



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

DESENVOLVIMENTO DE UMA PLATAFORMA DE ESTUDOS
EM ROBÓTICA MÓVEL COM REALIMENTAÇÃO VISUAL

Aluno: Walter Onofre Guerra Filho
Orientador: Antonio Marcus Nogueira Lima

Campina Grande - PB, agosto de 2007



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

Sumário

1	Introdução	2
2	Hardware utilizado	2
2.1	Robô	2
2.1.1	Alterações realizadas no Robô	3
2.2	Roteador Sem Fio	3
2.2.1	Modificações realizadas no roteador	5
3	Plataforma Desenvolvida	8
3.1	I-cybie	10
3.2	Roteador	10
3.3	PC	10
3.4	Interface de programação para a plataforma desenvolvida	11
3.4.1	libgrab_image	11
3.4.2	libicybie	11
3.4.3	librobot_vision	12
4	Exemplo de uso da plataforma	12
5	Conclusões	13

1 Introdução

A robótica móvel vem se expandindo rapidamente nos últimos anos. Seu estudo é de grande importância na formação de um engenheiro electricista não só pela modelagem dinâmica em si, mas também por todo o conhecimento em áreas correlatas necessário, como protocolos de comunicação, desenvolvimento de *software* para sistemas embarcados, entre outros.

Não existe no departamento de Engenharia Elétrica uma plataforma para estudos em robótica móvel. Caso algum aluno ou professor tenha interesse em estudar este assunto ou sistemas de controle através de realimentação visual, ele deve desenvolver o sistema de hardware necessário, o que atrasa bastante o estudo alvo.

Este trabalho visa suprir esta necessidade, criando de uma plataforma de estudos em robótica móvel, enfatizando os estudos de controle com realimentação visual.

Este trabalho está organizado como segue: Na Sessão 2 são descritos os componentes utilizados na plataforma; Na sessão 3 é descrita a interligação entre os componentes de *hardware* e os componentes de *software* desenvolvidos; Em seguida, na Sessão 4, é mostrado um exemplo simples de uso da plataforma e, finalmente, na Sessão 5 são apresentadas as conclusões e considerações finais.

2 Hardware utilizado

No projeto da plataforma, além do robô em si, outros dispositivos foram utilizados. O principal deles foi um roteador sem fio, que faz a interligação dos outros componentes e funciona como processador principal do sistema. Estes dois dispositivos, bem como as modificações realizadas em cada um deles, serão brevemente descritos a seguir.

2.1 Robô

Para a plataforma era desejado um robô que se locomovesse através de patas, não rodas. Embora o controle seja mais difícil num robô com patas, eles são muito mais versáteis em relação ao tipo de terreno em que podem se locomover do que os robôs movidos a rodas.

Além desta característica, algumas outras também eram desejadas, como:

- Disponibilidade de informações;
- Boa autonomia;
- Baixo custo;

A partir das características apresentadas, o robô escolhido foi o I-cybie, mostrado na Figura 1. O I-cybie, um cachorro-robô lançado no Brasil pela Estrela, é originalmente utilizado como brinquedo. Entretanto, com algumas alterações

de *software* e *hardware*, ele pode se tornar um robô cujo comportamento é programável [2].

As modificações realizadas no I-cybie serão melhor detalhadas na sessão seguinte.



Figura 1: Robô quadrúpede I-cybie

2.1.1 Alterações realizadas no Robô

A única modificação de *hardware* feita no robô foi adicionar uma porta serial. Para adicioná-la, o I-cybie deve ser aberto como mostrado na Figura 2. Na Figura 3 é mostrada uma visão superior da placa principal, com indicações de onde serão ligados os componentes. Em seguida, na Figura 4, é mostrado o local onde devem ser conectados o terra e o VCC. Por último, na Figura 5, são mostrados os conectores para o Tx e o Rx.

Como o VCC do I-cybie é de 3,6V e a tensão da serial do PC é de 12V, é necessário um circuito conversor. O diagrama esquemático deste circuito é mostrado na Figura 6.

No I-cybie original, existe um cartão de memória que pode ser substituído por um cartão programável. Com isto feito, e utilizando um SDK *open source* disponível [1], foi possível programar o comportamento desejado do robô.

2.2 Roteador Sem Fio

O roteador sem fio utilizado foi um roteador da marca Asus, modelo WL-500g Deluxe, mostrado na Figura 7. Este roteador tem como principais características:

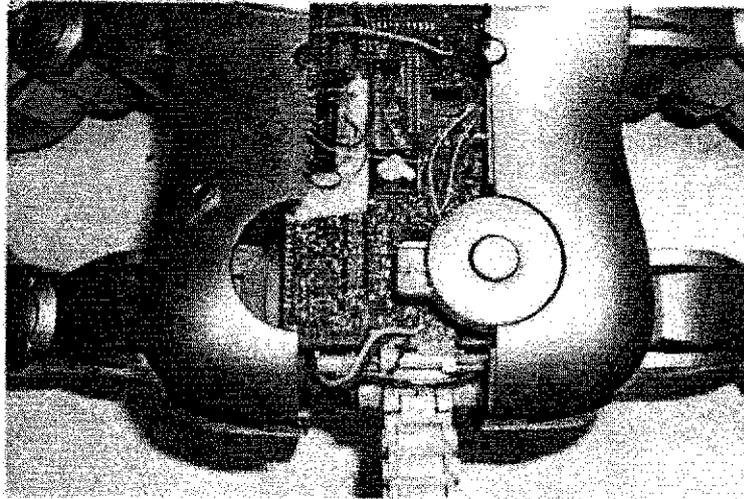


Figura 2: I-cybie aberto para a colocação da porta serial

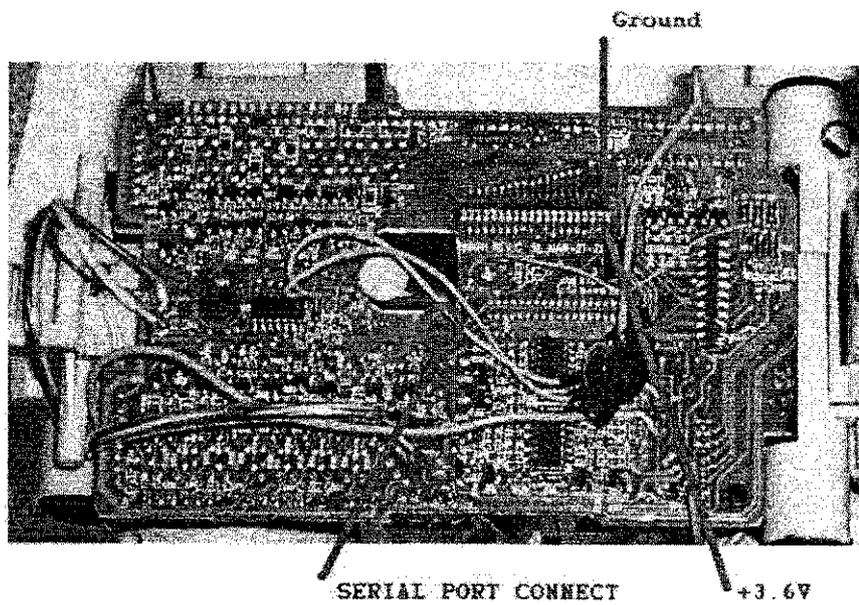
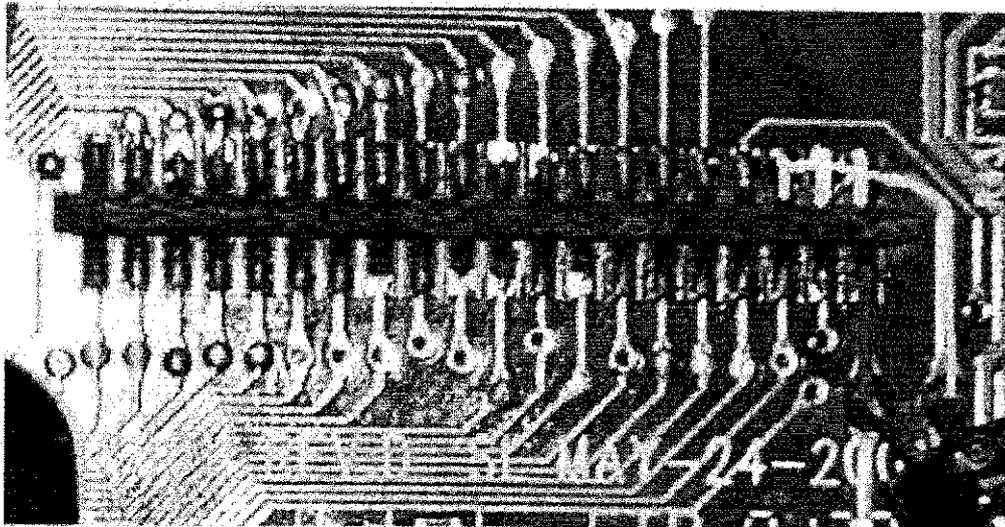


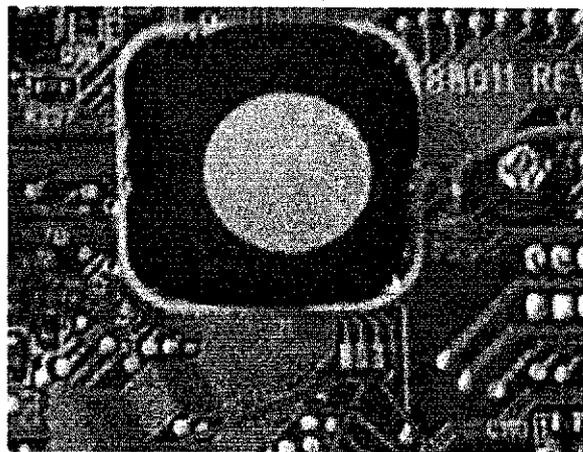
Figura 3: Visão superior da placa, com indicação de onde serão soldados os componentes



Red = +3.6V

Blue = GND

Figura 4: Local onde devem ser soldados o terra e o VCC



TxD0 RxD0 /CTS0

Figura 5: I-cybie aberto para a colocação da porta serial

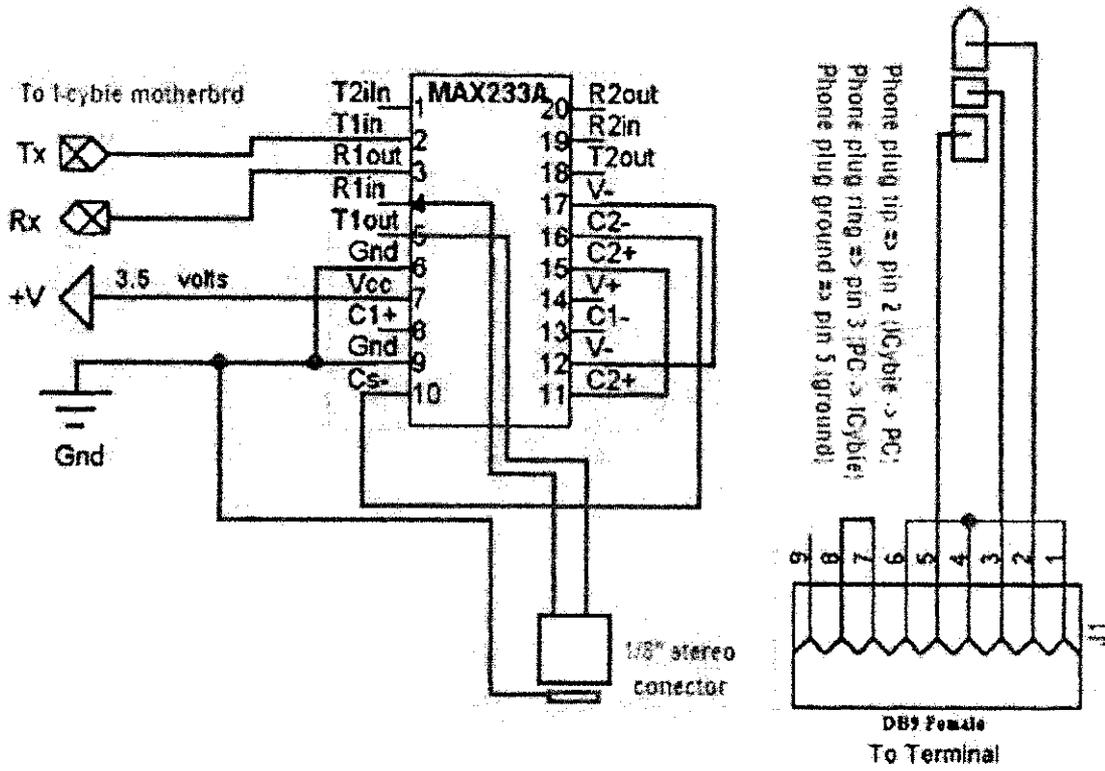


Figura 6: I-cybie aberto para a colocação da porta serial

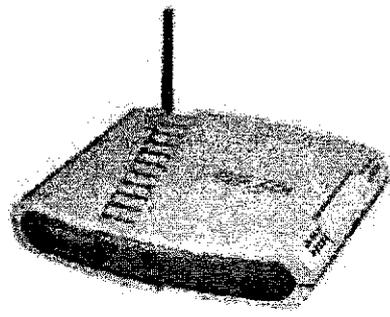


Figura 7: Roteador Asus WL-500g Deluxe

- Processador MIPS de 200MHz;
- 32MB de memória RAM;
- 4MB de memória Flash;
- Chipset Broadcom 5365;
- Duas portas USB;
- Duas portas seriais.

2.2.1 Modificações realizadas no roteador

Para que o roteador pudesse ser utilizado na plataforma, algumas alterações de *hardware* e *software* foram realizadas.

A primeira delas foi alterar o *firmware* padrão para um sistema Linux, de forma a tornar possível o desenvolvimento de outros aplicativos. A distribuição escolhida foi a OpenWrt [3], pois ela provê, além de uma imagem pronta do sistema operacional, com *kernel* e sistema de arquivos, um *Image Builder* que possibilita desenvolver uma imagem customizada, com *kernel* e programas definidos pelo usuário.

Além dessas formas de instalação, o OpenWrt também provê um SDK (*Software Development Kit*), que possibilita desenvolver novos programas no computador e compilá-los para a arquitetura do roteador, num processo chamado de cross compilação. O processo de instalação do Openwrt no roteador Asus WL-500g Deluxe é mostrado em [4].

A segunda modificação necessária foi a instalação de uma porta serial. O roteador possui, internamente, os pinos de saída de duas portas seriais, mas nenhuma delas está disponível externamente. Além disso, elas possuem tensão de 3,6V, diferente dos 12V utilizados pelo computador. Para a conversão de tensão necessária foi utilizado uma placa com um circuito integrado MAX232 e 5 capacitores, igual a utilizada no I-cybie. Esta placa foi presa à placa principal utilizando cola quente. Os conectores presentes no roteador são mostrados na Figura 8.

Por padrão, o OpenWrt utiliza a primeira porta serial como saída padrão do console de sistema operacional. Então, para o envio de comandos foi utilizada a segunda porta serial, que estava livre. A presença desta primeira porta é importante pois é somente através dela que tem-se acesso ao processo de inicialização do sistema e ao gerenciador de *boot*.

Além dessa duas modificações, uma terceira ainda precisou ser realizada. Ela foi necessária pois as duas portas USB disponíveis estão conectadas no mesmo barramento, o que impossibilita a obtenção das imagens das duas câmeras simultaneamente. Este problema foi resolvido facilmente pois o roteador possui pinos para mais duas portas USB internamente, conectadas em um outro barramento. Então, um segundo conector com mais duas portas foi colocado, deixando o roteador com quatro portas USB, com cada par conectado em um barramento.

Com as câmeras ligadas em barramentos diferentes, a aquisição das duas imagens simultaneamente (ou muito próximo disto) tornou-se possível.



Figura 8: Conectores para as portas seriais presentes no roteador

3 Plataforma Desenvolvida

Para o desenvolvimento da plataforma de estudos, foram pensados alguns casos de uso típicos, mostrados na Figura 9. Os dois casos de uso superiores são os de mais alto nível, definindo objetivos globais. Já os três casos de uso mais abaixo, são as atividades necessárias para que as tarefas superiores sejam alcançadas.

A plataforma proposta é composta por um cachorro robô modelo I-cybie, um roteador wireless ASUS WL-500g e duas webcams Philips. O modo de operação básico da plataforma é:

- O roteador captura as imagens das duas câmeras utilizando o barramento USB;
- Então, ele processa as imagens e define as próximas ações de controle;
- Finalmente, o roteador manda os comandos de controle ao cachorro, que os interpreta e executa.

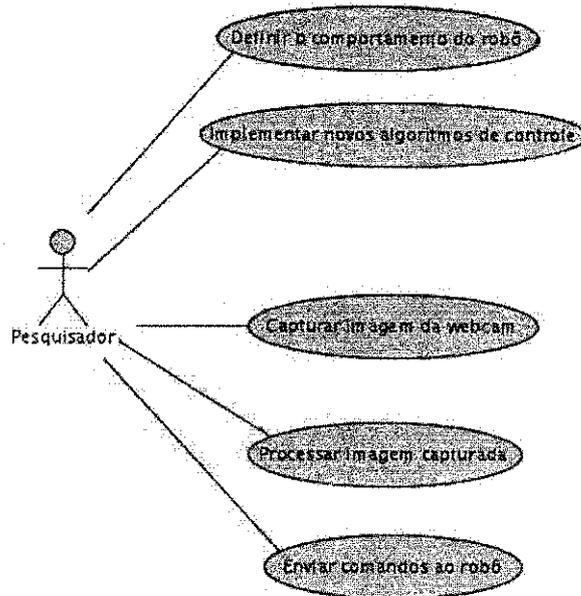


Figura 9: Casos de Uso para a plataforma proposta

Na Figura 10 é mostrado um diagrama esquemático de toda a plataforma. Além disso, são mostrados também os protocolos de comunicação utilizados na interligação dos diversos componentes. A seguir, serão detalhados os módulos que a compõe e como eles estão interligados.

3.1 I-cybie

No I-cybie é executado um interpretador de comandos. Este interpretador tem a única função de escutar a serial e executar os comandos enviados pelo roteador. Nele, nenhum processamento de informações, no sentido de definir ações de controle, é executado.

A decisão de tirar todo o processamento de comportamento do cachorro foi tomada devido à necessidade de processamento de imagens, o que não é viável utilizando o processador interno, além da quantidade de dados que precisaria ser transmitida (como as imagens, por exemplo) caso a decisão de controle fosse tomada no robô.

3.2 Roteador

É no roteador que grande parte do processamento é realizado. Ele é a peça central, que interliga todos os outros componentes.

Para permitir esta integração, foram desenvolvidas bibliotecas para acesso às câmeras, processamento das imagens obtidas e comunicação com o robô.

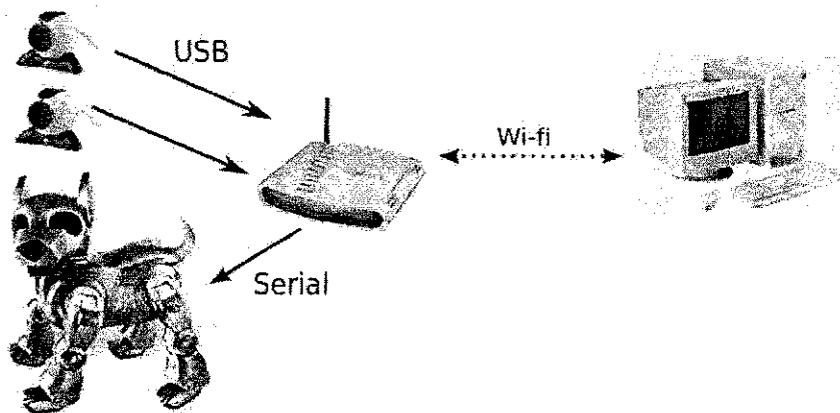


Figura 10: Diagrama da plataforma proposta

3.3 PC

O computador também tem acesso às imagens providas pelas câmeras, de forma que dependendo do poder de processamento exigido pelo algoritmo de controle, este pode ser executado direto no computador, enviando as instruções de alto nível para o roteador que, por sua vez, as transforma em instruções de mais baixo nível e as envia ao robô.

3.4 Interface de programação para a plataforma desenvolvida

Para desenvolver estratégias de controle para a plataforma, segue-se a seguinte sequência de desenvolvimento:

1. Desenvolver o algoritmo de controle. São disponibilizadas algumas bibliotecas para facilitar este desenvolvimento;
2. Compilar para MIPS utilizando o SDK provido pelo OpenWrt. A plataforma também provê *Makefiles* que automatizam o processo;
3. Copiar o executável para o roteador por *scp*;
4. Rodar o executável no roteador, através de um shell *ssh*.

Para facilitar o desenvolvimento de algoritmos de controle e prover acesso aos módulos da plataforma, foram desenvolvidas três bibliotecas. Cada uma delas foi desenvolvida com o objetivo de atender a um dos casos de uso propostos como requisitos.

3.4.1 libgrab_image

Esta biblioteca é utilizada pra capturar uma imagem da webcam presa ao robô. Ela captura um frame e armazena na variável *buffer*. O *buffer* de saída deve ser alocado antes da chamada da função. Esta abordagem permite que vários frames sejam capturados sobrescrevendo o mesmo *buffer*, o que evita várias chamadas às funções de alocação dinâmica de memória, que são bastante custosas computacionalmente.

```
void grab_image(unsigned char *buffer);
```

3.4.2 libicybie

É a biblioteca libicybie é quem provê acesso ao robô. Através do método *icybie_send_command* é possível enviar comandos de forma simples e direta. A enumeração mostrada abaixo define quais os comandos atualmente disponíveis.

```
void icybie_send_command(int command);
```

```
enum {  
    BLINK_GREEN, BLINK_RED, BLINK_YELLOW, HANDSHAKE, SIT  
};
```

3.4.3 librobot_vision

A terceira biblioteca desenvolvida foi a *librobot_vision*. Ela provê funções de alto nível para processamento de imagens. Nesta primeira versão, foram desenvolvidas apenas duas funções: a primeira para detectar se um cartão verde está na frente da câmera e a segunda para detectar se existe um cartão vermelho. Estas funções serão úteis pois a partir delas pode-se tomar decisões baseado somente na extração de características de alto nível das imagens obtidas da webcam.

```
int is_green_card_present(unsigned char *buffer);  
int is_red_card_present(unsigned char *buffer);
```

4 Exemplo de uso da plataforma

Para demonstrar o uso da plataforma, foi desenvolvida uma aplicação que reconhece alguns elementos posicionados à frente da câmera e, para cada um deles, executa uma ação distinta no robô. Os elementos escolhidos foram cartões nas cores vermelha e verde. Quando um dos elementos é detectado, o robô realiza uma ação, como piscar os olhos.

O código-fonte do programa é mostrado abaixo. Embora bastante pequeno, este programa faz uso das três bibliotecas, mostrando como utilizar cada uma das funções disponíveis.

```

1 #include <iostream>
2 #include <cstdlib>
3 #include "libs/grab_image.h"
4 #include "libs/robotVision.h"
5 #include "libs/icybie.h"
6
7 using namespace std;
8
9 int main() {
10     // Aloca o buffer que vai armazenar as imagens
11     unsigned char *buffer = (unsigned char *)malloc(320*240*3);
12
13     while (true) {
14         // Captura uma imagem
15         grab_image(buffer);
16
17         // Verifica quais cartões estão presentes
18         int red = is_red_card_present(buffer);
19         int green = is_green_card_present(buffer);
20
21         // Executa uma ação caso algum esteja presente
22         if (red) {
23             icybie_send_command(BLINK_RED);
24         }
25         else if (green) {
26             icybie_send_command(BLINK_GREEN);
27         }
28
29     }
30
31     // Libera a memória alocada
32     free(buffer);
33 }

```

Podem ser feitos alguns comentários sobre este código:

- Os *includes* das linhas 3, 4 e 5 são os referentes às bibliotecas desenvolvidas;
- Na linha 15 é chamada a função da biblioteca `libgrab_image`;
- Nas linhas 18 e 19 são chamadas funções da biblioteca `librobot_vision`;
- Nas linhas 23 e 26 são chamadas funções da biblioteca `libicybie`;

5 Conclusões

Neste trabalho foi apresentada uma plataforma de estudos em robótica móvel utilizando realimentação visual. Foram definidos os requisitos do projeto e apresentadas as soluções encontradas para satisfazê-los.

Foi desenvolvido também um pequeno exemplo de uso, de forma a ilustrar o processo de desenvolvimento para a plataforma proposta.

Estão previstos como continuação do trabalho uma expansão da biblioteca de acesso ao robô aumentando o número de comandos que podem ser passados. Também serão desenvolvidos novas funções para o processamento de imagens.

Referências

- [1] <http://aibohack.com/icybie/icsdk.htm>. Acessado em 27/08/2007.
- [2] <http://aibohack.com/icybie/index.html>. Acessado em 27/08/2007.
- [3] <http://openwrt.org>. Acessado em 27/08/2007.
- [4] <http://wiki.openwrt.org/openwrtdocs/hardware/asus/wl500gd>. Acessado em 27/08/2007.