

ÍNDICE

	Pág.
Introdução	2
Execução do Projecto	
Lista de Material	3
Iniciação e Pesquisa	4 a 5
Execução Propriamente Dita	
Programação	7 e 8
Transmissão de Dados	9 a 17
Estrutura Mecânica e Electrónica	17 a 20
Ligações	21
Análise Crítica e Global da Apresentação do Projecto	22
Agradecimentos e Sites	23 e 24

INTRODUÇÃO

O meu projecto chama-se ExplorUino®, posso dizer que é uma Sonda Móvel, ou seja é um carro que se move com comandos do utilizador que são dados a partir do computador com um software próprio. O carro tem também um sensor de temperatura e esse valor pode ser lido no software.

Escolhi este projecto por vários motivos, mas o principal foi o meu gosto pela robótica e pelo desenvolvimento de ideias novas que podem resolver problemas no dia-a-dia. Este projecto pode ser utilizado por variadas coisas, por exemplo, pelos Bombeiros, Polícia, Forças Armadas, etc. Um projecto deste tipo pode ser aplicado em catástrofes naturais, como por exemplo terramotos, em que um ser humano não consegue entrar em determinados sítios, e assim pode ser substituído por um carro como este.

Este projecto utiliza duas plataformas de programação, o Python e Linguagem C++. O Python é uma linguagem de alto nível e no projecto é responsável pela parte do software no computador para controlar o carro, utilizei também duas bibliotecas no python, o Pygame (ler os dados do teclado e imprimir imagens no ecrã) e o Pyserial (enviar e receber dados pela porta serie (usb)). A Linguagem C++, utilizei para programar o carro, ou seja o Arduíno.

EXECUÇÃO DO PROJECTO

LISTA DE MATERIAL

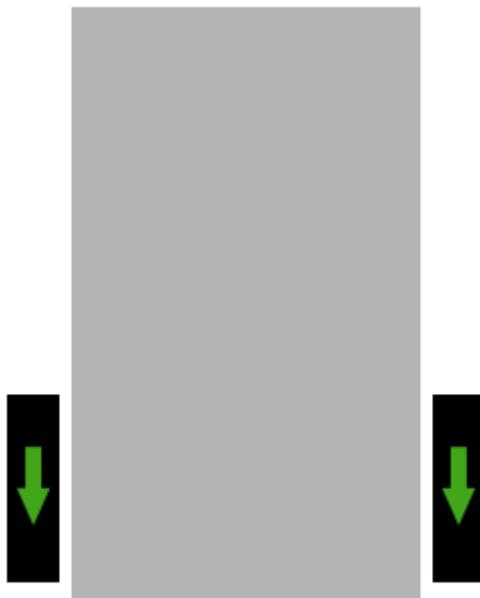
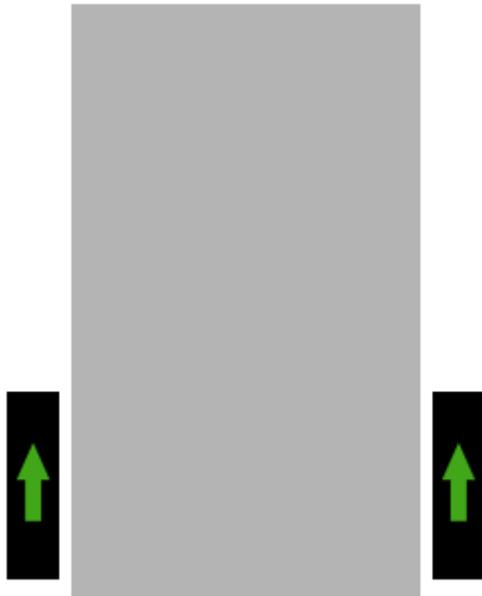
- Arduino Duemilanove
- Arduino Xbee shield without RF module
- 2 XBee 1mW Wire Antenna (802.15.4)
- XBee Explorer USB
- 2 Servos Motores desbloqueados
- Lm35
- L293D
- Bateria
- Acrílico
- Madeira
- Parafusos
- Porcas
- Anilhas
- Rodas
- Fios
- Ferro de soldar
- Solda
- Interruptor
- Berbequim

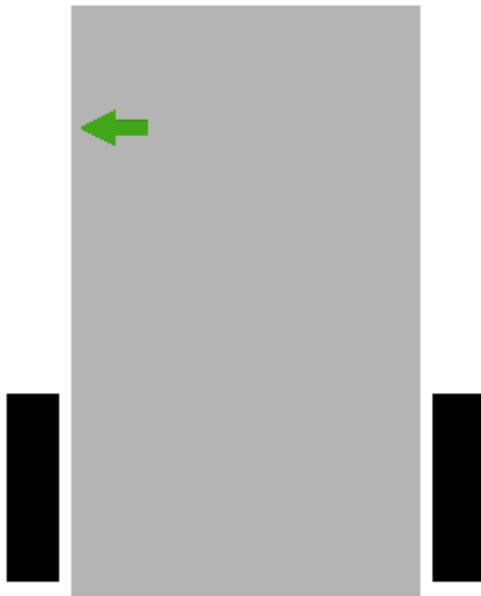
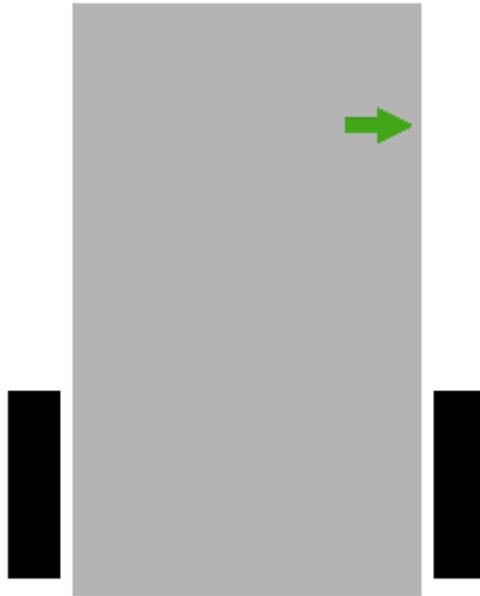
INICIAÇÃO E PESQUISA

No início do ano quando estive a pensar na PAP eu não sabia bem o que tinha de fazer, sabia que tinha de ser um robô. Mas perguntava-me a mim próprio... “Vou fazer um robô mas que vai fazer o robô?”, essa era a grande questão e andei a procura na internet de ideias de aplicações para o tal robô. Até que no You Tube vi um filme de um projecto da Universidade de Viana do Castelo que era um carro comandado remotamente e que fazia muitas coisas, tinha sensores de temperatura, humidade, pressão, tinha uma câmara e era controlado a partir do computador com o software. A partir daí já sabia o que ia fazer, mas depois surgiram outras perguntas... “como vou fazer?”, “o que vou utilizar para controlar o carro?”, “mil e uma” dúvidas surgiram na minha cabeça.

Depois comecei a pesquisar na internet como resolver esses problemas e decidi controlar o carro a partir do Arduino que é um projecto open-source de protótipos de electrónica baseados numa plataforma de hardware e software flexível e fácil de utilizar. É destinado a artistas, designers, hobbyistas e qualquer tipo de pessoa interessada em criar objectos ou ambientes interactivos. O microcontrolador em que se baseia é programável usando a linguagem Arduino (baseada em C/C++), e também aceita código directamente em C/C++, bem como o ambiente do Arduino que é baseado em Processing. Os projectos em Arduino podem ser stand-alone ou podem comunicar com outros circuitos, ou até mesmo com outros softwares a correr num computador (por exemplo, Java, Flash, Processing, MaxMSP). Encontrei também linguagem de programação muito acessível e muito versátil, pois pode-se “mexer” com muita coisa a partir daquela linguagem e era fácil de trocar informação com o arduino. Essa linguagem é o Python que me permitiu desenvolver toda a parte de controle do carro a partir do teclado do computador e ainda ter um software visível para saber o que se estava a passar no carro, ou seja:

- Saber a direcção e sentido do carro





- Saber a temperatura do local onde o carro se encontra

TEMPERATURA
29.3 °C

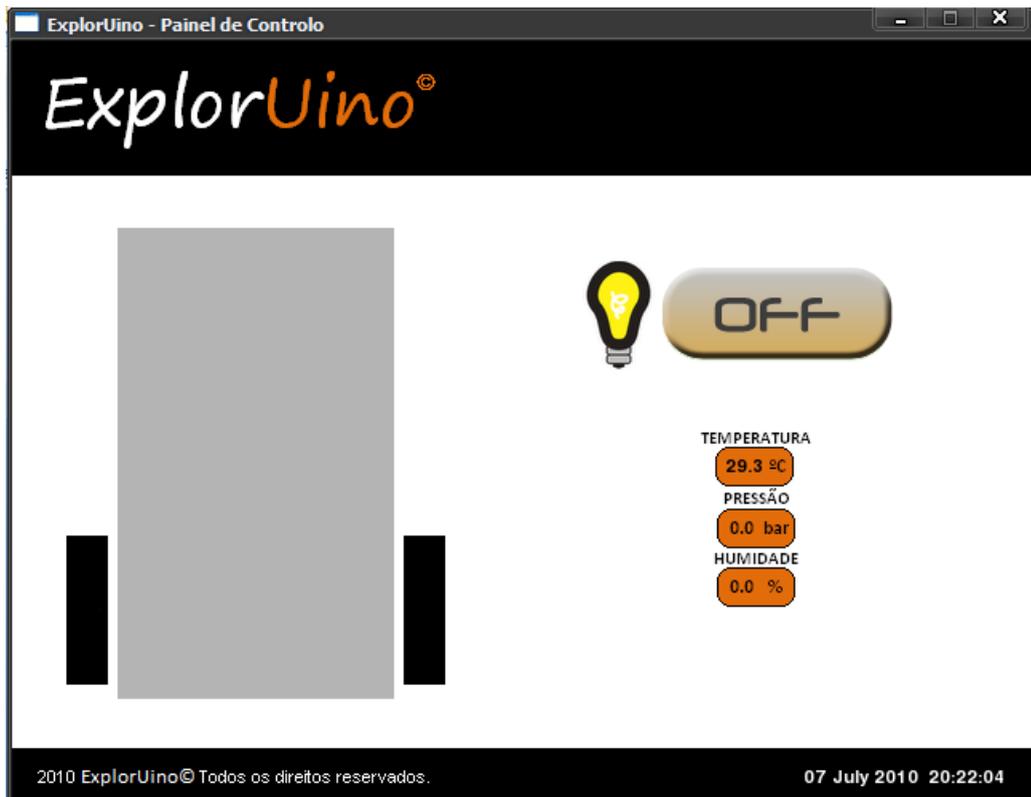
EXECUÇÃO PROPIAMENTE DITA

(PROGRAMAÇÃO)

A primeira coisa que fiz foi o desenvolvimento do programa em Python para o controlo do carro. Então essa programação consiste basicamente em ler os dados do teclado e imprimir as imagens conforme a tecla que estiver a ser pressionada, enviar essa ordem pela porta serie (usb) para o carro e receber os dados que vem do sensor que está no carro.

Essa informação vai ser enviada por strings, ou seja em forma de palavras ou bytes, pela porta usb, neste caso por exemplo, se carregar na seta para cima o que vai enviar vai ser o numero 2 e esse numero vai ser enviado em ASCII para o carro.

- Janela de visualização do software feito em Python



Numa segunda fase foi o desenvolvimento do programa para o Arduino que foi feito em linguagem C++. Esse programa consiste em receber os dados que são enviados do computador, para controlar a direcção e sentido do carro, e em enviar o valor do sensor de temperatura para o computador.

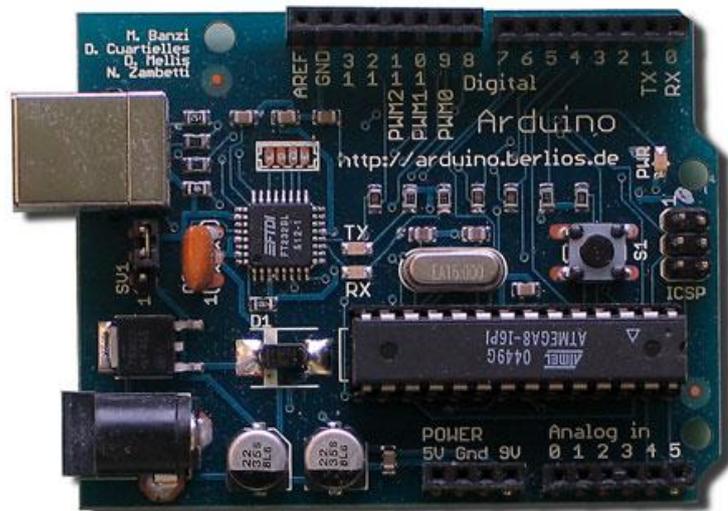
Os dados que recebo para o controlo dos movimentos do carro vem no código ASCII. Então, para saber o valor que estou a receber vou fazer esta comparação:

```
if ((inString[0] == 0 + 0x30) && (inString[1] == 0 + 0x30))
```

Logo, essa é a maneira como vou saber o valor que estou a enviar no computador.

O funcionamento é "inString[0] == 0" o "0" é o valor que vou comparar para saber se recebo aquele 0, ou não. Como os dados vem em ASCII vou ter de somar aqueles "0x30" que é um valor em Hexadecimal que equivale em decimal ao número 48 que, se formos ver na tabela ASCII é o numero 0. E assim somando esse valor conseguimos saber qualquer valor de 0 a 9 em ASCII.

Decimal	HEX	Carácter
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9



Exemplo:

Saber o número 7

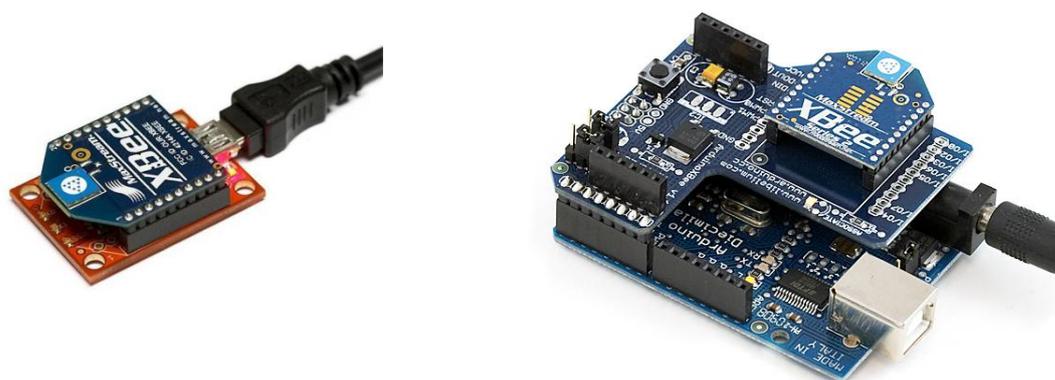
$$7 + 48 = 55$$

Se formos ver a tabela o numero 55 em decimal vai ser equivalente ao carácter número 7.

(TRANSMISSÃO DOS DADOS)

O carro vai andar de forma independente, ou seja, “sem fios”. Os dados vão ser transmitidos por rádio frequência pelo Xbee . O Xbee é um pequeno, mas completo, transmissor/receptor ZigBee. Há duas versões disponíveis: o Xbee e o Xbee PRO. São versões parecidas e perfeitamente compatíveis entre si. A única diferença reside na potência de transmissão: 1 mW para o módulo Xbee e 63 mW para o módulo Xbee PRO. O alcance do produto final depende da potência de transmissão.

Eu vou utilizei dois Xbee's serie 1 de 1mW que tem um alcance de 100 metros em terreno aberto e 30 metros em zonas fechadas. Um Xbee para ligar ao Computador e outro para ligar ao Arduino. Para ligar ao computador utilizei uma placa chamada Xbee Explorer usb e para o Arduino utilizei uma shield própria que existe para se poder ligar o Xbee ao Arduino.

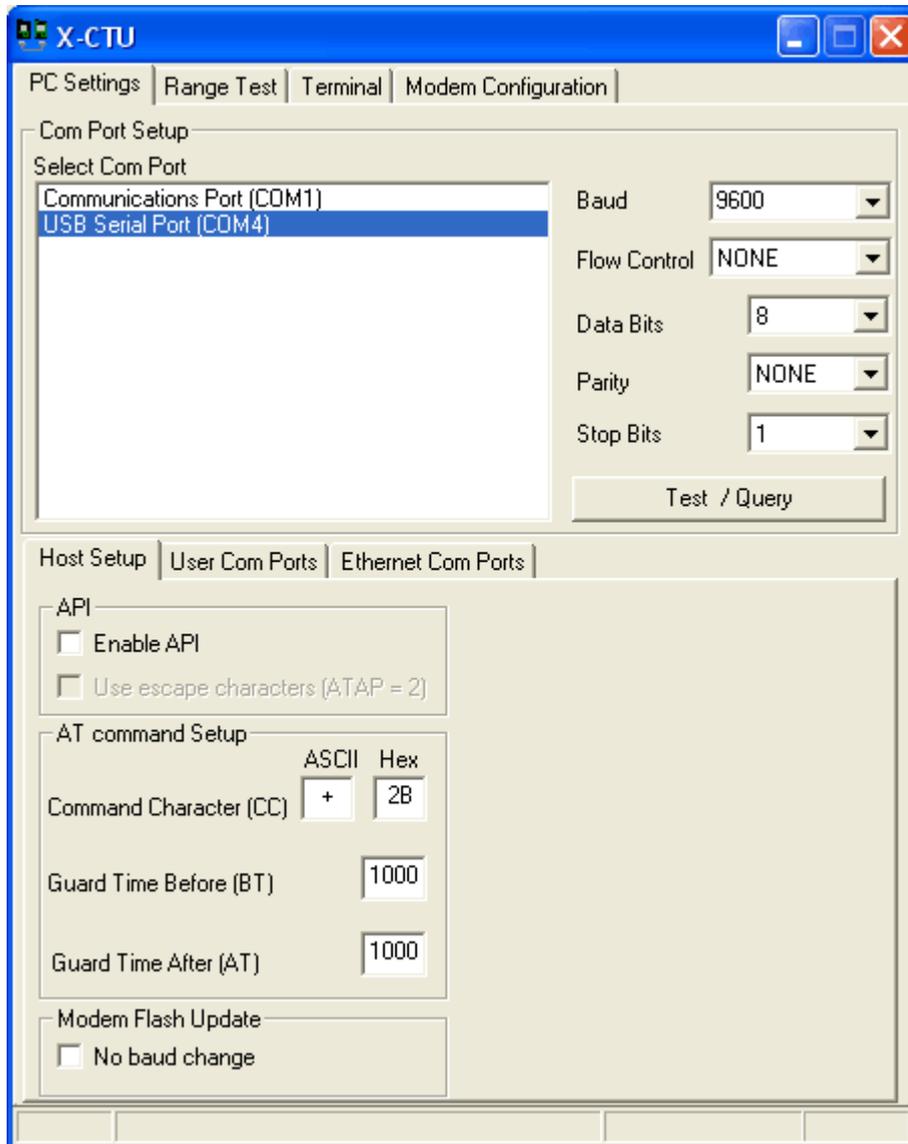


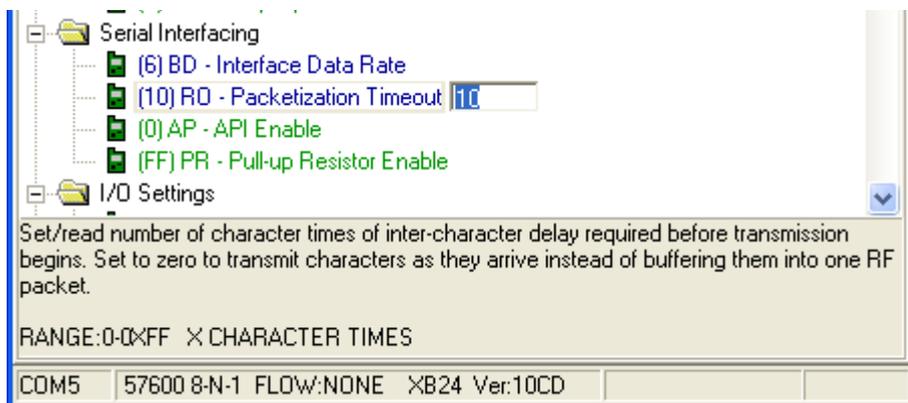
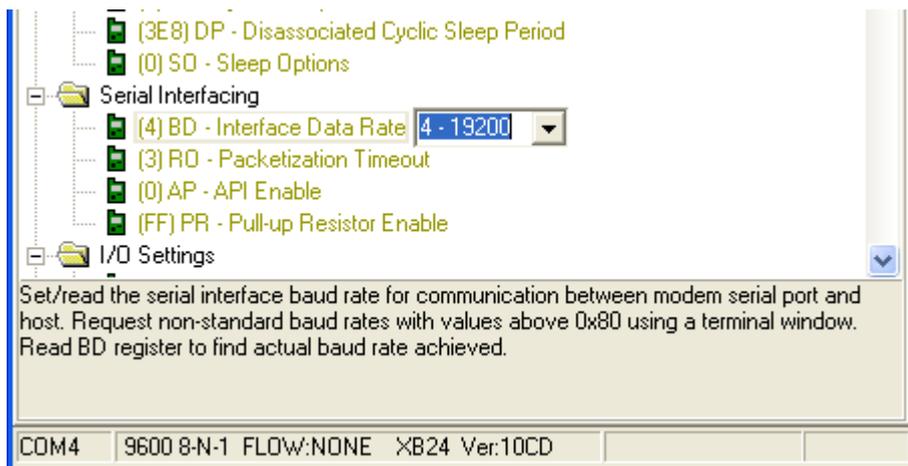
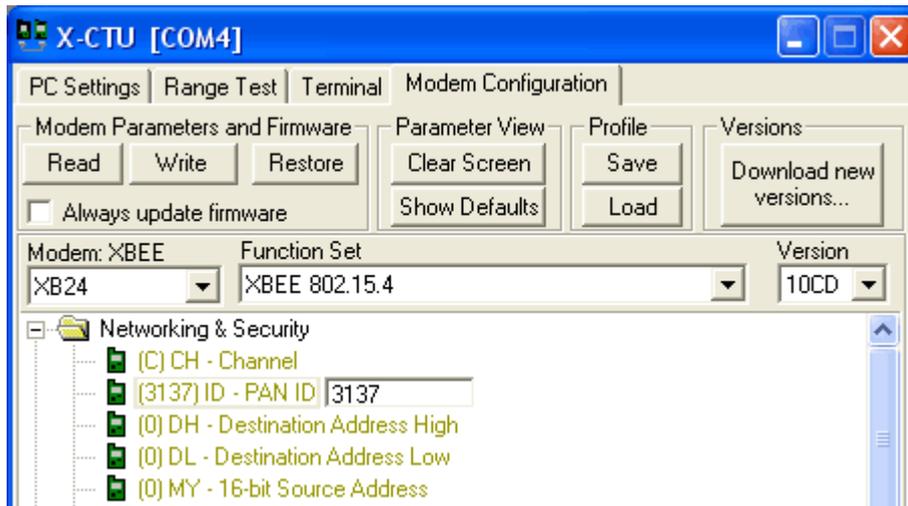
Os Xbee são muito versáteis e podem ser utilizados mais que um xbee numa aplicação e então podem ser configurados de várias maneiras: podem estar numa rede em estrela, em árvore, etc.

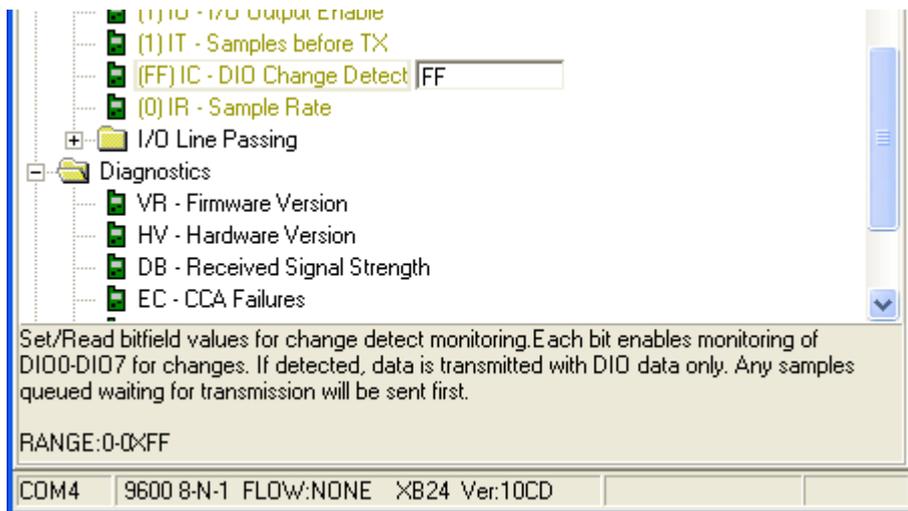
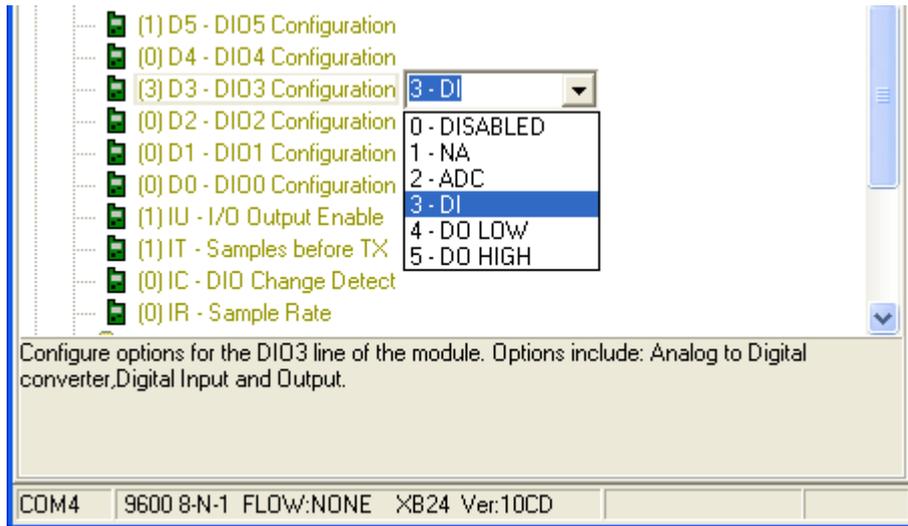
Para configurar os Xbee's utilizei um programa open source chamado X-CTU. Com esse programa conseguimos configurar os Xbee's de forma a que seja necessário.

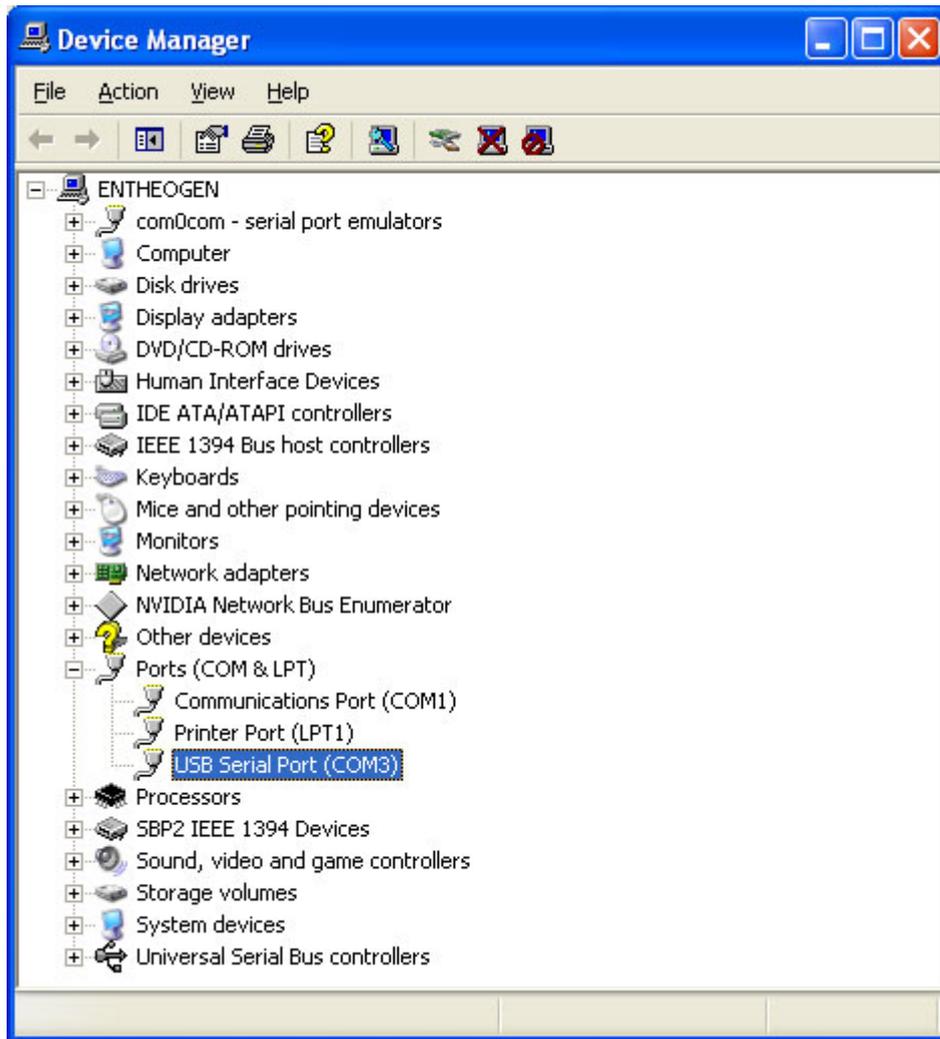
Para configurar os dois Xbee's utilizei um tutorial feito passo a passo:

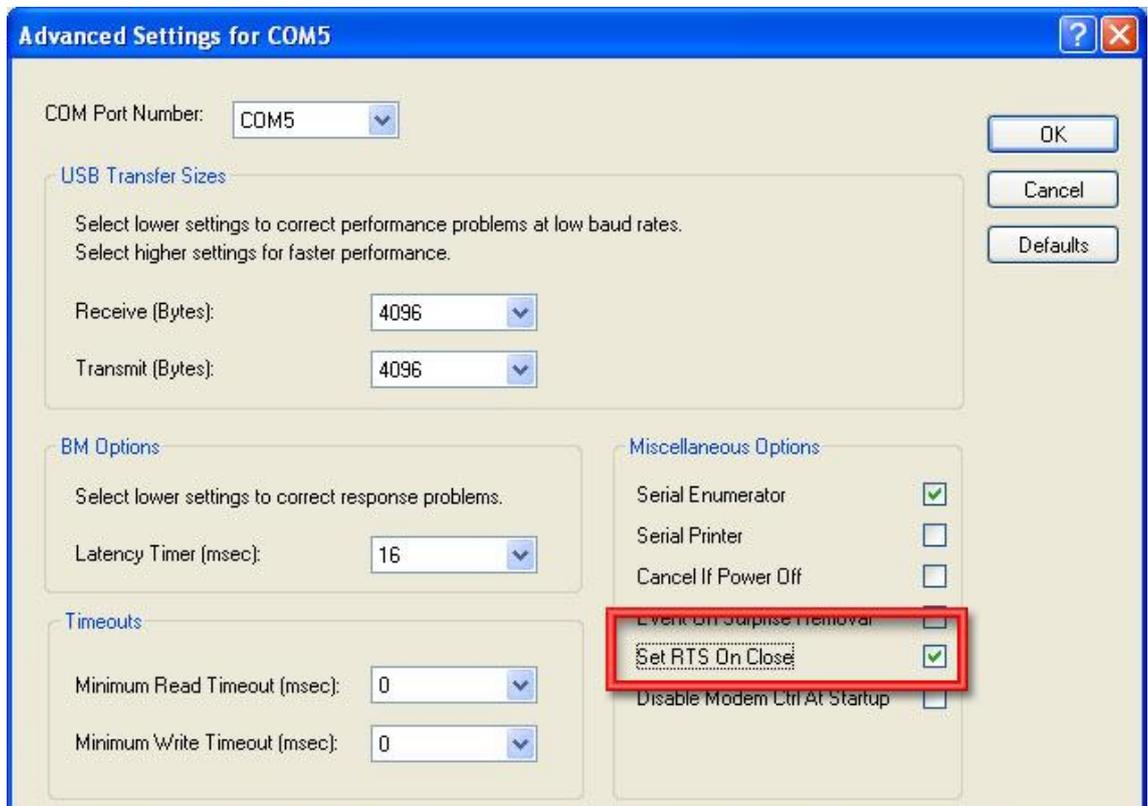
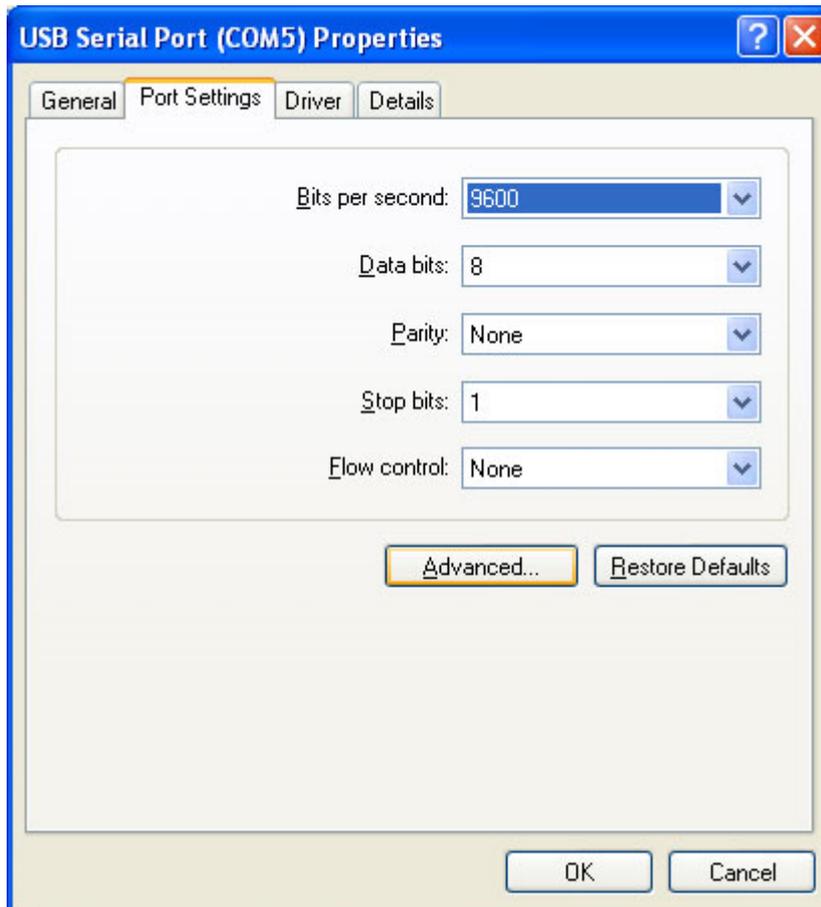
➤ Xbee computador



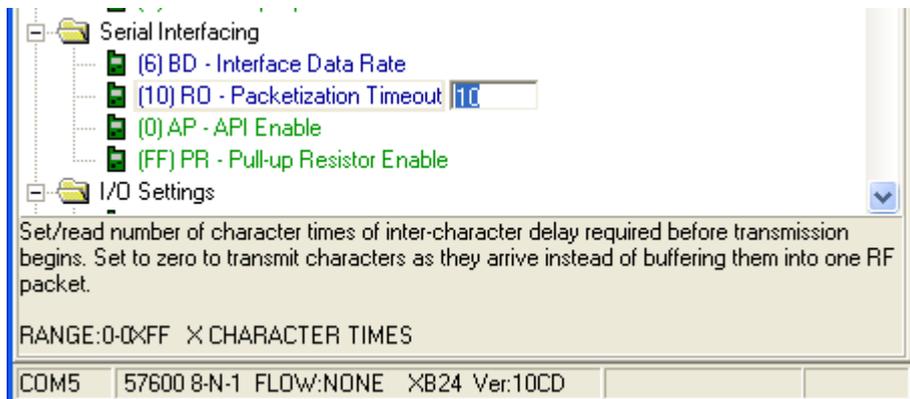
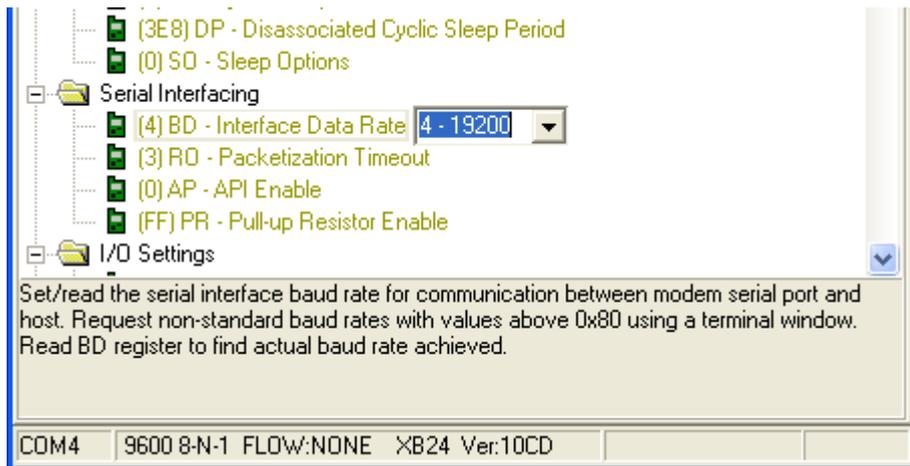
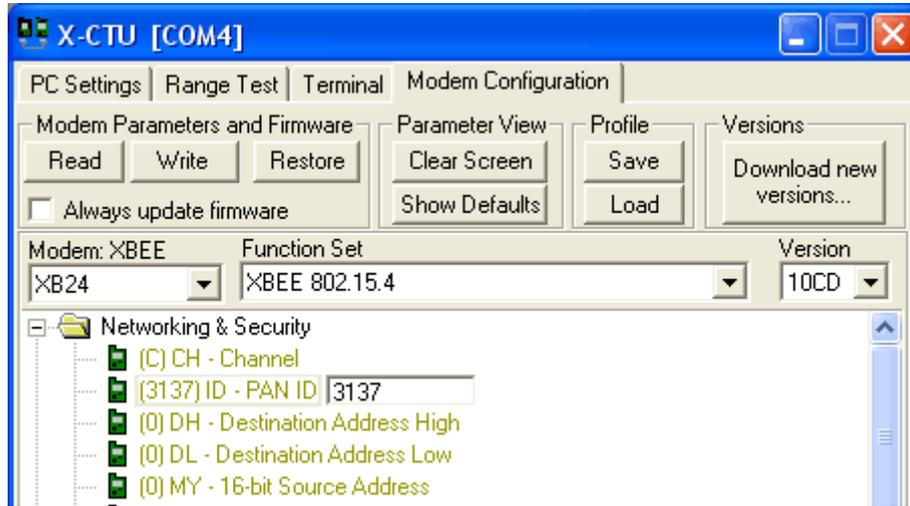








➤ Xbee Arduino



(1) D5 - DIO5 Configuration
 (0) D4 - DIO4 Configuration
 (0) D3 - DIO3 Configuration **0 - DISABLED**
 (0) D2 - DIO2 Configuration 0 - DISABLED
 (0) D1 - DIO1 Configuration 1 - NA
 (0) D0 - DIO0 Configuration 2 - ADC
 (1) IU - I/O Output Enable 3 - DI
 (1) IT - Samples before TX 4 - DO LOW
 (0) IC - DIO Change Detect 5 - DO HIGH
 (0) IR - Sample Rate

Configure options for the DIO3 line of the module. Options include: Analog to Digital converter, Digital Input and Output.

COM20 19200 8-N-1 FLOW:NONE XB24 Ver:10CD

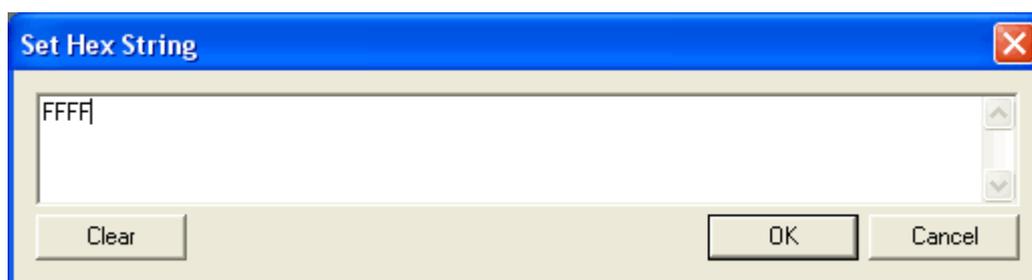
(0) D2 - DIO2 Configuration
 (0) D1 - DIO1 Configuration
 (0) D0 - DIO0 Configuration
 (0) IU - I/O Output Enable **0 - DISABLED**
 (1) IT - Samples before TX 0 - DISABLED
 (0) IC - DIO Change Detect 1 - ENABLED
 (0) IR - Sample Rate

Enables I/O data received to be sent out UART. The data is sent using an API frame regardless of the current ATAP mode.

COM20 19200 8-N-1 FLOW:NONE XB24 Ver:10CD

(0) IU - I/O Output Enable
 (1) IT - Samples before TX
 (0) IC - DIO Change Detect
 (0) IR - Sample Rate
 I/O Line Passing

- [FFFF] IA - I/O Input Address
- (FF) T0 - D0 Output Timeout
- (FF) T1 - D1 Output Timeout
- (FF) T2 - D2 Output Timeout
- (FF) T3 - D3 Output Timeout
- (FF) T4 - D4 Output Timeout
- (FF) T5 - D5 Output Timeout



Estes foram todos os passos que fiz para configurar os Xbee's. Tive algumas dificuldades em configurar mas depois consegui configurar a segunda tentativa.

(ESTRUTURA, MECÂNICA E ELECTRÓNICA)

A Ultima parte do projecto foi a realização da estrutura para o carro final, pois já tinha feito uma estrutura provisória de testes. A estrutura do carro é feita em acrílico e madeira para o suporte dos motores.

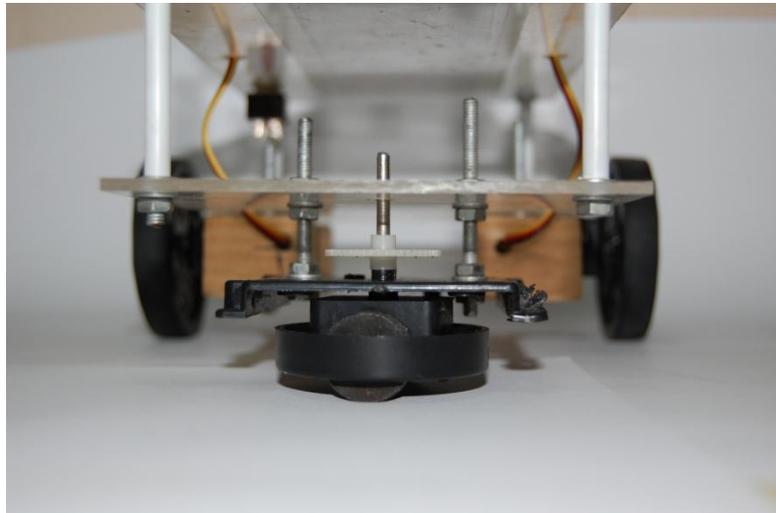
A estrutura foi das partes mais fáceis de realizar pois já tinha tudo projectado à muito tempo. No que tive mais dificuldades foi na roda da frente, pois não queria por um rolon porque ficava muito feio e também nestes projectos são muitos utilizados, então eu quis inovar um pouco. Andei durante uns dias a pensar e à procura nos carrinhos de brincar do meu irmão pequeno até que descobri um carro daqueles a pilhas que andam de um lado para o outro e esses movimentos são feitos de uma roda rotativa que eu achei a ideia perfeita e assim implementei-a.

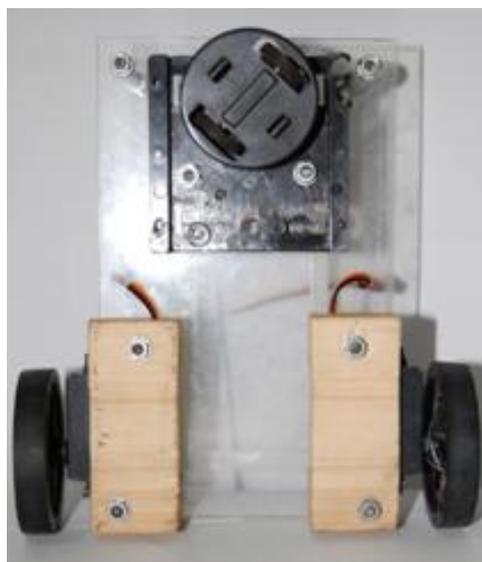
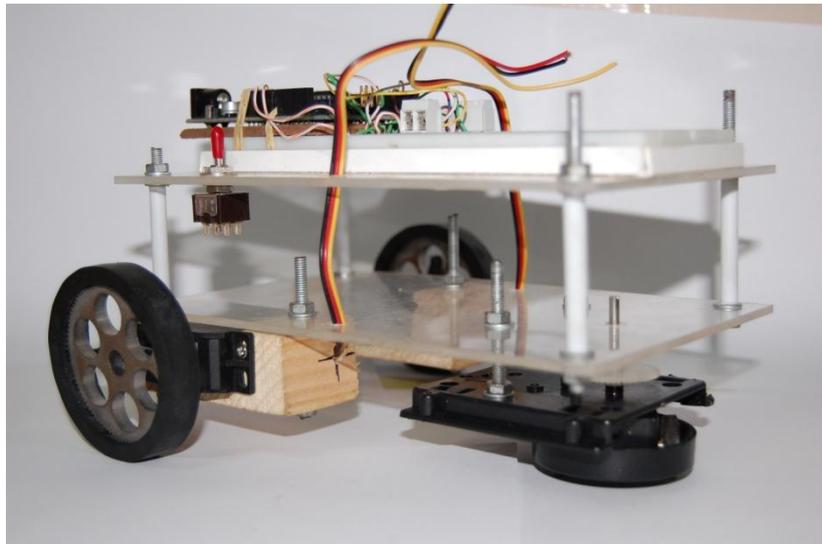
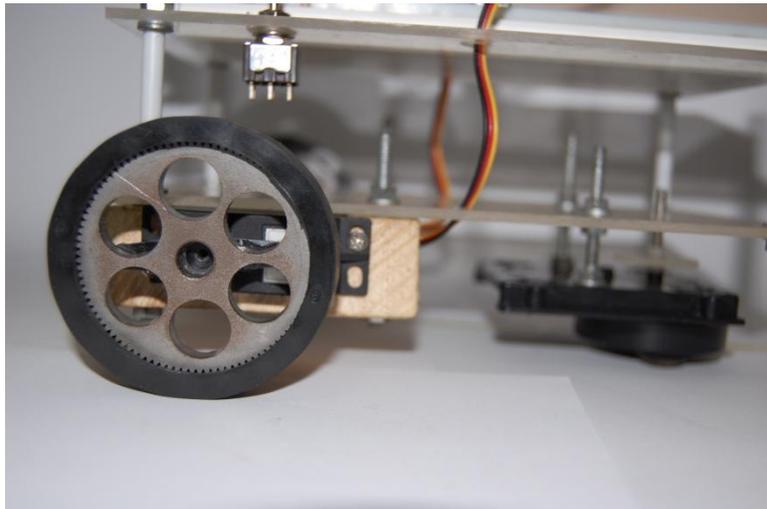
Resultados:

- Carro provisório



- Carro final





(LIGAÇÕES)

	Digitais	Analogicas
0 (rx)	xbee	temperatura
1 (tx)	xbee	
2	Luz	
3 (pwn)	Motor esq	
4	Motor esq	
5 (pwn)	pwm motor esq	
6 (pwn)	pwm motor dir	
7	motor dir	
8	Motor dir	
9 (pwn)		
10 (pwn)		
11 (pwn)		
12		
13		

ANALÍSE CRÍTICA E GLOBAL DA APRESENTAÇÃO DO PROJECTO

Neste projecto deparei-me com varias dificuldades durante toda a realização do projecto. Mesmo assim, consegui resolver todos os problemas, umas vezes por tentativa e erro, outros por pesquisa e outros pelo fórum Luso Robótica, mas de forma geral foram problemas fáceis de resolver e também foram problemas em que consegui aprender algo que não sabia e isso para mim foi muito bom.

A meu ver o meu projecto é muito interessante e tem muitas utilizações práticas. A minha avaliação é muito positiva, pois consegui superar tudo aquilo a que me propus e realizei o meu trabalho praticamente sozinho, o que fez com que aprendesse muito.

AGRADECIMENTOS E SITES

➤ AGRADECIMENTOS:

- Dário Oliveira (eu_mesmosp@hotmail.com)
- Ricardo Dias (microbyte – Luso Robótica)

➤ SITES:

- <http://lusorobotica.com>



- <http://www.rogercom.com/>



- <http://inmotion.pt/store/>



- <http://www.ladyada.net/>

ExplorUino[®]

OBRIGADO!

