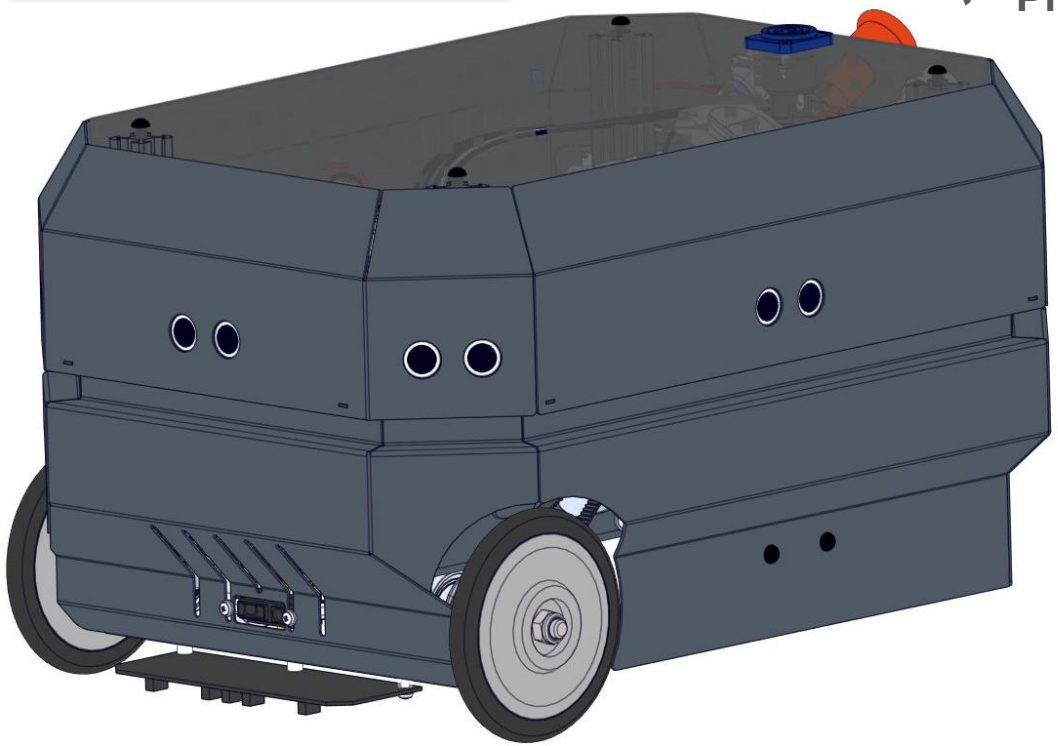
SOLIS

TECNOLOGIA

OPEN BOX

Plataforma de Treinamento em Robótica
SoBot Single



Professional Conception of Robotics

- ✓ **Mecânica:** Montagens e componentes
- ✓ **Eletrônica:** Processadores e sensores
- ✓ **Programação:** Softwares, firmwares e API’s

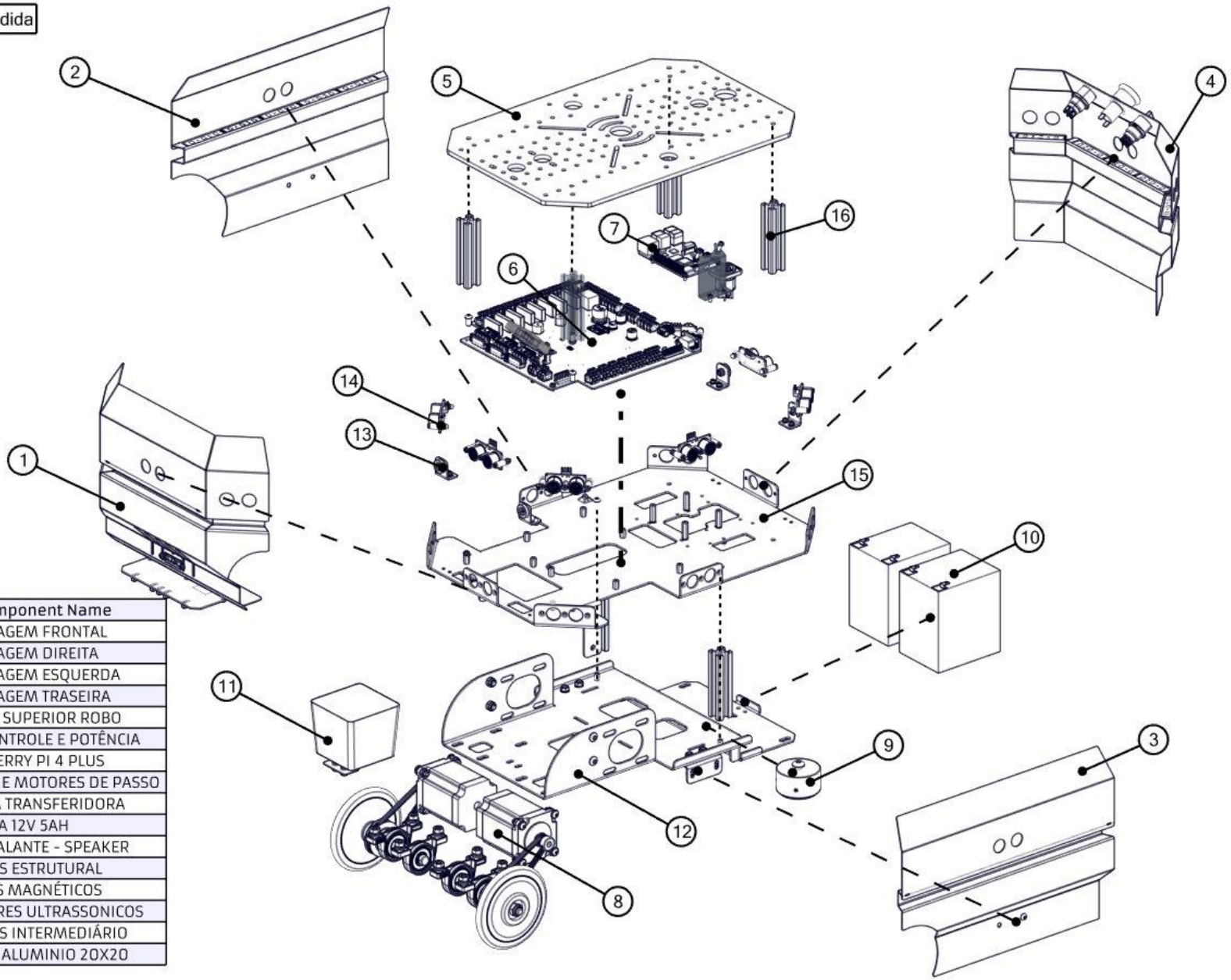
Applications

- ✓ Sistemas de controle com inteligência artificial
- ✓ Programação e navegação autônoma com detecção de obstáculos e planejamento de trajetória.

SOLIS TECNOLOGIA

ASSEMBLY

SoBot - Vista Explodida

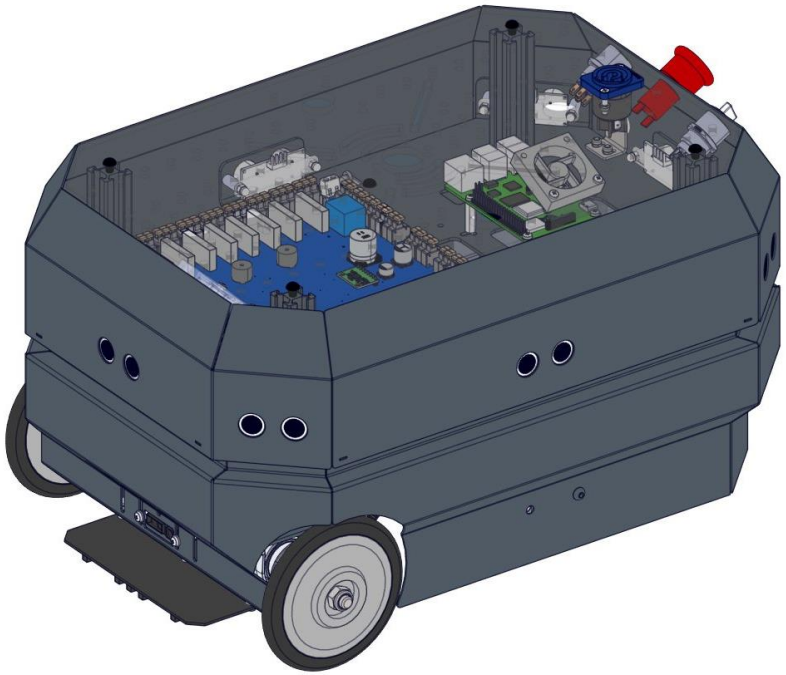


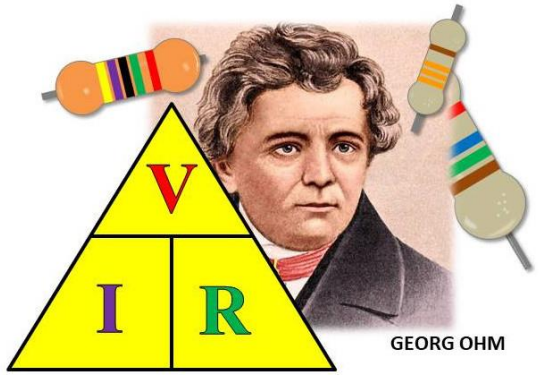
ID da BOM	SW Component Name
1	CARENAGEM FRONTAL
2	CARENAGEM DIREITA
3	CARENAGEM ESQUERDA
4	CARENAGEM TRASEIRA
5	TAMPA SUPERIOR ROBO
6	PCI-CONTROLE E POTÊNCIA
7	RASPBERRY PI 4 PLUS
8	RODAS E MOTORES DE PASSO
9	ESFERA TRANSFERIDORA
10	BATERIA 12V 5AH
11	ALTO FALANTE - SPEAKER
12	CHASSIS ESTRUTURAL
13	BOTÕES MAGNÉTICOS
14	SENSORES ULTRASSONICOS
15	CHASSIS INTERMEDIÁRIO
16	PERFIS ALUMINIO 20X20

SOLIS TECNOLOGIA

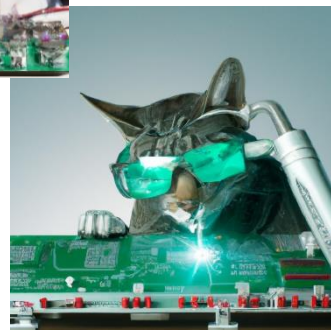
Conceitos Fundamentais

- ✓ **Montagem Mecânica:** subconjuntos e acessórios
- ✓ **Configuração Eletroeletrônica:** Processadores e sensores
- ✓ **Programação:** Desenvolvimento de firmwares e integrações



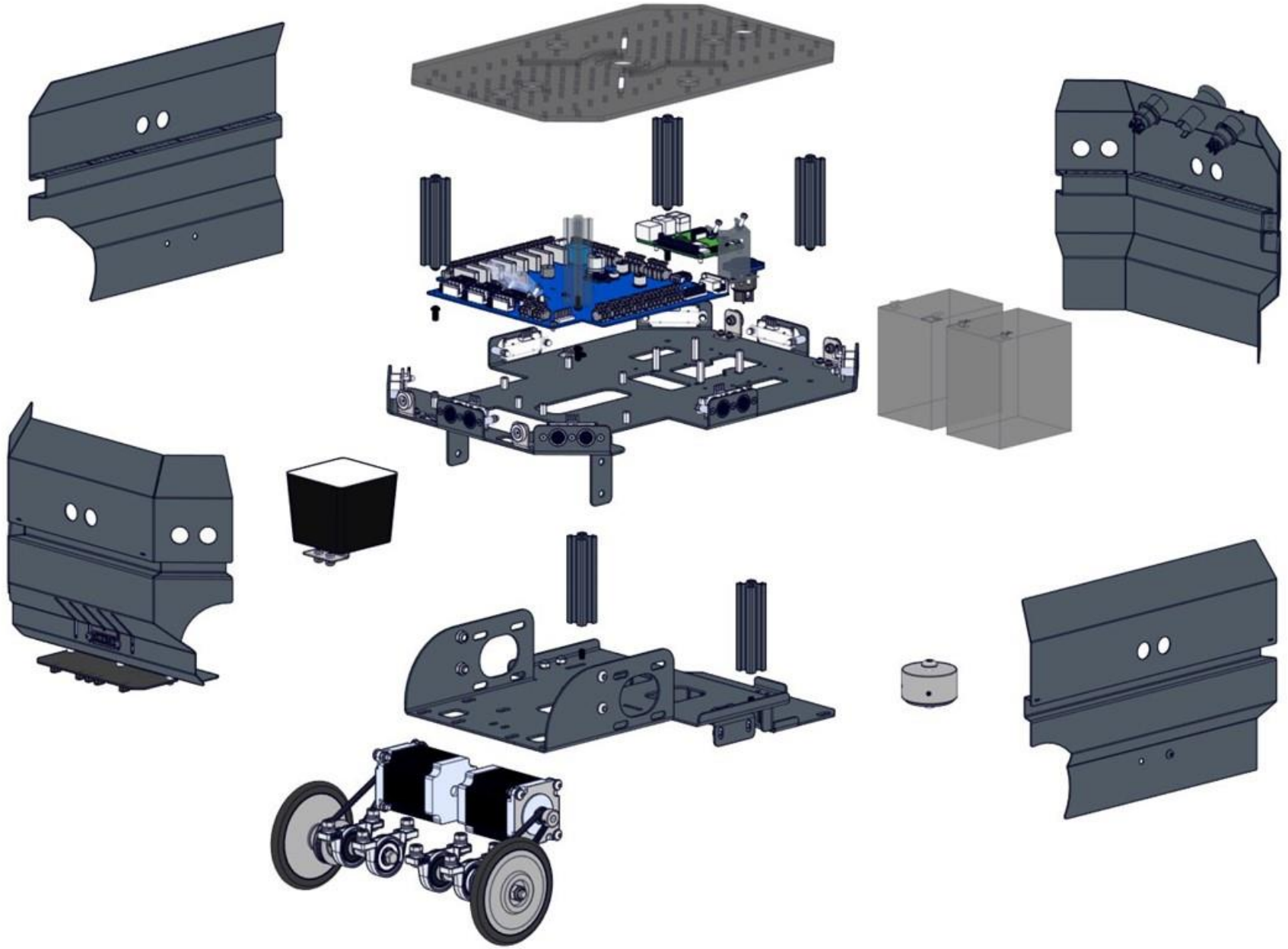


GEORG OHM



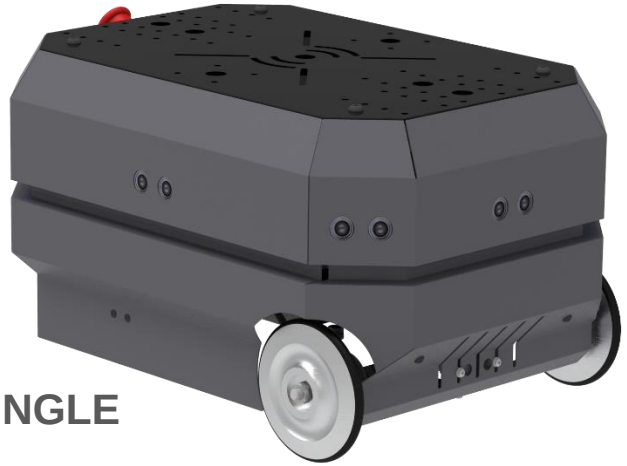
SOLIS TECNOLOGIA

MONTAGEM MECÂNICA



SOLIS
TECNOLOGIA

MONTAGEM MECÂNICA



SINGLE

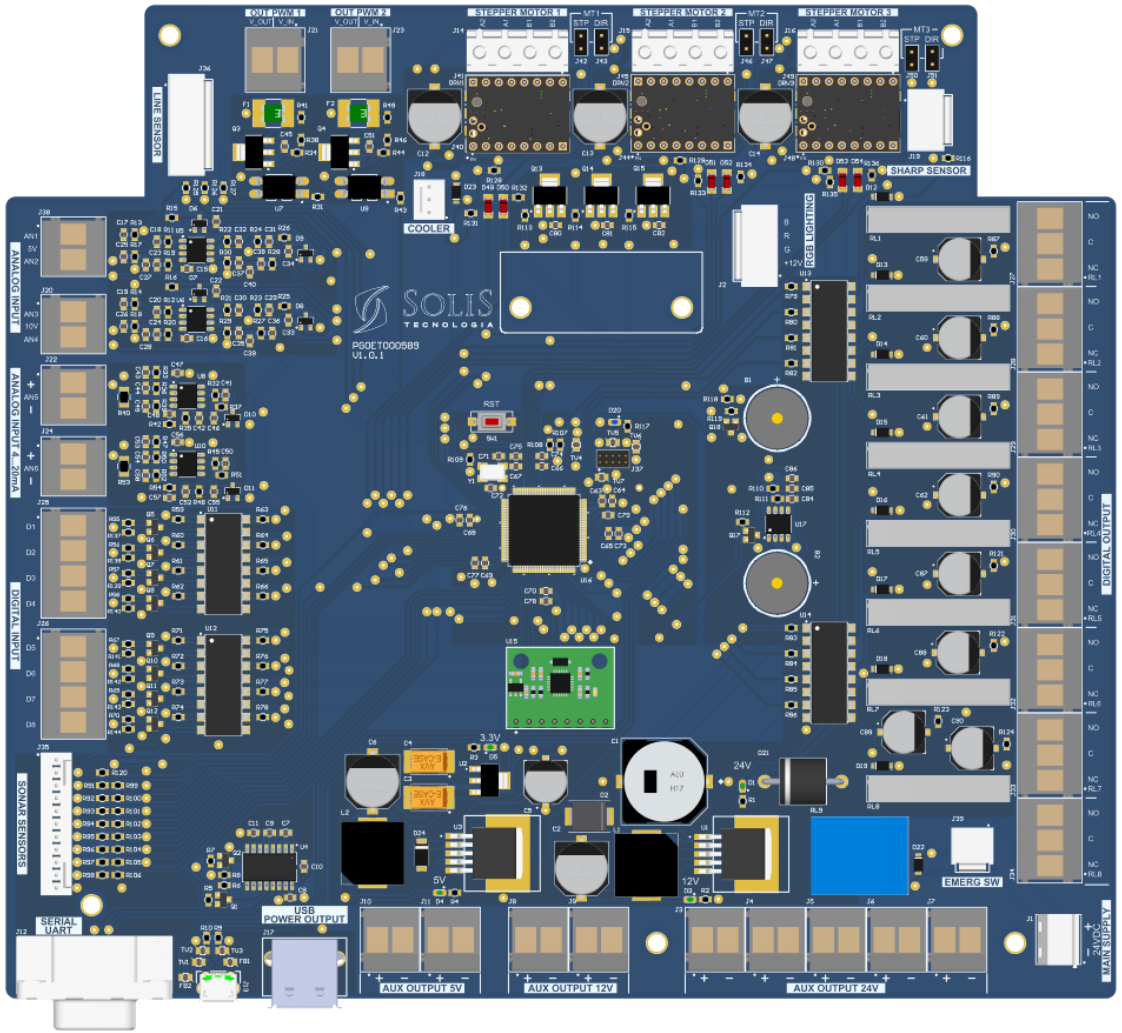
PALLET

MAGNETIC

AGRI-TECH

SOLIS TECNOLOGIA

CONFIGURAÇÃO ELETROELETRÔNICA

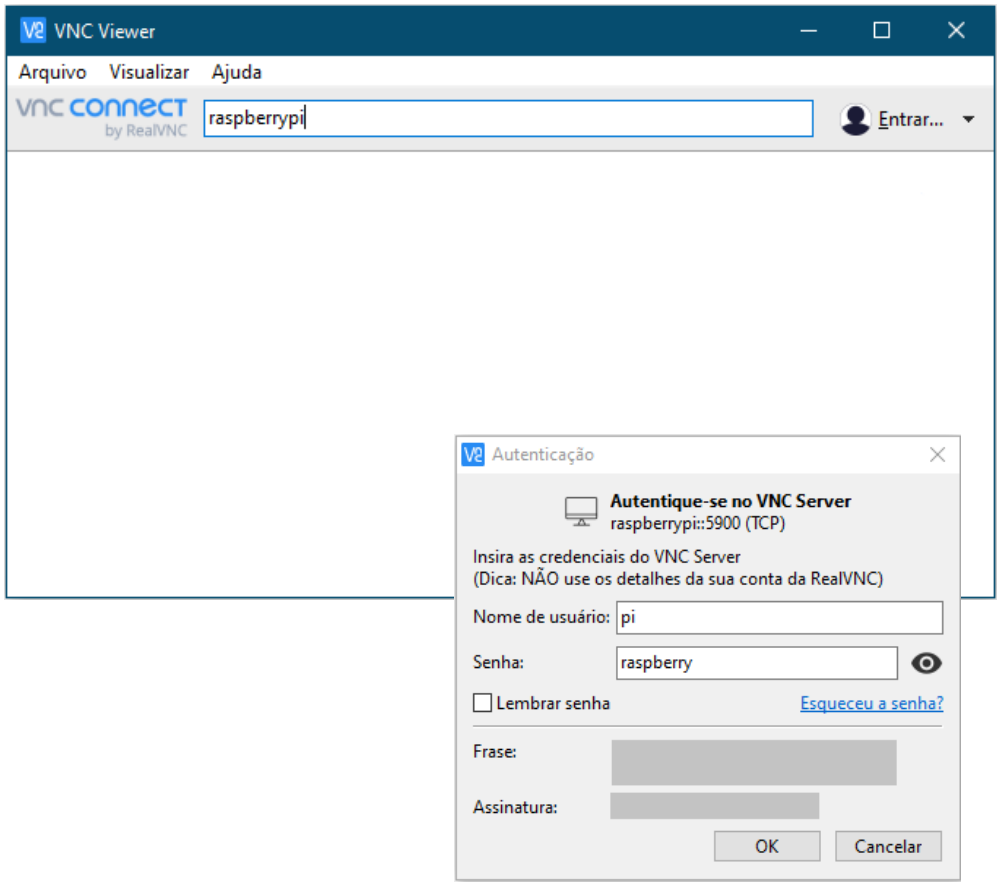


SoBot PCI: Eletrônica Embarcada

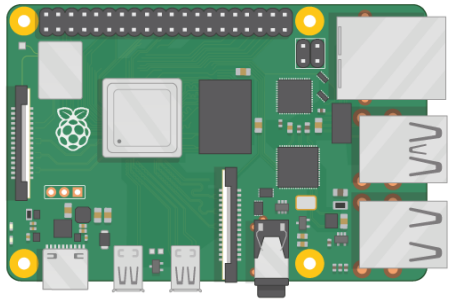
SOLIS TECNOLOGIA

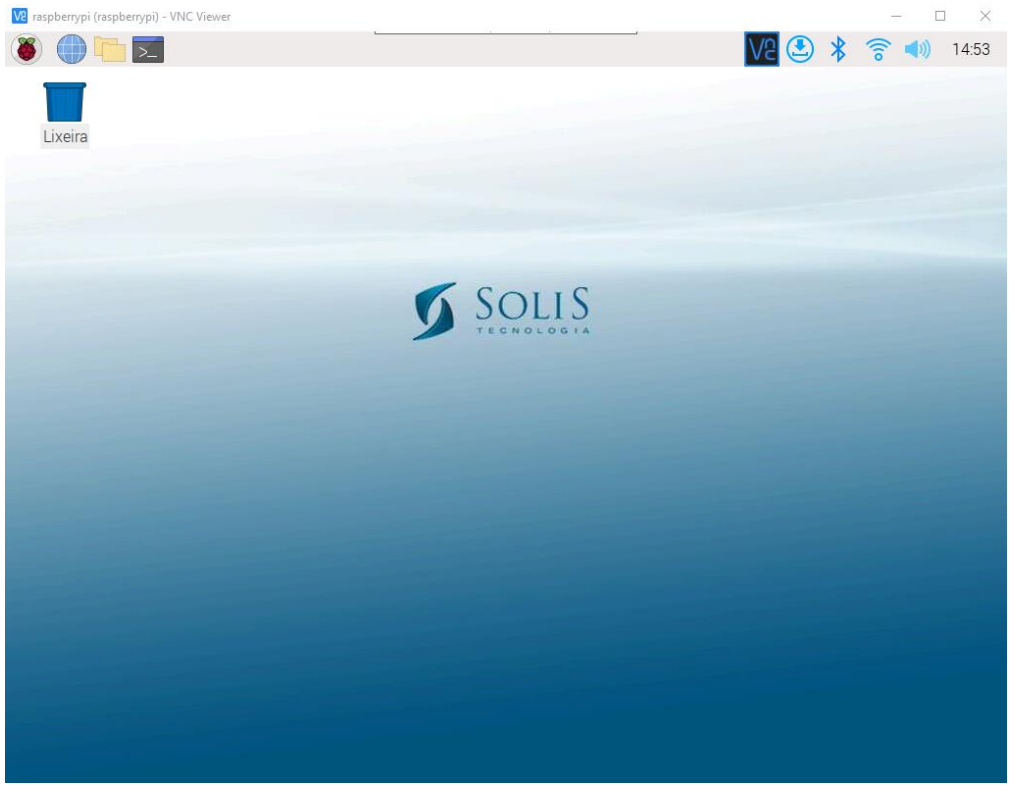
CONEXÃO REMOTA COM A RASPBERRY

Software VNC Viewer



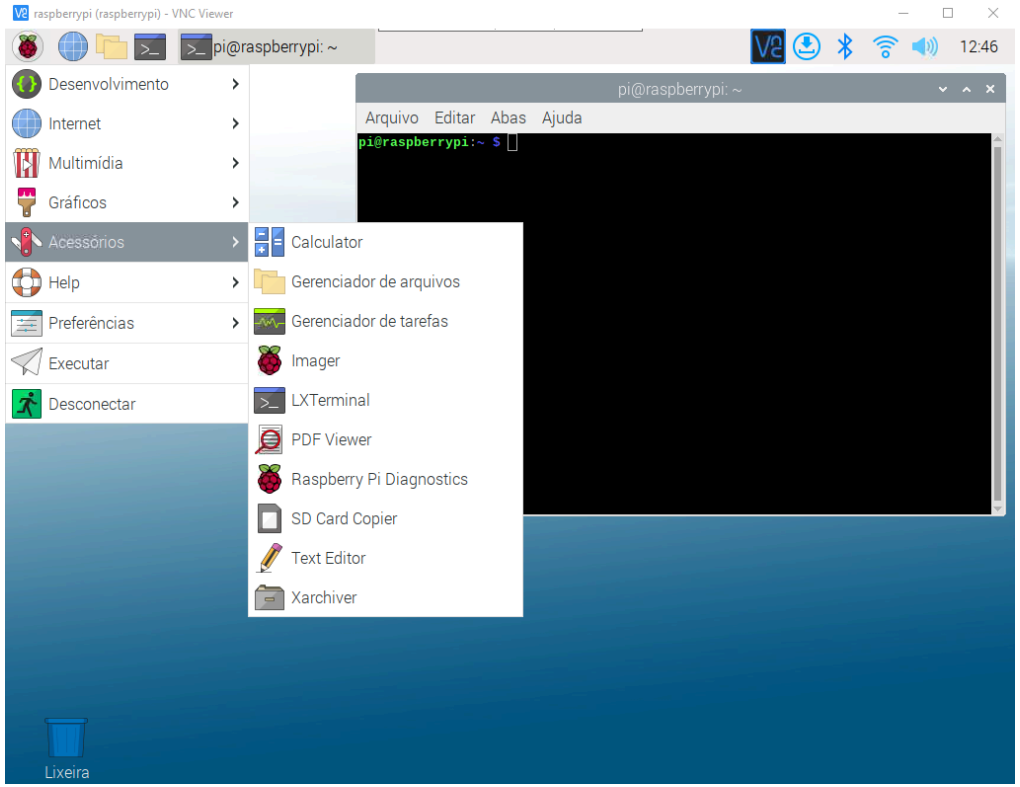
Raspberry Pi 4B





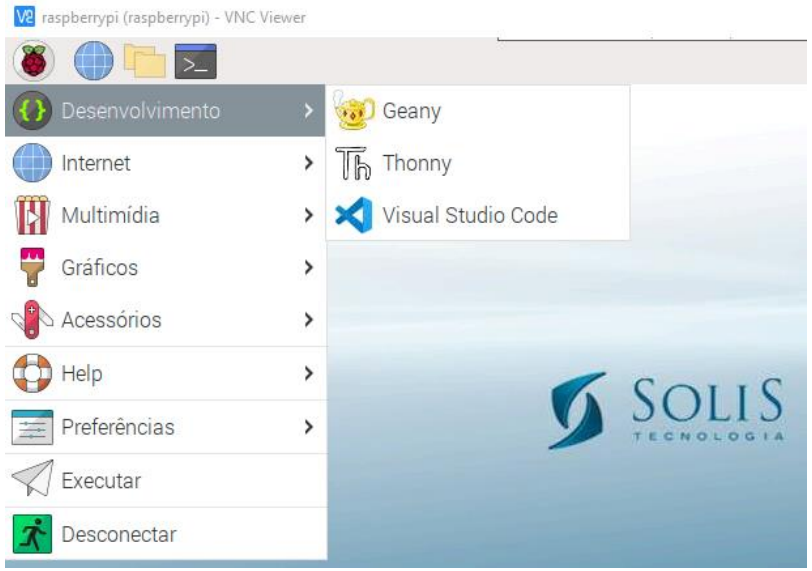
SOLIS
TECNOLOGIA

COMUNICAÇÃO ENTRE A PLACA DE CONTROLE E POTÊNCIA DO SOBOT E A RAPBERRY PI



Terminal de comandos

Visual Studio Code



SOLIS TECNOLOGIA

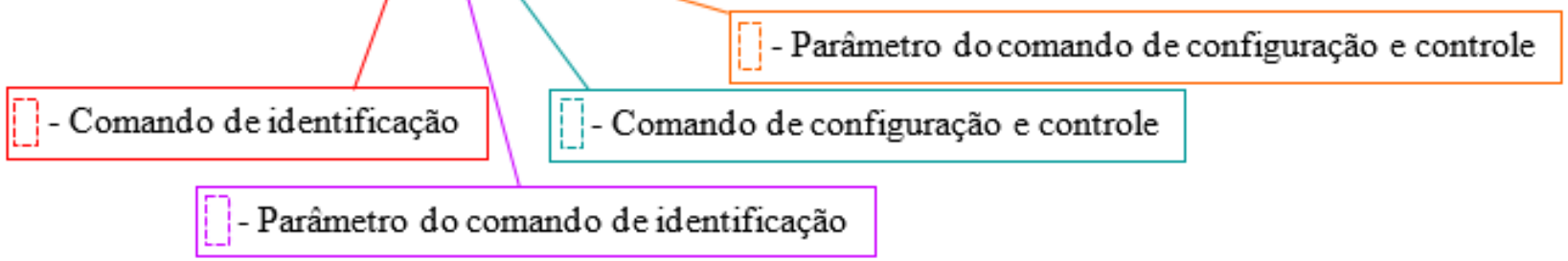
GUIA DE REFERÊNCIA DOS COMANDOS

Lista de comandos

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

ESTRUTURA DO COMANDO

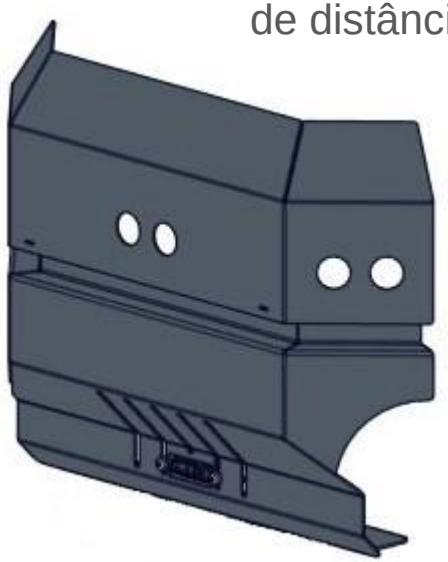
```
1  # Exemplo de movimentação do SoBot para frente
2
3  serial.write(b"MT0 E1")                          # Habilita os motores
4
5  serial.write(b"MT0 D1000 AT5000 DT5000 V10")     # Movimenta o SoBot
6
7  serial.write(b"MT0 E0")                          # Desabilita os motores
8
```



SOLIS TECNOLOGIA

DEMONSTRAÇÃO DO SENSORES SONAR E INFRA VERMELHO DE DISTÂNCIA

Sensor Infra Vermelho de distância



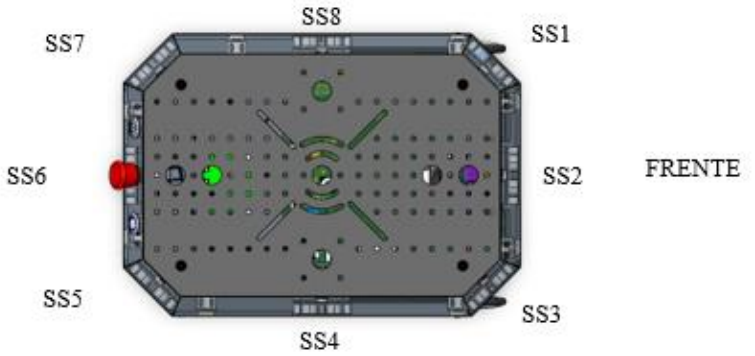
COMANDO		PARÂMETRO	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SI	**SENSOR INFRARED** Comando de identificação utilizado para realizar a leitura do sensor infravermelho de distância.		Sem parâmetro.

```
# Solicitação de status do sensor infravermelho
serial.write(b"SI")
```

Terminal Serial

```
SI 050
```

Sensores Sonar



COMANDO		PARÂMETROS	
CÓDIGO	DESCRIÇÃO	VALOR	DESCRIÇÃO
SS	**SENSOR SONAR** 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

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

Terminal Serial

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

SOLIS TECNOLOGIA

MANUTENÇÃO BÁSICA

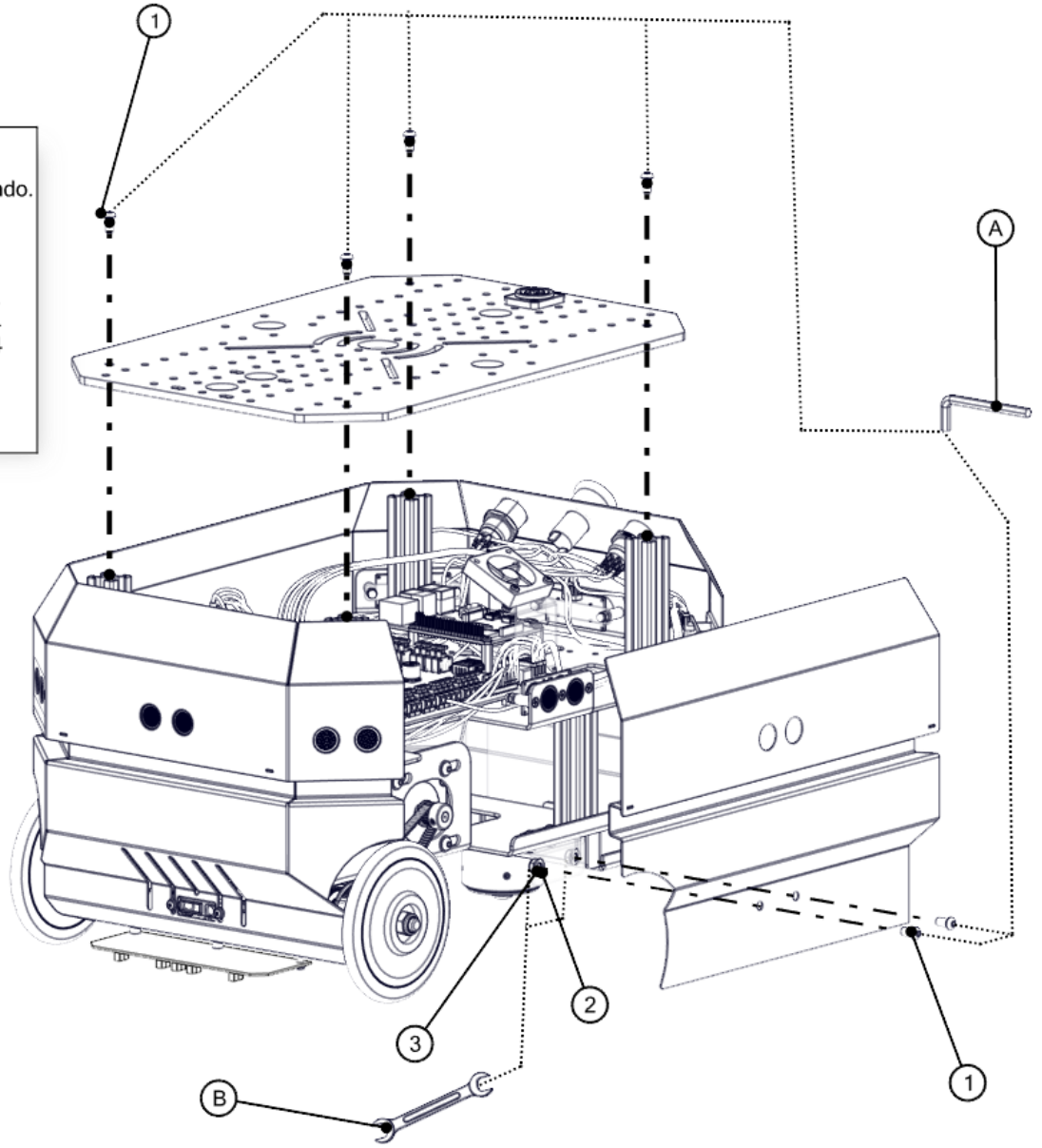
Manutenção Básica

Abertura/Fechamento Tampa Superior:

Considerar os componentes indicados na imagem ao lado.
1- Parafuso Allen Cab. Abaulada M5x10 | 04

Abertura/Fechamento Carenagem Lateral:
1- Parafuso Allen Cab. Abaulada M5x10 | 04
2- Porca Sextavada M5 | 04
3- Arruelas Lisas M5 | 04

Ferramentas necessárias para montagem:
A- 1 Chave Allen Nº3
B- 1 Chave Fixa/Chave Estrela 8mm



SOLIS TECNOLOGIA

MONTAGEM DA PCB SENSORES SEGUIDOR DE FAIXA

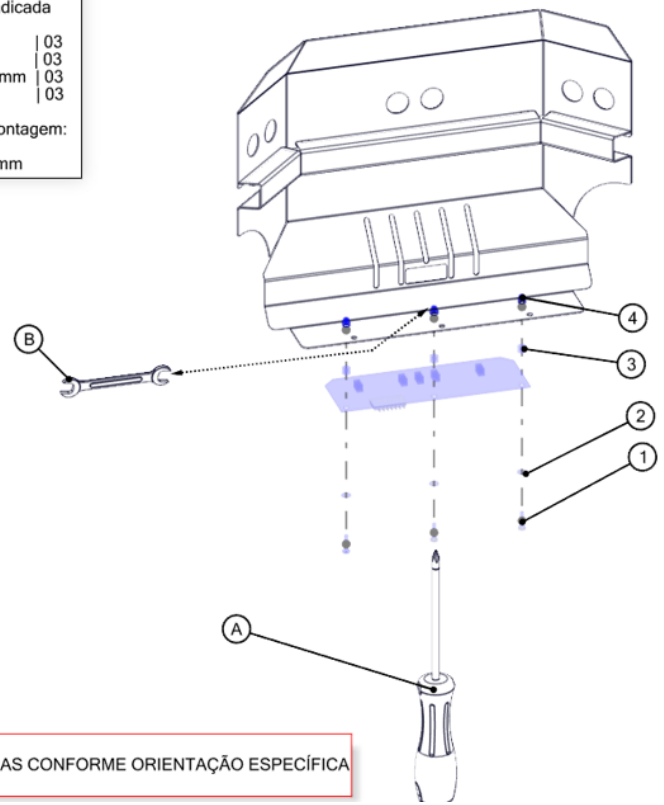
INSTALAÇÃO DA PCI - LEITOR DE LINHA:

Remover a Carenagem Frontal do SoBot

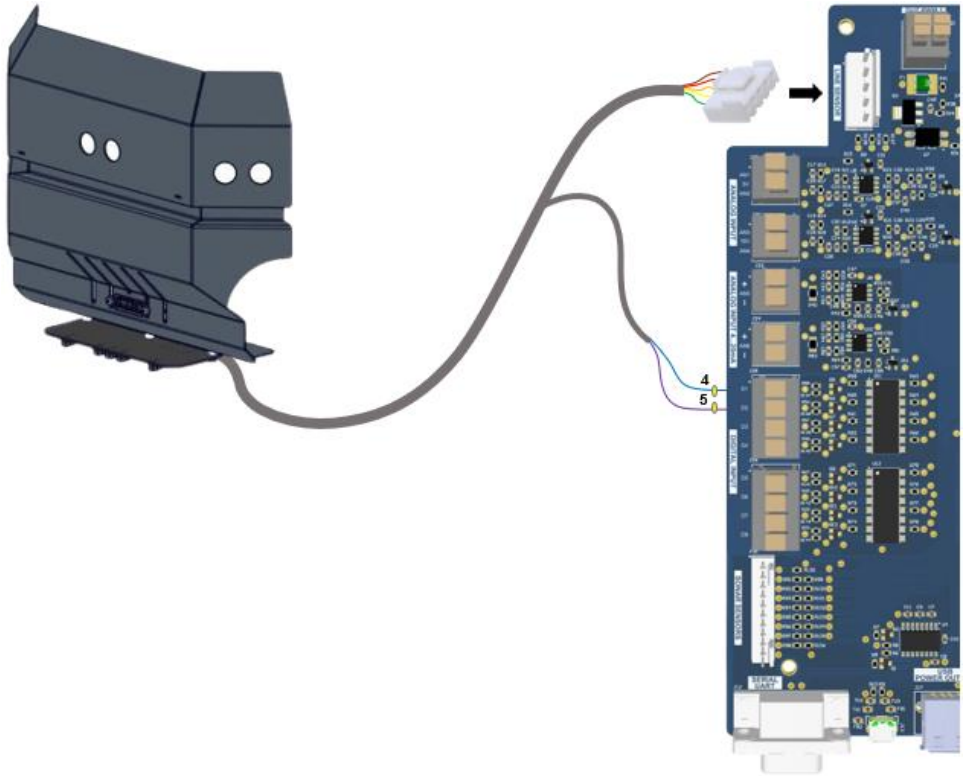
Montar componentes na ordem indicada

1- Parafusos M3x16	03
2- Arruelas de Nylon M3	03
3- Espaçadores de Nylon M3x4,7mm	03
4- Porcas Parlock M3	03

Ferramentas necessárias para montagem:
A- Chave Phillips 1/8 x 3"
B- Chave Fixa/Chave Estrela 5,5mm



IMPORTANTE:
REALIZAR CONEXÕES ELÉTRICAS CONFORME ORIENTAÇÃO ESPECÍFICA
ANTES DE FECHAR ROBÔ.



PROGRAMA DEMO PARA PCB SENSORES SEGUIDOR DE FAIXA

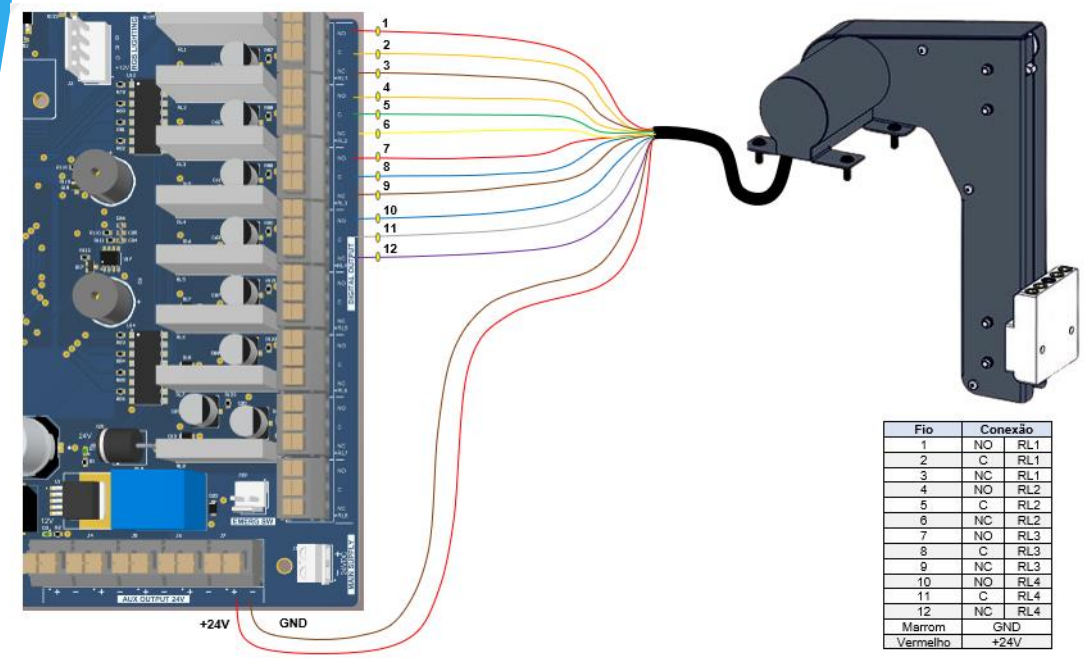
117 lines (100 sloc) | 3.51 KB

```python
#!/usr/bin/python3
"""
Solis Robot - SoBot

Line_Sensor.py: Programming example for the Solis robot to move following a line.

Created By   : Vinicius M. Kawakami
Version      : 1.0

Company: Solis Tecnologia
"""

from time import sleep
import serial

data_line = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

flag_fw = 0
flag_enable = 1
count_bl = 0

black = 49
white = 48

# Set serial port
usb = serial.Serial('/dev/ttyACM0', 57600, timeout=0, dsrdtr=False)
usb.flush()     # Waits data configuration

usb.write(b"LT E1 RD0 GR50 BL0")    # Turn on led tape in green

sleep(2)
usb.write(b"MT0 MC AT100 DT100 V2") # Parameter settings for continuous mode
sleep(1)
usb.write(b"MT0 ME1")               # Enables wheel motors on mode continuous
sleep(1)

while flag_enable == 1:

    usb.write(b"SL")            # Send command to read line sensor
    sleep(0.1)                  # Wait to return datas
    data_line = usb.readline()  # Read data
    print(data_line)

    # Check if sensor 2 is reading black line or if all sensors are reading black
    if(((data_line[4] == white) and (data_line[10] == black) and (data_line[16] == white)) or
     ((data_line[4] == black) and (data_line[10] == black) and (data_line[16] == black))):
        if(flag_fw == 0):
            flag_fw = 1
            usb.write(b"LT E1 RD0 GR50 BL0")
            usb.write(b"MT0 MF")    # Moving to forward

    # Check if sensor 1 and 2 is reading black
    elif((data_line[4] == black) and (data_line[10] == black) and (data_line[16] == white)):
        flag_fw = 0
        count_bl = 0
        usb.write(b"LT E1 RD0 GR0 BL50")
```

Link: https://github.com/SolisTecnologia/SoBot-Line-Follower

SOLIS TECNOLOGIA

MONTAGEM DO ELEVADOR COM PALLET NO SOBOT



Fio	Conexão	
1	NO	RL1
2	C	RL1
3	NC	RL1
4	NO	RL2
5	C	RL2
6	NC	RL2
7	NO	RL3
8	C	RL3
9	NC	RL3
10	NO	RL4
11	C	RL4
12	NC	RL4
Marrom	GND	
Vermelho	+24V	

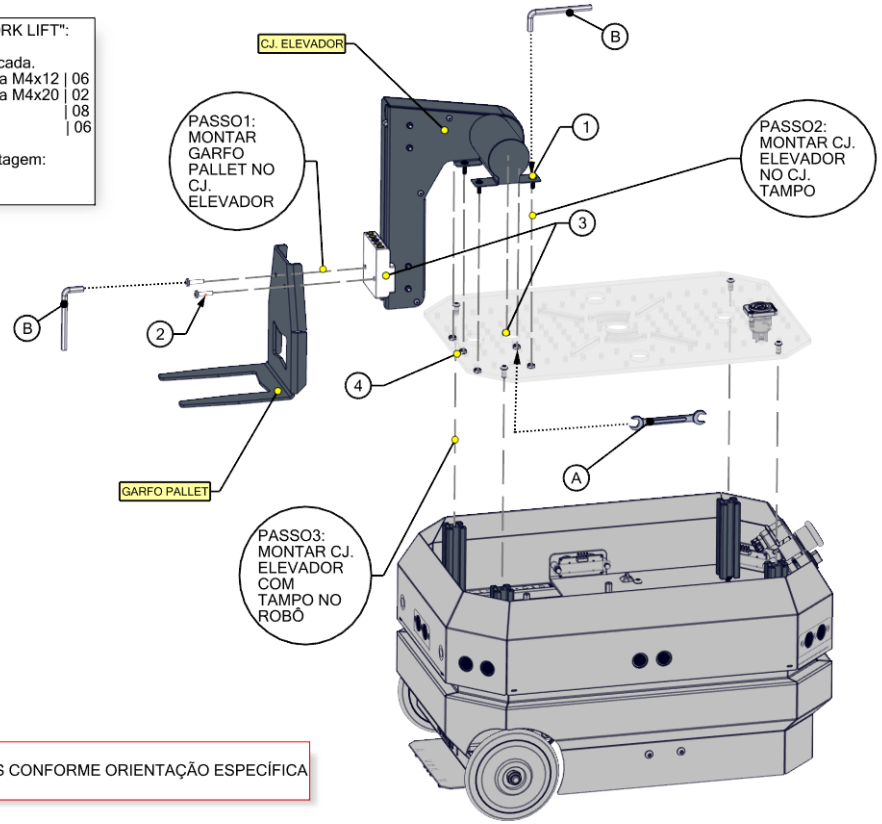
Ligação Elétrica

INSTALAÇÃO DO CONJUNTO "FORK LIFT":

Montar componentes na ordem indicada.
1- Parafusos Allen Cabeça Abaulada M4x12 | 06
2- Parafusos Allen Cabeça Abaulada M4x20 | 02
3- Porcas Sextavadas M4 | 08
4- Arruelas Lisas M4 | 06

Ferramentas necessárias para montagem:
A- Chave Fixa/Chave Estrela 7mm
B- Chave Allen Nº2,5

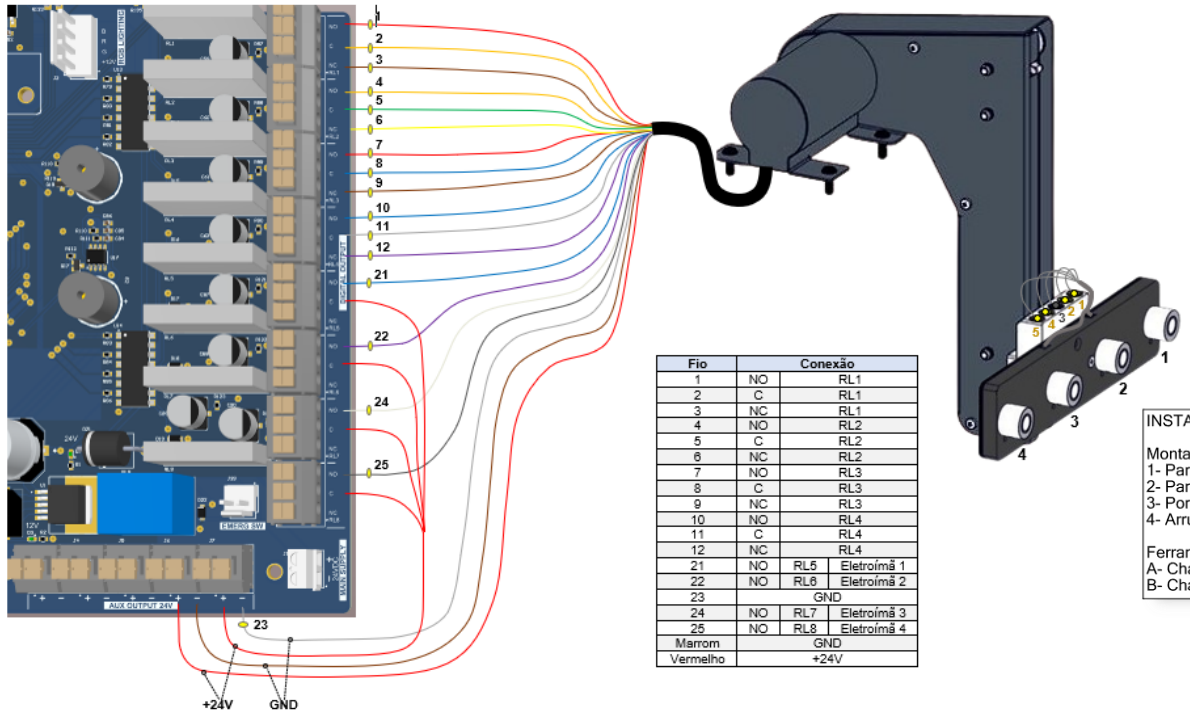
Montagem Mecânica



IMPORTANTE:
REALIZAR CONEXÕES ELÉTRICAS CONFORME ORIENTAÇÃO ESPECÍFICA ANTES DE FECHAR O ROBÔ.

SOLIS TECNOLOGIA

MONTAGEM DO ELEVADOR COM ELETROÍMÃ NO SOBOT



Fio	Conexão		
1	NO	RL1	
2	C	RL1	
3	NC	RL1	
4	NO	RL2	
5	C	RL2	
6	NC	RL2	
7	NO	RL3	
8	C	RL3	
9	NC	RL3	
10	NO	RL4	
11	C	RL4	
12	NC	RL4	
21	NO	RL5	Eletroímã 1
22	NO	RL6	Eletroímã 2
23	GND		
24	NO	RL7	Eletroímã 3
25	NO	RL8	Eletroímã 4
Marrom	GND		
Vermelho	+24V		

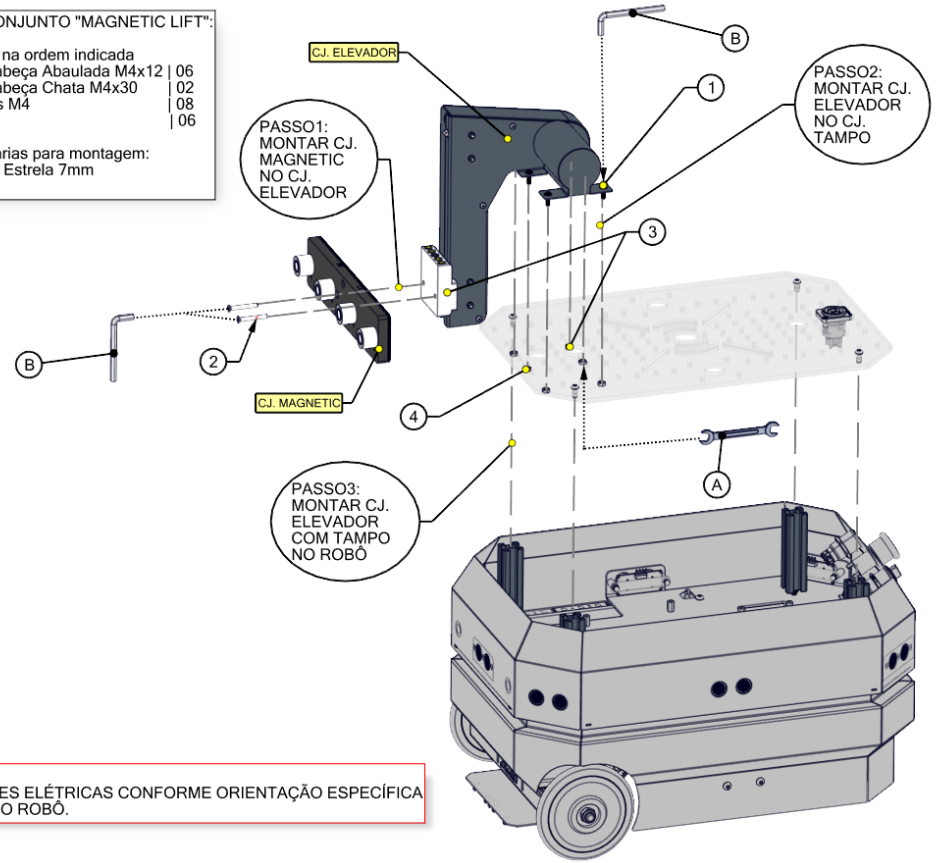
Ligação Elétrica

Montagem Mecânica

INSTALAÇÃO DO CONJUNTO "MAGNETIC LIFT":

Montar componentes na ordem indicada
1- Parafusos Allen Cabeça Abaulada M4x12 | 06
2- Parafusos Allen Cabeça Chata M4x30 | 02
3- Porcas Sextavadas M4 | 08
4- Arruelas Lisas M4 | 06

Ferramentas necessárias para montagem:
A- Chave Fixa/Chave Estrela 7mm
B- Chave Allen Nº2,5



IMPORTANTE:
REALIZAR CONEXÕES ELÉTRICAS CONFORME ORIENTAÇÃO ESPECÍFICA ANTES DE FECHAR O ROBÔ.

SOLIS TECNOLOGIA

PROGRAMA DEMO DO ELEVADOR COM PALLET E ELETROÍMÃ

Demo Elevador com Pallet

Projetos > Demo > Pallet > Controle_Pallet.py > ...

```python
"""

import inputs
import serial

flag_start = 0
flag_BT_RZ = 0
flag_BT_Z = 0

# Find the Logitech F710 controller ID connected to the Raspberry Pi
gamepad = inputs.devices.gamepads[0]
print(gamepad)

# Configure the serial port
usb = serial.Serial('/dev/ttyACM0', 57600, timeout=0, dsrdtr=False)
usb.flush()

# Configure wheel parametres
usb.write(b"WP MT1 WD99,84")
usb.write(b"WP MT2 WD99,54")
usb.write(b"WP DW264,95")

# Set the motion proportional gain
usb.write(b"PG SO2,3 CA3,22 DF5,28 RI-6")

# Configure operating parametres in continuous mode
usb.write(b"MT0 MC MD0 AT800 DT800 V8")

while True:

    events = inputs.get_gamepad()   # Checks if there was any control event

    for event in events:

        # Checks if it is event of type "KEY"
        if event.ev_type == "Key":
            print(f"Evento code: {event.code}")
            print(f"Evento state: {event.state}")

            # Check if the event code is "BTN_START" in state 1
            if event.code == "BTN_START" and event.state == 1:
                print("Botão Start pressionado")
                if flag_start == 0:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

pi@raspberrypi:~/Documentos/Projetos $

Python Python

Link: https://github.com/SolisTecnologia/SoBot-Control-Pallet

Demo Elevador com Eletroímã

Projetos > Demo > Eletroimã > Controle_Eletroima.py > ...

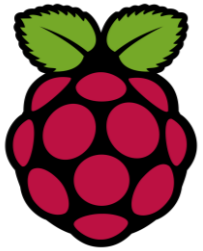
```python

import inputs
import serial

flag_start = 0
flag_BT_RZ = 0
flag_BT_Z = 0
flag_BT_A = 0
flag_BT_B = 0
flag_BT_X = 0
flag_BT_Y = 0

# Find the Logitech F710 controller ID connected to the Raspberry Pi
gamepad = inputs.devices.gamepads[0]
print(gamepad)

# Configure the serial port
usb = serial.Serial('/dev/ttyACM0', 57600, timeout=0, dsrdtr=False)
usb.flush()

# Configure wheel parametres
usb.write(b"WP MT1 WD99,84")
usb.write(b"WP MT2 WD99,54")
usb.write(b"WP DW264,95")

# Set the motion proportional gain
usb.write(b"PG SO2,3 CA3,22 DF5,28 RI-6")

# Configure operating parametres in continuous mode
usb.write(b"MT0 MC MD0 AT800 DT800 V8")

while True:
    events = inputs.get_gamepad()   # Checks if there was any control event
    for event in events:

        # Checks if it is event of type "KEY"
        if event.ev_type == "Key":
            print(f"Evento code: {event.code}")
            print(f"Evento state: {event.state}")

            # Check if the event code is "BTN_START" in state 1
            if event.code == "BTN_START" and event.state == 1:
                print("Botão Start pressionado")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

pi@raspberrypi:~/Documentos/Projetos $

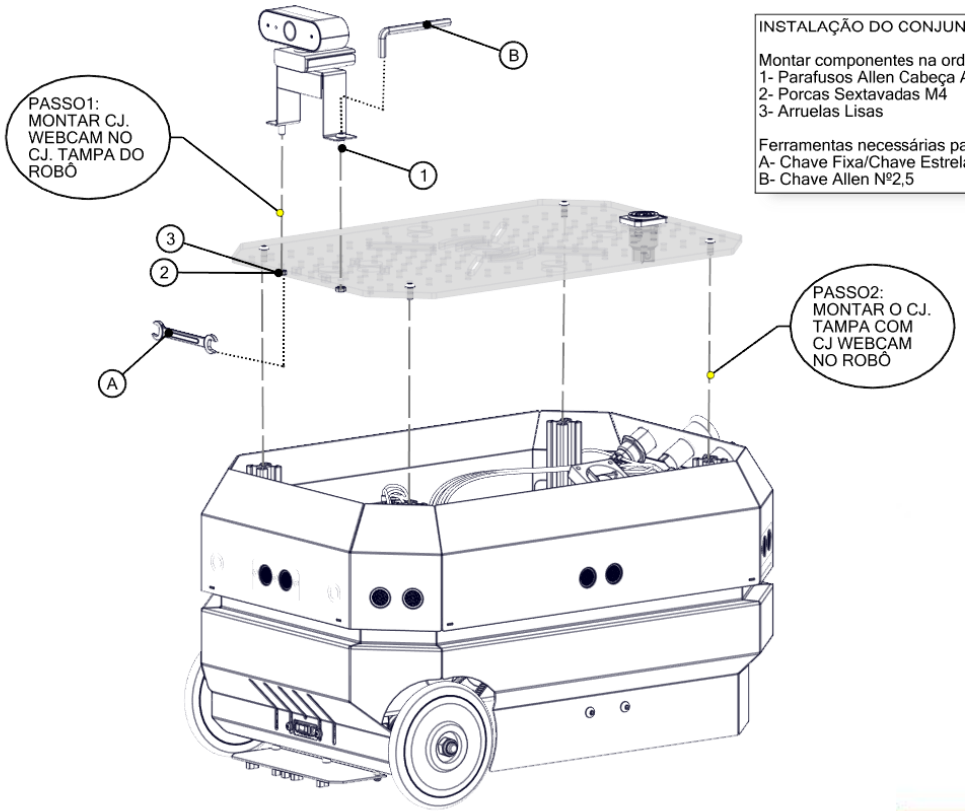
Python Python

Link: https://github.com/SolisTecnologia/SoBot-Control-Electromagnets



SOLIS TECNOLOGIA

MONTAGEM DO MÓDULO AGRO NO SOBOT



INSTALAÇÃO DO CONJUNTO CJ. WEBCAM:

Montar componentes na ordem indicada

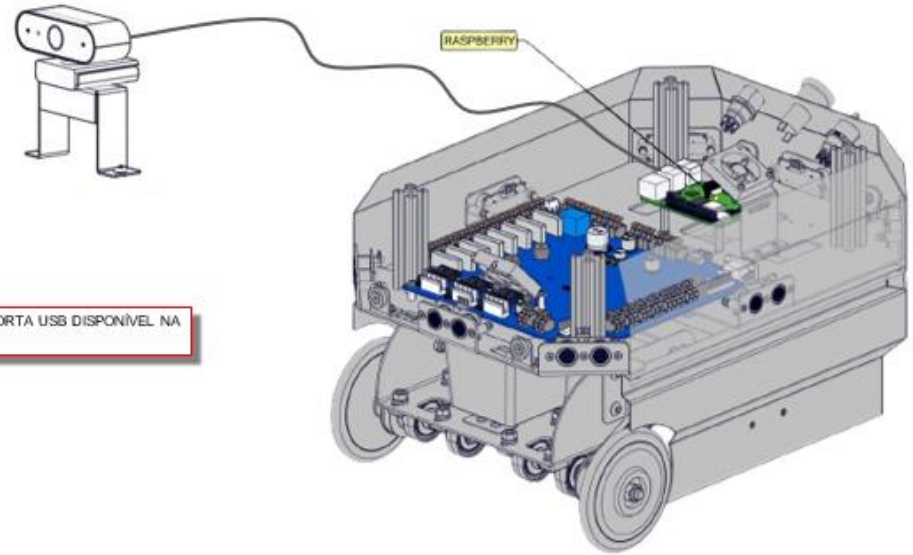
1- Parafusos Allen Cabeça Abaulada M4x12	02
2- Porcas Sextavadas M4	02
3- Arruelas Lisas	02

Ferramentas necessárias para montagem:
A- Chave Fixa/Chave Estrela 7mm
B- Chave Allen Nº2,5

IMPORTANTE:
REALIZAR CONEXÕES ELÉTRICAS CONFORME ORIENTAÇÃO ESPECÍFICA ANTES DE FECHAR ROBÔ.

Montagem Mecânica

Ligação Elétrica



CONECTAR A WEBCAM EM UMA PORTA USB DISPONÍVEL NA RASPBERRY.

SOLIS TECNOLOGIA

MONTAGEM E DESMONTAGEM DO RESERVATÓRIO AGRO

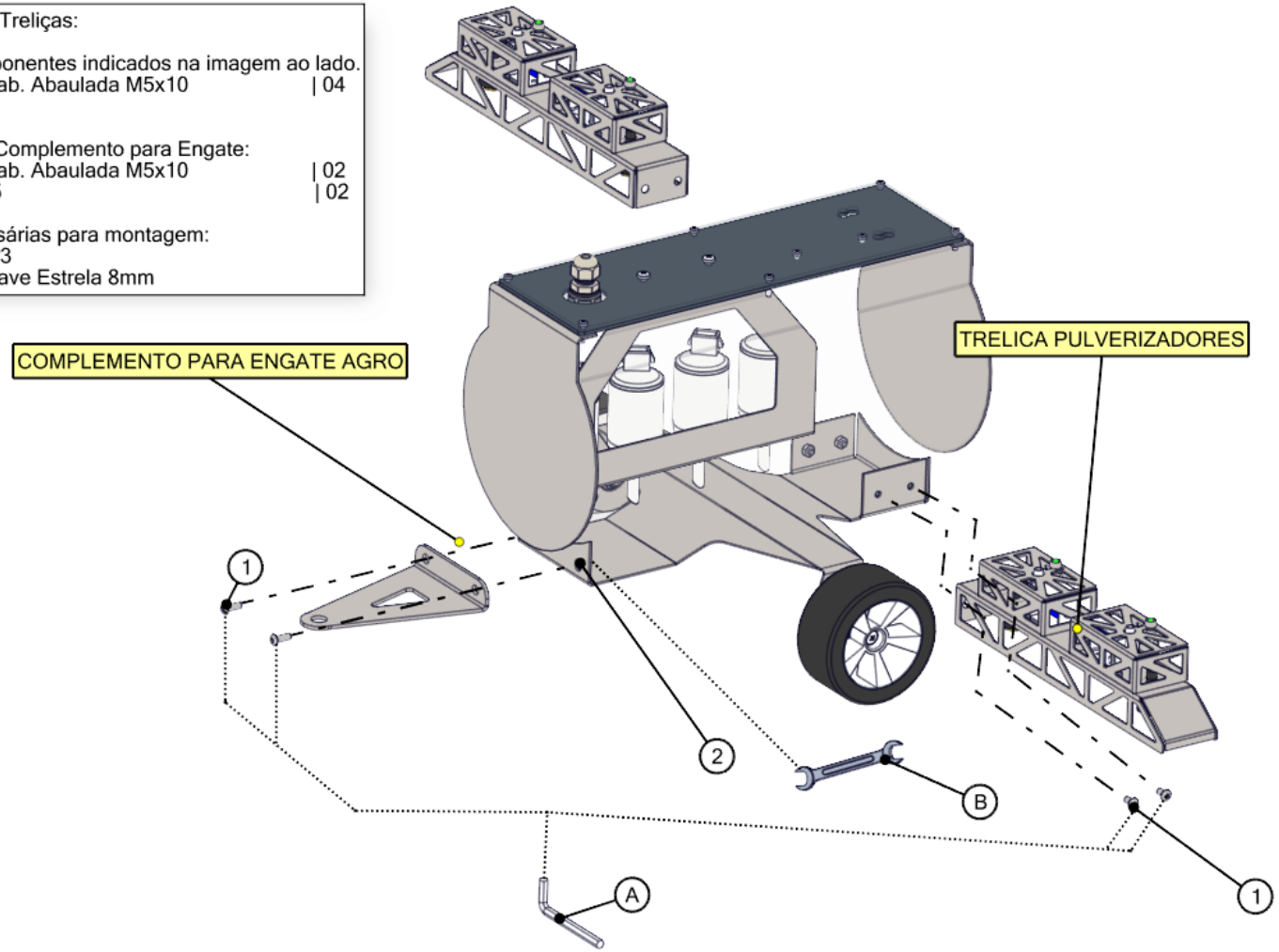
SoBot Agri-Tech
Montagem e Desmontagem para Maleta de Acessórios

Adicionar/Remover Treliças:

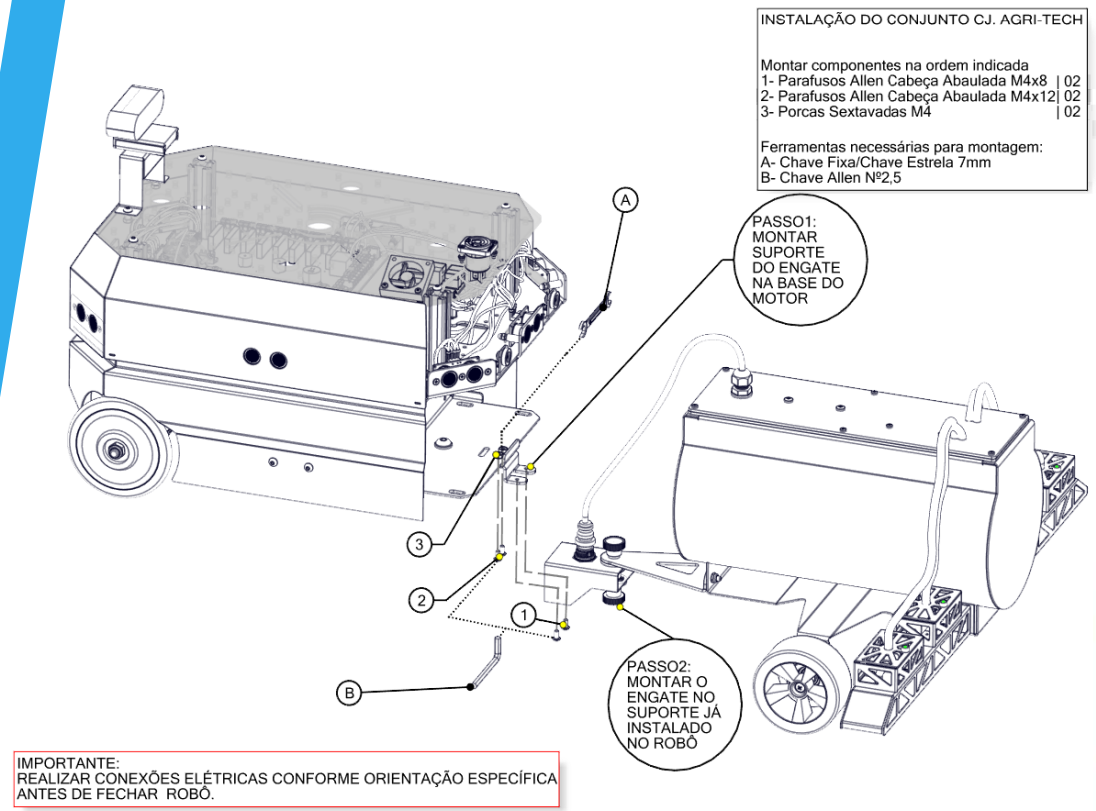
Considerar os componentes indicados na imagem ao lado.
1- Parafuso Allen Cab. Abaulada M5x10 | 04

Adicionar/RemoverComplemento para Engate:
1- Parafuso Allen Cab. Abaulada M5x10 | 02
2- Porca parlock M5 | 02

Ferramentas necessárias para montagem:
A- 1 Chave Allen Nº3
B- 1 Chave Fixa/Chave Estrela 8mm

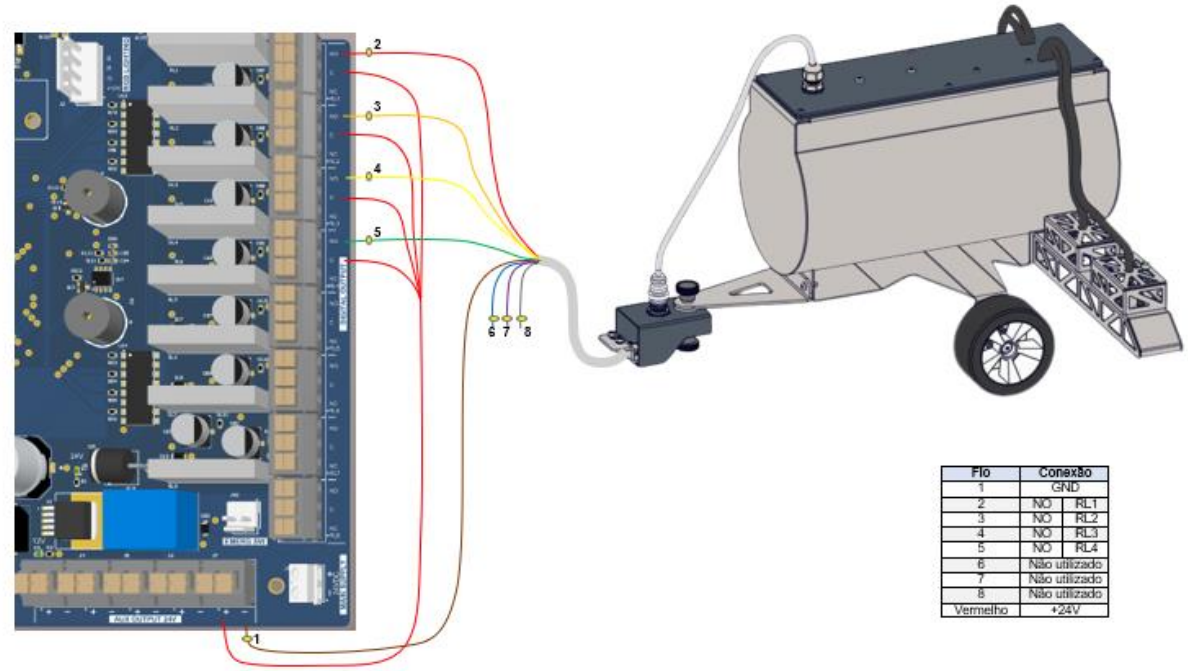


MONTAGEM DO MÓDULO AGRO NO SOBOT



Montagem Mecânica

Ligação Elétrica



Fio	Conexão	
1	GND	
2	NO	RL1
3	NO	RL2
4	NO	RL3
5	NO	RL4
6	Não utilizado	
7	Não utilizado	
8	Não utilizado	
Vermelho	+24V	

SOLIS TECNOLOGIA

PROGRAMA DEMO PARA O MODULO AGRO

Agri-Tech.py

Projetos > Demo > Agri-tech > Agri-Tech.py > ...

```
'''
#################################
|   |   Funçãos Principal
#################################
'''
flag_D1 = 0
flag_D2 = 0
flag_D3 = 0
flag_D4 = 0
tracker = EuclideanDistTracker()

camera = cv2.VideoCapture(0, cv2.CAP_V4L2)

camera.set(3, 640)
camera.set(4, 360)
camera.set(5, 25)  #set frame
camera.set(cv2.CAP_PROP_BRIGHTNESS, 128)
camera.set(cv2.CAP_PROP_CONTRAST, 32)
camera.set(cv2.CAP_PROP_AUTO_WB, 1)
camera.set(cv2.CAP_PROP_AUTO_EXPOSURE, 3)

kernel = np.ones((3, 3), np.uint8)

###### VERDE ESCURO #####
#lower = np.array([70, 100, 100])
#upper = np.array([77, 200, 200])
###### VERDE CLARO #####
#lower = np.array([44, 100, 120])
#upper = np.array([55, 200, 220])
###### VERDE CLARO FUNDO BRANCO #####
#lower = np.array([44, 85, 50])
#upper = np.array([55, 230, 160])
lower = np.array([40, 50, 120])
upper = np.array([80, 100, 255])
##### VERMELHO #####
#lower2 = np.array([0, 85, 150])
#upper2 = np.array([15, 255, 255])
##### VERMELHO FUNDO BRANCO #####
lower2 = np.array([0, 50, 80])
upper2 = np.array([10, 230, 255])
lower1 = np.array([170, 50, 80])
upper1 = np.array([180, 230, 255])
##### ROXO FUNDO BRANCO #####
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
pi@raspberrypi:~/Documentos/Projetos $
```

Python Python

Link: https://github.com/SolisTecnologia/SoBot-Agri-Tech

tracker.py

Projetos > Demo > Agri-tech > tracker.py > EuclideanDistTracker > update

```
#!/usr/bin/python3

import math


class EuclideanDistTracker:
    def __init__(self):
        # Store the center positions of the objects
        self.center_points = {}
        # Keep the count of the IDs
        # each time a new object id detected, the count will increase by one
        self.id_count = 0


    def update(self, objects_rect):
        # Objects boxes and ids
        objects_bbs_ids = []

        # Get center point of new object
        for rect in objects_rect:
            x, y, w, h = rect
            cx = (x + x + w) // 2
            cy = (y + y + h) // 2

            # Find out if that object was detected already
            same_object_detected = False
            for id, pt in self.center_points.items():
                dist = math.hypot(cx - pt[0], cy - pt[1])

                if dist < 25:
                    self.center_points[id] = (cx, cy)
                    #print(self.center_points)
                    objects_bbs_ids.append([x, y, w, h, id])
                    same_object_detected = True
                    break

            # New object is detected we assign the ID to that object
            if same_object_detected is False:
                self.center_points[self.id_count] = (cx, cy)
                objects_bbs_ids.append([x, y, w, h, self.id_count])
                self.id_count += 1

        # Clean the dictionary by center points to remove IDS not used anymc
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
pi@raspberrypi:~/Documentos/Projetos $
```

Python Python

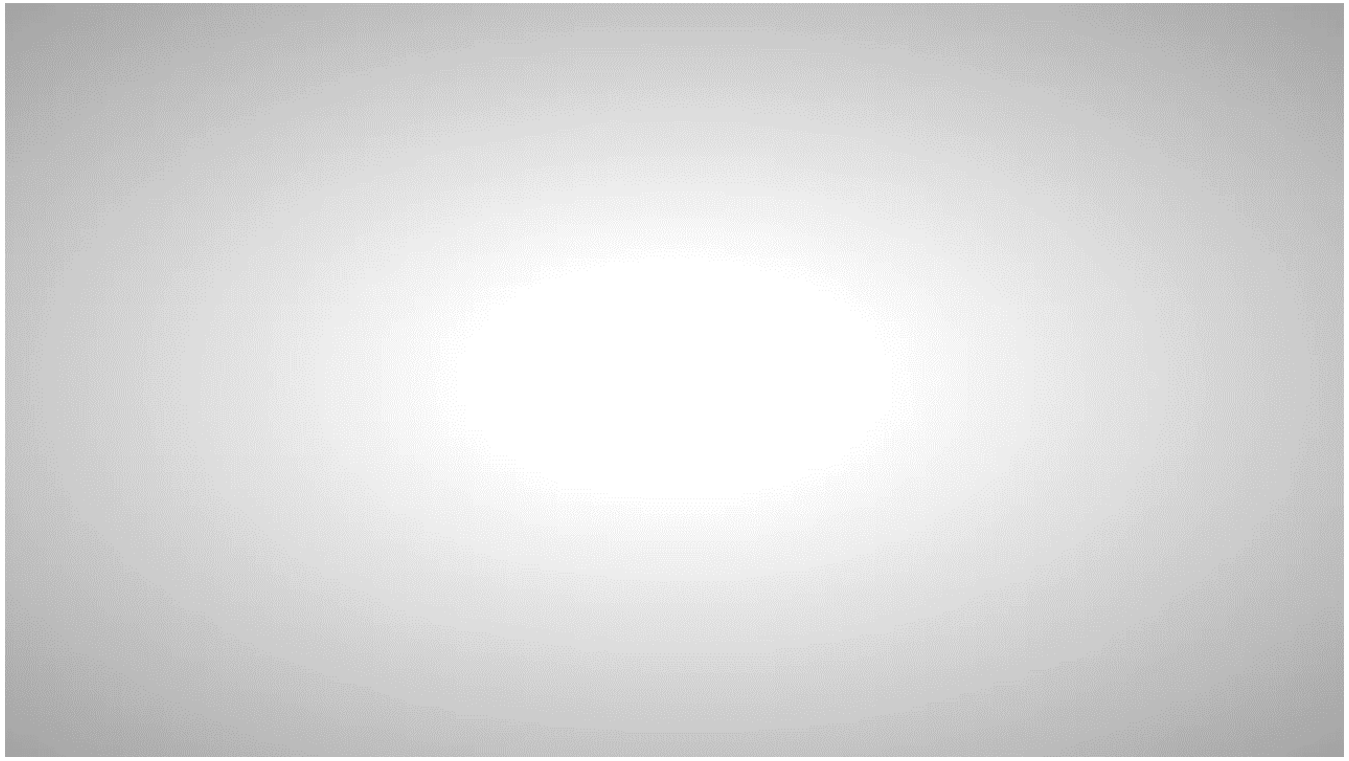
SOLIS TECNOLOGIA

PROJETOS ESPECIAIS / AGROCOMPUTAÇÃO

Crescendo 16% ao ano, este é o setor que mais sofre com falta de profissionais qualificados no país

Em plena expansão, mercado de tecnologia para o campo deve valer mais de US$8 bi em 2026 – e já tem cinco vagas abertas para cada profissional disponível no mercado; veja como se qualificar

Mercado de agricultura digital deve movimentar mais de US$ 8 bi até 2026 (Halfpoint Images/Getty Images)



Robótica Móvel Industrial, Automação de Processos, Mecatrônica, Logistic Control – Warehouse, STEAM

Eletrônica, Mecânica, Desenvolvimento de Software, Inteligência Artificial: Machine Learning, Visão Computacional

SOLIS TECNOLOGIA

CONTATOS

SOLIS
TECNOLOGIA

CONTATO

Solis Tecnologia e Consultoria Empresarial Ltda
Avenida Angélica, 2.627 - Térreo - São Paulo - SP
Fone: +55 (11) 4304-0786
Email: contato@solistecnologia.com.br
www.solistecnologia.com.br

Acesse o QR code para maiores informações.