

ZZ-Gripper v2.0

Robotica Movel Inteligente - 2017.2

Desiree Santos

November 7, 2017

1 Projeto

- Definicao
- Especificacao - V1.0
- Especificacao - V2.0

2 Especificacao

3 Resultados Preliminares

4 Resultados Planejados

5 Plano de Trabalho

Desenvolvido em 2014 no Laboratorio de Sistemas Autonomos(LSA) - FACIN/PUCRS, visando aplicacoes pick-and-place integrada com o robo Turtlebot.



Pick and Place

Desenvolver uma garra inteligente para o Turtlebot para que ele possa carregar objetos do ambiente.

- Microcontrolador: Arduino nano
- Driver: TB6612-Dual Motor Driver
- Atuadores: 2 Graus de liberdade
- Sensores:
 - Sensor de força
 - Sensor ultrasonico
 - Sensor de corrente
 - Chave fim de curso
- Força aplicada a objeto: 500g
- Peso maximo de sustentao: 700g

Desenvolvido protocolo de comunicacao para captar os dados do microcontrolador arduino e publicar no tpico ROS, afim de controlar o robo turtle.

- **Microprocessador: Raspberry Pi**
- Driver: TB6612-Dual Motor Driver
- Fonte de energia 12v 1A
- Atuadores: 2 Graus de liberdade
- Sensores:
 - **Laser**
 - Sensor de força
 - Sensor ultrasonico
 - Sensor de corrente
 - Chave fim de curso
- Força aplicada a objeto: 500g
- Peso maximo de sustentao: 700g

- Estrutura do pacote:
 - Linguagem unificada - Python
 - Modulo para controle do hardware
 - Modulo para manipular imagens
 - Modulo para integraao do ROS
 - Script de instalao para dependencias
 - Modelagem do ambiente

- Hardware:

- 1 Sonar
- 2 Controle da garra(2 servos motores)
- 3 Controle ga garra(motor de passo)
- 4 Sensor de fora com modulo Analogico-Digital
- 5 RaspCam - camera para raspberry
- 6 Diagrama de componentes - fritzing
- 7 Documentao tecnica do projeto eletrico, firmware e do driver

- Software:

- 1 OpenCV
- 2 Raspberry(Zero W — Rasp 3) com ROS kinetic

Resultados Planejados

- Hardware:

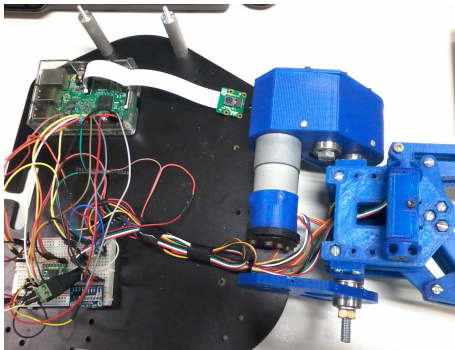
Integrados com a rasp

- 1 Sonar
- 2 Controle da garra(2 servos motores)
- 3 Controle ga garra(motor de passo)
- 4 Sensor de fora com mdulo Analogico-Digital
- 5 RaspCam - camera para raspberry
- 6 **Laser**
- 7 **Encode/Decode**
- 8 **Migrar da protoboard para PCB ***
- 9 **GUI para controlar a garra ***

- Software:

- 1 OpenCV
- 2 Raspberry(Zero W — Rasp 3) com ROS kinetic
- 3 **Node para controle integrado com a camera**
- 4 **Driver ROS: interface baseada em ROS actions**

Resultados Preliminares



Resultados Preliminares

```
1  #!/usr/bin/python
2  import time
3  import RPi.GPIO as GPIO
4  import Adafruit_ADS1x15
5
6  # Set which GPIO pins the drive outputs are connected
7  HAND_LEFT = 13
8  HAND_RIGHT = 16
9  STBY = 22
10 PWMB = 23
11 BIN1 = 24
12 BIN2 = 25
13 TRIG = 20
14 ECHO = 26
15
16 # MOTOR LEFT/RIGHT
17 GPIO.setmode(GPIO.BCM)
18 GPIO.setup(HAND_LEFT, GPIO.OUT)
19 GPIO.setup(HAND_RIGHT, GPIO.OUT)
```

Cronograma

Semana	Tarefa	Encode/Decode	Laser	PCB	ROS	Documentação
6 -nov	Novembro					
13 -nov						
20 -nov						
27 -nov						
4 -dez	Dezembro					