



*Federação das Indústrias do Estado da Bahia*

**CENTRO UNIVERSITÁRIO SENAI CIMATEC**

**Engenharia Elétrica**

**Projeto Theoprax de Conclusão de Curso**

**Desenvolvimento do robô de inspeção.**

Apresentada por: Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

Orientador: Prof. Marco Reis, M.Eng.

Setembro de 2018

Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

## **Desenvolvimento do robô de inspeção.**

Projeto Theoprax de Conclusão de Curso apresentada ao , Curso de Engenharia Elétrica do Centro Universitário SENAI CIMATEC, como requisito parcial para a obtenção do título de **Bacharel em Engenharia.**

Área de conhecimento: Interdisciplinar

Orientador: Prof. Marco Reis, M.Eng.

Salvador  
Centro Universitário SENAI CIMATEC  
2016

Dedico este trabalho a ...

---

## Agradecimentos

---

Salvador, Brasil  
dia de Setembro de 2018

Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

---

## Resumo

---

Escreva aqui o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

**Palavras-chave:** Palavra-chave 1, Palavra-chave 2, Palavra-chave 3, Palavra-chave 4, Palavra-chave 5

---

## Abstract

---

Escreva aqui, em inglês, o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

**Keywords:** Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.1.1	Objetivos Específicos . . . . .	1
1.2	Justificativa . . . . .	1
1.3	Requisitos do cliente . . . . .	2
1.4	Organização do Projeto Theoprax de Conclusão de Curso . . . . .	2
<b>2</b>	<b>Conceito do Sistema</b>	<b>3</b>
2.1	Estudo do estado da arte . . . . .	3
2.2	Descrição do sistema . . . . .	3
2.2.1	Especificação técnica . . . . .	3
2.2.2	Arquitetura geral do sistema . . . . .	3
2.2.3	Arquitetura de software . . . . .	4
2.3	Desdobramento da função qualidade . . . . .	4
2.3.1	Requisitos técnicos . . . . .	4
<b>3</b>	<b>Materiais e Métodos</b>	<b>5</b>
3.1	Especificação dos componentes . . . . .	5
3.1.1	Estrutura analítica do protótipo . . . . .	5
3.1.2	Lista de componentes . . . . .	5
3.2	Diagramas mecânicos . . . . .	5
3.3	Modelo esquemático de alimentação e comunicação . . . . .	5
3.3.1	Diagramas elétricos . . . . .	5
3.3.2	Esquemas eletrônicos . . . . .	6
3.4	Especificação das funcionalidades . . . . .	6
3.4.1	Fluxo das informações . . . . .	6
3.4.2	Motion Planning . . . . .	6
3.4.2.1	Definição da funcionalidade . . . . .	6
3.4.2.2	Dependências . . . . .	6
3.4.2.3	Premissas Necessárias . . . . .	6
3.4.2.4	Descrição da Funcionalidade . . . . .	7
3.4.2.5	Saídas . . . . .	8
3.4.3	Actuation . . . . .	9
3.4.3.1	Definição da funcionalidade . . . . .	9
3.4.3.2	Dependências . . . . .	9
3.4.3.3	Premissas Necessárias . . . . .	9
3.4.3.4	Descrição da Funcionalidade . . . . .	9
3.4.3.5	Saídas . . . . .	10
3.4.4	Power Management . . . . .	11
3.4.4.1	Definição da funcionalidade . . . . .	11
3.4.4.2	Dependências . . . . .	11
3.4.4.3	Premissas Necessárias . . . . .	11
3.4.4.4	Descrição da Funcionalidade . . . . .	11
3.4.4.5	Saídas . . . . .	12
3.4.5	System Integrity Check . . . . .	13

---

3.4.5.1	Definição da funcionalidade . . . . .	13
3.4.5.2	Dependências . . . . .	13
3.4.5.3	Premissas Necessárias . . . . .	13
3.4.5.4	Descrição da Funcionalidade . . . . .	14
3.4.5.5	Saídas . . . . .	15
3.5	Interface do Usuário . . . . .	15
3.6	Simulação do sistema . . . . .	15
<b>4</b>	<b>Resultados</b>	<b>16</b>
4.1	Testes unitários . . . . .	16
4.2	Testes integrados . . . . .	16
4.3	Avaliação da prontidão tecnológica . . . . .	16
4.4	Trabalhos futuros . . . . .	16
<b>5</b>	<b>Conclusão</b>	<b>17</b>
5.1	Considerações finais . . . . .	17
<b>A</b>	<b>QFD</b>	<b>18</b>
<b>B</b>	<b>Diagramas mecânicos</b>	<b>19</b>
<b>C</b>	<b>Diagramas eletro-eletrônicos</b>	<b>20</b>
<b>D</b>	<b>Wireframes</b>	<b>21</b>
<b>E</b>	<b>Logbook</b>	<b>22</b>
	<b>Referências</b>	<b>23</b>



---

## Lista de Tabelas

---

---

## Lista de Figuras

---

3.1	Fluxograma de funcionamento da funcionalidade de Motion Planning . . .	8
3.2	Fluxograma da funcionalidade Actuation . . . . .	10
3.3	Fluxograma de funcionamento da funcionalidade de Power Management . .	12
3.4	Fluxograma da rotina para checagem do sistema . . . . .	14

---

## Lista de Siglas

---

THEOPRAX

WWW ..... World Wide Web

---

## Lista de Simbolos

---

$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble
$\partial$	.....	Bla bla bla
$\Pi$	.....	ble ble ble

---

## Introdução

---

O mundo é - e sempre foi - um mundo de rede. Todavia apenas nas últimas duas décadas a teoria de redes tornou-se um tópico que atraiu atenção de pesquisadores e da mídia (refletida nos trabalhos de (BARABÁSI, 2003), (WATTS, 2003), (NEWMAN; WATTS, 2006)), especialmente em relação às redes sociais: os relacionamentos entre os terroristas do 11/9, a forma como a SARS se espalhou em 2002/03 e o mito dos "6 graus de separação" entre dois indivíduos. Até mesmo a forma como a obesidade se espalha pode ser explicada através da análise de redes. O aumento da popularidade dos sites de rede social como Facebook, Google+ ou LinkedIn (ou a Plataforma Lattes brasileira) aumenta a nossa percepção de rede formada por nossos amigos, colegas e família e isso constitui a base invisível de nossa vida social.

### **1.1    *Objetivos***

Nesta seção os objetivos principal (também pode-se utilizar a palavra meta) da monografia de graduação ou especialização, dissertação de mestrado ou tese de doutorado são apresentados.

#### *1.1.1    Objetivos Específicos*

Nesta seção os objetivos específicos (também pode-se utilizar a palavra meta) da monografia de graduação ou especialização, dissertação de mestrado ou tese de doutorado são apresentados.

### **1.2    *Justificativa***

O pesquisador/estudante deve apresentar os aspectos mais relevantes da pesquisa ressaltando os impactos (e.g. científico, tecnológico, econômico, social e ambiental) que a pesquisa causará. Deve-se ter cuidado com a ingenuidade no momento em que os argumentos forem apresentados.

### ***1.3 Requisitos do cliente***

asjdfkjasjdlfjsdlk;f

### ***1.4 Organização do Projeto Theoprax de Conclusão de Curso***

Este documento apresenta  $x$  capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como este projeto theoprax de conclusão de curso está estruturado;
- **Capítulo 2 - Nome do capítulo:** XXX;
- **Capítulo 5 - Conclusão:** Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

---

## Conceito do Sistema

---

Quanto maior for a rapidez de transformação de uma sociedade, mais temporárias são as necessidades individuais. Essas flutuações tornam ainda mais acelerado o senso de turbilhão da sociedade.

(Alvin Toffler)

Quanto maior for a rapidez de transformação de uma sociedade, mais temporárias são as necessidades individuais. Essas flutuações tornam ainda mais acelerado o senso de turbilhão da sociedade.

(Alvin Toffler)

### ***2.1 Estudo do estado da arte***

flkjaskldkfjaskldkfjs

### ***2.2 Descrição do sistema***

lasdjflsadjf

#### ***2.2.1 Especificação técnica***

lakjfldksjfdslakjf

#### ***2.2.2 Arquitetura geral do sistema***

lksajdfldksdajflk;

### 2.2.3 *Arquitetura de software*

## **2.3 *Desdobramento da função qualidade***

asdfsda<sup>sf</sup>

### 2.3.1 *Requisitos técnicos*

asdfsad<sup>fsd</sup>



---

## Materiais e Métodos

---

asdfasdfsdf

### **3.1 Especificação dos componentes**

asjdfkldjsaf

#### *3.1.1 Estrutura analítica do protótipo*

askjfsdalkjf

#### *3.1.2 Lista de componentes*

asfkjdsahfkjs

### **3.2 Diagramas mecânicos**

asdfsdaf

### **3.3 Modelo esquemático de alimentação e comunicação**

asdfadsfsdfs

#### *3.3.1 Diagramas elétricos*

asdfsdaf

### 3.3.2 Esquemas eletrônicos

asdfsda

## 3.4 Especificação das funcionalidades

asdfsdfsdfs

### 3.4.1 Fluxo das informações

asdfsaf

### 3.4.2 Motion Planning

#### 3.4.2.1 Definição da funcionalidade

A funcionalidade de *Motion Planning* é responsável por realizar o planejamento da trajetória do Robô, utilizando o software *MoveIt!* que realiza o cálculo da cinemática inversa para encontrar a melhor forma de ultrapassar os obstáculos.

#### 3.4.2.2 Dependências

O software *moveit* pode utilizar o modelo matemático da cinemática inversa do robô ou um arquivo do tipo URDF. O nome URDF é uma sigla para *Unified Robot Description Format*, esse arquivo é uma especificação em XML utilizada para descrever robôs. Modelos em URDF apresentam uma simplicidade na descrição do robô, e para o caso do Robô *Elir*, utilizar o modelo URDF possibilitará uma aproximação fiel ao modelo real do robô, assim para o cálculo da cinemática inversa será utilizado o seu modelo URDF e não o seu modelo matemático.

#### 3.4.2.3 Premissas Necessárias

Para o correto funcionamento dessa funcionalidade as seguintes premissas são necessárias:

- A configuração dos limites de giro das juntas do robô estarão compatíveis com os comandos enviados
- O modelo URDF do robô estará adequado com o modelo físico
- O pacote gerado pelo *MoveIt! Setup Assistant* estará configurado adequadamente

#### 3.4.2.4 Descrição da Funcionalidade

A movimentação do robô na linha acontecerá por movimentos de translação e transposição de obstáculos. A translação na linha será feita por controladores de torque nas rodas do robô, enquanto a transposição dos obstáculos utilizará o moveit. Por meio da ferramenta *MoveIt! Setup Assistant*, se utiliza o modelo do robô para criar um pacote do ROS com os principais arquivos pelo moveit. A configuração correta do moveit possibilita que se utilizem as funções da sua biblioteca para o cálculo da trajetória, levando em consideração também obstáculos no caminho.

O moveit fornece uma *user interface* que recebe o end-effector, a nomenclatura atribuída ao node feito em python que recebe o *end-effector* é `moveit_commander`. O *node* responsável por fazer a integração da user interface com os parâmetros recebidos pelo *ROS Parameter Server* com o *end-effector* para fazer os cálculos é denominado `move_group`. O *node* `move_group` também pode receber parâmetros como leituras dos sensores do robô e nuvens de pontos.

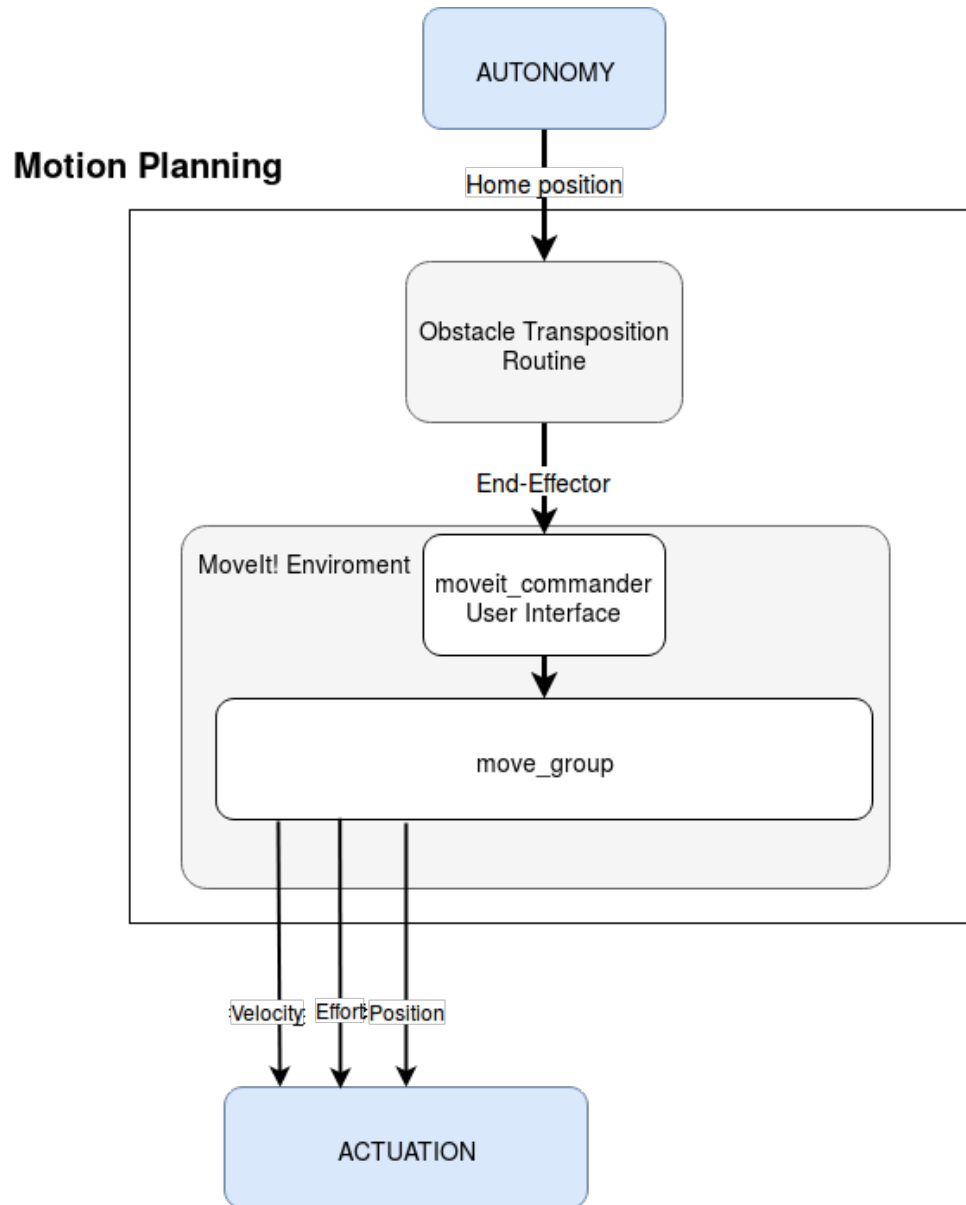


Figura 3.1: Fluxograma de funcionamento da funcionalidade de Motion Planning

Fonte: Própria

#### 3.4.2.5 Saídas

Por meio da compatibilização do *MoveIt!* com o *ROS*, a saída dessa funcionalidade são os comandos de velocidade, esforço e posição para cada junta do robô.

### 3.4.3 Actuation

#### 3.4.3.1 Definição da funcionalidade

A funcionalidade de Actuation tem como objetivo mover a estrutura física do robô, possibilitando o controle dos movimentos das juntas, garras e unidades de tração.

#### 3.4.3.2 Dependências

Essa funcionalidade depende das funcionalidades de *Power Management* e *Motion Planning*. O *Power Management* será responsável por fazer alimentação dos motores, possibilitando controlar a corrente máxima fornecida para cada grupo. A dependência em relação à funcionalidade de *Motion Planning* está atrelada principalmente com o software *MoveIt!*, que ao receber um *end-effector*, realiza o cálculo de trajetória e envia os comandos de velocidade, esforço e posição para os controladores das juntas, garras e unidades de tração.

#### 3.4.3.3 Premissas Necessárias

Para o correto funcionamento desse módulo, devem ser consideradas as seguintes premissas:

- Os motores devem estar configurados de acordo com o padrão de ID determinado pela equipe, fazendo parte da mesma malha de controle;
- Os controladores das juntas, garras e unidades devem estar configurados de acordo com os comandos que serão recebidos pelo *MoveIt!*;
- Os 3 grupos de motores estarão em malhas de alimentação de 12V individuais.

#### 3.4.3.4 Descrição da Funcionalidade

O ROS disponibiliza uma série de drivers para compatibilização dos motores dynamixel, possibilitando a criação de controladores específicos no seu ambiente. Serão criados os controladores referentes as juntas e unidades de tração do robô. Os controladores receberão comandos de *velocity* e *position* do *MoveIt!* junto com os comandos para movimentar o

robô na linha. Após os comandos serem recebidos pelos controladores, eles serão enviados para o *hardware* do robô, de acordo do padrão de comunicação dos motores, por meio de comunicação serial.

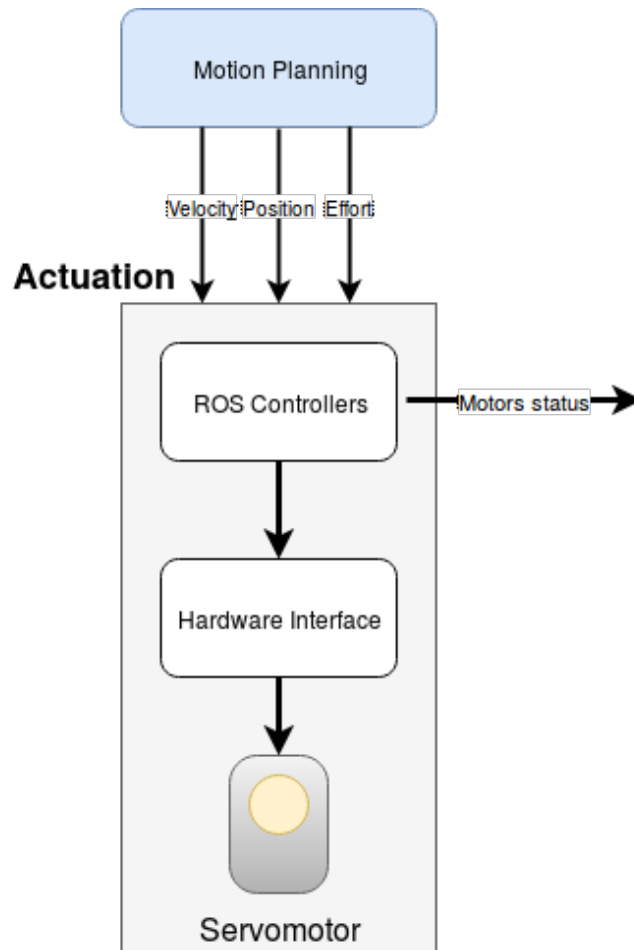


Figura 3.2: Fluxograma da funcionalidade Actuation

Fonte: Própria

#### 3.4.3.5 Saídas

A saída desta funcionalidade é o movimento da estrutura física do robô, que estará de acordo com o planejamento de trajetória do *MoveIt!* e com as instruções para operação na linha

### 3.4.4 *Power Management*

#### 3.4.4.1 *Definição da funcionalidade*

A funcionalidade de *Power Management* é responsável por administrar o fornecimento de energia para os dispositivos eletrônicos do robô, nos níveis adequados de tensão e corrente.

#### 3.4.4.2 *Dependências*

Essa funcionalidade depende da comunicação serial por meio da biblioteca *rosserial* e da operacionalização do firmware embarcado no hardware (placa) de acordo com as necessidades do projeto.

#### 3.4.4.3 *Premissas Necessárias*

Para o correto funcionamento desse módulo de *Power Management*, devem ser consideradas as seguintes premissas:

- A placa multiplexadora estará conectada diretamente ao módulo de *Power Management*
- Todos os dispositivos estarão conectados nas suas respectivas entradas
- A placa deverá ser alimentada por 2 baterias
- A placa estará conectada diretamente na NUC, por meio de uma USB

#### 3.4.4.4 *Descrição da Funcionalidade*

A placa de *Power Management* fornece diversos recursos para integração com o ROS. Seu firmware, além de realizar as medições e controle dos níveis de tensão e corrente para alimentação do robô, estará adaptado com as seguintes funcionalidades para que haja integração do hardware com o ROS:

- *Publishers* que contém os status das portas em níveis de tensão e corrente; avisos de surtos de corrente ou sobre-corrente; disponibilidade do hardware de *Power Management*

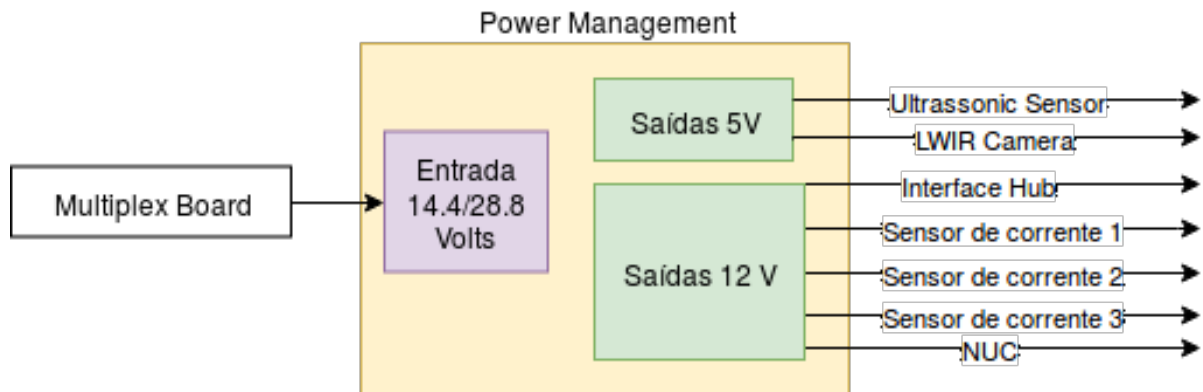


Figura 3.3: Fluxograma de funcionamento da funcionalidade de Power Management

Fonte: Própria

- *Serviços* para realizar a verificação dos níveis de corrente; definição dos limites de corrente nas portas; realização de comandos on-off

O conjunto de baterias fornecerá a energia para o sistema, a placa de *Power Management* irá administrar a distribuição da energia para os seguintes componentes:

- Grupos de servo motores
- Grupo de sensores de corrente
- NUC
- Interface HUB
- Câmera LWIR
- Sensor ultrassônico

#### 3.4.4.5 Saídas

A funcionalidade irá disponibilizar a energia para o robô e as seguintes estruturas no ambiente ROS:

- Tópicos com informações de tensão e corrente nas portas
- Tópico para aviso de sobre-corrente
- Tópico para informar disponibilidade da placa
- Serviços para ler e configurar limite de corrente das portas
- Serviço para ligar ou desligar energia em uma porta



### 3.4.5 *System Integrity Check*

#### 3.4.5.1 *Definição da funcionalidade*

É a funcionalidade responsável por checar a integridade do sistema antes do início da missão, verificando os subsistemas e suas variáveis.

#### 3.4.5.2 *Dependências*

A funcionalidade receberá informações dos seguintes componentes

- Sensor de Temperatura
- Servomotores
- Câmera IR
- Câmera Stéreo
- IMU
- Sensor de Proximidade
- Placa de Power Management
- Sonar
- Baterias

Todas as informações serão enviadas por meio do ambiente ROS, na forma de *Services* ou *Publishers*.

#### 3.4.5.3 *Premissas Necessárias*

As premissas necessárias para o funcionamento dessa funcionalidade são:

- Os subsistemas do robô irão disponibilizar o seu status no ambiente ROS por meio de tópicos ou serviços
- A checagem fará parte do planejamento de missão

#### 3.4.5.4 Descrição da Funcionalidade

A checagem da integridade do sistema é uma funcionalidade essencial para garantir o sucesso da missão e preservar a integridade do robô. O ROS facilita essa comunicação entre os subsistemas, possibilitando que seja criada uma rotina de checagem antes de cada missão.

Será disponibilizado no sistema uma rotina para iniciar a missão. Ao receber o comando para início de missão, os sistemas serão checados sequencialmente, utilizando estrutura de *Services* e *Publishers* do ROS. Caso algum sistema apresente falha, a missão não se iniciará e o erro será mostrado no *terminal* e registrado no arquivo de *log*. Se todos os sistemas estiverem em funcionamento, se iniciará a missão. O fluxograma da funcionalidade está ilustrado na figura 3.4.

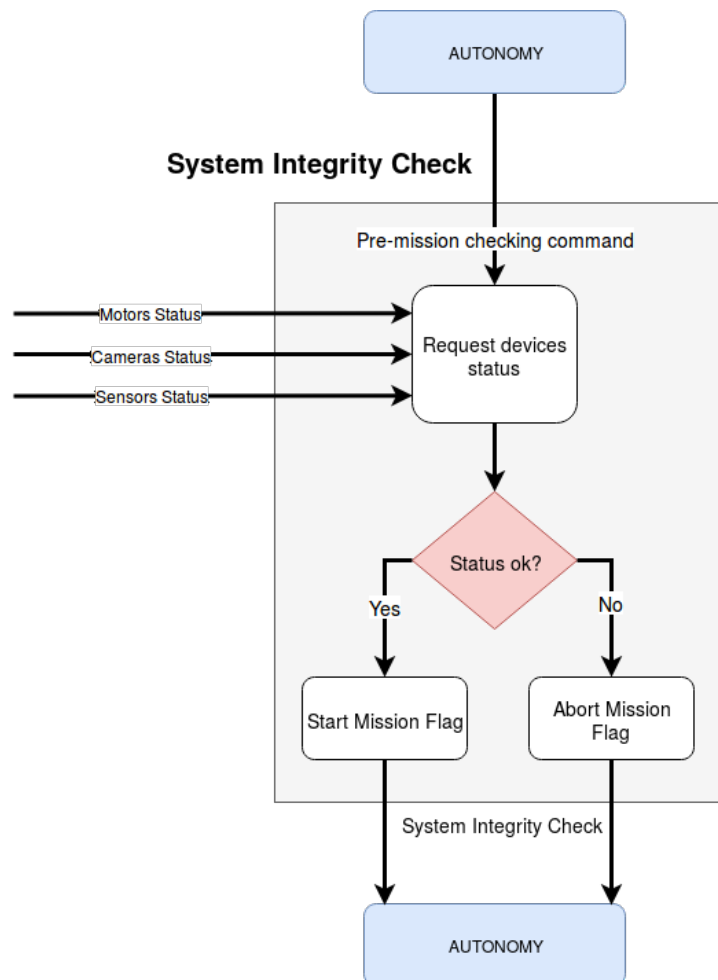


Figura 3.4: Fluxograma da rotina para checagem do sistema

Fonte: Própria

#### *3.4.5.5 Saídas*

No início da rotina de inspeção, a funcionalidade será responsável por enviar o sinal inicia a missão. Caso todos os sistemas checados estejam funcionando, a inspeção ocorrerá normalmente, se algum sistema apresentar defeitos, o defeito será mostrado no *terminal*, registrado em log e a missão será abortada.

### **3.5 Interface do Usuário**

asdfadsfsdfs

### **3.6 Simulação do sistema**

asdfadsfsdfs

---

## Resultados

---

asdfsdfsdf

### ***4.1 Testes unitários***

asdfadsfsdfs

### ***4.2 Testes integrados***

asdfadsfsdfs

### ***4.3 Avaliação da prontidão tecnológica***

asdfadsfsdfs

### ***4.4 Trabalhos futuros***

asdfadsfsdfs

---

## Conclusão

---

Chegou a hora de apresentar o apanhado geral sobre o trabalho de pesquisa feito, no qual são sintetizadas uma série de reflexões sobre a metodologia usada, sobre os achados e resultados obtidos, sobre a confirmação ou rechaço da hipótese estabelecida e sobre outros aspectos da pesquisa que são importantes para validar o trabalho. Recomenda-se não citar outros autores, pois a conclusão é do pesquisador. Porém, caso necessário, convém citá-lo(s) nesta parte e não na seção seguinte chamada **Conclusões**.

### **5.1** *Considerações finais*

Brevemente comentada no texto acima, nesta seção o pesquisador (i.e. autor principal do trabalho científico) deve apresentar sua opinião com respeito à pesquisa e suas implicações. Descrever os impactos (i.e. tecnológicos, sociais, econômicos, culturais, ambientais, políticos, etc.) que a pesquisa causa. Não se recomenda citar outros autores.

**QFD**

---

---

## Diagramas mecânicos

---

---

## Diagramas eletro-eletrônicos

---



## Wireframes

---

---

## Logbook

---

---

## Referências Bibliográficas

---

BARABÁSI, A. L. *Linked: A Nova Ciência dos Networks*. São Paulo: Leopardo Editora, 2003. [1](#)

NEWMAN, A.-L. B. M.; WATTS, D. J. *The Structure and Dynamics of Networks*. Princeton, NJ, USA: Princeton University Press, 2006. [1](#)

WATTS, D. J. *Six Degrees: The Science of a Connected Age*. New York: W W Norton & Co., 2003. [1](#)

*Desenvolvimento do robô de inspeção.*

Michael Faraday

John Nash

James Clerk Maxwell

Nikola Tesla

Salvador, Setembro de 2018.