

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

**STÉFANI ANTONELLI GOERCK DE FREITAS**

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE EMBARCADO PARA A  
EQUIPE DE FUTEBOL DE ROBÔS ARARANGUÁ INTRUDERS**

**Araranguá, 22 de fevereiro de 2013**

STÉFANI ANTONELLI GOERCK DE FREITAS

DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE EMBARCADO PARA A EQUIPE DE FUTEBOL  
DE ROBÔS ARARANGUÁ INTRUDERS

Trabalho de Curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação. Sob a orientação do Professor Anderson Luiz Fernandes Perez.

**Araranguá, 2013**

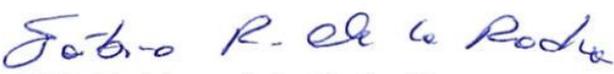
**Stéfani Antonelli Goerck de Freitas**

**Desenvolvimento de um Sistema de Controle Embarcado para a Equipe de Futebol de Robôs Araranguá Intruders**

Trabalho de Curso submetido à Universidade Federal de Santa Catarina, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.



Professor Anderson Luiz Fernandes Pérez, Dr.  
Presidente da Banca - Orientador



Professor Fábio Rodrigues de La Rocha, Dr.  
Membro



Professor Roderval Marcelino, Dr.  
Membro

**Araranguá, 22 de fevereiro de 2013**

*Dedico este trabalho ao meu esposo  
Lucas, aos meus pais Julsemar e Jussara, e a  
minha irmã Natasha.*

## AGRADECIMENTOS

*Agradeço a Deus pela vida, salvação e força as quais me entregou todos os dias sem nenhum merecimento, simplesmente pelo seu infindável amor. À meu esposo por tanto carinho, compreensão e companheirismo os quais ofereceu por todo essa caminhada de estudos. À meus pais pela base incorruptível que formaram em mim, pela presença e pelo apoio. À minha irmã pela amizade e paciência. À minha vó pelo cuidado incondicional. À Universidade Federal de Santa Catarina. Ao meu orientador por conduzir todo esse projeto com paciência e dedicação. Aos meus colegas do LARM, especialmente ao Marcelo. Aos professores da banca pelas contribuições.*

*Em Cristo os corações podem ser enriquecidos  
da plenitude da inteligência porque Nele estão  
escondidos todos os tesouros da sabedoria e  
da ciência.*

***Colossenses 2.2-3;***

## RESUMO

Este trabalho descreve o desenvolvimento de um sistema de controle embarcado para a equipe de futebol de robôs Araranguá Intruders. O futebol de robôs possui várias categorias e uma delas é a F180. A sistemática do jogo é baseada em robôs, que estão em campo, e um computador externo que envia os comandos que devem ser executados por cada robô. Nesse caso os robôs são jogadores passivos, tendo o papel apenas de agir de acordo com a ação que lhe foi enviada. Este projeto visa fazer com que os robôs tenham um comportamento mais “inteligente”, fazendo com que estes sejam agentes ativos dentro de campo, participando intensamente do jogo. Para isso foi necessário o desenvolvimento de um sistema embarcado que levasse em consideração não apenas informações vindas do computador técnico, mas também de sensores embarcados nos robôs que os ajudassem na percepção do ambiente. Esse sistema usou técnicas de interrupção e de tempo real. O sistema desenvolvido permite que o robô execute ações, baseadas em dados recebidos do computador técnico e do seu próprio sistema em resposta às ativações dos sensores. O sistema embarcado foi projetado com dois níveis de controle, as tarefas de tempo real estão no primeiro nível formando uma arquitetura hierárquica, enquanto os sensores e o controle dos atuadores estão no segundo nível de controle, formando uma arquitetura reativa. O sistema embarcado foi testado e validado e os resultados demonstram que o número de troca de mensagens entre o computador técnico e o robô diminuiu, assim tem-se mais tempo computacional para o processamento da estratégia, tornando a equipe mais competitiva.

Palavras-chave: futebol de robôs, sistema embarcado, robótica móvel, categoria F180, microcontrolador PIC.

## ABSTRACT

This work describes the development of an embedded control system for the Araranguá Intruders robot soccer team. The robot soccer has several categories and one of them is the F180. The game is based on robots, which are in the field and an external computer that sends the commands to be executed by each robot. In this case robots are passive players acting only in accordance with the received command. This project aims to make the robots to behave more *intelligent*, making them to be active agents in the field, participating intensely game. For this reason an embedded system was developed that takes not only information from the computer coach, but also from sensors embodied in the robot that would help the perception of the environment. The developed system is based on two methodologies, real-time tasks and interrupts. The system allows that robot perform actions, based on data received from the computer coach and data coming from their own system in response to activations of the sensors. The embedded system is designed with two levels of control, real time tasks are the first level of control, forming a hierarchical architecture, while the sensors and control actuators are on the second level of control, forming a reactive architecture. The embedded system has been tested and validated, the results, prove that the number of message exchange between the computer and the robot is less than classical architectures, so it has more computational time to strategy processing, doing the team more competitive.

Keywords: robot soccer, embedded systems, mobile robots, F180 category, PIC microcontroller.

## LISTA DE ILUSTRAÇÕES

Figura 1: Representação de um agente.....	20
Figura 2: Arquitetura deliberativa. ....	21
Figura 3: Arquitetura reativa. ....	21
Figura 4: Arquitetura híbrida.....	22
Figura 5: Estrutura de um jogo da categoria F180. ....	25
Figura 6: Dimensões máximas dos robôs.....	26
Figura 7: Padrão para a identificação dos robôs da categoria F180.....	26
Figura 8: Esquemático da categoria F180. ....	27
Figura 9: Fluxo de informação. ....	28
Figura 10: Movimentos básicos de cada robô.....	30
Figura 11: Robô da equipe Araranguá Intruders utilizado na competição de 2011...32	
Figura 12: Influência de Sistemas Embarcados sobre a Computação Ubíqua. ....	37
Figura 13: Fluxo de um Projeto de Sistema Embarcado. ....	39
Figura 14: Desenvolvimento de Projeto em Níveis.....	40
Figura 15: Intermediação feita pelo Sistema Embarcado entre Ambiente Virtual e Real.....	42
Figura 16: Sistema Conectado a Sensores e Atuadores. ....	43
Figura 17: Gráfico de Utilidade de Tarefas mediante Prazo. ....	47

Figura 18: Arquitetura de hardware dos robôs da equipe Intruders.....	50
Figura 19: Vista expandida do controle dos motores.....	52
Figura 20: Arquitetura de Software dos robôs da equipe Araranguá Intruders.....	53
Figura 21: Esquemático da simulação do controle de Nível 1.....	58
Figura 22: Função de Simulação do Rádio e Comunicação SPI.....	59
Figura 23: Esquemático da simulação do controle de Nível 2.....	60
Figura 24: Esquema de posicionamento das rodas e motores no robô.....	61
Figura 25: Funções de Controle. ....	63
Figura 26: Tratamento das Interrupções.....	64
Figura 27: Esquemático do programa de exemplo.....	71
Figura 28: Programa para aceder três LEDs sem o uso de funções de tempo real. ...	72
Figura 29: Programa para aceder três LEDs com o uso de funções de tempo real. ...	73

## LISTA DE TABELAS

Tabela 1: Linguagens de programação utilizadas em sistemas embarcados. Dados de 2006.....	35
Tabela 2: Sistemas embarcados e sua utilização em cada mercado.....	36
Tabela 3: Características das Tarefas de Tempo Real do Sistema Embarcado.....	57
Tabela 4: Relação de cores dos LEDS com direção dos motores presentes na simulação.....	62

## LISTA DE ABREVIATURAS E SIGLAS

CPU	<i>Central Processing Unit</i>
CCS	<i>Custom Computer Services</i>
FIRA	<i>Federation of International Robot-Soccer Association</i>
GPS	<i>Global Position Systems</i>
IBM	<i>International Business Machines</i>
LED	<i>Light Emitting Diode</i>
PIC	<i>Peripheral Interface Controller</i>
PID	Proporcional Integral Derivativo
PWM	<i>Pulse Width Modulation</i>
RTOS	<i>Real Time Operation System</i>
SPI	<i>Serial Peripheral Interface</i>
SSL	<i>Small Size League</i>
TCC	Trabalho de Conclusão de Curso
UFSC	Universidade Federal de Santa Catarina
VSS	<i>Very Small Size</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>14</b>
1.1.	<i>Objetivos.....</i>	15
1.1.1.	<i>Geral.....</i>	15
1.1.2.	<i>Específicos.....</i>	15
1.2.	<i>Metodologia.....</i>	15
1.3.	<i>Justificativa e Motivação.....</i>	16
1.4.	<i>Organização do Trabalho.....</i>	17
<b>2</b>	<b>Futebol de Robôs.....</b>	<b>19</b>
2.1.	<i>Sistemas Robóticos.....</i>	19
2.2.	<i>Arquiteturas de Software.....</i>	20
2.3.	<i>Robôs Móveis.....</i>	22
2.4.	<i>Futebol de Robôs.....</i>	23
2.5.	<i>Categoria F180.....</i>	25
2.6.	<i>Equipe Araranguá Intruders.....</i>	30
<b>3</b>	<b>Sistemas Embarcados.....</b>	<b>33</b>
3.1.	<i>Definição de Sistemas Embarcados.....</i>	33
3.2.	<i>Metodologia de Desenvolvimento.....</i>	39
3.3.	<i>Sistemas Embarcados e Sistemas de Tempo Real.....</i>	44
<b>4</b>	<b>Sistema de controle embarcado para o futebol de robôs.....</b>	<b>49</b>
4.1	<i>Arquitetura de Hardware dos Robôs da Equipe Araranguá Intruders.....</i>	49
4.2	<i>Arquitetura de Software dos Robôs da Equipe Araranguá Intruders.....</i>	52
<b>5</b>	<b>Análise do Sistema de Controle.....</b>	<b>57</b>
5.1.	<i>Controle de Nível 1.....</i>	57
5.2.	<i>Controle de Nível 2.....</i>	60
<b>6</b>	<b>CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS..</b>	<b>65</b>
6.1.	<i>Propostas para Trabalhos Futuros.....</i>	66
	<b>REFERÊNCIAS.....</b>	<b>67</b>
	<b>Apêndice I.....</b>	<b>70</b>

## 1 INTRODUÇÃO

O desenvolvimento da tecnologia tem sido apresentado para a sociedade de diversas maneiras e tem se tornando real na vida cotidiana. O uso de robótica tem se popularizado muito, os robôs falam, cantam, dançam e até têm a aparência dos seres humanos. Entretanto, existe uma tecnologia que é ainda mais presente, e que se difunde de maneira discreta, são os sistemas embarcados. Estes já fazem parte do cotidiano das pessoas, em muitos casos, de modo imperceptível. Por crescerem e se difundirem em grande número os sistemas embarcados são um campo da tecnologia que merece a devida atenção.

O uso de sistemas embarcados percorre todas as áreas de domínio humano, em atividades simples ou complexas eles têm se apresentado eficientes e importantes, e assim seu desenvolvimento começa a ser mais valorizado e estudado. Os sistemas embarcados têm a capacidade de tornar um equipamento simples na casa de alguém um aparelho tecnológico que irá realizar a tarefa desejada com ainda mais eficiência.

Os sistemas embarcados relacionam-se fortemente com a robótica móvel e com a inteligência artificial. Várias entidades criam cenários que são tidos como problemas primários a serem estudados, sendo um destes cenários o futebol de robôs. Este ambiente foi escolhido por lidar com os mais diversos tipos de conhecimento e por contar com uma grande abrangência de informações, é um ambiente desafiador já que todo este escopo deve ser dominado para formar uma equipe adequada e competitiva (ROBOCUP, 1998).

No futebol de robôs existem conceitos que vão desde o projeto e a construção do hardware até o projeto e o desenvolvimento do software, e um ponto que une esses dois extremos são os sistemas embarcados, um software que é integrado ao hardware e que usa todos os seus recursos para a realização de uma tarefa. Por isso a robótica móvel, com o foco no futebol de robôs, é uma prática de estudo difundida por todo o mundo, existem muitas inovações que são desenvolvidas nestas competições. Um desafio dentro da categoria F180 do

Futebol de Robôs é tornar os robôs, normalmente passivos, agentes ativos em campo. Isso pode ser possível através do desenvolvimento de um novo sistema de controle embarcado para os robôs. Dotar o robô de algum grau de inteligência seria interessante já que aumentaria a competitividade da equipe, equilibrando adequadamente responsabilidades entre os robôs e o computador técnico.

## **1.1. Objetivos**

Para um melhor entendimento dos objetivos deste trabalho, esta seção está separada em objetivo geral e objetivos específicos.

### **1.1.1. Geral**

Desenvolver um sistema de controle embarcado para a equipe de futebol de robôs Araranguá Intruders que permita que os robôs sejam agentes ativos em campo, tornando a equipe mais competitiva.

### **1.1.2. Específicos**

1. Estudar o estado da arte no desenvolvimento de sistemas embarcados, sobretudo aqueles aplicados ao futebol de robôs;
2. Estudar microcontroladores e linguagens de programação para sistemas microcontrolados;
3. Especificar um sistema de controle embarcado para a equipe de futebol de robôs Araranguá Intruders que considere, além das informações provenientes do computador técnico, informações internas aos robôs;
4. Projetar um sistema embarcado que seja de tempo real e que atenda os requisitos definidos em (3);
5. Desenvolver o sistema embarcado projetado em (4);
6. Testar o sistema desenvolvido em (5) e avaliar os resultados obtidos.

## **1.2. Metodologia**

O sistema será desenvolvido na linguagem de programação C, com o compilador CCS (Serviços Personalizados de Computador, do Inglês *Custom Computer Services*) para a

família de microcontroladores PIC18F que atualmente é utilizada pelos robôs da equipe Aranguá Intruders.

O ambiente utilizado para testar o sistema embarcado e simular os circuitos eletrônicos dos robôs da equipe Intruders foi o simulador de circuitos Proteus na interface ISIS. Nele todo o circuito eletrônico foi desenvolvido e todos os testes foram realizados. Para o desenvolvimento do software foi utilizado o PCW, um ambiente de desenvolvimento integrado da CCS para microcontroladores PIC (Interface Periférica de Controle, do Inglês *Peripheral Interface Controller*).

### 1.3. Justificativa e Motivação

O futebol de robôs possui uma organização muito grande, existem diversas categorias e cada uma conta com algumas particularidades. Uma categoria muito conhecida é a *Small Size* ou F180 da Robocup, a estrutura desta categoria é baseada em um campo em que ficam cinco robôs de cada time, acima do campo ficam duas câmeras que captam todas as imagens que são processadas por um sistema de visão computacional o *SSL Vision (Small Size League)* que envia o resultado deste processamento ao computador técnico de cada equipe, que então determina uma estratégia de jogo que é enviada aos robôs jogadores.

O futebol de robôs exige grande mobilidade dos robôs jogadores, seu sistema embarcado precisa ler informações do rádio e controlar os atuadores, que são os motores e, se houver, o sistema de chute. Além disso, o sistema de controle interno dos robôs deve monitorar todo o aparato sensório, uma vez que as informações lidas dos sensores são utilizadas na tomada de decisão para a execução de uma ação do robô (PAZOS, 2002).

Quando um robô possui a capacidade de planejar e decidir suas ações pode-se afirmar que ele possui algum grau de inteligência. No caso do futebol de robôs, na maioria das equipes, os sistemas embarcados que controlam os robôs são apenas reativos, tornando-os agentes passivos durante todo o jogo. A arquitetura proposta neste trabalho apresenta um diferencial e uma inovação para o futebol de robôs, que é dotar o robô de capacidade de decisão, fazendo com que ele seja mais que um receptor, um robô móvel inteligente. Dessa maneira o sistema garantirá certo grau de autonomia ao robô, uma vez que muitas das decisões são baseadas em informações internas e ações individuais.

Existem vários objetivos para tornar o robô um agente ativo durante o jogo, o objetivo científico é a descoberta de conhecimento e aprofundamento na área de sistemas embarcados,

buscando atingir o máximo potencial desse campo da ciência. O objetivo para o futebol de robôs é aumentar a competitividade da equipe, pois além de retirar algumas responsabilidades excessivas do computador técnico, o robô poderá ser mais preciso e eficiente.

A equipe pode se tornar mais competitiva, pois poderá contar não apenas com a entrada de informações da câmera, mas também através dos sensores do robô, tornando as jogadas muito mais precisas.

Tornando o robô um agente ativo é possível diminuir o número de troca de mensagens entre computador técnico e robô, isso é uma vantagem já que quanto mais mensagens mais processamento de ambos os lados é realizado. Assim as energias dos sistemas podem ser voltadas apenas para o jogo e não apenas para a comunicação, isso é relevante, pois tanto o robô quanto o computador técnico gastam tempo para poder se comunicar, este tempo poderia ser utilizado na seleção da estratégia no caso do computador técnico ou na leitura dos sensores no caso do robô.

No caso do futebol de robôs os estudos são realizados para explorar ao máximo o potencial de cada equipe. A maioria dos avanços acontece no sistema estrategista, porém, a evolução no sistema embarcado pode ser um grande passo, já que um robô ativo possui inúmeras vantagens sobre um robô meramente passivo em campo.

Além de que um robô ativo faz parte da realização do sonho da Robocup que é organizar um jogo em meados do ano de 2050 entre robôs humanoides totalmente autônomos e a seleção campeã da Copa do Mundo mais recente, e ver a equipe robótica vencer. Assim, manter embarcada a tomada de decisões e a entrada de informações, faz parte de um objetivo maior, não apenas de uma equipe, mas de toda a instituição responsável.

Além de realizar os objetivos propostos para o futebol de robôs, manter algumas funcionalidades embarcadas no robô contribui para a tendência tecnológica que a humanidade tem caminhado. Um exemplo disso é a automação veicular que tem crescido em grandes proporções, e nesse caso o objetivo é que as responsabilidades, antes de um motorista humano, passem para o controle do sistema. O sistema embarcado realiza a tomada de decisões para executar, por exemplo, o piloto automático.

#### **1.4. Organização do Trabalho**

Este trabalho está organizado em seis capítulos, sendo o primeiro este, a introdução, que norteia a respeito do tema escolhido abordando os principais aspectos que serão explana-

dos, fundamentando a importância do futebol de robôs e orientando a respeito dos objetivos, justificativa e motivação do presente trabalho.

O Capítulo 2 descreve o Futebol de Robôs, onde é definido o conceito de sistemas robóticos, das arquiteturas de controle, das características principais dos robôs móveis e, por fim, o futebol de robôs com ênfase na categoria *Small Size* da Robocup.

O Capítulo 3 descreve as principais características dos sistemas embarcados abordando os aspectos relacionados ao projeto, implementação e aplicação de tais sistemas. O capítulo também aborda os sistemas embarcados de tempo real.

No Capítulo 4 é descrito o projeto e o desenvolvimento do Sistema de Controle Embarcado para os robôs da Equipe de Futebol de Robôs Araranguá Intruders. O sistema está descrito de acordo com seus blocos funcionais.

No Capítulo 5 é realizada a análise do sistema de controle descrito no Capítulo 4. É detalhado o desenvolvimento e os resultados atingidos separadamente nos dois níveis de controle do sistema. Também é feita a comparação entre o sistema ideal e o realizado neste trabalho.

No Capítulo 6 são apresentando os resultados finais que demonstram que os objetivos determinados foram alcançados. O capítulo também traz algumas perspectivas para trabalhos futuros visando dar continuidade ao sistema desenvolvido neste TCC (Trabalho de Conclusão de Curso).

No Apêndice I são apresentadas mais informações sobre tarefas de tempo real utilizadas pelo compilador CCS. Para um melhor entendimento foi elaborado um exemplo que compara um programa com funções sem o uso de tempo real e um programa com tarefas de tempo real.

## 2 FUTEBOL DE ROBÔS

Este capítulo discorre sobre a definição e conceito de sistemas robóticos, as arquiteturas de controle, as características principais dos robôs móveis e, por fim, o futebol de robôs com ênfase na categoria *Small Size* da Robocup.

### 2.1. Sistemas Robóticos

Um robô é mais do que hardware e software combinado para um determinado fim, possui um sistema de controle que organiza todos os seus recursos e limitações com a finalidade de executar uma tarefa pré-determinada. Dependendo da aplicação os sistemas de controle robóticos são simples ou muito complexos, vários quesitos determinam esse nível de complexidade, sendo um dos mais importantes a maneira com que o robô irá relacionar-se com o meio, pelo fato de que um robô sofre influência e também influencia seu ambiente, suas entradas e saídas são muito variáveis e organizá-las aumenta a complexidade do desenvolvimento do sistema.

A arquitetura do sistema robótico é determinada de acordo com a organização, separação e comunicação de cada um de seus elementos. Um sistema, para ser completo, deve possuir duas partes principais: uma lógica, composta pelos módulos de software, e outra física, que é formada pelos componentes de hardware (processadores, sensores, atuadores, entre outros) (MAETA, 2001).

Os sistemas robóticos são utilizados nas mais diversas condições e suas tarefas também possuem uma grande variedade, em muitos casos existem restrições do ambiente que os resultados devem respeitar. Uma das restrições de um sistema robótico são os requisitos de tempo. Em algumas situações o sistema deve responder imediatamente, ou seja, ser considerado de tempo real. Portanto o módulo de software deve, além de gerenciar todo o, deve ainda atender a requisitos de tempo.

Os sistemas robóticos também podem possuir autonomia e podem ser representados como agentes. Um agente é definido como a entidade que influencia e é influenciada pelo meio, percebe o ambiente através de sensores e age sobre ele através de atuadores (RUSSEL; NORVIG, 2004).

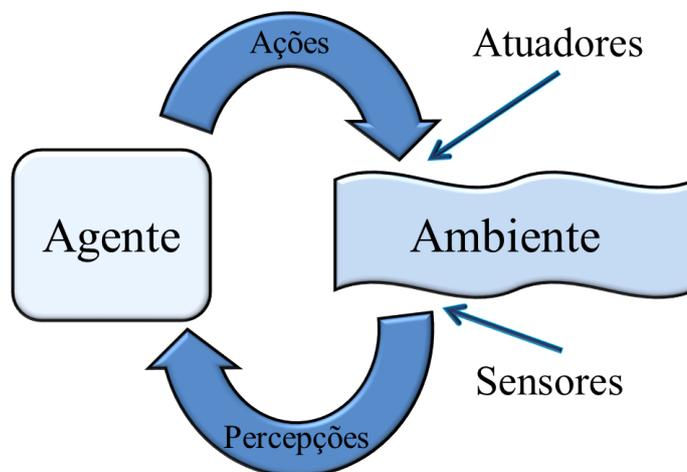


Figura 1: Representação de um agente.

A Figura 1 ilustra a interação de um agente com o ambiente através de sensores e atuadores. Os sensores e atuadores não são padronizados, os agentes os escolhem de acordo com suas características fundamentais e de acordo com os objetivos finais, apesar das diferenças o papel é sempre o mesmo, manter o agente interagindo sempre com o ambiente.

## 2.2. Arquiteturas de Software

Muito ouve-se falar sobre a arquitetura de hardware, porém pouco se observa a respeito da arquitetura do software, este último, também possui padrões e paradigmas a serem analisados. Existem três grandes arquiteturas para software em sistemas robóticos, a deliberativa ou hierárquica, a reativa e a híbrida, que é a junção das anteriores.

A deliberativa realiza suas ações internas de maneira lógica e sequencial, primeiro percebendo o ambiente, depois planejando a ação que será realizada e, por fim, ocorre a ação de fato. Nesta arquitetura que esta representada na Figura 2, uma tarefa é dividida em outras sub-tarefas que são executadas uma por vez, a reorganização e junção das subdivisões caracteriza o objetivo do sistema robótico. Esta arquitetura apresenta resultados positivos se o ambiente for muito dinâmico já que situações inesperadas forçam o sistema a perceber e planejar novamente para então agir.

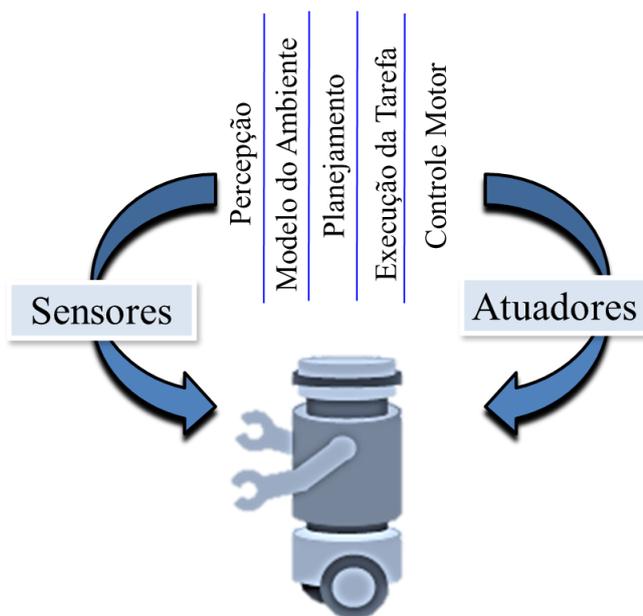


Figura 2: Arquitetura deliberativa.  
Fonte: Adaptado de (Sousa, 2007).

O sistema também pode ser baseado no conceito de ação-reação, no qual suas ações serão relativas aos estímulos recebidos do ambiente, este conceito é conhecido como arquitetura de software reativa, que está representada na Figura 3. Os estímulos gerados e percebidos pelo robô não serão processados para a avaliação e elaboração de um plano, nessa arquitetura não há fase de planejamento, a saída é totalmente relacionada à entrada.

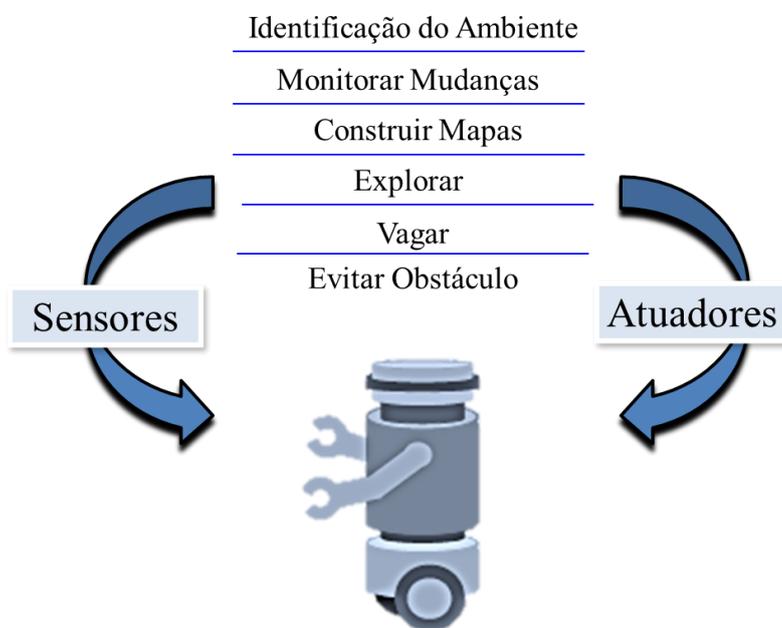


Figura 3: Arquitetura reativa.  
Fonte: Adaptado de (Sousa, 2007).

Já na arquitetura híbrida, que foi desenvolvida posteriormente às duas arquiteturas anteriores, o conceito principal é baseado na ideia de que um sistema pode em alguns pontos precisar de planejamento e em outros deve apenas reagir aos estímulos do ambiente. De acordo com esse pensamento partes do sistema devem ser deliberativo, enquanto outras devem ser reativas.

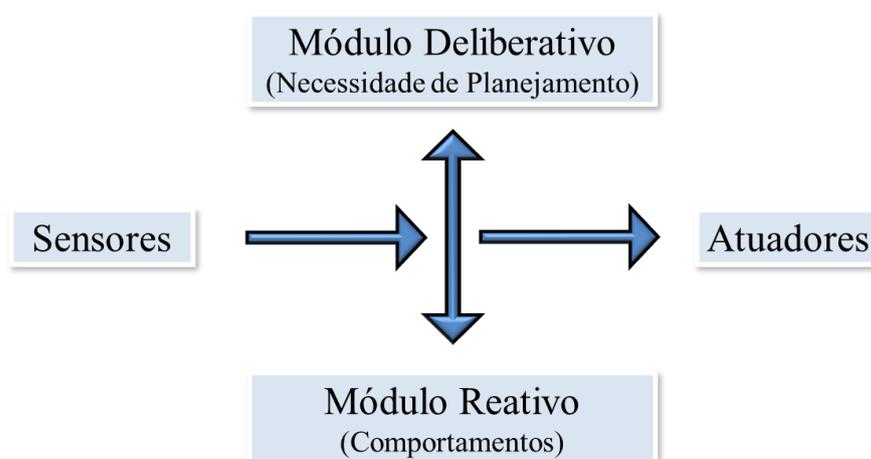


Figura 4: Arquitetura híbrida.

Na Figura 4 pode ser analisado como é a estrutura da arquitetura híbrida. O sistema continua recebendo informações de seus sensores e enviando informações para os atuadores, mas estes podem estar ligados tanto a módulos deliberativos como a módulos reativos. No caso do Futebol de Robôs na categoria F180 da Robocup, esta arquitetura é muito utilizada já que o planejamento é essencial para a estratégia, enquanto que para o robô o conceito de ação-reação é adequado.

### 2.3. Robôs Móveis

Os sistemas robóticos são um campo muito abrangente e podem ser aplicados em vários setores da economia, como o caso de usinas e indústrias. Em muitos casos era necessário que o robô tivesse mobilidade, portanto, aos poucos foi se desenvolvendo a área dos robôs móveis.

Segundo Dudek e Jenkin (2000) a constituição do robô suprime as barreiras de hardware e software, ele deve ser analisado de acordo com quatro itens, e, a partir deles, pode-se desenvolver o projeto dos módulos do sistema. Estes itens relacionam-se a fase inicial de projeto de um sistema robótico.

1. Locomoção: capacidade que o robô tem de se locomover pelo ambiente.

2. Percepção: como o robô percebe a si mesmo, seu comportamento e ações, e o ambiente.
3. Raciocínio: capacidade de mapear e controlar as informações adquiridas e reagir segundo as mesmas.
4. Comunicação: forma como o robô se comunica, seja com o ambiente, outros sistemas robóticos, ou ainda, um operador humano.

Para um robô móvel é imprescindível que seu nível de locomoção seja positivo. A maneira como ele vai se mover determina muito a respeito de como será realizada a tarefa proposta. O meio para o qual o robô é projetado é significativo, podendo haver locomoção em ambientes terrestres, aéreos e aquáticos, dentro dessas possibilidades os robôs ainda possuem várias maneiras de se mover como no ambiente terrestre em que ele pode andar, correr, saltar, pular ou ainda rolar. Os ambientes também podem ser variáveis apresentando combinações tais quais as citadas em (SIEGWART, NOURBAKHSI, 2004).

Os itens descritos anteriormente são importantes para definir o comportamento do robô diante de uma determinada situação, já que englobam todos os módulos que constituem um sistema robótico. E pelo fato de que os ambientes em que o robô pode estar inserido serem muito diferentes entre si e sofrerem variações constantes, os sistemas robóticos são primeiramente testados e desenvolvidos em ambientes controlados. Em muitos casos mesmo depois de concluídos os robôs continuam em ambientes controlados, já que sua aplicação oferece esse quesito. Esse é o caso dos robôs para o futebol de robôs, o ambiente em que eles atuam é composto por regras, é delimitado e normatizado e pode, também, ser controlado fornecendo ao robô informações essenciais.

## 2.4. Futebol de Robôs

O futebol de robôs foi citado pela primeira vez por Alan Mackworth, no artigo intitulado *On Seeing Robots* (MACKWORTH, 1993), esta ideia surgiu como uma iniciativa de pesquisadores e cientistas para manter o desenvolvimento de novas tecnologias.

Por muitos anos o grande foco de pesquisa foi o jogo de Xadrez e este estudo chegou ao ápice com a criação de um computador chamado Deep Blue pela IBM (Negócios Internacionais de Máquina do Inglês *International Business Machines*). Este computador superou um

grande desafio, disputou um jogo de Xadrez com o campeão mundial Garry Kasparov em 1997 e o venceu (CAMPBELL, 2002).

Atualmente existem novas propostas de áreas para pesquisas e uma delas é o futebol de robôs. Muitos desafios de Inteligência Artificial foram propostos com o objetivo de manter pessoas motivadas a descobrir e desenvolver inovações que tragam algum ganho para a humanidade (MACKWORTH, 1993).

De acordo com Salim (2004) o futebol de robôs é uma proposta que traz grandes ganhos já que é um ambiente propício para testes de novos algoritmos, desenvolvimento de novas arquiteturas de hardware e validação de várias teorias incluindo de agentes autônomos. A proposta de desenvolver agentes robóticos totalmente autônomos também é testada e estudada dentro do ambiente do futebol, o objetivo é que em algum momento os robôs possam perceber, interagir e aprender com os ambientes, por mais instáveis e dinâmicos que eles sejam (SAHOTA, 1994).

Segundo Kraetzchmar (1998) e Cheng (2001) para se formar uma equipe de futebol de robôs são necessários muitos recursos, na parte de software o sistema sempre deve buscar facilitar o papel do agente, então podem ser citados sistemas de controle inteligente para auxiliar o robô na descoberta do ambiente, sistemas que relacionem informações de vários sensores, já que a informação é provida de várias entradas deste gênero, e ainda, visão computacional com processamento de imagens, redes neurais e computação evolutiva para a realização da estratégia, colaboração multiagentes e sistemas baseados em tempo real. Já na parte de hardware é preciso definir o projeto do robô, para isso é relevante o conhecimento sobre dispositivos mecatrônicos e características de sensores, atuadores e efetadores.

Os recursos que devem estar disponíveis para a formação de uma equipe de futebol de robôs demonstra o crescimento intelectual que essas competições representam para a sociedade. A Robocup<sup>1</sup> e a FIRA<sup>2</sup> (Federação Internacional de Futebol de Robôs, do Inglês *Federation of International Robot-Soccer Association*) são as instituições que promovem os campeonatos por todo o mundo. Elas surgiram nos anos 90 e o objetivo maior é a promoção de meios de manter a Inteligência Artificial em evolução.

A partir da popularização desse desafio e dos campeonatos que foram ao longo dos anos sendo realizados em várias partes do mundo, muitas categorias foram surgindo, popula-

---

<sup>1</sup> Site Oficial: <http://www.robocup.org/>

<sup>2</sup> Site Oficial: <http://www.fira.net/>

rizando a diversidade no meio da robótica. As categorias variam de modelo e tamanho do robô, regras e até as tecnologias apropriadas, como, por exemplo, no caso de que em uma categoria a visão se dá com uma câmera embarcada no robô, enquanto em outras há uma câmera global que alimenta a visão computacional. Dentre as modalidades presentes na Robocup pode-se citar *Humanoid League*, *Four Legged League*, *Simulation League*, *Middle Size League*, *Small Size Robot League (F180)*, *Very Small Size Robot League*. Devido ao fato de a Equipe de Futebol de Robôs Araranguá Intruders disputar competições de robótica na categoria F180, na Seção 2.5 são descritas as principais características desta categoria.

## 2.5. Categoria F180

A categoria F180 foi escolhida devido a possibilidade de desenvolvimento de novas tecnologias no campo da robótica móvel. Por possuir mais robôs e estes serem maiores que a categoria *Very Small Size* torna possível o estudo e utilização de muitos equipamentos diferentes e inovadores na parte de hardware. Esta categoria também representa um desafio para o desenvolvimento de software, uma vez que deve-se desenvolver o sistema embarcado dos robôs e o sistema de geração de estratégias.

A categoria *Small Size (ou F180)* concentra-se na cooperação multi-agente, pois a equipe deve estar trabalhando como um todo para o mesmo objetivo. O ambiente é altamente variável, a dinamicidade é produzida pelos movimentos da bola e ainda as ações dos robôs jogadores. Na Figura 5 é possível ter uma visão geral de como é a estrutura para um jogo.



Figura 5: Estrutura de um jogo da categoria F180.

Para formar uma equipe deve-se ter de um até seis robôs, sendo obrigatoriamente um deles o goleiro. As dimensões dos robôs são reduzidas, podendo ser no máximo 15 centímetros de altura e 18 centímetros de diâmetro, conforme ilustração da Figura 6.

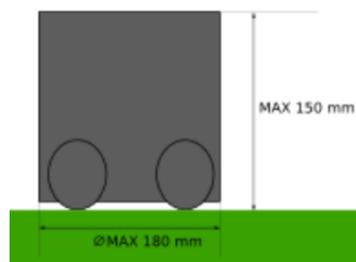


Figura 6: Dimensões máximas dos robôs.

Fonte: Extraída de (Robocup, 2012).

O peso varia muito e é de caráter particular de cada equipe. O campo tem 6,5 metros de comprimento por 4,5 metros de largura, é da cor verde, enquanto que a bola utilizada é do padrão de golfe e da cor alaranjada. A identificação vai acima dos robôs, e é formada por vários círculos, cada um com uma cor, a combinação das cores forma a identidade de cada robô. As regras determinam onde estará cada círculo e as cores escolhidas são para facilitar o processamento de imagens. As identificações possíveis podem ser vistas na Figura 7.

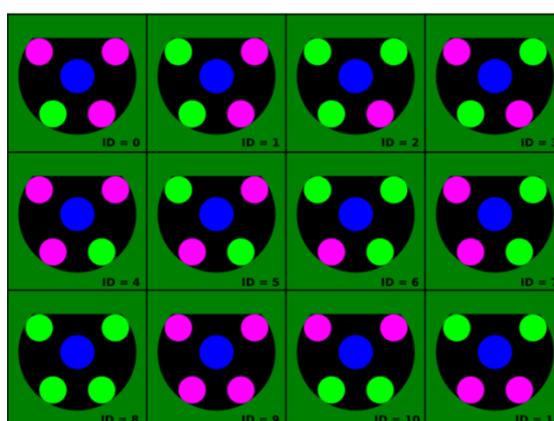


Figura 7: Padrão para a identificação dos robôs da categoria F180.

Fonte: Extraída de (Robocup, 2012).

Nesta categoria existe um dispositivo de visão global que fica posicionada acima do campo, esse dispositivo é formado por duas câmeras, cada uma responsável por gerar imagens de uma metade do campo. Essas imagens são processadas por um computador com o sistema de visão computacional instalado (*SSL Vison*). Este será responsável pelo processamento das imagens buscando identificar o posicionamento da bola e de todos os jogadores em campo. Essas informações são enviadas pela rede aos computadores técnicos de cada equipe.

O computador conhecido como técnico é o que recebe as informações do *SSL Vision*. A partir dessas informações determina uma estratégia adequada e envia, a cada robô jogador, a ação que deve ser realizada. Ou seja, nesse computador técnico está todo o sistema de estratégia, segundo as informações recebidas do sistema de visão computacional *SSL Vision*, ele escolhe a melhor estratégia para seus robôs jogadores, sempre com o intuito de vencer o jogo, fazendo gols e defendendo-se do adversário. Na Figura 8 pode ser observada essa descrição.

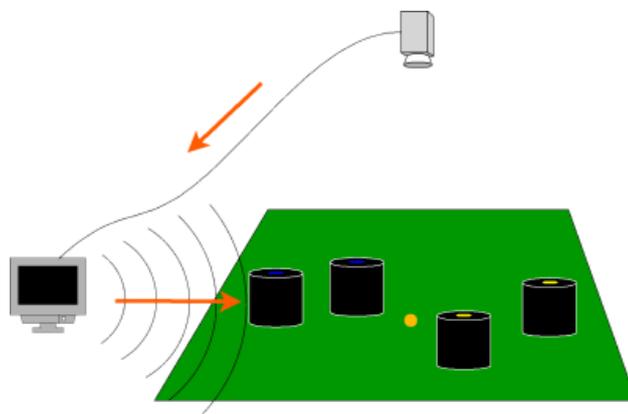


Figura 8: Esquemático da categoria F180.

Fonte: Extraído de (Robocup, 2012).

A Figura 8 ilustra uma situação onde a câmera está enviando informações diretamente ao computador que controla os robôs, dentro do especificado nesta Figura o mesmo computador realiza tanto o processamento da imagem quanto a geração de estratégia. Essa arquitetura pode ser utilizada, porém não é a mais comum. A partir de 2008, a Robocup disponibilizou um programa de processamento de imagem a todas as equipes, o *SSL Vision*, e essas podem escolher se preferem utilizar seu próprio programa de processamento de imagem ou utilizar o *SSL Vision*.

A Robocup escolheu disponibilizar esse programa de processamento de imagens, pois não havia mais inovação, todas as equipes usavam programas semelhantes e para evitar o tempo investido em algo que já estava bem consolidado a Robocup criou esse programa e o disponibilizou. A partir disso as equipes adotaram o *SSL-Vision*<sup>3</sup> e mesmo não sendo uma regra passou a ser o meio utilizado por todos os participantes para tratar as imagens.

<sup>3</sup> Site Oficial do Sistema SSL-Vision: <http://small-size.informatik.uni-bremen.de/sslvision>

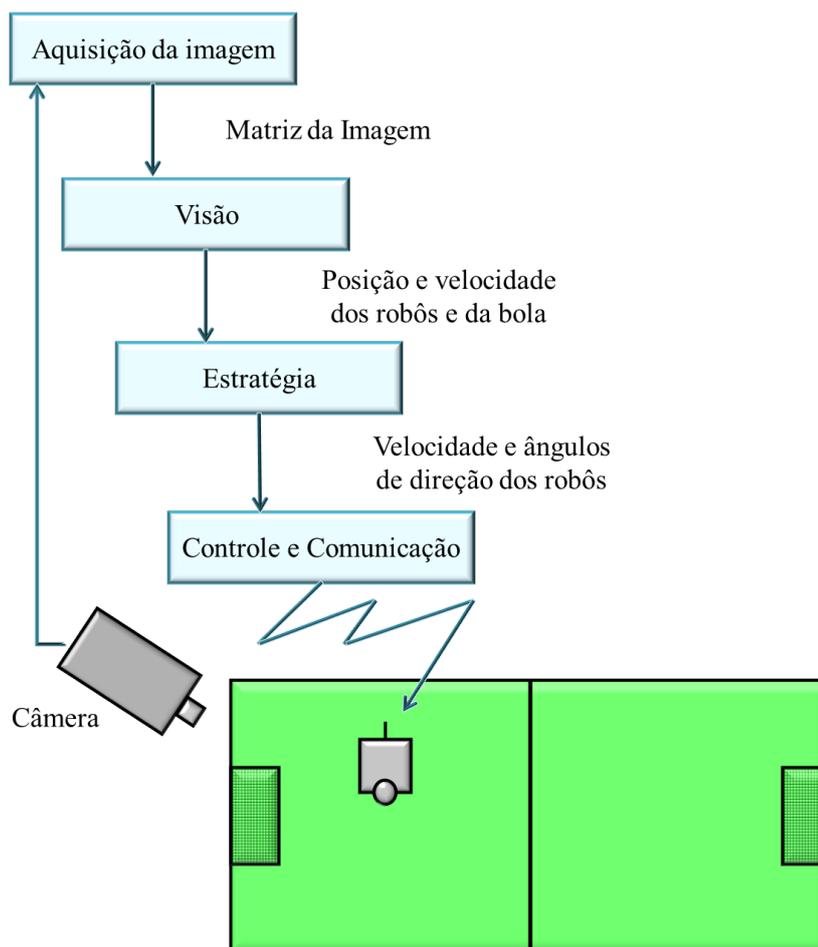


Figura 9: Fluxo de informação.

Fonte: Adaptada de (Sobreira, 2003).

Na Figura 9 pode ser visualizado o fluxo da informação, a partir do momento que a estratégia foi escolhida o computador técnico envia pelo rádio a ação individualmente a cada robô. Para saber a orientação de cada robô o sistema de processamento de imagem relaciona o formato do robô, sua identidade e as coordenadas do campo.

Percebe-se que nessa categoria o grau de autonomia dos robôs é limitado, já que recebem a ordem de movimentos que devem ser tomados de outro sistema maior, neste caso, o computador técnico. Porém alguns comportamentos não necessitam do controle do computador técnico ficando sob o comando do robô móvel, entre esses comportamentos pode se citar todo o controle do aparato sensório e todas as decisões que são relativas destes. Como, por exemplo, o domínio de bola, se o sensor de presença indicar que a bola está próxima o próprio robô decide se ativa ou não o sistema de domínio.

Segundo Novak e Springer (2004) o objetivo sempre é minimizar o processo de comunicação, o foco deve ser manter baixo o número de troca de mensagens. Para que isso

ocorra é necessário um bom projeto para definir quais funcionalidades ficam no robô e quais ficam no computador técnico.

A arquitetura apresentada pela categoria F180 oferece a vantagem para o reconhecimento do ambiente já que a câmera central capta todas as informações do campo, o *SSL Vision* as interpreta e repassa ao computador técnico que acaba tendo uma visão bem clara, facilitando assim a definição de estratégias. Por outro lado, em alguns momentos, podem deixar a desejar no tempo de resposta do robô para a execução da ação que foi enviada pelo computador técnico, já que o robô não pode oferecer a reação ao comando em tempo real.

Atualmente, existe a tendência de utilizar a arquitetura de software híbrida, na estrutura do futebol de robôs, isso pode ser afirmado pelo fato de partes do sistema serem tratados com mais prioridade que outros, terem requisitos diferentes e precisarem até, estar alocados separadamente.

É o que ocorre no caso das tarefas do sistema que possuem restrição temporal, como o desvio de obstáculos, o momento do chute e a condução da bola, estas devem ser tratadas com o máximo de agilidade e, portanto, devem estar embarcadas no robô e podem ser de tempo real. Enquanto as ações que precisam de planejamento são realizadas pelo computador técnico, como a geração de estratégia e o controle de navegação dos robôs (BIAZUZ, 2008).

Os movimentos possíveis de cada robô variam de acordo com o número e estrutura das rodas. Na Figura 10 podem ser visualizados os movimentos básicos comuns a todos os robôs. Dependendo da arquitetura de cada robô o acionamento dos motores e a movimentação pode apresentar algumas diferenças, já que a categoria não determina o número de rodas utilizado e nem sua posição com relação ao robô.

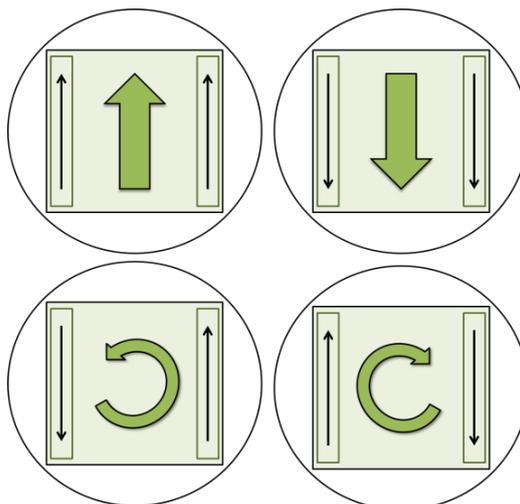


Figura 10: Movimentos básicos de cada robô.

Os robôs recebem as informações do sistema de estratégia, porém, como já foi dito, também recebem informações de sensores que podem estar embarcados neles próprios. O esquema de um robô que recebe informação de ambas as entradas é significativo no momento da competição. Os sensores influenciam na decisão, isso é útil, por exemplo, quando um robô está próximo a bola, o computador técnico não precisa mandar que ele a domine, o domínio pode ser pré-definido a partir do sensor de presença, ou seja, sempre que o robô estiver perto da bola vai dominá-la, só será diferente se a estratégia determinar que ele chute, aí então, ele executa a ação recebida do módulo de comunicação que sempre tem prioridade.

É no momento de projetar a arquitetura de controle da equipe que se deve escolher qual funcionalidade vai ser tratada em que módulo, por ser muito essencial para o sistema como um todo esta etapa deve ser testada e validada.

## 2.6. Equipe Araranguá Intruders

A Equipe de Competições Tecnológicas Araranguá Intruders surgiu no ano de 2011 pela iniciativa de professores e alunos da Universidade Federal de Santa Catarina (UFSC) para desenvolver, inicialmente, robôs para competições de futebol. A equipe Araranguá Intruders participa na categoria F180 da Robocup e sua missão é *participar e conquistar títulos em competições de robótica a nível nacional e internacional valorizando o conhecimento e contribuindo na formação técnica de seus membros* (Intruders, 2012).

O futebol de robôs foi escolhido pela proposta de desenvolvimento, formulação de conhecimento e inovações. Os grandes investimentos para manter crescente o interesse em

Inteligência Artificial e seus respectivos conceitos chama a atenção de vários pesquisadores e mantém esse campo da ciência como um dos auxiliares no desenvolvimento de novas tecnologias.

O projeto de um sistema embarcado para uma equipe de futebol de robôs não necessita de alto investimento financeiro para seu desenvolvimento, entretanto o sistema embarcado é apenas uma das partes que compõe uma equipe de futebol de robôs. Logo, observa-se que para formar uma equipe de futebol de robôs existe a necessidade de apoio financeiro. A produção de robôs e os recursos necessários para o desenvolvimento do software de geração de estratégias refletem em custos relativamente altos se comparados aos valores de investimento em sistemas embarcados. No caso da equipe Araranguá Intruders contou-se com apoio financeiro de patrocinadores.

Adquirindo o conhecimento necessário, gradativamente foram sendo determinados pontos com relação ao projeto dos robôs e paralelamente os equipamentos foram sendo adquiridos. A partir dessa etapa os robôs começaram a ser produzidos e os softwares relacionados desenvolvidos.

Os robôs da equipe Araranguá Intruders contam com um sistema embarcado produzido para o microcontrolador PIC18F452, este foi escolhido por ser de baixo custo e atender os requisitos de hardware dos robôs. Este microcontrolador possui 40 pinos e alguns deles tem funções especiais. Os pinos são separados por portas, que são nomeadas com letras do alfabeto, cada porta possui 8 pinos correspondentes que vão de 0 a 7. Alguns pinos são analógicos e outros digitais.

O PIC18F452 possui potencial satisfatório para a aplicação para o qual foi escolhido. Os pinos que são diferenciados, como os analógicos e os utilizados para o controle de PWM (Modulação por Largura de Pulso, do Inglês *Pulse Width Modulation*), são reduzidos, porém suficientes para os robôs jogadores de futebol.

A Figura 11 ilustra um dos robôs que participaram da IX Competição Brasileira de Robótica, realizada em outubro do ano passado na cidade de São João Del Rei - MG. Ao todo foram produzidos seis robôs idênticos.

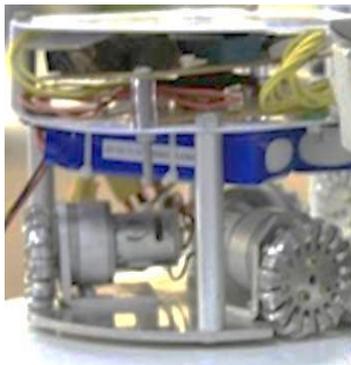


Figura 11: Robô da equipe Araranguá Intruders utilizado na competição de 2011.

Cada robô da Equipe Araranguá Intruders utilizado na competição de 2011 possuía a seguinte especificação de hardware: microcontrolador PIC18F452, três motores de corrente contínua com caixa de redução, driver dos motores baseado em circuitos de ponte H, sistema de comunicação da Telecontrolli de 433 MHz. As rodas e o chassi são de alumínio de 3mm e 4mm, respectivamente.

## **3 SISTEMAS EMBARCADOS**

Este capítulo descreve as principais características dos sistemas embarcados abordando os aspectos relacionados ao projeto, implementação e aplicação de tais sistemas. O capítulo também aborda conceitos relativos aos sistemas embarcados de tempo real.

### **3.1. Definição de Sistemas Embarcados**

Os sistemas embarcados são um tipo diferente de tecnologia, não são perceptíveis como no caso dos computadores, mas fazem parte da vida cotidiana de muitas pessoas. A rotina do homem está cada vez mais repleta de tecnologia e os sistemas embarcados são a grande maioria dessa tecnologia, apesar de muitos nem ao menos percebem sua utilização diária.

Sistema embarcado ainda não é um termo popular, porém esta tecnologia é muito utilizada pelas pessoas. O próprio nome já consta em si uma definição, ele baseia-se em sistemas que estão embarcados, ou seja, dentro de um equipamento, isso significa que existe um programa que faz parte do aparelho. Esses sistemas são interessantes, pois são utilizados para maximizar as funcionalidades de um determinado equipamento.

Os sistemas embarcados são a reunião de hardware, software e periféricos que tem um objetivo comum, realizar uma tarefa específica. A capacidade computacional destes sistemas está voltada a atender um único objetivo. São fortemente acoplados ao hardware e se adequam as características deste.

Um sistema embarcado utiliza todos os seus recursos e existe para cumprir um propósito especial, como, por exemplo, controlar as funções de aquecimento de micro-ondas ou de resfriamento do ar-condicionado (CARRO; WAGNER, 2003). O termo embarcado ou embutido é utilizado justamente por executar sobre plataformas de hardware, fazendo parte dele. O dispositivo eletrônico esconde o sistema dentro de si.

Um equipamento que possui um sistema embarcado possui seu projeto de hardware elaborado juntamente com o projeto de software, isso ocorre, pois é inerente a sistemas embarcados a forte ligação entre o software e o hardware. Ambas as partes adequam-se a características e limitações da outra para otimizar ao máximo todos os recursos disponíveis. Sistemas embarcados também podem fazer parte de sistemas maiores objetivando maximizar o potencial do dispositivo para aumentar a funcionalidade e eficiência.

Os sistemas embarcados são projetados de acordo com critérios e especificidades de cada aplicação. Segundo Wolf (2001) os requisitos específicos para cada projeto são: o objetivo do sistema, o tempo de resposta, o consumo de energia, o desempenho, a pouca disponibilidade de memória, a portabilidade e a confiabilidade necessitando assim, de uma configuração única para cada aplicação.

Por existirem tantos pontos a serem observados na construção de um sistema embarcado para que ele seja adequado à tarefa proposta, é importante explorar a arquitetura de hardware e software avaliando as consequências no resultado final. Esta prévia avaliação deve corresponder ao sistema real, portanto é necessário a utilização de bons estimadores e métodos que tragam resultados concretos, para que a avaliação seja suficiente deve-se explorar o espaço de projeto no nível mais abstrato possível (GERVINI, 2003).

De acordo com Taurion (2005), algo que pode determinar o sucesso e a qualidade de um sistema embarcado, seja ele para qual aplicação for é o projeto e o ambiente de construção, estes pontos são relevantes para o resultado final. Pelas muitas limitações encontradas na área dos sistemas embarcados a maioria dos softwares não adota conceitos como orientação a objetos, e são escritos em linguagens de programação estruturadas como a linguagem C, como pode ser percebido na Tabela 1.

Tabela 1: Linguagens de programação utilizadas em sistemas embarcados. Dados de 2006.

(Extraído de CMP United Business Media 2006).

Linguagem	Uso (%)
C	51
C++	30
Assembly	8
Java	3
BASIC	1
UML, MatLab ou outra linguagem de modelagem	3
LabView	2
Outras	5

Os sistemas embarcados são projetados, na maioria das vezes, para microcontroladores que são pequenos dispositivos que contam com quesitos computacionais essenciais, tais como memória, unidade central de processamento e dispositivos de entrada e saída, provendo assim, condições de se elaborar um sistema para atender uma demanda específica.

O sistema embarcado lida com várias restrições, o software, especificamente precisa contornar o limite de memória disponível, o desempenho do processador e o consumo de energia, isso deve ser analisado para determinar a utilização de uma plataforma para tal aplicação. De acordo com a aplicação que deve ser desenvolvida a plataforma e o microprocessador serão escolhidos.

Da mesma maneira em que o software presente em um sistema embarcado pode variar muito, o hardware também apresenta tal característica, a arquitetura de um sistema computacional embarcado pode conter blocos dedicados, memórias, um ou mais processadores e interfaces para periféricos. É essencial que todos esses componentes tenham boa comunicação entre si, esta pode ocorrer através de uma estrutura de interligação que pode ser um barramento ou até uma rede complexa (BENINI, 2002).

Geralmente o hardware de um sistema embarcado é formado por microprocessador, memória e periféricos, tornando claro que ele é um sistema computacional, as diferenças estão nos requisitos e nas aplicações alvos para os quais é projetado (BOSA, 2009). O software é, normalmente, armazenado em uma memória que compõe a plataforma que está sendo utilizada. Ele é projetado e compilado para a arquitetura do microcontrolador escolhido.

A capacidade computacional de um sistema microprocessado é suficiente para muitas aplicações como pode ser observado na Tabela 2. Os exemplos de sistemas embarcados vão

desde uma simples máquina de lavar até a automação completa de uma residência. Isso demonstra as potencialidades dos sistemas embarcados, apesar de serem pouco notados eles têm uma grande capacidade computacional. Inclusive possuem integração com a Internet, esta é uma tendência natural, já que os sistemas embarcados representam a maior fatia do mercado de processadores e a interconexão em redes TCP/IP é realizada com simplicidade e sucesso (MACHADO; SIQUEIRA, 2004).

Tabela 2: Sistemas embarcados e sua utilização em cada mercado.

Extraído de (Noergaard, 2005).

Mercado	Dispositivo Embarcado
Automotivo	Sistema de Ignição
	Controle de Motor
	Freios (ABS)
Eletrônica de Consumo	Televisores Analógicos e Digitais
	DVDs, Vídeo cassetes
	<i>Personal Data Assistants</i> (PDAs)
	Eletrodomésticos (Refrigeradores, Microondas, Torradeiras)
	Brinquedos, Jogos
	Telefones, <i>Pagers</i> e Celulares
	Câmeras
	<i>Global Positioning Systems</i> (GPS)
Controle Industrial	Robótica e Controle de Sistemas (Manufatura)
Medicina	Bombas de Infusão
	Próteses
	Equipamento de Diálise
	Monitores Cardíacos
Redes de Comunicação	Roteadores
	<i>Hubs</i>
	<i>Gateways</i>
Automação de Escritórios	Equipamentos de Fax
	Copiadoras
	Impressoras
	<i>Scanners</i>

Conforme Vianna (2009) a proliferação de aplicações para sistemas embarcados é resultado de uma convergência para a computação ubíqua, esse conceito é baseado na hipótese de que a computação pode transcender o escopo dos computadores convencionais e torna-se parte no cotidiano das pessoas. O cientista Marc Weiser foi o primeiro a usar este termo, ele

acredita que neste mundo, deveremos conviver com computadores e não apenas interagir com eles (WEISER, 1991).

Muitas características dos sistemas embarcados podem ser encontradas na computação pervasiva ou ubíqua. A Computação Pervasiva visa o uso da informação em todos os lugares em todo o tempo. Na Figura 12, pode-se ver uma representação gráfica de como a Computação Ubíqua é influenciada pelas Tecnologias da Comunicação e pelos Sistemas Embarcados (MARWEDEL, 2003).

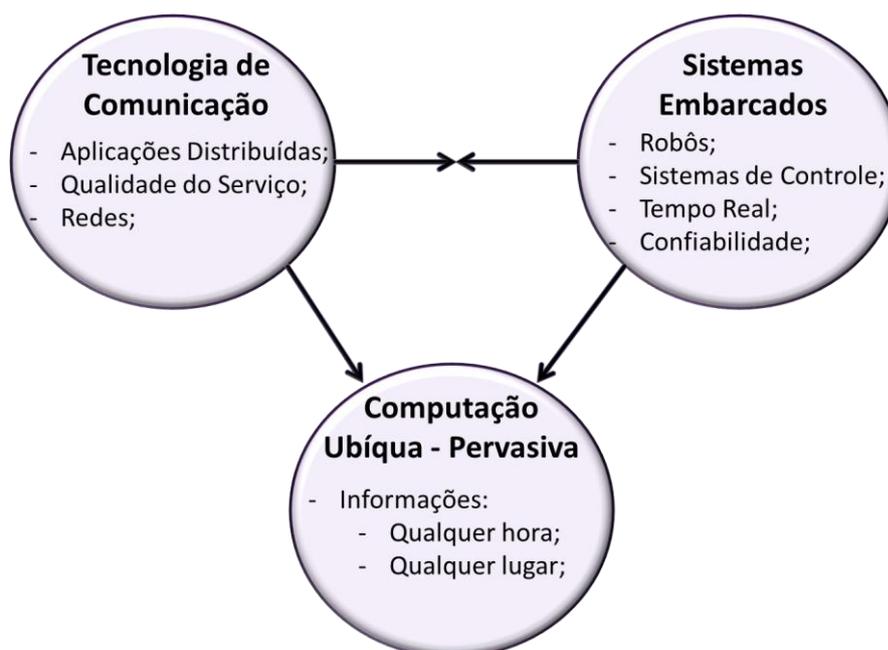


Figura 12: Influência de Sistemas Embarcados sobre a Computação Ubíqua.

Adaptado de (Marwedel, 2003).

De fato os sistemas embarcados estão hoje em todo lugar e na maioria das vezes não são ao menos notados, existe a tendência de que possa haver a convivência com essa tecnologia mesmo sem percebê-la. Na maioria dos casos os sistemas embarcados são discretos, a prioridade é manter o foco no objetivo final para o qual foram projetados, ou seja, a preferência está na tarefa a ser realizada e não na maneira como será feita.

Este paradigma entre a tarefa e o método é algo muito discutido, pois por muitos anos a tecnologia evoluiu para facilitar a maneira de realizar a tarefa com eficiência. Os computadores são facilitadores no desenvolvimento de atividades enquanto os sistemas embarcados são catalizadores na realização da tarefa. Ou seja, enquanto o foco dos computadores é auxiliar o usuário oferecendo-lhe diversas maneiras práticas de realizar a mesma tarefa com mais

eficiência, o objetivo dos sistemas embarcados não é ajudar o usuário e sim realizar a própria tarefa.

Os computadores oferecem diversas ferramentas de edição de texto, um usuário que necessite desenvolver um documento textual contará com o auxílio do computador e este último o auxiliará a produzir um texto da melhor maneira possível. Os sistemas embarcados não oferecem opções de como realizar a tarefa para a qual foram criados, simplesmente a executam. O sistema embarcado de um ar condicionado não ajudará o usuário a refrigerar o ambiente de diversas maneiras, simplesmente, assim que acionado, diminuirá ou aumentará a temperatura do recinto o mais eficientemente possível.

O conceito de tarefa e método leva a outra definição de que, sistemas embarcados são todos os produtos em que a parte computacional do sistema não é o principal do produto, ou seja, sistemas embarcados servem para deixar o aparelho ou equipamento ainda melhor em sua atividade. As pessoas compram um equipamento não pelo sistema que está embarcado, mas pelas funcionalidades que ele oferece. Como foi dito, o objetivo de sistemas embarcados não é a maneira de se realizar uma dada tarefa, mas tornar a realização da tarefa otimizada. Devido a grande área em que sistemas embarcados podem atuar a diversidade de possibilidades para o desenvolvimento torna a tarefa inicial de projetar o sistema um grande desafio, como já foi dito, deve-se tratar as limitações de hardware e software, escolher bem o microcontrolador e definir bem os objetivos para o qual o sistema está sendo criado. A partir de tantos pontos a serem observados e da imensa diversidade na implementação surgiram as plataformas que visam facilitar o desenvolvimento dos sistemas embarcados.

Segundo Bosa (2009) uma plataforma é uma arquitetura de hardware e software específica para determinado domínio da aplicação. Elas são parametrizáveis quanto ao número e definição de componentes. Uma plataforma possui todos os dispositivos para o desenvolvimento de um sistema e usá-las é muito benéfico, já que, normalmente já estão de acordo com normas e padrões, deve-se, portanto, estudar e escolher uma plataforma que satisfaça as condições do projeto.

O projeto de um sistema embarcado pode incorporar sofisticadas funcionalidades, como algoritmos complexos, no caso de um motor de automóvel as operações realizadas pelo microprocessador são complicadas, existem funções de filtragem para otimizar o desempenho do carro e ainda minimizar a poluição e utilização de combustível. Outra funcionalidade interessante que pode estar integrada ao projeto são as interfaces, são utilizados com frequência

microprocessadores para controlar a interface como usuário, estas podem incluir menus e muitas outras opções, um exemplo que pode ser citado é o equipamento de navegação conhecido como GPS (Sistema de Posicionamento Global, do Inglês *Global Positioning System*) (WOLF, 2001).

### 3.2. Metodologia de Desenvolvimento

Uma metodologia pode auxiliar o desenvolvedor de sistemas embarcados. Existe um fluxo que a informação deve seguir, na Figura 13 pode ser visualizada uma metodologia típica para o desenvolvimento de projeto de sistemas embarcados criada por Carro e Wagner (2003).

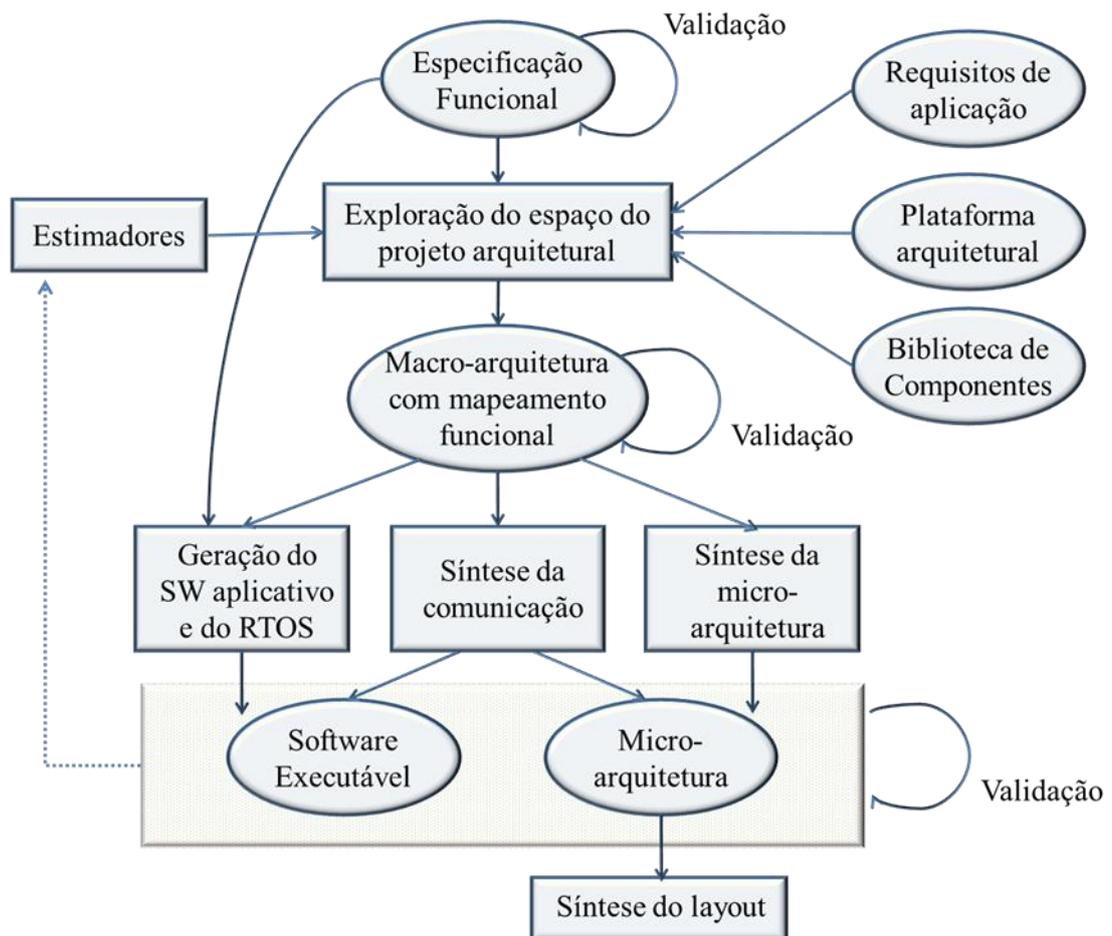


Figura 13: Fluxo de um Projeto de Sistema Embarcado.

Extraído de (Carro e Wagner, 2003).

Tudo se inicia pela especificação funcional, processo de definição das funções do sistema, ou seja, a delimitação da tarefa que será executada pelo sistema embarcado. É válido ressaltar que sistemas embarcados são dedicados a um objetivo único, portanto esse processo de escolha das funções que serão executadas não é trivial, pois deve-se levar em conta as prio-

ridades do aparelho que o sistema gerenciará e o público-alvo para o qual está sendo desenvolvido. Esta etapa inicial deve ser feita em uma linguagem formal e deve ser o mais abstrata possível, nesse momento não podem ser valorizadas ou descritas as implementações do sistema, pois o foco é na demarcação do alvo. Deve ao final passar pelo processo de validação, se esse for positivo, então é continuado o fluxo.

Em seguida é realizada uma exploração do espaço do projeto arquitetural, esse momento leva em consideração muitas informações, a validação da especificação funcional, os requisitos da aplicação, a plataforma arquitetural e as bibliotecas de componentes. Esses conjuntos de informação servem como auxílio para a determinação da arquitetura ideal para o projeto.

Se os estimadores indicarem um resultado aceitável para os projetistas é idealizada uma macro-arquitetura, esta terá uma descrição rápida de quais e como serão cada um dos componentes que integrará o sistema, tanto de hardware, como de software. Juntamente com isto, também é formulado um mapa de todas as funções do sistema, é a partir desse mapeamento que o desenvolvimento do software pode ser iniciado. Assim que esta etapa passar com sucesso pela validação se tornará possível desenvolver os componentes e integrá-los. Essa ordem entre as ações é ilustrada no esquema da Figura 14.

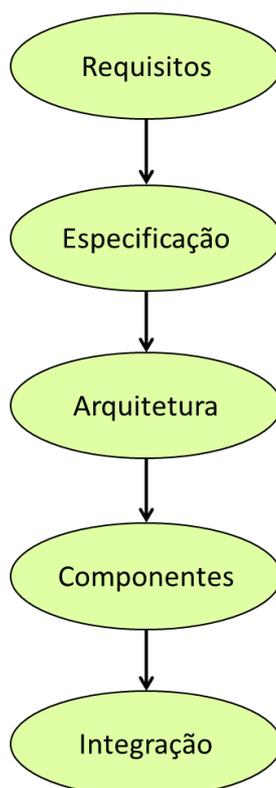


Figura 14: Desenvolvimento de Projeto em Níveis.

Adaptada de (Wolf, 2001).

A fase de arquitetura é muito essencial, um bom projeto do mapeamento funcional e da macro-arquitetura torna o desenvolvimento dos componentes mais simples, tornando harmoniosa a integração destes. Os componentes somente serão iniciados se a validação da fase anterior for aceita não somente pelo cliente, mas também pela equipe técnica.

A etapa de desenvolvimento dos componentes acontece de maneira paralela, ou seja, o software, o hardware e a comunicação serão criados e implementados juntos. Esta etapa, portanto, é subdividida em três processos de criação: a geração do software e aplicativo de tempo real, se este for necessário à aplicação; a síntese de comunicação, como o sistema irá se comunicar com o ambiente e como será a comunicação dos componentes entre si; e, por fim, a síntese da micro-arquitetura, ou mais comumente chamado, desenvolvimento e escolha do hardware.

A partir de todo esse fluxo de processo para desenvolver um sistema embarcado é possível visualizar quais informações são necessárias e em que momento, este processo torna o desenvolvimento de sistemas algo organizado e claro. Utilizando uma metodologia para servir de auxílio torna-se mais simples identificar os erros e as possíveis melhorias, mantendo o desenvolvimento de sistemas embarcados um processo otimizado e alcançável.

Os custos tecnológicos para o desenvolvimento de sistemas embarcados são considerados baixos, pois, o avanço da tecnologia em áreas como a engenharia elétrica e a micro eletrônica tornou possível a miniaturização e o barateamento do hardware para sistemas computacionais, dando a oportunidade de expansão aos sistemas embarcados (VIANNA; MACHADO, 2009). A partir disso torna-se plausível a multiplicação de sua utilização no meio da sociedade.

Levando-se em consideração que os sistemas embarcados são imprescindíveis em muitas atividades humanas, e por serem a maioria das tecnologias que circundam o homem, acredita-se que precisam continuar evoluindo, buscando novas soluções e resultados com as mesmas condições.

Os sistemas embarcados podem trazer lucros através de economias, pode-se citar, por exemplo, um sistema que controla toda a rede elétrica de um prédio desligando e ligando luzes e equipamentos. Essa economia também pode ocorrer nas casas, pois existe a possibilidade de conexão de um ambiente virtual com o real e a intermediação ser realizada pelo sistema embarcado, conforme o exemplo ilustrado na Figura 15. O sistema pode receber comandos do

ambiente virtual e realiza-los no real, também pode perceber que não há ninguém no ambiente e apagar as luzes, por exemplo.

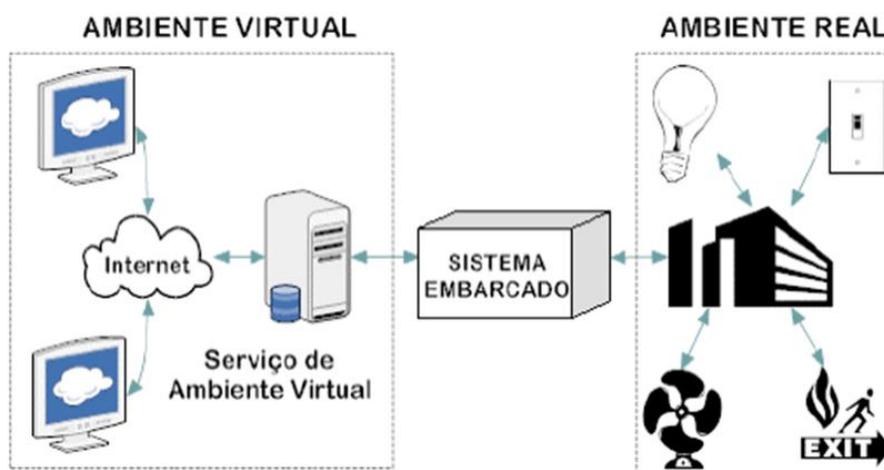


Figura 15: Intermediação feita pelo Sistema Embarcado entre Ambiente Virtual e Real.

Extraído de (Vianna; Machado, 2009).

A evolução da microeletrônica e o barateamento das CPUs (Unidade Central de Processamento, do Inglês, *Central Processing Unit*) viabilizaram o emprego de sistemas computadorizados nos diversos equipamentos. Os cérebros desses aparelhos modernos são os computadores que estão neles embutidos (CAIXETA, 2006). Eles são a excelente combinação de hardware e software para executar uma tarefa específica e este fato é um dos grandes diferenciais do sistema embarcado para outros sistemas computacionais.

Pelo benefício na integração da computação com os eletrodomésticos comuns muito começou a ser estudado para que em ambientes monitorados por sistemas embarcados estes pudessem não só controlar, mas também interferir no ambiente. Foi dito que a conexão entre os ambientes virtuais e reais pode ser interessante do ponto de vista financeiro e isso é um fato, a maneira como isso é realizado é através da conexão não apenas de entradas aos sistemas embarcados, mas também saídas para que possa haver intervenção por parte do sistema ao ambiente.

Da mesma maneira que sistemas computacionais possuem entrada, processamento e saída, os sistemas embarcados também, e estas entradas podem vir de um ambiente, de sensores ou de outro sistema, enquanto as saídas podem ser emissões de sinais ou ainda podem estar conectadas a atuadores. Estes atuadores são os que influenciam verdadeiramente no espaço delimitado, são os que vão atuar sobre o ambiente. Cada sistema embarcado possui sua apli-

cação, portanto cada um pode ou não ter atuadores, como também os atuadores podem ser diferentes de um sistema para outro.

Na Figura 16 pode ser visto um sistema embarcado que possui sensores e atuadores. Ele recebe informações de um ambiente virtual e também dos sensores e reage através dos atuadores. O sistema deve ser capaz de gerenciar instruções de acionamento de dispositivos recebidas do servidor, executar os acionamentos, verificar possíveis falhas no acionamento, checar variáveis de ambiente e informar ao servidor as modificações encontradas (COSTA; VIANNA, 2008).

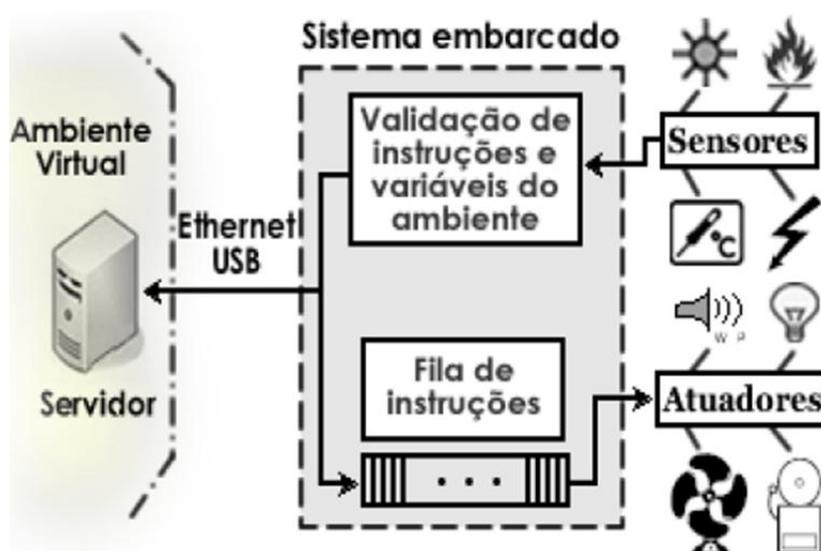


Figura 16: Sistema Conectado a Sensores e Atuadores.

Extraído de (Costa; Vianna, 2008).

Nesse caso o sistema é mais complexo do que um sistema embarcado de um ar condicionado, por exemplo, isso é apenas uma comparação para que possa ser percebido como sistemas embarcados estão em todo o lugar e podem ter inúmeras aplicações. As entradas e saídas do seu processamento podem trazer realmente ótimos resultados, mas para que isso ocorra é fundamental que tenha ocorrido um bom planejamento do sistema, tanto na parte de software como de hardware.

Um sistema embarcado é formado por uma entrada, uma CPU e uma saída. Essa unidade de processamento pode ser um microcontrolador, e a escolha desse microcontrolador é muito importante para o sistema embarcado, uma vez que o próprio microcontrolador possui desempenho, que muitas vezes, é suficiente a determinados aparelhos com funções específicas. Porém, na maioria dos casos, um sistema embarcado é desenvolvido para aumentar a efi-

ciência e eficácia do aparelho no desempenho da função específica. Na concepção de um sistema embarcado, procura-se adotar um microcontrolador cujas portas de comunicação sejam suficientes em número e compatibilidade com os outros componentes controlados no sistema (SANTOS, 2010).

A escolha do microcontrolador deve ser determinada de acordo com os requisitos do sistema a ser desenvolvido ainda no início do desenvolvimento do projeto. O tratamento de erros, que percorre todo o desenvolvimento do sistema embarcado, deve ser considerado, principalmente, quando este está lidando diretamente com a vida de alguém.

Pode-se citar um sistema embarcado que vai atuar em um marca-passo, este não pode ter erro algum, já que qualquer espécie de falha pode gerar consequências graves e até irreversíveis, a qualidade do sistema deve ser altíssima. Por outro lado, um jogo utilizado pelo celular não pode ter erros para não frustrar o usuário, porém suas consequências são bem mais amenas. A complexidade envolvida, os prazos definidos e os custos são totalmente diferentes nas situações citadas, o que influencia em escolhas diferentes para cada aplicação. Enquanto um deve ter mais qualidade e pode ter maior preço, o outro pode ter qualidade inferior, porém preço muito mais acessível (TAURION, 2005).

Se o sistema tiver outro propósito como troca de informação com as pessoas, o projeto deve ser bem diferente, será através de interfaces como teclados ou displays que o usuário poderá interagir com o programa. Mesmo havendo interação, por se tratar de um sistema embarcado nunca o usuário terá acesso direto à aplicação que foi embutida no dispositivo.

Um sistema embarcado não possui muita flexibilidade, pois é produzido para um objetivo único, entretanto, mesmo possuindo tal característica as atualizações são muito desejadas. Isso ocorre pela aceitação do fato de que a tecnologia irá evoluir e o sistema deve aceitar novas funções para não ser rotulado como ultrapassado. Assim torna-se importante que este tipo de sistema seja flexível para o uso de novas versões.

### **3.3. Sistemas Embarcados e Sistemas de Tempo Real**

Os sistemas embarcados possuem modos de funcionamentos, isso está relacionado à como eles vão desempenhar suas funções no meio para o qual foram projetados. Este ponto é de fundamental importância na avaliação de como desenvolver o sistema para determinada

aplicação. Existem dois modos de funcionamento: o Reativo e o Controle em Tempo Real (FARINES; FRAGA; OLIVEIRA, 2000).

O modo de funcionamento Reativo é quando o sistema espera pelas influências do meio no qual está inserido, ou seja, ele tem seu funcionamento como resposta a eventos externos. Isso pode acontecer enviando algum sinal de rádio para um robô, por exemplo. Nesse caso existe a necessidade da entrada de dados, e o sistema precisa receber e saber interpretar cada um desses dados.

Por esperar a influencia externa não tem limites de tempo de entrada, porém existem os limites de tempo para saída, ou seja, ele precisa apresentar a reação quando ativado, isso deve acontecer, normalmente, após os dados de entrada começaram a ser executados. Os eventos externos podem ser periódicos, como nos controles de *loop*, ou assíncronos, como no caso de pressionar um botão.

O modo de funcionamento de controle em Tempo Real é quando um sistema requer que os resultados cheguem em um prazo específico. Resultados que chegam depois do prazo expirado são inúteis. Pode-se citar como exemplo um robô explorador, que está em uma superfície estranha para desbravamentos científicos, seu papel é coletar informações, mas ao mesmo tempo em que coleta os dados precisa avaliar o ambiente em que se encontra, ele precisa receber informações sobre obstáculos que possivelmente podem atrapalhar seu objetivo, como uma rocha que impeça o caminho, se ele receber a informação de que existe uma rocha a frente depois de já ter colidido com ela, a informação foi inútil, pois deveria ter sido recebida e interpretada antes, para que o impacto não acontecesse.

Esta noção de programação dentro de prazos é simples, mas também exigente, isto pode ser afirmado, pois se a resposta certa for recebida fora do prazo ela deixa de estar correta. Ou seja, se o programa não produz a saída requerida dentro do prazo, então o programa não funciona, mesmo que a saída produzida é funcionalmente correta, portanto esse conceito torna o sistema rígido e válido (WOLF, 2001).

A computação é dita de tempo real quando realiza a gerencia de recursos com o objetivo de atingir o término da operação diante de uma restrição de tempo. Assim o tempo faz parte da lógica do sistema, o tempo real e físico, absoluto ou relativo. O sistema precisa ter a garantia de que as tarefas serão executadas dentro das restrições temporais, independente se o

sistema estiver sobrecarregado, isso se chama determinismo, um dos pontos fundamentais em um sistema de tempo real. As tarefas podem variar em prioridade e também em periodicidade, ou seja, algumas tarefas vão ser executadas de maneira síncrona, com uma frequência já estabelecida, enquanto outras são esporádicas, ou seja, assíncronas. Para tratar de tantos requisitos e ainda conseguir garantir o determinismo, a maioria dos sistemas embarcados de tempo real é baseada em multitarefa que é o conceito de várias tarefas concorrendo entre si (MAETA, 2001).

Em algumas aplicações a utilização de sistemas com multitarefa é essencial, como, por exemplo, no caso de sistemas multimídia. Se as tarefas não conseguirem executar de forma sincronizada no prazo estipulado a percepção de toda a apresentação pode estar comprometida. O conceito de multitarefa determina que várias atividades precisam ocorrer ao mesmo tempo, para que todas executem dentro de seus prazos o sistema deve controlá-las, a execução deve ser concorrente mesmo que as taxas de limite de tempo sejam diferente.

O requisito de tempo real também é muito utilizado na indústria automobilística, como no caso do sistema controlador de velocidade de cruzeiro em automóvel. Este sistema regula a velocidade do carro percebendo as rotações do eixo dianteiro, o velocímetro, a chave de controle, o acelerador e o pedal do freio, claramente ele tem restrições temporais entre outras como desempenho e segurança (SHAW, 2001).

Os sistemas de tempo real podem ser de dois tipos: críticos, que é quando as tarefas devem responder em um prazo específico, porque se não o fizerem podem levar a sérias consequências, normalmente ligado à vida de pessoas. E, o não crítico (brando), que é quando as tarefas podem ser executadas no limite de tempo, mas se não o fizerem não causarão consequências graves.

No gráfico da Figura 17 é possível visualizar o contraste entre os sistemas de tempo real crítico, não-crítico e os sistemas operacionais de propósito geral com relação a utilidade da tarefa se o prazo for ultrapassado e a tarefa não ter sido cumprida. Como já foi explanado sistemas de tempo real são aqueles que possuem prazos de execução para as tarefas, nesta categoria existe uma subdivisão que os organiza em críticos e não-críticos. Para os críticos a tarefa fora do prazo é inútil, enquanto que para os não-críticos ela ainda pode ter utilidade, mesmo que reduzida.

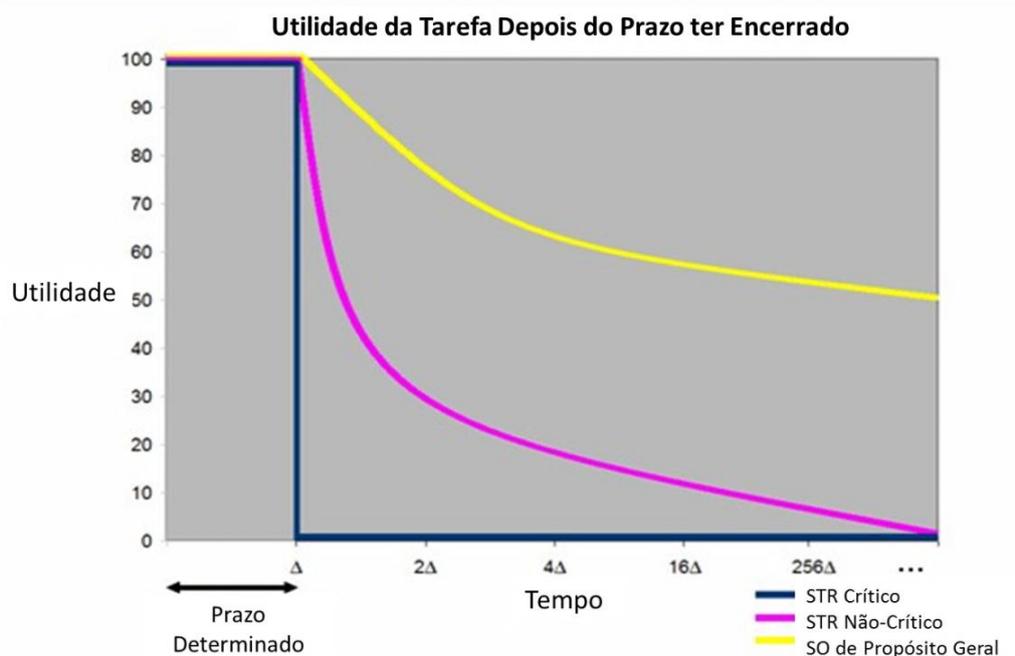


Figura 17: Gráfico de Utilidade de Tarefas mediante Prazo.

Extraído de (National Instruments, 2011).

Para o sistema de tempo real crítico a partir do momento em que o prazo foi ultrapassado a utilidade da tarefa passa a ser zero, ou seja, deixa de ter utilidade já que o tempo limite foi ultrapassado. No caso de sistemas de tempo real não-críticos a tarefa perderá a utilidade conforme o tempo de atraso for aumentando com relação ao prazo principal.

Atualmente o número de sistemas com requisitos de tempo real tem crescido muito, existem sistemas mais simples como os sistemas embarcados em utilidades domésticas, tais como videocassetes e micro-ondas, como também na outra extremidade do quesito complexidade os sistemas militares de defesa e de controle de tráfego aéreo e ferroviário. Muitos equipamentos eletromecânicos incorporam sistemas embarcados para terem sua funcionalidade aumentada, agregando valor e exibindo um diferencial diante dos competidores (CAIXETA, 2006).

Todos os sistemas embarcados, independente do seu modo de funcionamento devem ser confiáveis. Segundo Marwedel (2003) esses sistemas devem ser a prova de falhas e já que existe a interação com o meio o impacto deve ser sempre positivo. Para considerar o sistema confiável quatro itens fundamentais devem ser observados:

1. Fiabilidade: define a probabilidade de que o sistema não irá falhar;

2. Manutenibilidade: também pode ser entendida como Recuperação, já que se refere à probabilidade de que uma falha no sistema será reparada em certo intervalo de tempo.
3. Disponibilidade: o sistema deve estar sempre disponível. Para aumentar esse quesito deve-se elevar os dois anteriores.
4. Segurança: este termo possui dois aspectos, primeiramente que se o sistema falhar não causará dano algum ao ambiente em que está, ou seja, o meio deve estar sempre seguro. E posteriormente, que o sistema deve manter seus dados com confidencialidade e garantir a comunicação com autenticidade.

Um sistema que engloba esses pontos torna-se mais interessante também do ponto de vista econômico, já que representam vantagens para os clientes. Um equipamento que possua um sistema embarcado considerado confiável é ainda mais valorizado. Os sistemas embarcados devem atender a tarefa para os quais foram projetados, seu sistema de controle deve identificar e atingir o objetivo por completo, porém vale ressaltar que a confiabilidade do sistema interfere se o resultado atingido foi com sucesso ou apenas com suficiência.

Os sistemas embarcados são muito importantes no meio social, representam a maior parte da tecnologia que o homem tem contato diariamente, apenas por isso já é uma área altamente relevante e interessante. Levando em conta que existem meios de facilitar seu projeto através de metodologias e que seu desenvolvimento tem um baixo custo torna-se um campo ainda mais atraente. Porém todos esses pontos positivos não devem permitir que um sistema por pressa, seja utilizado sem ser confiável mantendo inseguro o ambiente em que se encontra, portanto, apesar dos pontos para o sistema atingir um estado confiável tornarem mais complexo e lento seu desenvolvimento é um ponto essencial e que deve ser mantido.

## 4 SISTEMA DE CONTROLE EMBARCADO PARA O FUTEBOL DE ROBÔS

Este capítulo descreve o sistema de controle projetado para a Equipe de Futebol de Robôs Araranguá Intruders. Para um melhor entendimento do capítulo será utilizada uma abordagem *bottom-up*, ou seja, de baixo para cima, sendo inicialmente explicado o hardware que compõem os robôs. Em seguida será explicado o sistema de controle.

### 4.1 Arquitetura de Hardware dos Robôs da Equipe Araranguá Intruders

O projeto dos robôs da equipe Araranguá Intruders conta com 6 (seis) robôs holonômicos compostos por 4 (quatro) motores de corrente contínua, 2 (dois) microcontroladores, 1 (um) rádio transceptor, sistema de chute e domínio de bola e sensores. Esta configuração de hardware foi a proposta para ser utilizada em 2012 na X Competição Brasileira de Robótica em Fortaleza – CE. A Figura 18 ilustra o projeto da arquitetura de hardware de cada robô da equipe Intruders.

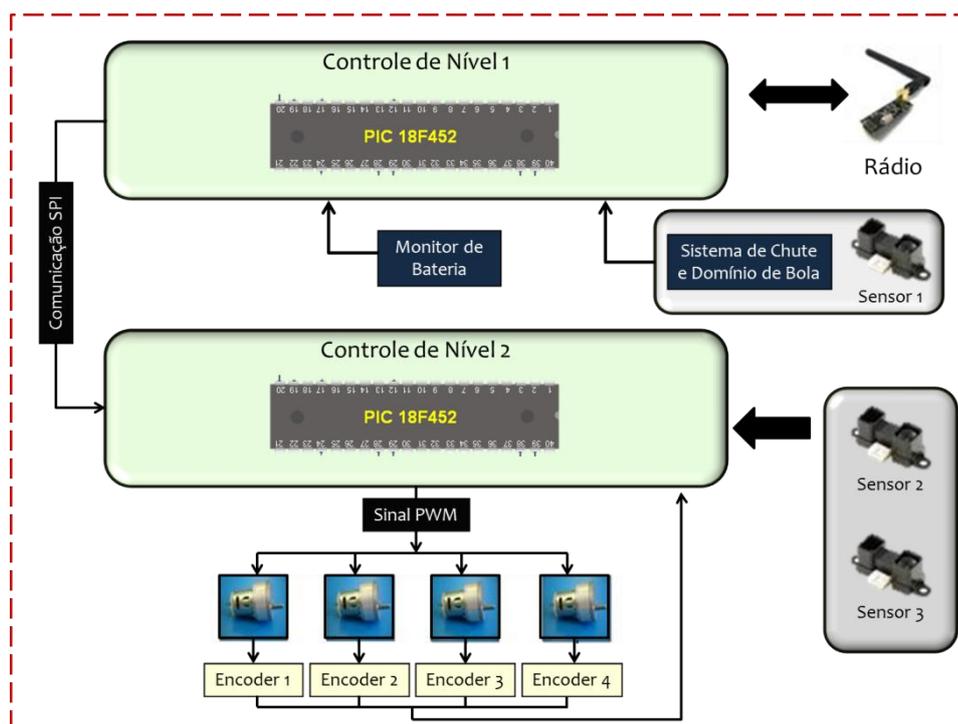


Figura 18: Arquitetura de hardware dos robôs da equipe Intruders.

A arquitetura de hardware é composta por 2 (dois) níveis de controle, sendo cada nível controlado por um microcontrolador PIC modelo 18F452. Essa divisão permite aumentar a capacidade de processamento e o tempo de resposta.

O primeiro nível é composto pelo sistema de comunicação (rádio), o sistema de chute e domínio de bola e a comunicação com o microcontrolador do segundo nível, intitulado de microcontrolador escravo.

O *sistema de comunicação* é composto por um módulo de rádio transceptor nRF24L01 de 2.4 GHz da Nordic responsável pelas comunicações entre o robô e o computador técnico e vice versa. O fato de haverem muitas mensagens foi um dos principais motivos pelo qual foi escolhido utilizar dois microcontroladores ao invés de apenas um. O objetivo é fazer com que todas as mensagens sejam processadas o quanto antes possível, devido ao processamento de uma mensagem ser uma atividade de alta prioridade, muitas outras funções podem ser deixadas de ser executadas.

O *sistema de chute e domínio de bola* é composto por um solenoide que serve para “chutar a bola”, um motor de corrente contínua acoplado a um rolete que serve para manter a bola dominada e um sensor de infravermelho que acusa a presença da bola próxima ao rolete de domínio. O microcontrolador, ao receber o sinal do sensor de infravermelho, aciona o rolete de domínio de bola.

Outra função que é realizada pelo primeiro nível é a comunicação com o microcontrolador do segundo nível, esta é uma comunicação SPI (Interface Serial Periférica, do Inglês, *Serial Peripheral Interface*) e o conteúdo da mensagem é a direção e a velocidade linear de cada roda acoplada aos motores de corrente contínua.

O segundo nível é composto pelo monitor de bateria, controle dos motores e sensores de infravermelho.

O *monitor de bateria* avalia periodicamente o nível da carga da bateria. Essa função foi adicionada, pois existem robôs que gastam mais bateria do que outros durante a partida. Com o monitoramento da bateria o robô pode avisar o computador técnico para ser substituído ou trocar de função, já que normalmente todos os robôs possuem os mesmos itens de hardware. Com isto é possível obter informações do robô sem precisar tirá-lo de campo, já que um dos pontos que aumenta a competitividade é possuir o time completo em campo para jogar. Os robôs da equipe Intruders utilizam baterias de LiPo de 3 (três) células de 11.1V 2200 mAh e 14.8V 5000 mAh.

O *controle dos motores* realiza a ativação dos motores individualmente. O microcontrolador do primeiro nível repassa ao segundo nível a informação referente a direção e a velocidade de cada motor. Essa informação é um valor que determina a variação PWM (Modulação por Largura de Pulso, do Inglês, *Pulse Width Modulation*) de cada motor. Como os motores não são todos iguais há casos de variação da rota, por exemplo, uma roda pode girar mais do que a outra e o robô não ir para a direção que deveria. Para tratar este problema cada motor possui um *encoder*, um sensor responsável por enviar o número de rotações do motor para o microcontrolador.

O controle dos motores é um sistema em malha fechada, ou seja, quando o microcontrolador envia a velocidade linear para cada motor ele não finaliza o seu controle, pelo contrário, ele aguarda confirmações do *encoder*. Como pode ser visto na Figura 19 o comando é passado aos motores, os *encoders*, por sua vez, monitoram a rotação das rotas e repassam essa informação ao microcontrolador, que conclui se a rota está de acordo com o que foi definido pelo técnico. Portanto quando se trata do controle dos motores, o microcontrolador do segundo nível recebe informação do microcontrolador do primeiro nível e também dos *encoders*.

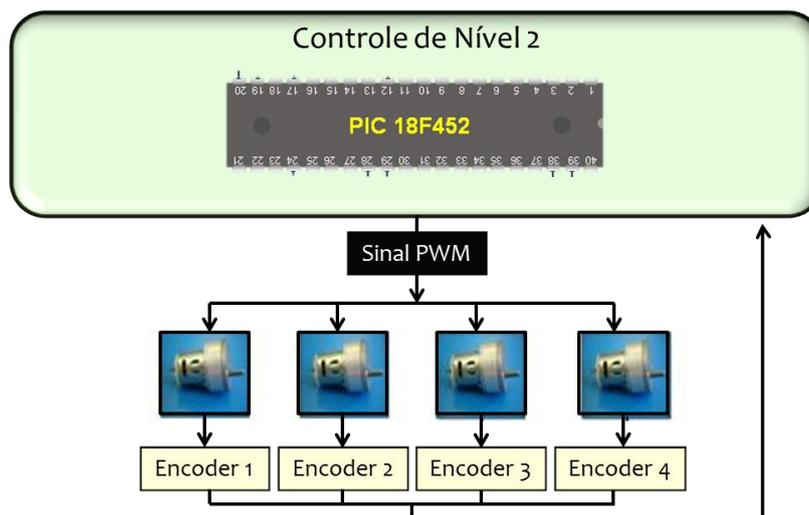


Figura 19: Vista expandida do controle dos motores.

Os *sensores de infravermelho* localizados frontalmente são uma das estratégias escolhidas para diminuir o número de troca de mensagens entre o robô e o computador técnico. Existe uma situação clássica em que o robô está indo em frente e existe um obstáculo adiante, ao invés do robô esperar pela mensagem de que deve desviar, ele mesmo pode chegar a esta conclusão através dos sensores frontais.

Se o computador técnico não possuir a responsabilidade de desvios pode concentrar suas operações na realização de gols e em defesas mais estratégicas. O principal objetivo é que o robô não venha a sofrer nenhuma colisão que possa prejudicar seu estado. O microcontrolador do segundo nível controla essa função, pois se perceber que pode colidir acionará os motores para fazer um desvio de rota, assim o próprio sistema gera um comando que se executado evita a possível batida, tornando esse controle de rota mais rápido e eficiente.

#### 4.2 Arquitetura de Software dos Robôs da Equipe Araranguá Intruders

Um Sistema de Controle coordena as ações realizadas pelo robô. Toda e qualquer informação, seja recebida ou enviada passa pelo sistema de controle. Na maioria dos casos os sistemas embarcados presentes nos robôs das equipes de futebol de robôs são apenas reativos, reagindo aos comandos recebidos do técnico, porém, o sistema de controle proposto neste trabalho possui uma arquitetura híbrida, logo algumas diferenças serão significativas.

O controle de software possui várias maneiras de ser desenvolvido, como foi detalhado na Seção 3.2 do Capítulo 3. Para o projeto do sistema de controle embarcado dos robôs da Equipe Araranguá Intruders a metodologia escolhida foi a desenvolvida por Carro e Wagner (2003). O sistema é composto pela combinação entre interrupções e tarefas de tempo real.

Para um melhor detalhamento do sistema de controle optou-se por compatibilizar a arquitetura de software com a arquitetura de hardware descrita na Seção 4.1. Desta forma as funções do sistema de controle foram separadas em dois níveis. A Figura 20 ilustra o sistema de controle projetado em que é possível perceber a divisão das funcionalidades.

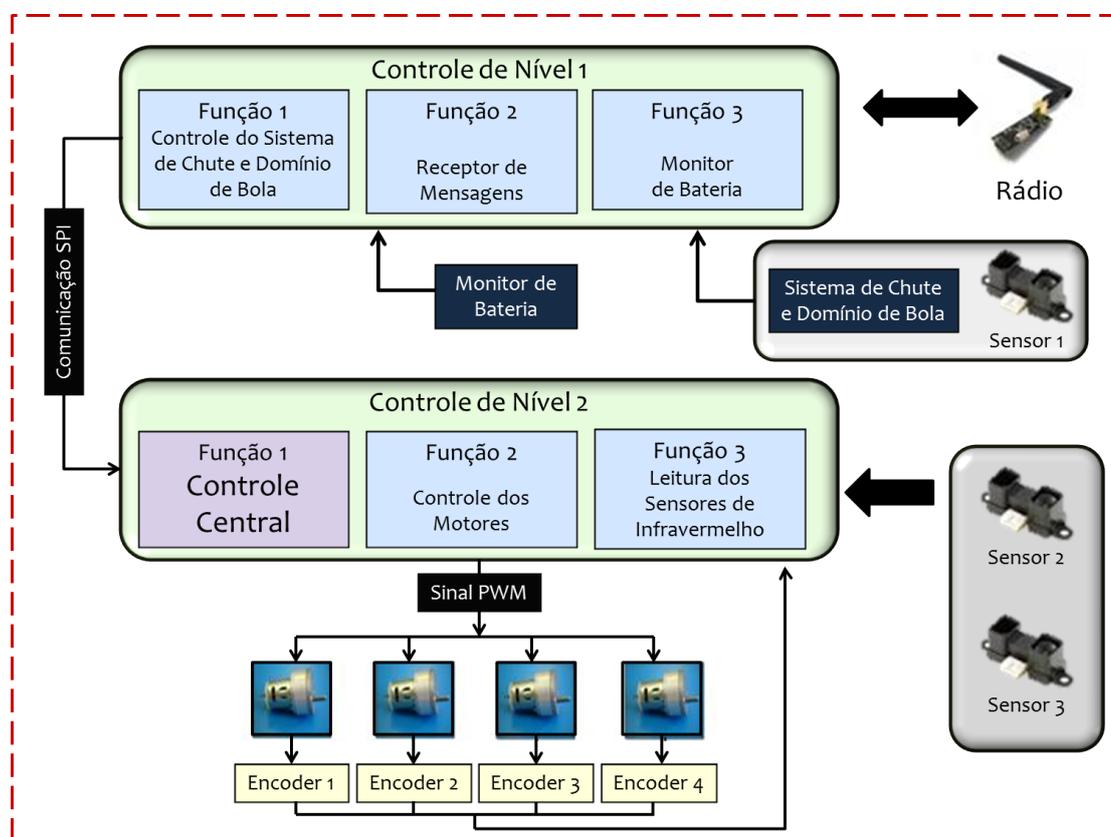


Figura 20: Arquitetura de Software dos robôs da equipe Araranguá Intruders.

O *Controle Nível 1* possui duas tarefas principais, a Função 1 é responsável pelo Controle do Sistema de Chute e Domínio de Bola e a Função 2 responsável por enviar e receber mensagens. Além dessas duas tarefas o Nível 1 do sistema é responsável por repassar informações referente a direção e a velocidade linear de cada roda para o Nível 2 via comunicação SPI.

O *Controle Nível 2* é formado por quatro tarefas, a Função 1 é o Monitor de Bateria, a Função 2 é o Controle dos Motores, a Função 3 é a Leitura dos Sensores de Infravermelho e a Função 4 é o Controle Central. Esta última Função possui a mais alta prioridade no sistema e a partir dela as outras funcionalidades do sistema são ativadas.

No Controle Nível 1 a Função 2 de Recepção de Mensagens é a que ocupa a maior parte do processamento, e as outras funcionalidades acontecem a partir dela. Para o sistema de

controle será uma tarefa de tempo real, pois existe um prazo para o comando ser recebido, interpretado e comunicado, para que depois seja atendido.

As mensagens estarão chegando a todo o momento via rádio, então o primeiro passo é conhecer em que padrão de formatação a mensagem será enviada do computador técnico. Cada equipe projeta o seu próprio protocolo de comunicação, no caso da Equipe Araranguá Intruders o primeiro dado passado é a identificação do robô, porque os comandos são diferentes para cada um, portanto, por mais que o robô capte outras mensagens só pode executar a que for definida para ele mesmo. Outras informações que vão presente na mensagem é a direção e a velocidade linear de cada um dos motores.

A partir do momento que as mensagens são recebidas pelo sistema de comunicação devem ser imediatamente interpretadas, se a identificação não for a do robô, a mensagem é ignorada, se for para o robô em questão ela prossegue na interpretação. Entenda-se por comando a parte da mensagem destinada a determinar a ação do robô, portanto, dependendo do comando a tarefa reagirá de maneiras distintas.

Se o comando for para algum direcionamento a Função 2 repassa essa informação via comunicação SPI para o Controle Nível 2, porém, se for para chutar a Função 2 manda a informação para a Função 1, que é o controle do sistema de chute e domínio de bola.

O domínio de bola é muito importante para todos os robôs e a maioria das equipes já possuem esse dispositivo, pois é a única maneira que o robô tem para conseguir conduzir a bola, portanto, deve ser utilizado sempre que possível. Não é interessante manter o domínio ligado o tempo todo, pelo gasto excessivo de bateria, então achar o equilíbrio para essa funcionalidade é fundamental, e o encontrado nesse projeto foi o uso de uma interrupção, ou seja, assim que o sensor 1 indicar a presença da bola a interrupção é acionada e então o motor que controla o rolete de domínio é ligado.

A partir do momento em que a bola está dominada o computador técnico pode escolher o momento adequado para chutar, portanto este é um comando recebido pelo rádio. Quando a mensagem de chute for interpretada o microcontrolador ativará o sistema de chute.

No controle Nível 2 a Função 1, *Controle do Nível de Bateria*, será uma tarefa que monitorará periodicamente a bateria. Foi escolhido designar essa função como uma tarefa, pois a bateria não precisa ser monitorada constantemente, apenas em alguns momentos, assim a tarefa terá um intervalo grande entre uma verificação e outra.

A Função 2, *Controle dos Motores*, é uma função auxiliar que levará em conta duas entradas, o comando que será passado pela chamada da função e a informação vinda dos *encoders*. Inicialmente ocorrerá a interpretação do comando para o correto acionamento dos motores. As informações lidas a partir dos *encoders* são consideradas para o sistema fazer eventuais correções de rotas.

Fazer o controle da rota é identificar, através das informações dos *encoders*, se as rodas estão na rotação correta, ou seja, a rotação que deveriam estar para executar o comando desejado. Portanto, a função deve calcular o que fazer para corrigir a rota, reenviando a informação aos motores para que a execução do comando seja adequada.

Como a velocidade dos motores não será necessariamente a mesma em todos os momentos, o meio mais rápido de controlar a rota é acelerar ou desacelerar a roda que está realizando o desvio. Por exemplo, se o robô deveria ir em frente e na prática está em uma linha diagonal à direita, é porque a roda da esquerda está girando com mais velocidade que a da direita, assim deve-se, ou acelerar a roda direita ou desacelerar a esquerda.

Para resolver essa situação será utilizado o controle PID (Proporcional Integral Derivativo) de maneira que o valor desejado é comparado com o valor da atuação da variável do processo atual. O que indica que ele controla as rodas, como foi descrito anteriormente, comparando a saída atual com a desejada pelo Controle de Nível 2.

Esta escolha de aceleração ou desaceleração é definida a partir dos limites abrangidos pela velocidade, isto ocorre porque se a roda direita já está em rotação máxima e ainda está mais devagar que a esquerda o possível é a desaceleração do lado esquerdo. A ação comandada pela função *Controle dos Motores* continuará sendo a mesma, porém a velocidade repassada para cada motor sofrerá as alterações segundo o que for interpretado dos *encoders*.

A Função 3, *Leitura dos Sensores de Infravermelho*, foi projetada como uma interrupção. O sistema precisa dessa funcionalidade para não depender do computador técnico para desviar dos outros robôs ou outros obstáculos.

A interrupção ocorre assim que os sensores captarem a existência de algum obstáculo a frente, por ser uma interrupção será tratada imediatamente, o comando de desvio para os motores será gerado automaticamente e executado, depois de não haver mais obstáculos o comando enviado pelo computador técnico volta a ser executado. Portanto nesse momento, a informação do robô é executada independente do controle do sistema estrategista, por mais

que ele tenha enviado o comando de ir para frente, o robô atenderá primeiramente seus sensores, para depois executar o comando.

Para evitar que o robô entenda a bola como um obstáculo e acabe desviando dela ao invés de dominá-la e executar a jogada de acordo com o computador técnico, foi definido que sempre que a bola passasse na frente dos robôs o computador técnico enviaria uma mensagem alertando sobre essa situação. Portanto essa interrupção gerada pelos sensores só seria tratada se não houvesse a mensagem referindo-se a proximidade da bola.

A Função 4, *Controle Central*, é uma função especial que rege todo o Controle Nível 2, é ela que receberá as mensagens vindas da comunicação SPI e chamará a função de Controle dos Motores, também é ela que tratará a interrupção vinda da Leitura dos Sensores de Infravermelho. A Função 4 somente sai de execução quando precisa ceder processamento para o Monitor de Bateria.

O Controle Central é o responsável por chamar a Função 2 e passará para ela o que deve ser feito. O que deve ser feito depende da informação que foi recebida da comunicação com o Controle Nível 1, porém caso a Função 3 seja ativada, a informação que será passada para a Função 2 será gerada pelo próprio Controle Central para desviar do obstáculo encontrado. A Função 4 torna o robô mais ativo dentro de campo, já que tem autonomia para, em certos momentos, decidir não fazer o que o computador técnico mandou, mas sim, executar ações de acordo com as informações percebidas pelo robô.

## 5 ANÁLISE DO SISTEMA DE CONTROLE

Este capítulo analisa o sistema de controle descrito no Capítulo 4. Detalha o desenvolvimento e os resultados atingidos separadamente no Controle de Nível 1 e 2 do sistema. Também faz a comparação entre o sistema ideal e o realizado neste trabalho.

### 5.1. Controle de Nível 1

O controle de Nível 1 é a parte do sistema embarcado que possui arquitetura de software hierárquica, como visto no Capítulo 2, na Seção 2.2. É nessa parte do sistema que ficam as tarefas de tempo real, que controlarão, conforme o Capítulo 4, Seção 4.2, a recepção das mensagens recebidas do Computador Técnico, e o monitoramento da bateria. Maiores detalhes das tarefas podem ser visualizadas na Tabela 3.

Tabela 3: Características das Tarefas de Tempo Real do Sistema Embarcado.

Tarefa	Período	Prazo
<b>Simula_radio</b>	5 ms	5 ms
<b>Monitor_de_bateria</b>	10 s	5 ms

Os nomes expostos na primeira coluna da Tabela 3 são nomes correspondentes aos parâmetros das funções de tempo real do RTOS. Período corresponde ao Rate que é um parâmetro das diretivas do RTOS (Sistema Operacional de Tempo Real, do Inglês *Real Time Operation System*), ele delimita o prazo máximo que a tarefa deve entrar em execução, ou seja, a tarefa *simula\_radio* deve entrar em execução uma vez a cada cinco milissegundos. E a terceira coluna Prazo é correspondente ao parâmetro Max, que indica qual é o tempo máximo que a tarefa ficará executando, quanto tempo ela usará o processador. Mais informações a respeito do RTOS do CCS podem ser vistas no Apêndice I.



sistema emitirá um aviso, se o nível for crítico o sistema embarcado pode chegar ao ponto de não executar alguns comandos vindos do computador técnico.

O sistema de chute está sendo representado por um LED, no caso de um robô real ao invés do acionamento direto do LED quando a mensagem recebida for de chute, a interrupção será ativada e o solenoide será acionado. Quanto ao domínio, ao invés de um botão ativar a interrupção externa, será o sensor de infravermelho, e o que será acionado não será um LED, mas sim um motor que fará girar o rolete de domínio, mantendo a bola próxima.

Quanto ao fluxo da informação no sistema embarcado é importante salientar que, por não existirem no Proteus módulos de rádio que pudessem representar fielmente o sistema, foi utilizada uma função que gera números aleatórios, a partir desses números o sistema carrega uma mensagem e a repassa para outra função para que essa repasse o comando, via comunicação SPI para o Controle Nível 2. Se a mensagem gerada for a de chute, o Controle de Nível 1 se encarregará de ativar o sistema de chute ao invés de repassá-la ao Controle de Nível 2.

A função *simula\_radio* é uma tarefa para o sistema, ela entra em execução uma vez a cada cinco milissegundos. Esse tempo não pode ser considerado pelo sistema como um atraso, visto que o tempo para que o robô execute o último comando é maior do que o período em que a tarefa não está executando.

```

teste2.c
124 void simula_radio()
125 { int aux;
126   aux = rand()%5000;
127   switch (aux){
128     case 0: msg = 0b00011111;
129             break;
130     case 1: msg = 0b00011011; //0b00000110;
131             break;
132     case 2: msg = 0b00010110;
133             break;
134     case 3: msg = 0b00010011;
135             break;
136     case 4: msg = 0b00010010;
137             break;
138     case 5: msg = 0b00010000;
139             break;
140     case 6: msg = 0b00011000;
141             break;}
142   comunicacao_spi();}
143 void comunicacao_spi()
144 { //sistema de chute
145   if (msg==0b00010000)
146   {y=~y;
147     if(y)
148       output_high(pin_b6);
149     else
150       output_low(pin_b6); }
151   else //envia mensagem para Controle Nivel 2
152     spi_write (msg); //escreve na porta de comunicacao spi o valor
153 }

```

Figura 22: Função de Simulação do Rádio e Comunicação SPI.

Na Figura 22 pode ser visualizada a função que simula a execução do rádio e seu tratamento, no caso desta simulação as mensagens já estão prontas e apenas são escolhidas de acordo com o número gerado aleatoriamente. A função *comunicação\_spi* aproxima-se do sistema ideal, visto que filtra a mensagem antes de repassá-la e se não for o comando de chute, a repassa através da função *spi\_write*, como seria em um sistema real.

## 5.2. Controle de Nível 2

O controle de Nível 2 é a parte do sistema embarcado que possui arquitetura de software reativa. É nessa parte do sistema que ficam as relações do robô com o meio, relações provindas das entradas do robô, ou seja, a qualquer contato externo com o ambiente. É importante salientar que esse contato externo pode vir dos sensores ou do comando recebido do Controle Nível 1, que por sua vez é a mensagem interpretada do Computador Técnico.

A Figura 23 ilustra o esquemático da simulação no Proteus do sistema que cabe ao controle do Nível 2. Existe um microcontrolador PIC18F452 que controla, o sinal dos sensores infravermelho que tem o objetivo de evitar colisões, essa funcionalidade é composta por dois botões ligados, respectivamente aos pinos B0 e B1, e o sistema dos atuadores, formado por 8 LEDs, ligados aos pinos D0 a D7.

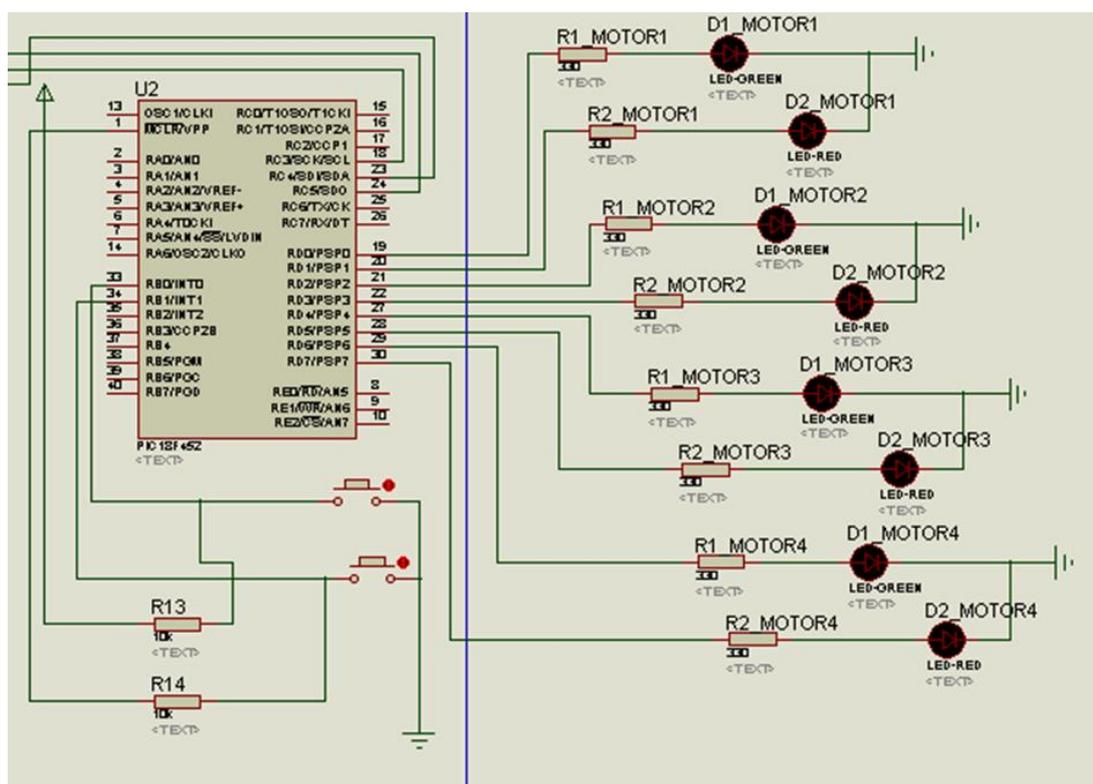


Figura 23: Esquemático da simulação do controle de Nível 2.

Os sensores de infravermelho são os sensores ideais para reconhecimento do ambiente, no caso dessa categoria do futebol de robôs, seria bem adequado o recebimento e interpretação do sinal, visto que o número de colisões seria muito reduzido. Para representar esse sistema foram utilizados dois botões, quando pressionados o sistema os trata da mesma maneira que trataria o sinal vindo do sensor.

Neste protótipo foram utilizados dois botões para simular os dois sensores que ficam nas laterais do robô. Se o sensor da esquerda captar alguma proximidade o sistema Nível 2 irá controlar os atuadores para irem a direita. Se for captado algo no sensor direito os atuadores irão a esquerda, se ambos estiverem pressionados a mensagem repassada a função de Controle Central será ir para trás. Neste caso, o botão, ao ser pressionado, caracteriza a aproximação de algo a cerca de 5 cm de distância no sistema real.

Como no caso da simulação não existe a possibilidade de ambos os botões serem pressionados ao mesmo tempo, foram feitas duas interrupções externas e cada uma muda a mensagem de comando de acordo com a posição do seu sensor. Por terem sido utilizadas interrupções externas, foi analisada a arquitetura do microcontrolador PIC18F452 para verificar quais eram os pinos com essa funcionalidade, atestou-se que eram os pinos da Porta B, por isso os botões foram colocados nos pinos B0 e B1.

Quanto ao sistema dos atuadores, parte-se do princípio que um motor tem sempre dois sentidos para girar a roda à qual está ligado, portanto foi utilizado dois LEDs para representar cada sentido do motor. Por exemplo, os LEDs ligados aos pinos D0 e D1 representam os dois sentidos possíveis de rotação do Motor 1 e isso ocorre na representação de todos os quatro motores. Os motores são distribuídos no robô segundo a Figura 24.

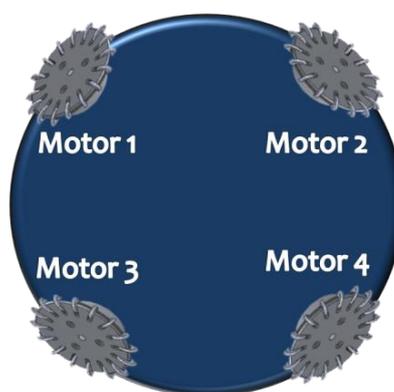


Figura 24: Esquema de posicionamento das rodas e motores no robô.

E para interpretar os resultados da simulação pode-se consultar a Tabela 4, que demonstra a combinação de cores visualizada no simulador e qual a direção representa para um robô real. Foram apresentadas apenas as principais direções, porém é possível desenvolver várias direções, já que as rodas do robô são holonômicas.

Tabela 4: Relação de cores dos LEDS com direção dos motores presentes na simulação.

		Motor 1		Motor 2		Motor 3		Motor 4	
		Frente	Trás	Frente	Trás	Frente	Trás	Frente	Trás
Robô	Frente								
	Trás								
	Direita								
	Esquerda								

O Controle Nível 2 tem seu fluxo de informação relacionado a função *controle\_central*. Isso porque esta função tem como parâmetro o comando que deve ser executado e então dispara as funções de controle dos atuadores de acordo com esse comando.

A função que recebe as mensagens diretamente do Controle Nível 1 é a *le\_msg*, esta função recebe via spi o comando e o interpreta, determinando qual é a ação que deve ser realizada. A partir disso ela chama a função *controle\_central* passando por parâmetro uma variável carregada do valor referente a ação que deve ser efetuada. Por sua vez, a função *controle\_central* entra em execução, interpreta o valor do parâmetro e executa a ação correspondente, como pode ser visualizado na Figura 25.

```

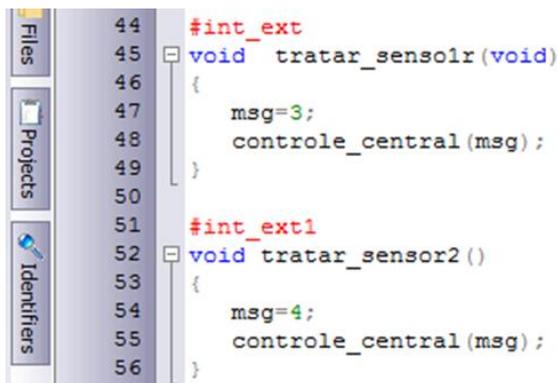
slave.c
100 void controle_central(msg)
101 {
102     switch(msg)
103     {
104         case 1: ir_frente();
105                 break;
106         case 2: ir_atras();
107                 break;
108         case 3: ir_direita();
109                 break;
110         case 4: ir_esquerda();
111                 break;
112         case 5: ir_frente();
113                 break;
114     }
115 }
116 void le_msg()
117 { if( spi_data_is_in() ){
118     dado = spi_read(); }
119
120     if (dado >= 28 && dado <= 31)
121         msg=1;
122     if (dado >= 24 && dado <= 27)
123         msg=2;
124     if (dado >= 20 && dado <= 23)
125         msg=3;
126     if (dado >= 16 && dado <= 19)
127         msg=4;
128     controle_central(msg);
129 }

```

Figura 25: Funções de Controle.

A função *controle\_central* também é executada pelo tratador de interrupções, o que ocorre é que quando uma interrupção é acionada, ela carrega a variável *msg* com o comando que vai satisfazer o tratamento da interrupção e chama a função *controle\_central*. Ou seja, se o sensor esquerdo, ou o botão ligado ao pino B0, for ativado, o robô deve ir para a direita, então a variável *msg* é carregada com o valor 3 e a função *controle\_central* é chamada, como seu parâmetro está selecionado com o valor 3, é chamada a função de controle dos atuadores para ir a direita.

A função *controle\_central* é chamada, portanto, pelos tratadores das interrupções, como pode ser visualizado na Figura 26, ou pela função *le\_msg* que é a recebedora dos comandos via SPI, que por sua vez, são os comandos vindos do computador técnico.



```
44 #int_ext
45 void tratar_sensor1r(void)
46 {
47     msg=3;
48     controle_central(msg);
49 }
50
51 #int_ext1
52 void tratar_sensor2()
53 {
54     msg=4;
55     controle_central(msg);
56 }
```

Figura 26: Tratamento das Interrupções.

Como havia sido descrito no Capítulo 4, os sensores serão tratados com prioridade em relação ao comando do sistema estrategista, visto que esta é uma forma de minimizar o número de mensagens e de proteger o robô. Portanto o controle dos sensores é feito por interrupções e os comandos do técnico são tratados por uma função que é executada dentro do laço *while(1)*.

## 6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

Neste trabalho foi apresentado o sistema de controle embarcado desenvolvido para a Equipe de Futebol de Robôs Araranguá Intruders. O sistema desenvolvido tinha o objetivo de tornar o robô um agente ativo, ou seja, com mais responsabilidades do que comumente acontece no futebol de robôs.

Para tornar o robô um jogador participativo ao invés de apenas reativo foi desenvolvido um sistema de controle embarcado que levasse em consideração não apenas informações advindas do computador técnico, mas também informações internas e externas lidas dos sensores embarcados no robô.

O sistema embarcado visa auxiliar o computador técnico no objetivo de vencer o jogo, para isso assumiu maiores responsabilidades, mas qualquer atitude é realizada não pelo bem de um robô, mas sim da equipe. Dessa maneira pode-se perceber, que mesmo dotado de sensores e tendo um papel mais ativo em campo, o sistema embarcado submete-se à estratégia do computador técnico.

Pelo fato do futebol de robôs ser um ambiente dinâmico, todas as ações e reações dos robôs ocorrem muito rapidamente, por conta disso, o sistema embarcado deve conciliar a quantidade de informações de entrada com as ações de saídas que devem ser condizentes com a estratégia da equipe.

As mensagens recebidas do computador técnico são muito constantes, mesmo assim o sistema embarcado conseguiu gerenciar os comandos e executou cada um deles. Mesmo tratando as informações vindas dos sensores, a capacidade computacional do microcontrolador deu ao robô o tempo suficiente para executar tanto o controle dos sensores e dos atuadores, quanto receber e agir de acordo com as mensagens do computador técnico.

As tarefas de tempo real foram imprescindíveis para a adequada recepção das mensagens, percebe-se que a partir dos resultados, o tempo de envio de mensagem do computador técnico é maior do que o período em que a tarefa não está executando.

O sistema embarcado cumpriu com os requisitos básicos de controle que se resumiram em executar adequadamente as ações recebidas e superou o que era pretendido quanto ao tratamento de sensores. Dotar o robô de mais fontes de informação fortaleceu o sistema embarcado e também o computador técnico, porque algumas responsabilidades passaram do controle do sistema externo para o interno.

O sistema demonstrou interdependência para com o computador técnico ao invés da dependência comum à categoria. E atestou-se que, nesse projeto o único tipo de mensagem que foi trocado entre o computador técnico e o robô foram as ações que compõe a estratégia para fazer o gol, atingindo o objetivo de manter a troca de mensagens no menor nível possível.

Assim a capacidade de processamento do computador técnico ficou mais focada na interpretação dos dados provenientes do sistema de visão e conseqüentemente na geração de estratégias, fazendo com que a equipe se torne mais competitiva.

### **6.1. Propostas para Trabalhos Futuros**

Nesta Seção são listadas algumas propostas para trabalhos futuros.

- 1) Testar o sistema implementado nos robôs reais;
- 2) Comparar a arquitetura proposta e os resultados apresentados com outros sistemas embarcados com arquitetura reativa, que não utilizam tarefas de tempo real e diferentes níveis de controle.
- 3) Usar um sistema operacional embarcado para gerenciar as tarefas de tempo real;
- 4) Comparar os resultados, principalmente o tempo de resposta, entre o sistema embarcado proposto e os obtido com o uso de um sistema operacional embarcado.

## REFERÊNCIAS

ARARANGUÁ INTRUDERS. **Missão.** Disponível em: <http://www.araranguaintruders.ufsc.br/> Acessado em: 10 out, 2012.

BENINI, Luca; MICHELI, Giovanni. **Networks on chips: A new SoC paradigm.** *Computer*, v. 35, n. 1, p. 70-78, 2002.

BLAZUZ, Claudio J. **Desenvolvimento de uma Arquitetura Híbrida e Distribuída para Sistemas Multiagentes e sua Aplicação no Futebol de Robôs.** Dissertação (Mestrado) – Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2008.

BORGES, Rodrigo W. **Aplicabilidade de Sistemas Operacionais de Tempo Real (RTOS) para sistemas embarcados de baixo custo e pequeno porte.** Dissertação (Mestrado) – Curso de Ciências, Programa de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2011.

BOSA, Jefferson L. **Sistema Embarcado para a Manutenção Inteligente de Atuadores Elétricos.** Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

CAMPBELL, Murray; HOANE, Joseph A. J; HSU, Feng-hsiung. Deep Blue. *Artificial Intelligence*, , v. 134, n. 1, p. 57-83, 2002.

CARRO, Luigi; WAGNER, Flávio Rech. **Sistemas computacionais embarcados.** Jornadas de atualização em informática. Campinas: UNICAMP, 2003.

CHENG, Gordon; ZELINSKY, Alexander. **Supervised autonomy: A framework for human-robot systems development.** *Autonomous Robots*, v. 10, n. 3, p. 251-266, 2001.

CMP United Business Media. *2006 Embedded System Design – State of Embedded Market Survey.* Pesquisa de Mercado, CMP United Business Media, 2006.

COSTA, Anna Helena Reali; PEGORARO, Renê. **Construindo robôs autônomos para partidas de futebol: o time GUARANÁ.** *SBA Controle & Automação*, Campinas, 2000.

DUDEK, Gregory; JENKIN, Michael. **Computational principles of mobile robotics.** Cambridge University Press, 2010.

FERREIRA, Ivo A; **Sistema de Controle e Supervisão de Sistemas Embebidos: Tipo SCADA**. Dissertação (Mestrado) – Curso de Engenharia Elétrica e de Computadores, Universidade do Porto, Porto, 2008.

GERVINI, Alexandre I. et al. **Avaliação de Desempenho, Área e Potência de Mecanismos de Comunicação em Sistemas Embarcados**. SEMISH'03–XXX Seminário Integrado de Software e Hardware, 2003.

KRAETZCHMAR, G. et al. **The ULM Sparrows: Research into Sensorimotor Integration, Agency, Learning, and Multiagent Cooperation**. ROBOCUP WORKSHOP. 1998

MACKWORTH, Alan, K. 1993. *On seeing robots*. Computer Vision: Systems, Theory, and Applications, World Scientific Press: Singapore, pp. 1–13.

MACHADO, Guilherme Bertoni; SIQUEIRA, Frank. **Integração de Sistemas Embebidos utilizando Web Services**. Universidade Federal de Santa Catarina, 2004.

MAETA, Silvio M. **Desenvolvimento da Infraestrutura Embarcada do Projeto AURORA**. Dissertação (Mestrado) – Curso de Ciência da Computação, Universidade Estadual de Campinas, Campinas, 2001.

MARWEDEL, Peter. *Embedded system design: Embedded systems foundations of cyber-physical systems*. Springer, 2010.

MURPHY, Robin R. **Introduction to AI Robotics**. The MIT Press: London, England, 2000.

NOERGAARD, Tammy. *Embedded System Architecture – A comprehensive guide for engineers and programmers*. Newnes, 2005.

NOVAK, Gregor; SPRINGER, Richard. *An Introduction to a Vision System used for a MirrorSOT Robot Soccer System*. IEEE International Conference on Computational Cybernetics, 2004.

PAZOS, Fernando. **Automação de Sistemas e Robótica**. Editora Axcel, 2002.

ROBOCUP; KITANO, Hiroaki (ed). **Robocup-97: Robot Soccer World Cup I**. Springer, 1998.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Editora Campus, 2004.

SALIM, Antonio; FUENTES, Olac; MUNOZ, Angélica. *Development of Local Vision-based Behaviors for a Robotic Soccer Player*. IEEE. Anais da Quinta Conferência do México em Ciência da Computação, 2004.

SAHOTA, M. K. & MACKWORTH, A. K. *Can Situated Robots Play Soccer?* Anais da Inteligência Artificial Canadense, 1994.

SANTOS, Túlio L. **Desenvolvimento de um Sistema Embarcado para Medição de Corrente**. Dissertação (Mestrado) – Curso de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2010.

SHAW, Alan C. **Sistemas e Software de Tempo Real**, 2001. Porto Alegre. Editora Bookman.

SIEGWART, Roland; NOURBAKHS, Illah R. **Introduction to Autonomous Mobile Robots**. 2004

SOBREIRA, Rodolfo M; SILVA, Francisco A; ROS, Renato L. **Futebol de Robôs, uma aplicação de robótica**. 2003.

TAURION, Cezar. **Software embarcado: oportunidades e potencial de mercado**. Rio de Janeiro. Brasport, 2005.

VIANNA, Alexandre SG; MACHADO, Liliane S. **Controle e Gerenciamento de Ambientes Reais Educacionais Através de Ambientes Virtuais**. International Conference on Engineering and Computer Education (ICECE2009), Buenos Aires, Argentina. 2009.

WEISER, Mark., *The Computer for the 21st Century*. Scientific American, 1991.

WOLF, Wayne. *Computers as Components: Principles of embedded computing system design*. 2001

## APÊNDICE I

Neste apêndice são apresentadas mais informações sobre tarefas de tempo real utilizadas pelo compilador CCS. Para um melhor entendimento foi elaborado um exemplo que compara um programa com funções sem o uso de tempo real e um programa com tarefas de tempo real.

As tarefas de tempo real são utilizadas em situações onde várias funcionalidades de um programa devem entrar em execução dentro de um prazo específico. O RTOS possui um núcleo central que se assemelha ao *kernel* de um SO, é ele o encarregado por controlar a execução das tarefas, se houver mais de uma tarefa, ele deve selecionar a tarefa que será executada segundo critérios estabelecidos no desenvolvimento da tarefa pelo programador.

O RTOS utilizado pelo compilador CCS permite que o microcontrolador PIC seja dotado de um sistema embarcado que possua tarefas de tempo real. Para tanto, são necessárias algumas diretivas iniciais para a execução das tarefas. A primeira é a diretiva de pré-processamento de delimitação de prazos e tempo de processamento `#USE RTOS` na qual os parâmetros em questão são o *timer* e o *minor\_cycle* que delimitam o menor ciclo de execução do RTOS.

Para o sistema, as tarefas são simples funções, porém, acima da sua assinatura inicial possui uma linha de código que determina que a função a seguir será uma tarefa. As funções que serão consideradas tarefas não devem ter nenhum parâmetro além dos específicos para o núcleo do RTOS e também não devem ter retorno.

Deve-se utilizar a diretiva de pré-processamento `#TASK`, com os parâmetros *rate*, *max*, *queue* que informam, respectivamente o prazo de execução da tarefa, quantas vezes ela deve entrar em execução dentro de um espaço determinado de tempo, o tempo máximo de execução que ela pode atingir e quantos bytes serão alocados para as mensagens recebidas da tarefa, por padrão é utilizado zero.

Depois de utilizar as duas diretivas acima e determinar quais funções serão tarefas de tempo real, basta realizar a chamada da função *rtos\_run* que coloca em execução o RTOS do CCS.

Da mesma maneira que a execução das tarefas de tempo real são acionadas pela função *rtos\_run*, elas também podem ser encerradas durante a execução de um programa. Se o sistema embarcado contiver tarefas de tempo real e funções, as tarefas de tempo real são executadas e depois de finalizadas, são encerradas, para que outras funcionalidades do programa entrem em execução. Para isso deve-se utilizar a função *rtos\_terminate* que encerra a execução do núcleo do RTOS.

As tarefas de tempo real estão sempre relacionadas a prazos específicos, porém através da função *rtos\_wait(Expre)* uma tarefa de tempo real pode ficar em estado de espera. Nesse caso, a tarefa é finalizada, deixando o processador livre, caso seu parâmetro, a expressão lógica *Expre* seja verdadeiro, ela volta a executar para então finalizar completamente.

Para um melhor entendimento do RTOS do CCS foi desenvolvido um programa que controla três LEDs, conforme esquemático da Figura 27. O programa dispõe de uma organização totalmente distinta, usando funções executadas a partir da função *main()* ou tarefas de tempo real executadas pelo função *rtos\_run()*.

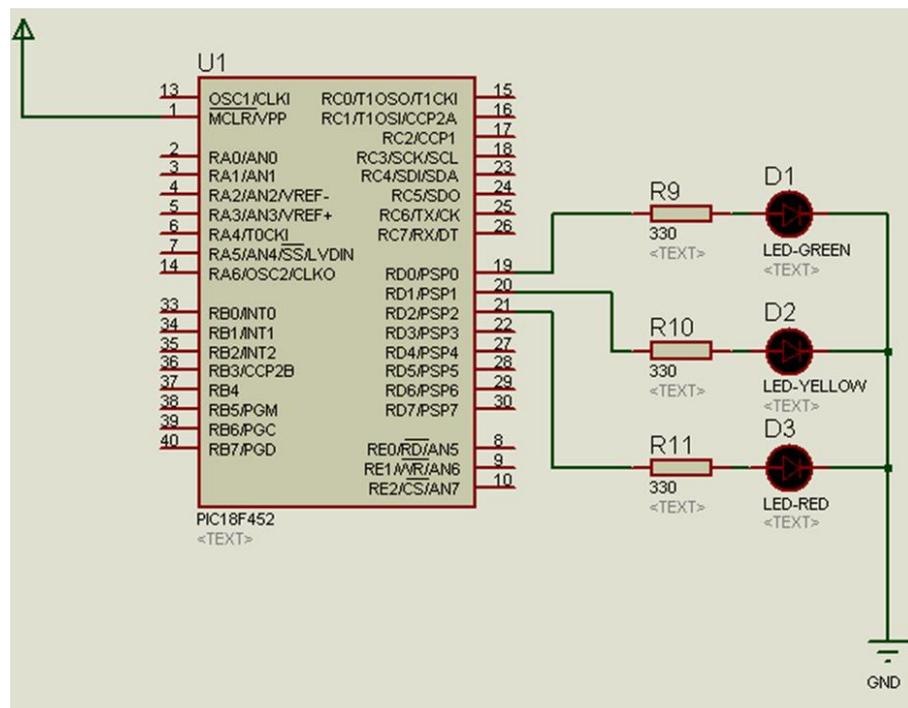


Figura 27: Esquemático do programa de exemplo.

A Figura 28 ilustra o código fonte do sistema embarcado para ligar os três LEDs. Neste caso, o programa foi desenvolvido somente com funções, ou seja, não é utilizada nenhuma tarefa de tempo real. Cada função do sistema entra em execução seguindo a ordem em que foram especificadas na função *main()*.

```

apendice_funcao.c
9  void liga_led1();
10 void liga_led2();
11 void liga_led3();
12
13 void main()
14 {
15     while(true){
16         liga_led1();
17         liga_led2();
18         liga_led3();
19     }
20 }
21
22 void liga_led1()
23 {
24     v1=~v1;
25     if(v1)
26         output_high(pin_d0);
27     else
28         output_low(pin_d0);
29 }

```

Figura 28: Programa para aceder três LEDs sem o uso de funções de tempo real.

Pode ser notado que as chamadas das funções são feitas pela função principal do programa, a função *main()*, porém, o código se torna menos organizado, com mais dificuldades de manutenção, e também não supre algumas necessidades, como no caso em que os três LEDs necessitassem ser acessos ao mesmo tempo. A maior diferença encontrada entre as funções e as tarefas de tempo real é a certeza da execução dentro de prazos específicos. Se as funcionalidades devem ser tratadas de forma periódica pode-se utilizar *timers* e interrupções, porém o controle sobre todas as funções se torna muito mais difícil para o programador, já que os cálculos de execução de cada tarefa devem ser projetados por ele para que não entrem em conflito.

No caso do RTOS as tarefas vão sendo simplesmente lançadas com seus respectivos parâmetros, e a execução dessas tarefas será responsabilidade do RTOS. O mesmo programa desenvolvido para o exemplo da Figura 28 foi adaptado para usar tarefas de tempo real. A Figura 29 ilustra o programa desenvolvido com o uso de tarefas de tempo real que são responsáveis por acender os LEDs.

```
apendice_tarefa.c
8
9 #use rtos(timer=0, minor_cycle=5ms) //Diretiva de RTOS
10
11 #task (rate=5ms, max=5ms) //Indica que a funcao seguinte eh uma tarefa
12 void liga_led1();
13
14 #task (rate=5ms, max=5ms)
15 void liga_led2();
16
17 #task (rate=5ms, max=5ms)
18 void liga_led3();
19
20 void main()
21 {
22     rtos_run(); //Inicia a operacao de RTOS
23
24     while(true){}
25 }
26
27 void liga_led1()
28 {
29     v1=~v1;
30     if(v1)
31         output_high(pin_d0);
32     else
33         output_low(pin_d0);
34 }
```

Figura 29: Programa para aceder três LEDs com o uso de funções de tempo real.

As tarefas entram em execução dentro do seu próprio prazo obedecendo as restrições de prazo estipulado no programa.

Freitas, Stéfani A. G.  
Tipo do Trabalho –  
Araranguá, \_\_/\_\_/2012.  
nº pág.

Concede-se à Universidade Federal de Santa Catarina – UFSC, a permissão para reproduzir cópias deste trabalho e emprestá-las tão somente para propósitos acadêmicos e científicos. Direitos reservados. Leis 9.609/98 e 9.610/98. Autoriza-se cópia, para utilização exclusivamente com finalidade didática, desde que com a citação da fonte.

---

Autor