08100
AI5 09800
```

As respostas das leituras das entradas analógicas solicitadas indicam que a entrada AI1 está lendo 1,55 Volts, a entrada AI3 está lendo 8,1 Volts e a entrada AI5 está lendo 9,8 miliamperes.

## 5.9. CONJUNTO DE COMANDOS PARA CONTROLE DAS SAÍDAS ANALÓGICAS

A Placa de Controle e Potência do SoBot, possui 2 saídas analógicas de controle individual no formato PWM com configuração de frequência de trabalho e porcentagem de duty-cycle ajustável.

**NOTA:** Os comandos para controlar as saídas analógicas do SoBot, são recebidos e armazenados para serem executado sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
AO	<b>OUTPUT ANALOG PWM</b> Comando de identificação utilizado para controle da saída analógica PWM.	1	Utilizado para controlar a saída analógica PWM 1.
		2	Utilizado para controlar a saída analógica PWM 2.
E	<b>ENABLE</b> Comando utilizado para habilitar a saída analógica PWM. Utilize 0 no parâmetro para desabilitar e 1 para habilitar a saída.	0	Utilizado para desabilitar a saída analógica.
		1	Utilizado para habilitar a saída analógica.
F	<b>FREQUENCY (Hz)</b> Comando utilizado para indicar a frequência de trabalho em hertz da saída pré-selecionada no comando de identificação.	0 até 1.000	0 até 1.000 Valor da frequência em hertz da saída analógica. <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
DC	<b>DUTY-CYCLE</b> Comando utilizado para indicar porcentagem de trabalho do duty cycle da saída pré-selecionada no comando de identificação.	0 até 100	Valor da porcentagem de duty cycle da saída analógica.

Tabela 20 – Conjunto de comandos para controle das saídas analógicas.

### Exemplo de uso:

- Comandos para controlar a saída analógica 1 ligando em 1.000Hz, 30% de duty cycle e desligando após 5 segundos.

1	# Comandos de controle da saída analógica 1
2	serial.write(b"AO1 E1 F1000 DC30") # Liga a saída analógica 01
3	sleep(5) # Aguarda 5 segundos
4	serial.write(b"AO1 E0") # Desliga a saída analógica 01

**NOTA:** O comando de identificação “AO” tem que ser o primeiro para que o SoBot entenda que é um comando de controle da saída analógica.

## 5.10. CONJUNTO DE COMANDOS PARA CONTROLE DA FITA DE LED

O SoBot possui sinalização luminosa por fita LED RGB ao redor de sua carenagem, onde é possível realizar o controle da intensidade luminosa de suas cores para indicar funções sendo executadas, ou simplesmente para mudar o visual do SoBot.



**NOTA:** Os comandos para controle da fita LED do SoBot, são recebidos e armazenados para serem executado sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
LT	<b>LED TAPE</b> Comando de identificação utilizado para controle da fita LED.		Sem parâmetro.
E	<b>ENABLE</b> Comando utilizado para habilitar o controle de intensidade da fita LED. Utilize 0 no parâmetro para desabilitar e 1 para habilitar o controle.	0	Utilizado para desabilitar o controle da fita LED.
		1	Utilizado para habilitar o controle da fita LED.
RD	<b>REED</b> Comando utilizado para controlar a intensidade da cor vermelha da fita LED. O range de controle da intensidade é de 0 até 255 onde 255 é a intensidade máxima da cor.	0 até 255	Valor da intensidade da cor.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
GR	<b>GREEN</b> Comando utilizado para controlar a intensidade da cor verde da fita LED. O range de controle da intensidade é de 0 até 255 onde 255 é a intensidade máxima da cor.	0 até 255	Valor da intensidade da cor.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
BL	<b>BLUE</b> Comando utilizado para controlar a intensidade da cor azul da fita LED. O range de controle da intensidade é de 0 até 255 onde 255 é a intensidade máxima da cor.	0 até 255	Valor da intensidade da cor.  <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.

Tabela 21 – Conjunto de comandos para controle da fita LED.

### Exemplo de uso:

- Comando para controlar a fita LED na cor vermelha:

1	# Comandos de controle da fita LED		
2	serial.write(b" LT E1 RD50 GR0 BL0")	# Liga a fita LED em vermelho	
3	sleep(5)	# Aguarda 5 segundos	
4	serial.write(b" LT E0")	# Desliga a fita LED	

**NOTA:** O comando de identificação “LT” tem que ser o primeiro para que o SoBot entenda que é um comando de controle da fita LED e se atente ao uso da intensidade máxima da fita LED devido ao seu consumo de energia principalmente quando estiver na bateria.

## 5.11. CONJUNTO DE COMANDOS PARA CONTROLE DO BUZZER

O SoBot possui sinalização sonora por Buzzer com oscilador interno e ao ser habilitado, realiza o acionamento intermitente de aviso.

Os comandos para controlar o sinalizador sonoro Buzzer do SoBot, são recebidos e armazenados para serem executado sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
BZ	<b>BUZZER</b> Comando de identificação utilizado para controle do Buzzer.		Sem parâmetro.
E	<b>ENABLE</b> Comando utilizado para habilitar e desabilitar o Buzzer com oscilador fixo. Utilize 0 no parâmetro para desabilitar e 1 para habilitar o controle.	0	Utilizado para desabilitar o sinalizador buzzer.
		1	Utilizado para habilitar o sinalizador buzzer.

Tabela 22 – Conjunto de comandos para controle do Buzzer.

### Exemplo de uso:

- Comandos para controlar o sinalizador buzzer

```
1 # Comandos de controle da fita LED
2 serial.write(b"BZ E1")           # Liga o buzzer
3 sleep(5)                         # Aguarda 5 segundos
4 serial.write(b"BZ E0")           # Desliga o buzzer
```

## 5.12. COMANDO PARA LEITURA DO SENSOR ACELERÔMETRO

O sensor acelerômetro integrado ao drive do SoBot, tem como resposta a aceleração gravitacional, a velocidade angular nos eixos X, Y e Z e a temperatura medida pelo sensor.

A resposta da aceleração gravitacional tem faixa de escala completa com range de mais ou menos 2g, onde 'g' é a unidade de medida da intensidade do campo gravitacional.

A resposta da velocidade angular é dada pelo sensor giroscópio que tem faixa de escala completa com range de mais ou menos 25 graus por segundo e a unidade de medida é °/s. A resposta da temperatura é dada pelo sensor interno do módulo acelerômetro e tem um range de medição de -40° até 85° Celsius.

**NOTA:** O comando para realizar a leitura do sensor acelerômetro do SoBot, é recebido e executado imediatamente.

COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SA	<b>SENSOR ACCELEROMETER</b> Comando de identificação utilizado para realizar leitura do sensor acelerômetro.		Sem parâmetro.

Tabela 23 – Comando para leitura do sensor acelerômetro.

### Exemplo de uso:

- Solicitação de leitura do sensor acelerômetro:

1	<i># Solicitação de status do sensor acelerômetro</i>
2	<code>serial.write(b"SA")</code>

Retorno da solicitação de leitura do sensor enviado no exemplo anterior:

Terminal Serial
AX0,00 AY0,00 AZ0,91 T32,6 GX10,0 GY-0,4 GZ-2,0

A resposta da leitura do sensor acelerômetro indica que a aceleração gravitacional em X é 0,00g, em Y é 0,00g e em Z é 0,91g, a temperatura local é de 32,6° Celsius e a velocidade angular em X é 10,0°/s, em Y é -0,4°/s e em Z é -2,0°/s.

## 5.13. CONJUNTO DE COMANDOS PARA COMUNICAÇÃO SERIAL

O SoBot possui uma interface de comunicação serial que pode ser controlada por comandos para que o SoBot possa interagir com dispositivos externos por meio da comunicação serial.

**NOTA:** Os comandos para controle da interface de comunicação serial do SoBot, são recebidos e armazenados para serem executado sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SC	<b>SERIAL COMMUNICATION</b> Comando de identificação utilizado para comunicação serial. Os bytes recebidos pela interface serial UART só são redirecionados para a interface USB caso a UART esteja habilitada. O número máximo de dados que podem ser enviados pela UART são 15 bytes. As configurações da interface UART são as seguintes:  Baud Rate – 9600 bps Modo de paridade – Desabilitado Número de Stop Bits – 1 Stop Bit		Sem parâmetro.
E	<b>ENABLE</b> Comando utilizado para habilitar e desabilitar a interface de comunicação serial. Utilize 0 no parâmetro para desabilitar e 1 para habilitar a interface.	0	Utilizado para desabilitar a interface serial.
		1	Utilizado para habilitar a interface serial.
SD	<b>SEND SERIAL</b> Comando utilizado para indicar o envio de caracteres pela serial. O parâmetro utilizado junto ao comando deve ser a quantidade de bytes que serão enviados em seguida.	0 até 15	Quantidade de bytes que serão enviados. <b>NOTA:</b> Se parâmetro maior que o valor máximo permitido, o mesmo será configurado com o valor máximo.
RS	<b>READ SERIAL</b> Comando utilizado para realizar a leitura de caracteres recebidos pela serial. <b>NOTA:</b> O SoBot consegue realizar a leitura de até 15 caracteres sequenciais.		Sem parâmetro. <b>NOTA:</b> Se recebido mais que 15 caracteres, os mesmos não são lidos.

Tabela 24 – Conjunto de comandos para comunicação serial.

Os caracteres de retorno de carro e nova linha também são armazenados na memória de leitura serial.

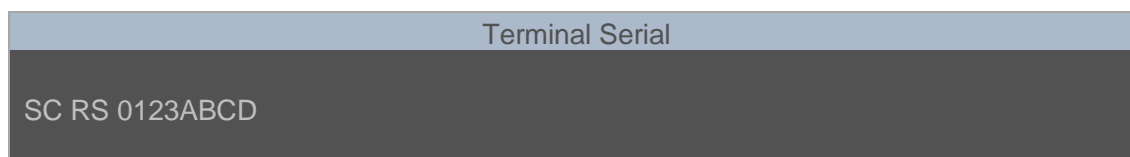
O comando de leitura dos dados recebidos só consegue ler os caracteres após o envio de dados para o dispositivo externo.

## Exemplo de uso:

- Comandos de controle de dados pela serial

1	# Comandos de controle da comunicação serial
2	serial.write(b"SC E1 SD5 HELLO") # Habilita a comunicação serial
3	serial.write(b"SC E1 SD5 READY") # Sinaliza pronto para leitura
4	sleep(5) # Aguarda 5 segundos
5	# Dispositivo externo envia dados "0123ABCD"
6	serial.write(b"SC E1 SD2 OK") # Envia 5 caracteres pela serial
7	serial.write(b"SC RS") # Verifica caracteres recebidos

Retorno da solicitação de leitura de caracteres recebido na serial enviado no exemplo anterior:

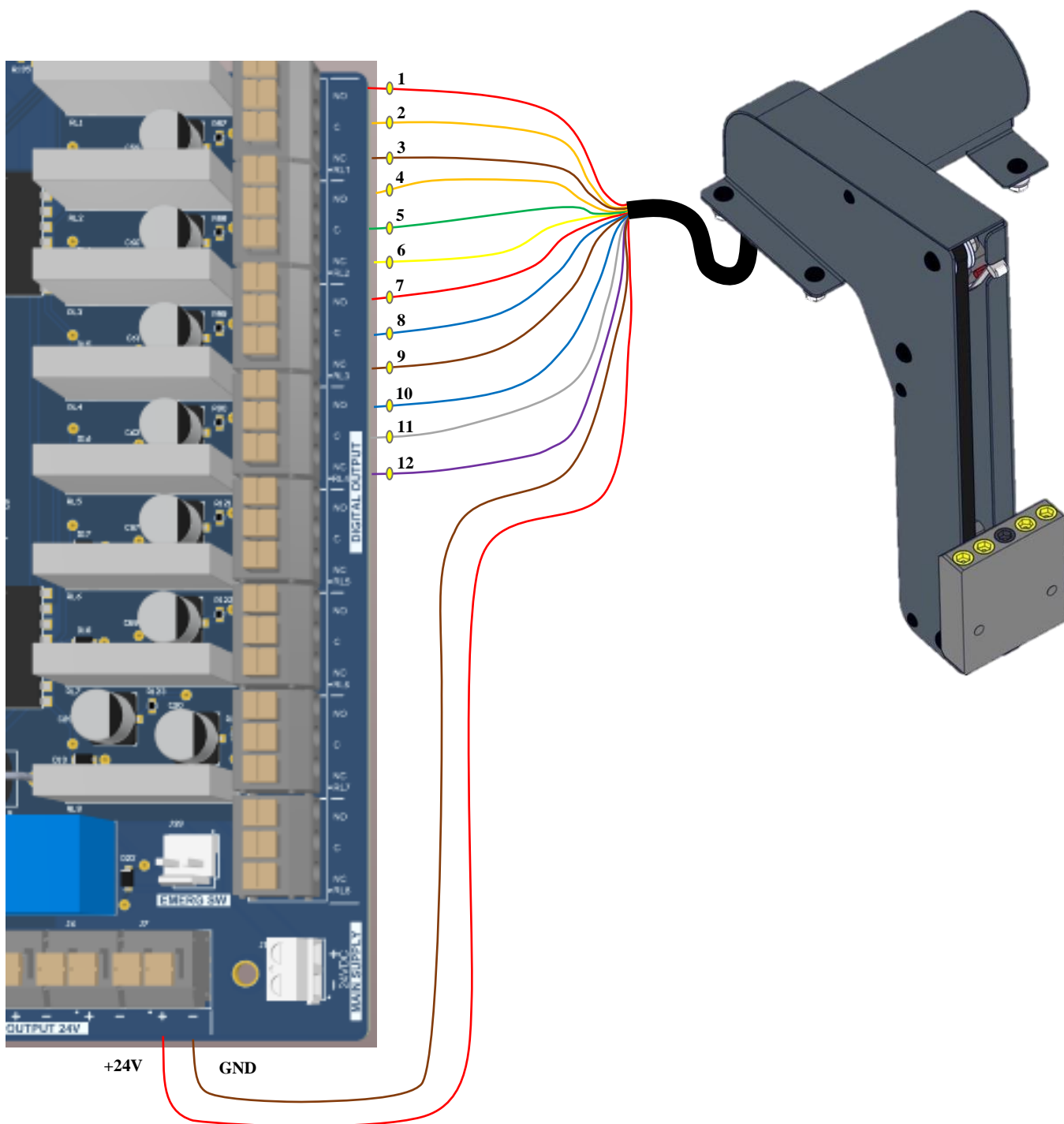


A resposta da leitura dos dados recebidos pela serial indica que foi armazenado dados com os seguintes caracteres "0123ABCD".

## 5.14. CONJUNTO DE COMANDOS PARA CONTROLAR O MÓDULO ELEVADOR

O SoBot possui um módulo do tipo elevador que pode ser utilizado para manipulação de objetos.

Para o funcionamento correto do módulo elevador, o mesmo deve ser ligado conforme diagrama elétrico a seguir:



**NOTA:** Os comandos para controle do módulo elevador do SoBot, são recebidos e armazenados para serem executado sequencialmente.

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
EL	<b>ELEVADOR</b> Comando de identificação utilizado para controlar o elevador ligados as saídas digitais.		Sem parâmetro.

(continua)



(conclusão)

COMANDOS		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
UP	<b>UP</b> Comando utilizado para movimentar o elevador para cima. O elevador irá se movimentar para cima até encontrar o fim de curso ou receber um comando de parar (ST -STOP).		Sem parâmetro.
DN	<b>DOWN</b> Comando utilizado para movimentar o elevador para baixo. O elevador irá se movimentar para baixo até encontrar o fim de curso ou receber um comando de parar (ST -STOP).		Sem parâmetro.
ST	<b>STOP</b> Comando utilizado para parar o movimento do elevador. O elevador irá se movimentar para cima até encontrar o fim de curso ou receber um comando de parar (ST -STOP)		Sem parâmetro.

Tabela 25 – Conjunto de comandos para controlar o módulo elevador.

**Exemplo de uso:**

- Comandos de controle do elevador:

1	# Comandos de controle do elevador		
2	serial.write(b"EL DN")	# Comando para Descer o elevador	
3	sleep(2)	# Aguarda 2 segundos	
4	serial.write(b"EL ST")	# Comando para Parar o elevador	
5	sleep(2)	# Aguarda 2 segundos	
6	serial.write(b"EL UP")	# Comando para Subir o elevador	

**5.15. COMANDO PARA TEMPO DE ATRASO “DELAY”**

O comando de “Delay” tem a função de incluir um comando de tempo entre os comandos que são armazenados para serem executados sequencialmente.

O comando de “Delay”, é recebido e armazenado para ser executado sequencialmente.

COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
DL	<b>DELAY</b> Comando utilizado para que o SoBot aguarde por um tempo antes de executar o próximo comando de ação. Unidade de medida utilizado no parâmetro é em milissegundos.	50 até 60000	Valor do tempo de delay para o próximo comando de ação. <b>NOTA:</b> Se parâmetro maior ou menor que o range estabelecido, o mesmo será configurado com o valor limite.

Tabela 26 – Comando para tempo de atraso “Delay”.

### Exemplo de uso:

- Comando utilizado em conjunto aos comandos do elevador para aguardar completar a descida e subida completo considerando 7 segundos:

1	# Comandos de controle do elevador com tempo de delay	
2	serial.write(b"EL DN")	# Comando para Descer o elevador
3	serial.write(b"DL7000")	# Aguarda tempo de 7 segundos
4	serial.write(b"EL UP")	# Comando para Subir o elevador
5	serial.write(b"DL7000")	# Aguarda tempo de 7 segundos
6	serial.write(b"EL ST")	# Comando para Parar o elevador

## 5.16. COMANDOS PARA OBTER RETORNO DE AÇÕES EXECUTADAS

A configuração de comando de retorno pode ser de forma geral e/ou individual para habilitar ou desabilitar o retorno de ações executadas.

**NOTA:** O comando para controlar o retorno e ações executadas, é recebido e executado imediatamente.

COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
CR	<b>COMMAND RETURN</b> Comando para controlar a confirmação de retorno geral de todas as ações do SoBot	0	Utilizado para desabilitar a confirmação de retorno geral.
		1	Utilizado para habilitar a confirmação de retorno geral.

Tabela 27 – Comando para obter retorno de ações executadas.

### Exemplos de uso:

- Comandos para Feedback / Status:

1	# Comandos de controle do retorno de comandos geral	
2	serial.write(b"CR1")	# Comando para habilitar o retorno geral
3		
4	serial.write(b"CR0")	# Comando para desabilitar o retorno geral

Para casos que o retorno não seja necessário, é possível controlar individualmente a confirmação da ação executada enviando o comando de identificação junto ao comando de retorno conforme exemplos a seguir:

- Para controle do retorno de comandos de movimentações das rodas:

```
1 # Comandos para controle do retorno de comandos dos motores das rodas
2 serial.write(b"MT0 CR1")      # Comando para habilitar o retorno
3
4 serial.write(b"MT0 CR0")      # Comando para desabilitar o retorno
```

Confirmação do retorno de comandos de movimentação das rodas recebidos e executados:

```
Terminal Serial
CR OK MT0
```

- Para controle de retorno da movimentação do Motor 3:

```
1 # Comandos para controle do retorno de comandos do motor 3
2 serial.write(b"MT3 CR1")      # Comando para habilitar o retorno
3
4 serial.write(b"MT3 CR0")      # Comando para desabilitar o retorno
```

Confirmação do retorno de comandos de movimentação do motor 3 recebidos e executados:

```
Terminal Serial
CR OK MT3
```

- Para controle de retorno da fita LED:

```
1 # Comandos para controle do retorno de comandos da fita LED
2 serial.write(b"LT CR1")      # Comando para habilitar o retorno
3
4 serial.write(b"LT CR0")      # Comando para desabilitar o retorno
```

Confirmação do retorno de comandos da fita LED recebidos e executados:

```
Terminal Serial
CR OK LT
```

- Para controle de retorno de todas as saídas digitais:

1	# Comandos para controle do retorno de comandos das saídas digitais
2	<code>serial.write(b"DO CR1")</code> # Comando para habilitar o retorno
3	
4	<code>serial.write(b"DO CR0")</code> # Comando para desabilitar o retorno

Confirmação do retorno de comandos das saídas digitais recebidos e executados:

Terminal Serial
CR OK DO1

A resposta do comando de retorno das saídas digitais tem a indicação de qual saída foi utilizada sendo de DO1 até DO8.

- Para controle de retorno de todas as saídas analógicas:

1	# Comandos para controle do retorno de comandos das saídas analógicas
2	<code>serial.write(b"AO CR1")</code> # Comando para habilitar o retorno
3	
4	<code>serial.write(b"AO CR0")</code> # Comando para desabilitar o retorno

Confirmação do retorno de comandos das saídas analógicas recebidos e executados:

Terminal Serial
CR OK AO1

A resposta do comando de retorno das saídas analógicas tem a indicação de qual saída foi utilizada sendo AO1 ou AO2.

- Para controle de retorno do Buzzer:

1	# Comandos para controle do retorno de comandos do Buzzer
2	<code>serial.write(b"BZ CR1")</code> # Comando para habilitar o retorno
3	
4	<code>serial.write(b"BZ CR0")</code> # Comando para desabilitar o retorno

Confirmação do retorno de comandos do Buzzer recebidos e executados:

Terminal Serial	
CR OK BZ	

- Para controle de retorno do modo elevador:

1	<i># Comandos para controle do retorno de comandos do elevador</i>
2	<code>serial.write(b"EL CR1")</code> <i># Comando para habilitar o retorno</i>
3	
4	<code>serial.write(b"EL CR0")</code> <i># Comando para desabilitar o retorno</i>

Confirmação do retorno de comandos do elevador recebidos e executados:

Terminal Serial	
CR OK EL	

- Para controle de retorno do tempo de delay:

1	<i># Comandos para controle do retorno de comandos do tempo de delay</i>
2	<code>serial.write(b"DL CR1")</code> <i># Comando para habilitar o retorno</i>
3	
4	<code>serial.write(b"DL CR0")</code> <i># Comando para desabilitar o retorno</i>

Confirmação do retorno de comandos do tempo de delay recebidos e executados:

Terminal Serial	
CR OK DL	

- Para controle de retorno dos parâmetros de configuração das rodas:

1	<i># Comandos para controle do retorno de comandos de configuração das rodas</i>
2	<code>serial.write(b"WP CR1")</code> <i># Comando para habilitar o retorno</i>
3	
4	<code>serial.write(b"WP CR0")</code> <i># Comando para desabilitar o retorno</i>

Confirmação do retorno de comandos de configurações das rodas recebidos e executados:

Terminal Serial	
CR OK WP	

- Para controle de retorno dos parâmetros de configuração do ganho proporcional:

1	# Comandos para controle do retorno de comandos de configuração do ganho
2	# proporcional de movimentação
3	serial.write(b"PG CR1") # Comando para habilitar o retorno
4	
5	serial.write(b"PG CR0") # Comando para desabilitar o retorno

Confirmação do retorno de comandos de configurações das rodas recebidos e executados:

Terminal Serial
CR OK PG

## 6. Código de erros

O SoBot pode identificar alguns possíveis erros em comandos recebidos que não fazem parte da sua lista de comandos pré-programados sendo comandos completamente desconhecidos ou apenas algum erro de sintaxe.

### 6.1. ERRO 0

Para comandos completamente desconhecidos, o SoBot retorna o seguinte código de erro:

Terminal Serial
ERROR=00

### 6.2. ERRO 1

Para comandos com erro de sintaxe, o código de erro acompanha o comando de identificação.

Seguem exemplos de retorno de erro 1 de sintaxe:

Retorno de erro no comando para controle dos **motores das rodas**:

Terminal Serial
ERROR=01 MT0

Retorno de erro no comando para controle do **motor 1:**

Terminal Serial
ERROR=01 MT1

Retorno de erro no comando para controle do **motor 2:**

Terminal Serial
ERROR=01 MT2

Retorno de erro no comando para controle do **motor 3:**

Terminal Serial
ERROR=01 MT3

Retorno de erro no comando para leitura dos **sensores sonares:**

Terminal Serial
ERROR=01 SS

Retorno de erro no comando para leitura das **entradas digitais:**

Terminal Serial
ERROR=01 DI

Retorno de erro no comando para acionar as **saídas digitais:**

Terminal Serial
ERROR=01 DO

Retorno de erro no comando para leitura das **entradas analógicas:**

Terminal Serial
ERROR=01 AI



Retorno de erro no comando para controle das **saídas analógicas**:

Terminal Serial
ERROR=01 AO

Retorno de erro no comando para controle da iluminação da **fita LED**:

Terminal Serial
ERROR=01 LT

Retorno de erro no comando para controle do sinalizador **sonoro Buzzer**:

Terminal Serial
ERROR=01 BZ

Retorno de erro no comando para enviar ou receber dados pela **serial**:

Terminal Serial
ERROR=01 SC

Retorno de erro no comando para controle do módulo **elevador**:

Terminal Serial
ERROR=01 EL

Retorno de erro no comando de tempo de **delay**:

Terminal Serial
ERROR=01 DL

Retorno de erro no comando de **configuração das rodas**:

Terminal Serial
ERROR=01 WP

Retorno de erro no comando de configuração do **ganho proporcional**:

Terminal Serial
ERROR=01 PG

### 6.3. ERRO 2

Código de erro para indicar quando está sendo realizado o controle dos motores por comandos de identificação “MT0” e é enviado um comando para controle do motor por pulsos ou vice versa.

Seguem exemplos de retorno do erro 2:

Retorno de erro no comando para controle dos motores das rodas quando os motores estão sendo controlados no modo pulsos por segundo e é enviado comando de identificação “MT0”:

Terminal Serial
ERROR=02 MT1 E1 MT2 E1

Retorno de erro no comando para controle dos motores das rodas quando os motores estão sendo controlados por comandos de identificação “MT0” e é enviado comando de pulsos por segundo:

Terminal Serial
ERROR=02 MT0 E1



INOVANDO NA FORMAÇÃO POR **COMPETÊNCIAS**