

PROTÓTIPO DE VEÍCULO ROBÓTICO DE BAIXO CUSTO REMOTAMENTE OPERADO PARA VÍDEO INSPEÇÃO

Autor: Rodrigo de Avila Carpe

Pontifícia Universidade Católica do Rio Grande do Sul

Curso de Engenharia Elétrica

Av. Ipiranga 6681, Prédio 30, CEP: 91619-900 – Porto Alegre – RS – Brasil

E-mail: rodrigo.a.carpe@gmail.com

Telefone: +555198112501

Orientador: Anderson Royes Terroso

Pontifícia Universidade Católica do Rio Grande do Sul

Curso de Engenharia Elétrica

Av. Ipiranga 6681, Prédio 30, CEP: 91619-900 – Porto Alegre – RS – Brasil

E-mail: terroso@pucrs.br

1. INTRODUÇÃO GERAL

Este trabalho aborda o desenvolvimento de um protótipo robótico operado remotamente, sem fio, para transmissão de vídeo em tempo real. Este capítulo é uma breve introdução do plano, com assuntos necessários para entendimento de todo o projeto.

1.1. Apresentação do problema

A inspeção visual em locais de difícil acesso ou locais que ofereçam risco físico às pessoas é um problema. Com o advento da copa do mundo *FIFA* 2014 e a realização das Olimpíadas no Brasil em 2016 surgem à preocupação com a segurança. Hoje em dia, unidades remotamente operadas são largamente utilizadas a nível mundial em eventos dessa magnitude. O jornal *Zero-Hora* de 22/11/2012 trás como destaque a matéria: "*Robô da Polícia Federal chama atenção em operação conjunta no aeroporto Salgado Filho*". Conforme a matéria destaca, a Polícia Federal do Rio Grande do Sul possui somente uma unidade deste robô, o que denota a nossa carência em dispositivos de segurança deste tipo, obviamente pelo custo elevado de sua aquisição. Observando que no mercado os preços de venda praticados de um robô de inspeção de vídeo remoto são muito altos, optou-se pelo desenvolvimento de um protótipo de baixo custo. Atualmente dispositivos robóticos fazem parte de nosso cotidiano e cada vez mais são percebidos. Vários são os objetivos da existência de sistemas robóticos: comodidade, segurança, saúde, acessibilidade, eficiência, economia entre outros. Nos dias atuais dispomos de recursos tecnológicos impensáveis há alguns anos atrás. Prova disso é a larga utilização de robôs na indústria, forças armadas, policiais e até mesmo em hospitais esse recurso já é utilizado, como foi publicado no Jornal *Folha de São Paulo* em 05/12/2011: "*Hospital do RIO será o primeiro da rede pública a utilizar robô em cirurgia*". Mais impensável ainda é que chegamos ao nível de termos um robô explorando o solo de outro planeta, o *Curiosity* que completou um ano em Marte a um custo aproximado de 2,5 bilhões dólares americanos [1] [2].

A evolução tecnológica dos últimos anos se deve em muito a invenção do transistor por *Bardeen* e *Brattain* em meados dos anos 50, nos Laboratórios da *Bell*

Telephone [3]. A partir daí, tudo que conhecemos hoje em termos de eletrônica, deriva dessa invenção. Obviamente estes 60 anos foram um pequeno lapso de tempo se comparado a existência da raça humana, porém, o desenvolvimento tecnológico cresceu à uma taxa exponencial. Hoje temos um *smartphone* na palma de nossas mãos com capacidade de processamento, memória, e recursos que o melhor dos computadores não possuía à 10 nos atrás. Toda essa tecnologia agrega um alto valor aos produtos que à utilizam. E é neste contexto que o presente trabalho se insere.

1.2. Proposta

Desenvolver um *VOR* (Veículo de Operação Remota) de baixo custo, capaz de enviar vídeo em tempo real. A operação poderá ser feita através de um dispositivo com sistema operacional *Android*® e comunicação via *Bluetooth*® ou via rádio controle. Quando operado via *Bluetooth*® o *VOR* enviará periodicamente para a tela do dispositivo *Android*®, dados dos sensores de temperatura e umidade relativa do ar, além da tensão da bateria principal. O protótipo robótico transportará uma câmera de vídeo em cores, montada sobre dois servos motores e terá movimentos de *PAN* (panorama) e *TILT* (inclinação) com 180° de liberdade no eixo horizontal e 110° no eixo vertical. A câmera permitirá além da inspeção, a própria pilotagem do *VOR* em primeira pessoa. As imagens geradas pela câmera serão transmitidas diretamente por um módulo de rádio na faixa de 1,2GHz com 700mW de potência até uma distância aproximada de 1000m do receptor em linha de visada. As imagens serão recebidas por um receptor *AV* (áudio e vídeo) e poderão ser exibidas em qualquer monitor com entrada *RCA* (o nome do conector advém da sigla da empresa que introduziu este conector, a *Radio Corporation of America*) ou na tela do computador através da utilização de uma placa de captura de vídeo, onde inclusive as imagens podem ser gravadas. Junto à câmera será montado um *led* (Light Emitting Diode) de alta intensidade com temperatura de cor entre 4500K e 6000K, visando à iluminação em cores das imagens geradas. A força motriz será obtida através de quatro conjuntos motores acoplados a caixas de redução, que terão sua velocidade e sentido de rotação controladas por uma placa *Arduino*. O veículo será equipado com rodas de 120mm de diâmetro, utilizadas em modelos *RC*

(Rádio Controlados) *off-road*. A interface entre o *uC* (microcontrolador) e os motores será feita por intermédio de duas placas *drivers* de potência baseadas no chip *LM298* que é uma ponte-H dupla com capacidade de corrente igual a 2 ampères. Os comandos do operador serão recebidos por um módulo *Bluetooth®* ou por um módulo receptor de rádio controle e o *VOR* automaticamente entenderá qual sistema está sendo utilizado. A alimentação será feita com duas fontes distintas de energia. Uma alimentará o *uC*, motores, iluminação e *drivers* de potência. A outra irá suprir a câmera e o transmissor *AV*. O chassi será projetado sob medida para o protótipo e fabricado em liga de alumínio.

1.3. Justificativa e importância do trabalho

O crescimento do PIB Brasil de 1,3% em 2001 para 7,5% em 2010 [4] reflete a atual atividade econômica do mercado nacional. Isso trás mais segurança para o capital em toda cadeia de produtos negócios e serviços no país. A atividade destes setores atrai investidores internacionais e rapidamente nos credenciamos para sediar eventos de grandes magnitudes tais como a copa do mundo e as olimpíadas. Estes tipos de eventos são particularmente visados por criminosos, terroristas e grupos extremistas que sabem que os olhos do mundo estarão voltados para tais realizações. Uma plataforma de inspeção robótica de baixo custo pode ser amplamente utilizada em eventos assim para fornecer uma primeira linha de defesa em situações táticas, além de uma grande gama de outras aplicações tais como: construção civil, empresas de segurança, operações da polícia e forças armadas, indústria em geral, alimentos, empresas de saneamento, controle de pragas, dutos de refrigeração, subestações e usinas de energia.

2. REFERENCIAL TEÓRICO

Este trabalho aborda um tema relativamente atual, que não define conceitos fechados para o desenvolvimento e onde a gama de possibilidades é quase infinita, assim, o lugar onde se pode encontrar mais informações é na *Internet*. Em buscas pela rede, encontraram-se algumas dezenas de veículos operados remotamente para as mais diversas finalidades, porém, este trabalho limita-se há referenciar somente *VOR's* terrestres voltados para vídeo inspeção. Existem projetos de veículos para inspeção que vão desde equipamentos montados com reciclagem de eletrônicos até equipamentos sofisticados ao custo de alguns milhares de dólares. Em especial destaca-se o mercado chinês onde é possível encontrar uma vasta gama destes dispositivos para as mais diversas aplicações. A seguir são apresentados os modelos pesquisados e uma breve descrição dos principais elementos utilizados neste projeto.

2.1. Abordagem de outros autores

O modelo *SuperDroid UM4 Ultra Mobile Surveillance & Inspection Robot* [5] é comercializado pela empresa *Robotshop*. Este *VOR* pode ser visto na Figura 1 e apresenta funcionalidades muito similares às propostas neste trabalho, porém, utiliza somente a opção de controle *RC* com 5 canais, o que limita bastante sua capacidade uma vez que dois canais são utilizados para a direção, dois canais para o movimento da câmera e sobra apenas um canal para controlar outro acessório opcional. Ao custo de *USD* \$3,836.51 e considerando o imposto de importação de 60%, este modelo não chegaria ao Brasil por menos de R\$ 15.040,00 (*USD* \$1.00 = R\$2,45).



Figura 1 - Fonte: <http://www.robotshop.com/superdroid-um4-spektrum-remote-3.html>

Outra opção interessante e que trás uma abordagem construtiva diferente é o *Trackbot Surveillance and Inspection Tracked Robot* [6], também comercializado pela empresa *Robotshop*. Tem como principal característica o conjunto de esteiras motrizes do tipo *Caterpillar*, muito indicadas para terrenos acidentados, que pode ser visto na Figura 2. Muito similar ao modelo apresentado anteriormente no que tange às suas funções, porém, não possui em sua versão básica o movimento da câmera de vídeo. Pode ser adquirido pelo valor de USD \$5,990.00 ou R\$ 23.480,00 no Brasil.



Figura 2 - Fonte: <http://www.robotshop.com/Images/xbig/en/trackbot-surveillance-inspection-tracked-robot-1-B.jpg>

O modelo *VOR-42* [7] apresentado na Figura 3 é fabricado no Brasil pela empresa *Brasil Robotics* e possui controle por cabo ou em sua versão '*W*', por rádio. O diferencial fica por conta do sistema de controle que é dedicado e desenvolvido pela empresa para quaisquer dos dois métodos de operação. Sua tração é feita através de esteiras e possui a função *hodômetro*, que permite saber a distância percorrida pelo robô. Por seu tamanho reduzido em relação aos demais modelos expostos aqui, é muito indicado para inspeção em tubulações industriais, podendo operar em tubulações com diâmetro a partir de 350mm. O preço do modelo *VOR-42W*, conforme informado em contato com a empresa é de R\$ 10.500,00.



Figura 3 - VOR42 - Fonte: http://www.brasilrobots.com.br/vor42/vor42_2.jpg

2.2. Principais componentes do projeto

As três opções de controle implementadas utilizadas neste protótipo, buscam cobrir a maior gama de aplicações. Quando operado via *Bluetooth®* tem-se o benefício da economia por utilizar qualquer aparelho celular ou tablete com *Android®* e a possibilidade de receber dados de telemetria dos sensores embarcados no *VOR*. O aspecto negativo fica por conta do alcance do sinal *Bluetooth®*. A operação via rádio *Futaba®* trás consigo anos de desenvolvimento em que as empresas fabricantes, como a *Futaba Electronics Manufacturing Corporation* chegaram a um alto nível de robustez de hardware e que em geral é pouco sujeito a falhas. Sendo voltado para o modelismo, mas em alguns casos aplicados em utilizações profissionais, este sistema de rádio pode operar em distâncias superiores a 1000m. Alguns modelos mais modernos possuem um recurso que permite a transmissão de dados do receptor (RX) para o transmissor (TX), conhecido por telemetria. O modelo utilizado neste protótipo não conta com este serviço. E finalmente a implementação de um controle via rádio serial com a utilização dos módulos *APC-220*, placa *Arduíno Pro MINI* e o controle de vídeo game utilizado no console *Playstation 2* da *Sony®*.

2.2.1. Rádio Bluetooth®

O Bluetooth® é um protocolo padrão de comunicação sem fio projetado para baixo consumo de energia com baixo alcance, baseado em microchips transmissores de baixo custo com o propósito de eliminar as conexões físicas entre dispositivos de diferentes fabricantes. Este protocolo tem como características base a faixa de operação ISM (Industrial, Scientific and Medical) centrada em 2.4 GHz e com largura de banda de 1 MHz. Inicialmente o Bluetooth foi desenvolvido por um consórcio de empresas denominado SIG (Special Interest Group) e pouco tempo depois foi adotado pelo IEEE dando origem ao padrão 802.15 para redes PAN (Personal Area Network) [8]. Através do módulo Bluetooth® conectado a dois pinos digitais da placa com o microcontrolador, é possível enviar e receber dados de forma serial. Uma característica deste sistema de comunicação é que a conexão ou pareamento entre dois dispositivos sempre é iniciado pelo denominado Master

(Mestre), que neste caso será um celular ou tablete, sendo que o outro dispositivo denomina-se *Slave* (Escravo).

Conforme o artigo citado conclui, a comunicação *Bluetooth®* do ponto de vista experimental é perfeitamente viável e funcional para o controle de um veículo teleoperado. A Tabela 1 apresenta as classes dos dispositivos *Bluetooth®* comerciais e seus respectivos alcances com base em suas potências de transmissão. Neste projeto será utilizado um módulo classe 2 similar ao apresentado na Figura 4.

Tabela 1 - Módulos Bluetooth® comerciais

Classe	Potência máxima permitida (mW/dBm)	Alcance (Aproximadamente)
Classe 1	100 mW (20 dBm)	até 100 metros
Classe 2	2.5 mW (4 dBm)	até 10 metros
Classe 3	1 mW (0 dBm)	~ 1 metro

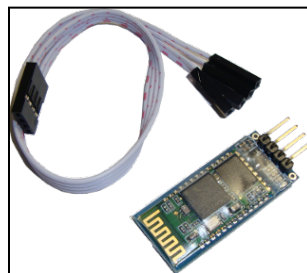


Figura 4 - Módulo Bluetooth® escravo - Fonte:http://serialink.com.br/loja/images/prod_modulo_bluetooth_gde.png

2.2.2. Rádio Futaba®

O sistema de rádio-controle é composto por dois módulos: TX que é um equipamento que permite controlar um veículo a distância por meio de transmissão de ondas de rádio e o RX que é o responsável por receber os sinais do TX e neste caso, enviá-los ao *uC* para serem tratados. O rádio utilizado neste projeto foi o modelo *FUTABA Sky Sport 6A* que pode ser visto na Figura 5.



Figura 5 - Radio FUTABA Sky Sport 6A Fonte: <http://www.fatlion.com/sailplanes/images/6skysport.gif>

As principais características deste modelo são: 6 canais, sendo canais de 1 a 4 utilizados nos *sticks* (alavancas com movimentos nos eixos X, Y e em alguns casos no eixo Z) proporcionais, canal 5 on/off, canal 6 *PWM* (Pulse-Width Modulation), operação em *FM* (Frequency Modulation) na faixa de 72MHz. Para utilização neste projeto, o módulo de transmissão do rádio foi substituído por um da faixa de 2,4GHz, diminuindo com isso o tamanho da antena. O módulo utilizado nessa conversão foi importado do fabricante *CORONA ELECTRONICS & TECHNOLOGY CO.,LTD* na China e pode ser visto na Figura 6. O rádio com o kit de conversão instalado é apresentado na Figura 7.



Figura 6 - Módulo TX CORONA CT8ZDSSS - Fonte: <http://www.corona-rc.com/Upload/2009121133214598.jpg>

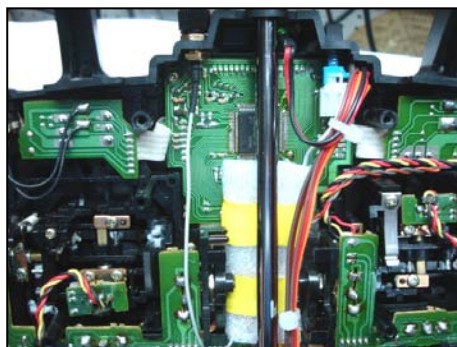


Figura 7 - Instalação do módulo Corona - Fonte: http://www.manorc.com.br/imagens/72%20para%202_4/Instalacao.JPG

O rádio *Futaba*® recebeu ainda um suporte de fibra de carbono que admite a fixação de monitores de *LCD* (Liquid Crystal Display) de até 7", permitindo assim uma melhor visualização das imagens geradas. A montagem final pode ser vista na Figura 8.



Figura 8 - Rádio com suporte de monitor montado - Fonte: O autor

Os sinais são recebidos pelo módulo RX que neste caso foi adquirido junto com o transmissor Corona já que é necessário um receptor que opere na mesma frequência do transmissor. O modelo utilizado foi o *CR6D* que pode ser visto na Figura 9.



Figura 9 - RX Corona CR6D - Fonte: O autor

2.2.3. Rádio APC-220

Talvez o método de controle mais interessante entre os propostos, do ponto de vista custo-benefício seja o módulo *APC-220*, pois trata-se de uma solução simples e econômica para comunicação de dados sem fio criando uma interface

UART/TTL transparente. A Figura 10 apresenta seu aspecto junto ao adaptador USB, que permite conectar o módulo diretamente ao microcomputador.



Figura 10 - Módulos APC 220 - Fonte: <http://image.made-in-china.com/2f0j00PKiaLrWlgDon/APC-220-Wireless-Communication-Module-RF-Transmitter-Module.jpg>

As principais características deste modelo são:

- Transmite em distâncias de até 1000m (linha de visada) @ 9600 bps;
- 256 bytes data buffer;
- Alta sensibilidade (-112dbm @ 9600 bps);
- Modulação GFSK;
- Interface UART/TTL;
- Watch dog integrado;
- Dimensões: 37x17x6.5 mm.

2.2.3.1. Microcontrolador para o controle remoto

O *Arduino Pro Mini* é uma placa com microcontrolador baseado no *ATmega168*. Possui 14 I/O's (pinos de entrada e saída) digitais, dos quais 6 podem ser usados como saídas *PWM*, 8 entradas analógicas, um oscilador *on-board*, um botão de *reset*, e os furos para fixação de pinos. Não possui um adaptador *FTDI* integrado, sendo necessária a utilização de um adaptador para fazer a carga dos programas. Existem duas versões, porém, o modelo utilizado neste projeto opera em 5V e 16 MHz. A Figura 11 apresenta seu aspecto.

Para movimentar o *VOR*, serão adquiridos quatro motores com caixa de redução acoplada. O modelo escolhido foi o *Akiyama AK555/12-R83-V2* visto na Figura 13 e suas especificações são:

- Velocidade: 83 RPM;
- Torque: 11,1 kgf.cm;
- Tensão: 12 VDC;
- Potência: 4,9 W;
- Corrente: 1,6 A;
- Relação da caixa de redução: 1:72.



Figura 13 - Micromotor DC com caixa de redução acoplada - Fonte: <http://www.yuzhou.com.br/loja/files/produtos/files/datasheetdomicromotordccomcaixadereducaoavelocidade83rpmi001638.pdf>

A técnica utilizada para movimentar o *VOR* consiste em comandar o giro dos motores de forma sincronizada para obter-se o resultado esperado, assim, para que o veículo mova-se para frente, será necessário que os dois motores instalados no lado esquerdo girem em sentido horário e os dois motores instalados do lado direito girem em sentido anti-horário. Para mover-se para trás, somente invertemos a rotação em relação ao movimento anterior. Para virar à esquerda, será necessário que todos os motores girem em sentido anti-horário. Se a curva for à direita, o inverso.

2.2.5. Servo motores

Os servo motores são dispositivos eletrônicos-mecânicos que recebem os sinais do receptor de rádio e os converte em movimento mecânico. Um exemplo de servo motor é mostrado na Figura 14.



Figura 14 - Servo motor - Fonte: <http://www.pictronics.com.br/images/artigos/servo/Hit servop.jpg>

A ideia por trás do funcionamento de um servo obedece a uma lógica simples. A informação recebida do módulo RX orienta o circuito eletrônico do servo qual a posição ele deve assumir através de um sinal *PWM* como exemplificado na Figura 15.

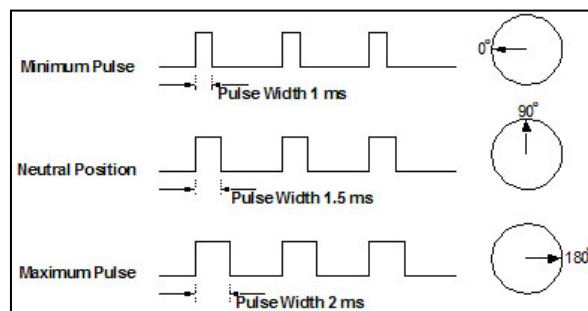


Figura 15 - PWM para controle do servo - Fonte: <http://www.pictronics.com.br/images/artigos/servo/pulso.jpg>

Através de uma caixa de redução, a energia mecânica na ponta do eixo do pequeno motor *DC (Direct Current)* tem seu torque aumentado muitas vezes em contrapartida à sua velocidade de rotação é diminuída consideravelmente. Conectado ao eixo externo do servo, existe um potenciômetro denominado "*Potenciômetro de feedback*" que tem a função de informar ao circuito eletrônico do servo se o mesmo atingiu a posição solicitada pelo RX.

2.2.6. Microcontrolador do VOR

O tipo de processador que melhor se aplica a este projeto é aquele que integra na mesma pastilha de silício, além da *CPU* (Unidade Central de Processamento), também circuitos de memória e uma diversidade de outros circuitos auxiliares dedicados a funções de entrada e saída tais como conversão analógica/digital (A/D) e saídas digitais moduladas por largura de pulso (PWM). Este tipo de processador é usualmente chamado de *microcontrolador* (uC). Devido à simplicidade dos seus circuitos internos, os microcontroladores com *CPU* do tipo *RISC* (Reduced Instruction Set Computing) tendem a operar com maior eficiência e com frequências de *clock* mais elevadas. Dentre uma vasta gama de *uC*'s disponíveis no mercado, optou-se pela utilização do *ATMEL ATmega 1280* que é um microcontrolador de 8bits, *RISC*, com 128kB de memória *flash*, 8kB *SRAM*, 4kB *EEPROM*, *PWM*, 4 *UARTs*, 86 pinos de I/O, conversor A/D de 10-bits, operando com *clock* de 16MHz. Este microcontrolador está embarcado na placa *Arduino Mega* visto na Figura 16. O *Arduino* é uma plataforma de prototipagem rápida desenvolvida com finalidade educativa, porém, ganhou muita força entre profissionais por tratar-se de uma plataforma open-source com milhares de colaboradores ao redor do mundo. No mercado é possível encontrar uma infinidade de *Shields* (escudos com sensores ou para funções específicas que podem ser sobrepostos ao *Arduino* compatíveis pino à pino) que possibilitam montagens rápidas de protótipos.

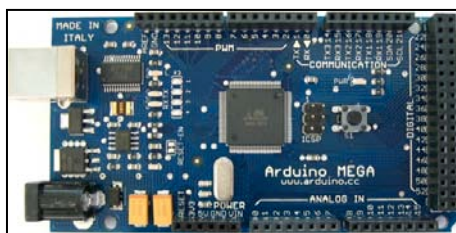


Figura 16 - Arduino MEGA 1280 - Fonte: <http://arduino.cc/en/uploads/Main/ArduinoMega.jpg>

3. METODOLOGIA

3.1. Introdução

Neste item será descrita a metodologia proposta com a utilização dos recursos disponíveis para ilustrar as principais etapas do projeto.

3.2. Fluxograma

A Figura 17 apresenta o comportamento esperado para as etapas de desenvolvimento e validação do projeto.



Figura 17 - Fluxograma das etapas do projeto - Fonte: Autoria Própria

3.3. Desenvolvimento do Hardware

Este trabalho demandará o desenvolvimento de hardware mecânico e eletrônico. Os subitens a seguir explanam sobre o assunto.

3.3.1. Chassis

A partir da definição do hardware eletrônico que será embarcado no projeto, tendo assim suas dimensões e características, foi possível então desenvolver o chassis e seus componentes. O material escolhido para a fabricação foi a chapa de alumínio com 2mm de espessura, por sua resistência mecânica comparada ao acrílico, por exemplo, capacidade de dissipação de calor, facilidade para moldagem e corte, além de sua durabilidade. O desenho foi elaborado a partir da ferramenta *AutoCad* da *Auto Desk em sua versão educacional*, os cortes(*) da chapa foram feitos por máquina *CNC PUNCIÓNADEIRA* e as dobras foram feitas manualmente. A Figura 18 apresenta o desenho pronto enviado à fabricação e na Figura 19 as peças já prontas.

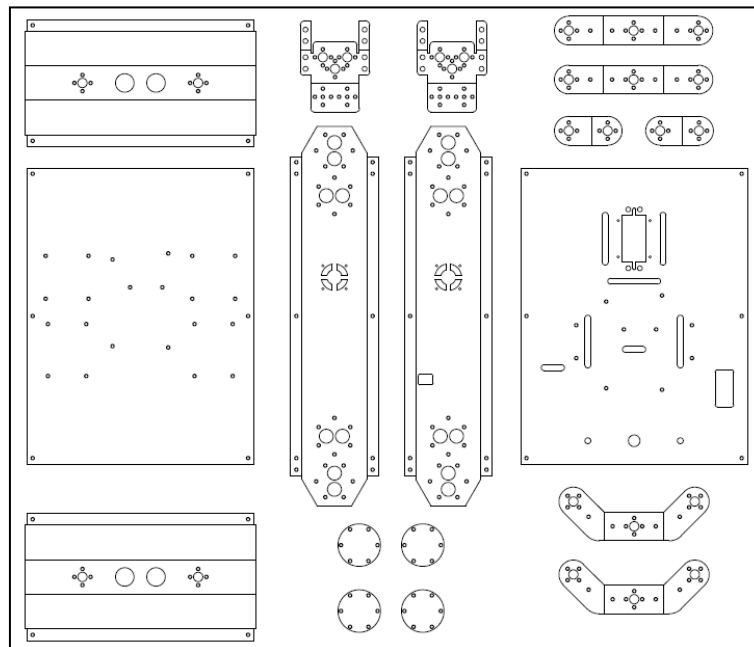


Figura 18 - Desenho das partes mecânicas - Fonte: Autoria Própria

(*) Agradecimento a colaboração do colega *Udo Neimann*, que gentilmente fabricou os componentes mecânicos necessários ao projeto.



Figura 19 - Fonte: Autoria Própria

3.3.2. Placa de distribuição de energia

No decorrer do desenvolvimento, em função da grande quantidade de periféricos e suas diferentes tensões de trabalho, observou-se a necessidade de uma placa para distribuição de energia. O esquemático e layout da placa foram elaborados com o auxílio da ferramenta *Eagle®* da *CadSoft*. A confecção foi feita através do processo de transferência térmica e corrosão em Cloreto Férrico (FeCl_3). As Figuras 20 e 21 apresentam respectivamente o diagrama elétrico e o layout.

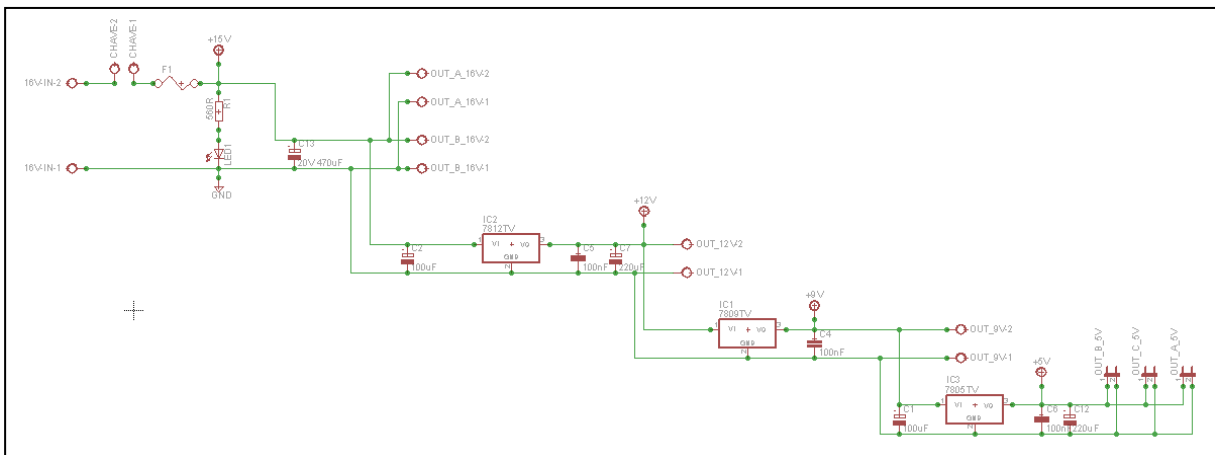


Figura 20 - Diagrama elétrico da placa de distribuição de energia - Fonte: Autoria Própria

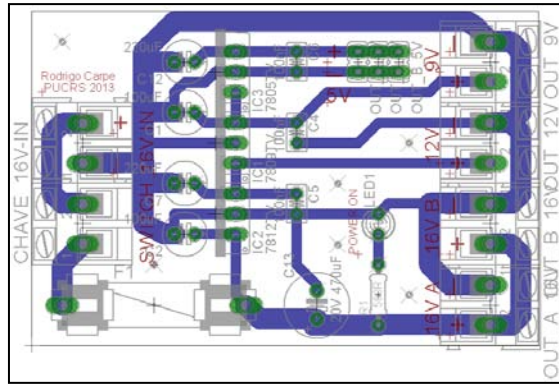


Figura 21 - Layout da placa de distribuição de energia - Fonte: Autoria Própria

A placa de distribuição de energia é baseada em reguladores de tensão lineares, suas ligações e componentes periféricos foram montados conforme recomendações das folhas de dados do fabricante [9]. Ao todo são três reguladores ligados em cascata. A bateria foi montada associando-se em série quatro células de *Li-Ion* (ions de lítio) de 3,7V e 2200mAh resultando em uma bateria de 14,8V e 2200mAh, nominais. Embora a tensão nominal de cada célula seja de 3,7V, elas carregam até 4,2V, resultando em uma tensão total de 16,8V. Para o ciclo de carga foi necessário adquirir um carregador "inteligente", que carrega as células individualmente, garantindo assim que todas finalizem o ciclo de carga com a mesma tensão. Essa tensão ingressa na placa através do *borne* 16V-IN, passa pela chave geral e por um fusível rápido de 2 ampères, após, vai para dois bornes de saída OUT_A_16V e OUT_B_16V, onde serão ligados os *drivers* de potência dos motores. Após o fusível existe uma conexão à entrada do LM7812 que tem sua saída conectada ao *borne* 12V_OUT, responsável pelo fornecimento de energia para os *drivers* de acionamento dos *fans* e *leds*. A saída do regulador de 12V também alimenta a entrada do LM7809 que tem sua saída ligada ao *borne* OUT_9V com a função de fornecer energia para o *uC*, além de rebaixar a tensão para o regulador LM7805 que através de barras de pinos, alimenta os servos e módulo de relés. Como os três reguladores são positivos e possuem massa à terra, uma boa opção é fixação dos mesmos diretamente no chassis de alumínio, ganhando-se assim a redução do tamanho da placa de energia e um dissipador de calor superdimensionado.

Uma segunda bateria 3S *Li-Po* (três células de polímero de lítio) com capacidade nominal de tensão igual a 11,1V e corrente de 2200mAh será utilizada

para alimentar a câmera e o *transmissor AV*, visando assim, menor interferência causada pela comutação das escovas nos coletores dos motores de tração. Totalmente carregada essa bateria tem sua tensão máxima fixada em 12,6V. Um dos relés será utilizado para comutar a alimentação do *link AV* e da câmera.

3.4. Desenvolvimento do Software

Certamente o maior tempo dedicado a este projeto foi relacionado ao desenvolvimento dos softwares. A abordagem será feita em três partes: Software do VOR, software do controle por módulo APC-220 e software para *android*®.

3.4.1. Software do VOR

O veículo tem suas funções intermediadas por um *Arduino Mega 1280*, como dito anteriormente. A programação foi feita de forma independente para operar com qualquer um dos métodos de controle propostos e por fim fez-se a fusão dos três programas, de tal forma que algumas rotinas e variáveis ocasionalmente foram compartilhadas. Resumidamente o fluxo do programa é apresentado na Figura 22.

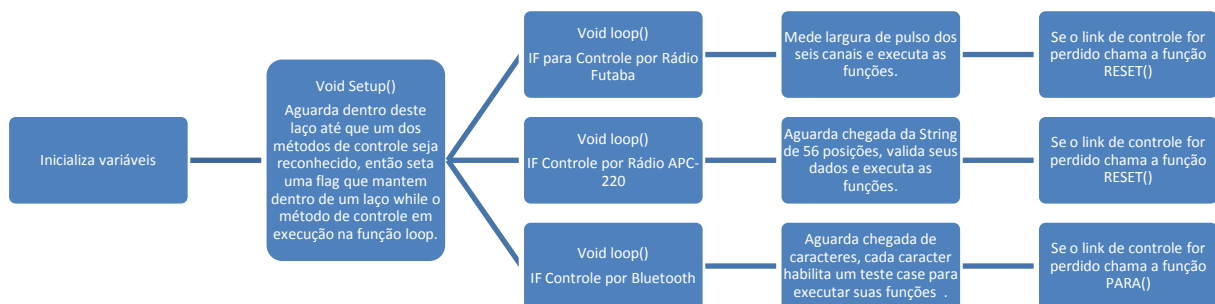
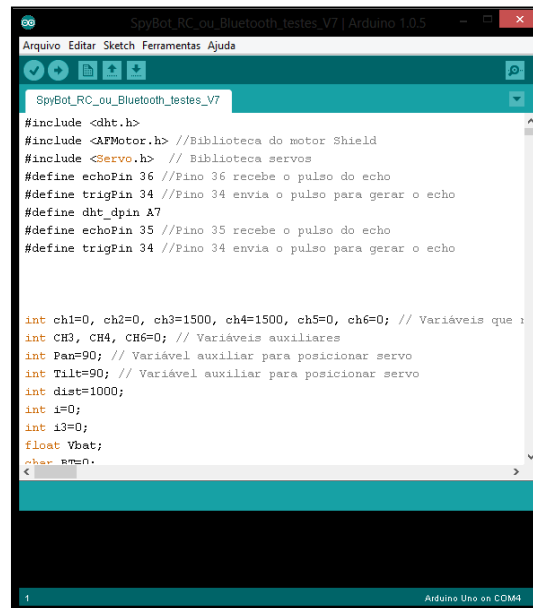


Figura 22 - Fluxo do programa para o VOR - Fonte: Autoria Própria

O compilador para a programação do uC é conhecido como *Sketch* e atualmente está na versão 1.0.5. Foi elaborado em C/C++ baseado em *Wiring*, e o ambiente de desenvolvimento é baseado em *Processing*, ambos *open-source*.

Abaixo pode-se ver na Figura 23 um exemplo do *Sketch* de programação do *Arduino*.



```
SketchBot_RC_ou Bluetooth_testes_V7 | Arduino 1.0.5
Arquivo  Editar  Sketch  Ferramentas  Ajuda
SpyBot_RC_ou Bluetooth_testes_V7
#include <dht.h>
#include <AFMotor.h> //Biblioteca do motor Shield
#include <Servo.h> // Biblioteca servos
#define echoPin 36 //Pino 36 recebe o pulso do echo
#define trigPin 34 //Pino 34 envia o pulso para gerar o echo
#define dht_dpIn A7
#define echoPin 35 //Pino 35 recebe o pulso do echo
#define trigPin 34 //Pino 34 envia o pulso para gerar o echo

int ch1=0, ch2=0, ch3=1500, ch4=1500, ch5=0, ch6=0; // Variáveis que
int CH3, CH4, CH6=0; // Variáveis auxiliares
int Pan=90; // Variável auxiliar para posicionar servo
int Tilt=90; // Variável auxiliar para posicionar servo
int dist=1000;
int i=0;
int i3=0;
float Vbat;
char s[57];
```

Figura 23 - Sketch de programação do *Arduino* - Fonte: Autoria Própria

Basicamente o programa inicializa todas as variáveis e constantes utilizadas no código. Um laço *while* monitora os três canais por onde os controles podem se comunicar, sendo eles: *Serial-1* para o módulo *APC220*, *Serial-2* para o *Bluetooth®* e seis pinos digitais, uma para cada canal do receptor do rádio *Futaba®*. Logo que um destes canais começa a receber dados, o programa define o valor de um *flag* e sai da função *setup*. Ao ingressar na função *loop* serão executados somente os comandos dentro do laço *if* que derem verdadeiro para o *flag* carregado no *setup*. Para o controle através do rádio *Futaba®*, o software foi escrito apenas para o *uC* do *VOR*. O (Apêndice I) apresenta o código completo.

3.4.2. Programa do controle remoto APC-220

Para a utilização do *APC-220* fez-se necessários dois módulos. Um conectado ao *Mega 1280* do *VOR* e o outro conectado a uma placa *Arduino Pro MINI* e esta a um controle de *Playstation 2*. Em síntese o *uC* faz a cada intervalo de 60ms uma varredura do estado de cada botão do controle e carrega uma *string* com 57 posições, iniciada por '*' e terminada por '#', com valores separados por ';'. Essa

string é então transmitida de forma serial para o módulo do *VOR*. O código para leitura do controle foi originalmente criado por *Bill Porter* [10], porém, só escreve na porta serial quando o estado de algum botão é alterado. Para utilização neste projeto, a biblioteca e o corpo do programa foram modificados para que os dados fossem transmitidos sistematicamente e o tamanho da *string* formatado na origem para ser reconhecida no destino. O (Apêndice II) apresenta o código completo utilizado no controle.

Tomando como exemplo a seguinte *string* enviada para comando do *VOR*, temos: `"*0;0;206;000;125;034;0;0;1;1;1;0;1;128;128;090;128;0;1#"`, abaixo pode-se ver o que cada posição representa no mapeamento do estado dos botões:

Tabela 2 - String do controle APC-220 - Fonte: Autoria Própria

Posição	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Dado	*	0	0	206	000	125	034	0	0	1	1	1	0	0	1	128	128	090	128	0	1	#

- 1 - Início da transmissão: ' * ';
- 2 - Botão *Start* : solto=0, pressionado=1;
- 3 - Botão *Select*: solto=0, pressionado=1;
- 4, 5, 6, 7 - Botões direcionais: *PWM* 000 a 255 e correspondem respectivamente a: cima, direita, esquerda, baixo;
- 8 - Clique do *stick* analógico esquerdo no eixo Z: solto=0, pressionado=1;
- 9 - Clique do *stick* analógico direito no eixo Z: solto=0, pressionado=1;
- 10 - Botão *L2* : solto=0, pressionado=1;
- 11 - Botão *R2* : solto=0, pressionado=1;
- 12 - Botão *Triângulo*: solto=0, pressionado=1;
- 13 - Botão *Circulo*: Quando pressionado envia 1 e volta a zero;
- 14 - Botão *Quadrado*: Quando pressionado e solto envia "1" e volta a zero;
- 15 - Botão *X*: Quando pressionado envia 1 e volta a enviar zeros, quando é solto envia novamente 1 e volta a enviar zeros;
- 16, 17 - Respectivamente eixos Y e X do *stick* analógico esquerdo: *PWM* 000 a 255 quando liberados enviam o valor central de 128;
- 18, 19 - Respectivamente eixos Y e X do *stick* analógico direito: *PWM* 000 a 255 quando liberados enviam o valor central de 128;
- 20 - Botão *L1*: solto=0, pressionado=1;
- 21 - Botão *R1*: solto=0, pressionado=1;

22 - Fim da transmissão: '#'.

A Figura 24 mostra a janela do terminal serial imprimindo a identificação do método de controle e a série de *strings* de comando recebidas pelo *uC* do *VOR*.

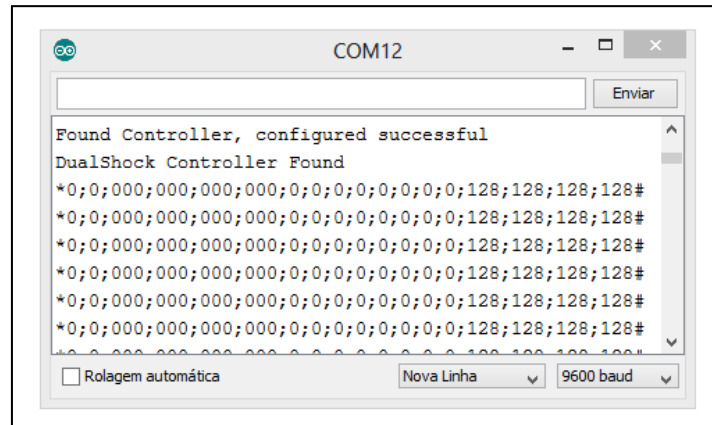


Figura 24 - Janela do terminal imprimindo o recebimento de dados - Fonte: Autoria Própria

3.4.3. Programa do controle via dispositivo Android®

Para o desenvolvimento do aplicativo [11] *Android*® utilizou-se a ferramenta criada pelo *Google Labs* e hoje mantida pelo *MIT (Massachusetts Institute of Technology)* denominada *App Inventor*. Essa interface permite a programação através de um conjunto composto por duas seções: o *App Inventor Designer* e o *App Inventor Blocks Editor*, cada uma com uma função específica. Para utilização dessa ferramenta é necessário fazer *login* no site com uma conta *Google*®. O *Designer* (Figura 25) como o próprio nome sugere trás a tela inicial do projeto, onde é possível "desenhar" o aplicativo, escolhendo a posição dos botões e imagens, inserindo fotos, *check boxes* e outros componentes disponíveis para a construção de um programa.

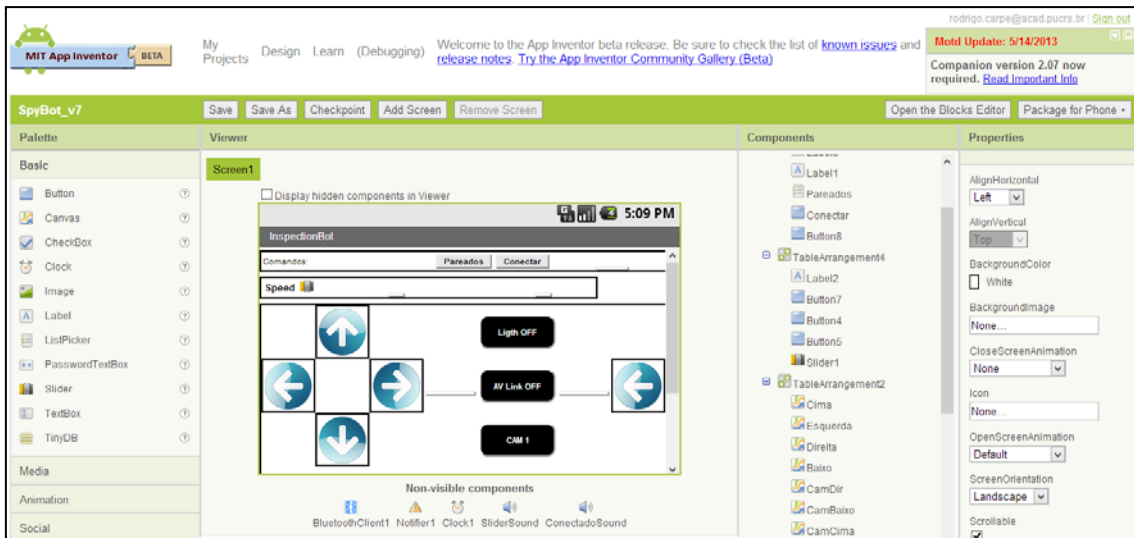


Figura 25 - App Inventor Designer - Fonte: Autoria Própria

Para utilizar o Editor de Blocos (Figura 26) é necessário configurar e instalar alguns aplicativos. O primeiro é o *Java* em sua versão atualizada e o segundo é o aplicativo de desenvolvimento do *App Inventor*. Ele está disponível para *Windows*, *Mac OS* e *GNU/Linux*.

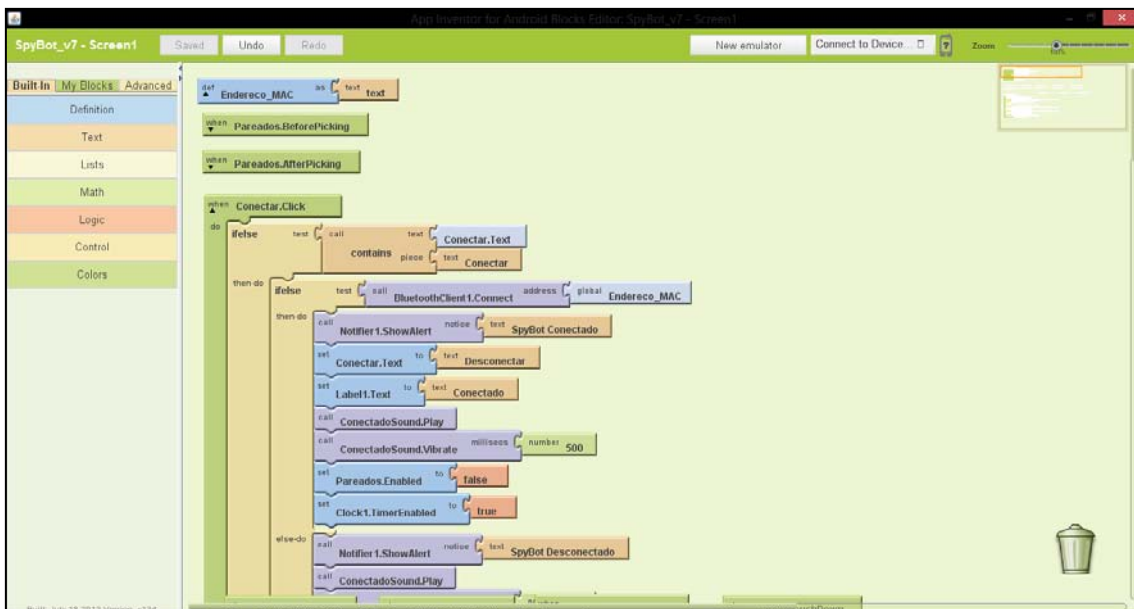


Figura 26 - Editor de Blocos do App Inventor - Fonte: Autoria Própria

O Editor de Blocos permite programar de forma simples adicionando lógica ao aplicativo, nele é possível definir as funções de cada componente criado no

Designer e como eles devem se comportar, implementado o código de forma gráfica, muito similar a um quebra-cabeça. Os testes podem ser simulados através de uma máquina virtual rodando *OS Android®* ou carregado diretamente para o aparelho.

Para os testes funcionais foram utilizados dois dispositivos *Android®*. O primeiro é o celular do fabricante Samsung modelo *Galaxy S3 GT-I9300* com *OS V.4.1.2* e o segundo, do mesmo fabricante é o tablete modelo *Galaxy TAB GT-P6210* com *OS V.3.2*. Os dispositivos são apresentados nas Figuras 27 e 28 respectivamente.



Figura 27 - Galaxy S3 -
Fonte:<http://www.samsung.com/br/consumer>



Figura 28 - Galaxy TAB - Fonte:
<http://www.samsung.com/br/consumer>

3.5. Montagem do protótipo

3.5.1. Mecânica

A montagem mecânica compreende a montagem da estrutura do chassis, fixação dos quatro motores de locomoção, rodas, além da cabeça com os movimentos de *PAN* e *TILT* para câmera de vídeo. A Figura 29 demonstra o início da montagem do protótipo com alguns componentes eletrônicos ainda em fase de testes. Os motores vistos na figura operam em 5V @ 193rpm, porém, mostraram-se incompatíveis em função do baixo torque (1,1kgf.m). Foram substituídos pelos especificados no subitem 2.2.4.



Figura 29 - Início da montagem eletrônica - Fonte: Aatoria Própria

A Figura 30 trás o aspecto do VOR pronto para iniciar as etapas de testes. As antenas dos módulos APC 220 e transmissor de vídeo foram posicionadas externamente para melhor desempenho. Para o módulo *bluetooth*® e antena do receptor do rádio *Futaba*® optou-se pela utilização de uma caixa plástica visando a melhor relação "proteção mecânica" VS "atenuação de sinal".



Figura 30 - Primeiros testes - Fonte: Aatoria Própria

3.5.2. Eletrônica

Para um melhor entendimento das conexões elétricas do projeto, apresenta-se um diagrama de blocos simplificado. Na Figura 31 as linhas em "azul" representam sinais e as linhas em "vermelho" representam fornecimento de energia.

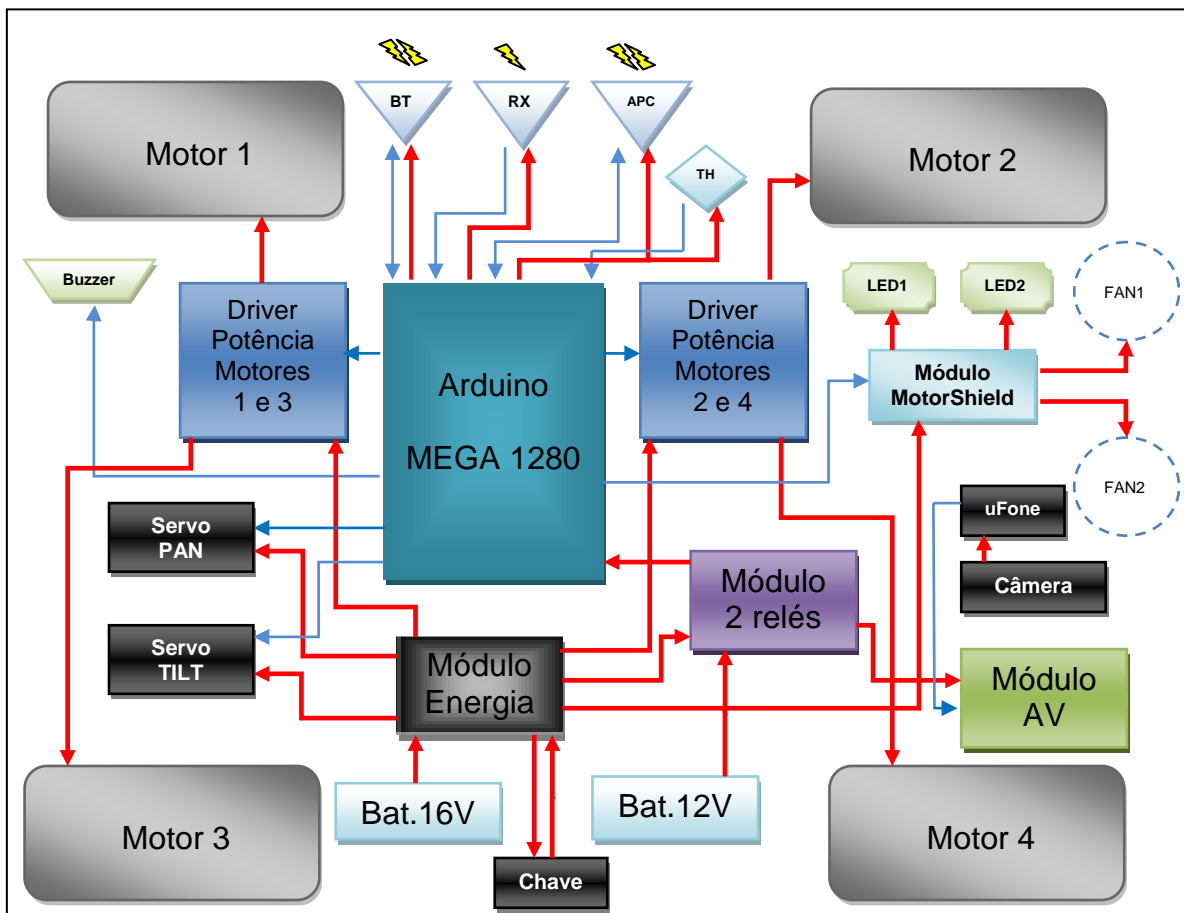


Figura 31 - Diagrama de conexões elétricas - Fonte: Autoria Própria

A eletrônica foi completamente embarcada, com destaque para a grande quantidade de fios e cabos utilizados, tornando-se em certos momentos difícil manter a organização e um padrão de cores para diferenciar fios de alimentação de fios de sinais. Isso ocorre em parte por tratar-se de um protótipo montado com *shields* de sobreposição e placas de uso genérico que acabam por ocupar um espaço maior que um *hardware* eletrônico dedicado ocuparia. Na Figura 32 pode-se ver a montagem eletrônica finalizada.

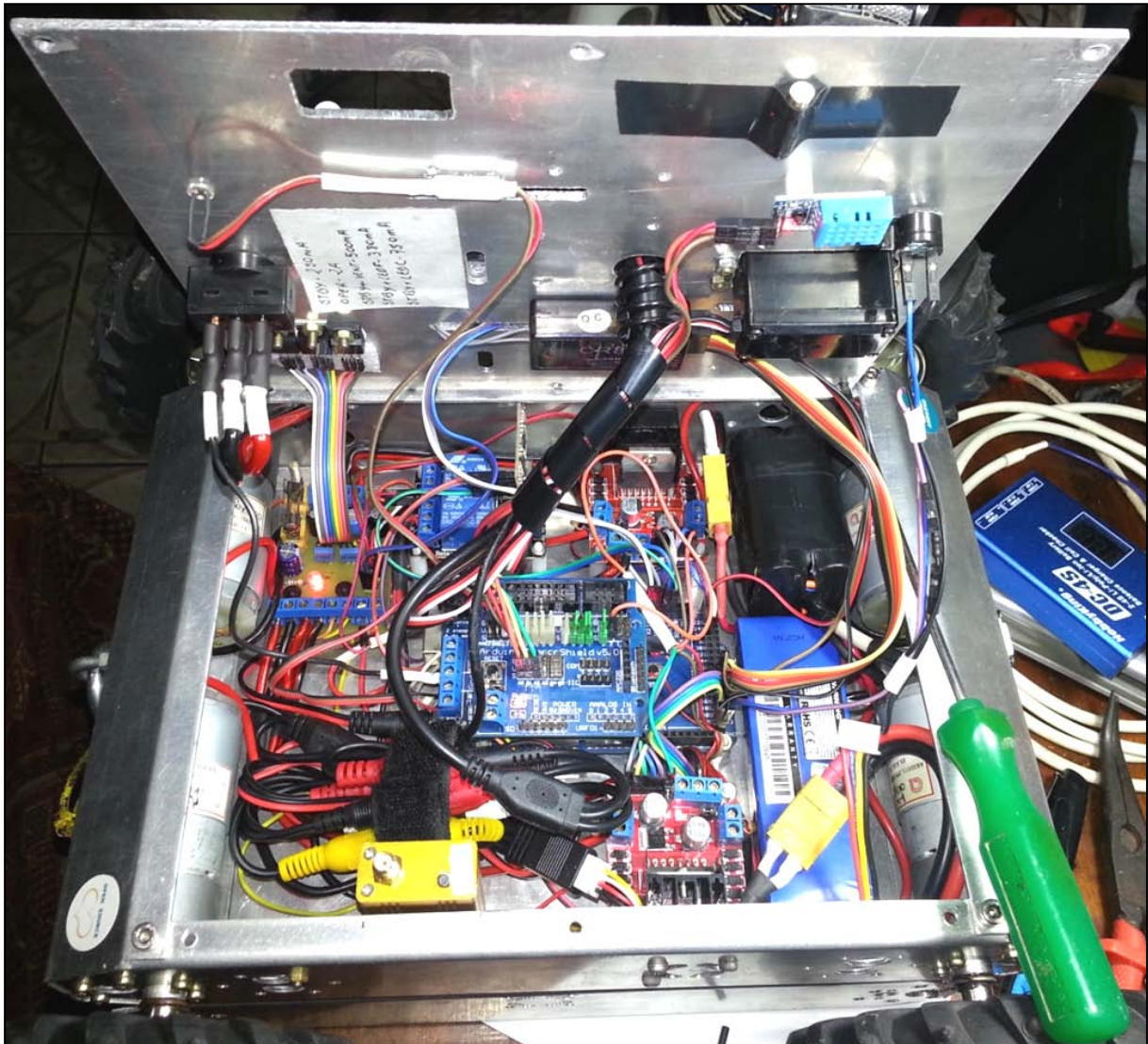


Figura 32 - Aspecto da montagem eletrônica - Fonte: Autoria Própria

4. RESULTADOS

No decorrer dos testes o modelo apresentou-se funcional, atendendo as expectativas. A proposta deste trabalho foi chegar em um protótipo funcional e de baixo custo. A Tabela 3 apresenta o preço dos componentes utilizados, considerando terem sido adquiridos no mercado nacional, sem frete, e referenciados ao site *Mercado Livre*. Nenhuma das opções inclui o monitor de vídeo.

Tabela 3 - Pacote básico de componentes VOR exceto controles

ÍTEM	QUANTIDADE	PREÇO (R\$)	OBSERVAÇÕES
Arduino Mega 1280	1	75,90	Mega 2560
Motor Shield	1	34,90	
Sensor Shield V5.0	1	24,90	
Drive Ponte H - L298	2	50,00	
Mini cooler	2	86,00	
Servo Standard	2	52,00	
Câmera	1	45,00	
Microfone amplificado	1	17,50	
Transmissor/Receptor AV	1	150,00	
LED 10W	1	12,00	
LED 1W	1	3,80	
Sensor de temp. e umid.	1	16,00	
Placa regulação tensão	1	35,00	
Motores	4	160,00	
Bateria	2	150,00	
Chassi e suportes PAN/TILT	1	350,00	Estimado
Rodas	4	300,00	
Módulo relés	1	24,90	
Carregador baterias	1	50,00	
Chaves, parafusos, fios e cabos	1	100,00	
TOTAL		1737,90	

A Tabela 4 apresenta os componentes necessários para implementação do controle via *Bluetooth®*, partindo-se da premissa que o operador já possua um dispositivo *Android®*.

Tabela 4 - Preços para o método de controle por Bluetooth

ÍTEM	QUANTIDADE	PREÇO (R\$)	OBSERVAÇÕES
Módulo Bluetooth Slave	1	34,90	
TOTAL		34,90	

Para implementação do sistema de controle através do rádio *Futaba®*, são necessários os hardwares mostrados na Tabela 5.

Tabela 5 - Preços para o método de controle por rádio Futaba

ÍTEM	QUANTIDADE	PREÇO (R\$)	OBSERVAÇÕES
Rádio Futaba 6 canais com receptor	1	850,00	Equivalente
TOTAL		850,00	

A Tabela 6 trás os preços para utilização do controle através do módulo de rádio APC-220.

Tabela 6 - Preços para o método de controle por rádio APC-220

ÍTEM	QUANTIDADE	PREÇO (R\$)	OBSERVAÇÕES
Rádio APC-220 TX / RX	1	149,00	
Controle Playstation 2	1	49,00	Genérico
TOTAL		198,00	

O método de controle via *Bluetooth®* é sem dúvida a maneira mais barata de implementação, e mostrou-se funcional e confiável. Considerando que o operador já possua um dispositivo *Android®*, tal como um *smartphone* ou *tablete*, possibilita o controle de inúmeros canais, porém, tem seu alcance limitado. Já o método de controle através do rádio *Futaba®* mostrou-se eficiente e estável, porém, é limitado

em 6 canais e seu custo é muito alto. Em uma análise de custo-benefício, o método de controle com melhor relação sem dúvida fica por conta do módulo de rádio APC-220 com controle de *Playstation 2*, pois permite o controle de até 20 canais em distâncias que alcançam 1000m e com custo reduzido. O valor total para implementação do protótipo VOR com este controle é de R\$ 1.935,90. É preciso ressaltar que mesmo tratando-se de um protótipo e de ter utilizado componentes genéricos seu preço final ainda permanece baixo, quando comparado aos modelos apresentados no início deste trabalho. Uma futura implementação com as placas eletrônicas dedicadas e a produção em série poderia reduzir consideravelmente seu preço.

5. CONCLUSÃO

Este trabalho englobou diversas áreas do curso de engenharia elétrica, permitindo aplicar os conhecimentos teóricos e exigindo a busca de novos conhecimentos para o desenvolvimento de um protótipo que forneceu rápido retorno visual de seu comportamento, sendo possível reprogramá-lo quantas vezes forem necessárias para que o robô atinja o resultado desejado.

Do ponto de vista experimental, o sistema desenvolvido atende as metas estabelecidas que agregam valor funcional ao VOR. É importante ressaltar que o modelo desenvolvido mostra o potencial da solução baseada em controle e integração de sistemas embarcados, além de dispositivos móveis, sobretudo, utilizando ambientes de desenvolvimento Open Source. Perspectivas futuras para o trabalho incluem o projeto de hardware eletrônico dedicado, elaboração de ferramentas acopláveis a estrutura, diminuição do chassi e telemetria visualizada diretamente no monitor de vídeo.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] <http://amyshirateitel.com/2012/09/28/the-cost-of-curiosity/>
- [2] <http://mars.nasa.gov/msl/mission/rover/>
- [3] http://www.nobelprize.org/nobel_prizes/physics/laureates/1956/
- [4] <http://seriesestatisticas.ibge.gov.br/series.aspx?vcodigo=SCN55&t=produto-interno-bruto-capita>
- [5] <http://www.robotshop.com/superdroid-um4-spektrum-remote-1.html>
- [6] <http://www.robotshop.com/trackbot-surveillance-inspection-tracked-robot-1.html>
- [7] <http://www.brasilrobots.com.br/vor42/index.html>
- [8] A STUDY OF BLUETOOTH APPLICATION FOR REMOTE CONTROLLING OF MOBILE EMBEDDED SYSTEMS - Daniel Mathias Silva dos Santos, Katielle Dantas Oliveira, Vinicius Carvalho Marques, Jorge Fredericson de Macedo Costa da Silva, Thiago Oliveira Rodrigues, Jose Wally Mendonça Menezes - 2012 IEEE Brazilian Symposium on Computing System Engineering - Disponível em: <http://propi.ifto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/4270/2567>
- [9] <http://pdf1.alldatasheet.com/datasheet-pdf/view/105712/FAIRCHILD/LM7812.html>
- [10] <http://www.billporter.info/2010/06/05/playstation-2-controller-arduino-library-v1-0/>
- [11] Apostila e notas de aula da disciplina de Sistemas Embarcados - Prof. Anderson Terroso 2013/1