

Comunicação via RF

Manoel Alexandre Vieira
Graduando em Eng. de Computação

Sumário

- **O que é Rádio Frequência?**
- **Qual a importância no Futebol de Robôs?**
- **Comunicação Serial de Dados**
- **Circuitos Utilizados**
- **Como Enviar Dados**
- **Como Receber Dados**
- **Demonstração**

O que é Rádio Frequência?

- Rádio é um recurso tecnológico das telecomunicações utilizado para propiciar comunicação por intermédio da transcepção de informações previamente codificadas em sinal eletromagnético que se propaga através do espaço.
- Os três elementos básicos que permitem a comunicação entre dois pontos são:
 - O Transmissor;
 - O Meio de Transmissão ;
 - O Receptor;

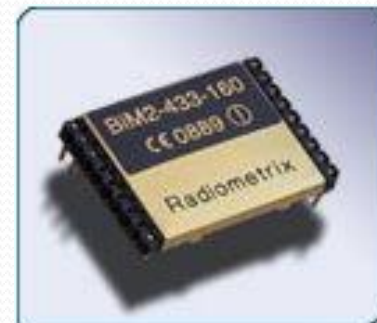
Dispositivos de Transmissão

➤ O transmissor: converte sinais sonoros, analógicos ou digitais em ondas eletromagnéticas, enviando-os para o espaço através de uma antena transmissora, para serem recebidos por um receptor;



➤ O Receptor: responsável pela decodificação dos sinais eletromagnéticos recebidos do espaço, captados pela antena, transformando-os em ondas sonoras, sinais digitais e/ou analógicos.

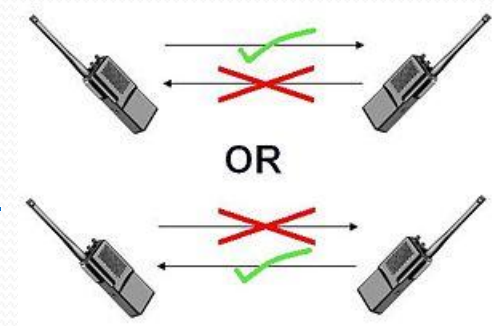
➤ O Transceptor: O radio-transceptor, funciona das duas formas, como transmissor e receptor.



Modos de Transmissão

➤ Simplex: quando a comunicação é unidirecional no meio de transmissão;

➤ Semi-Duplex: quando a transmissão é bidirecional, porém é executada alternadamente em cada sentido.

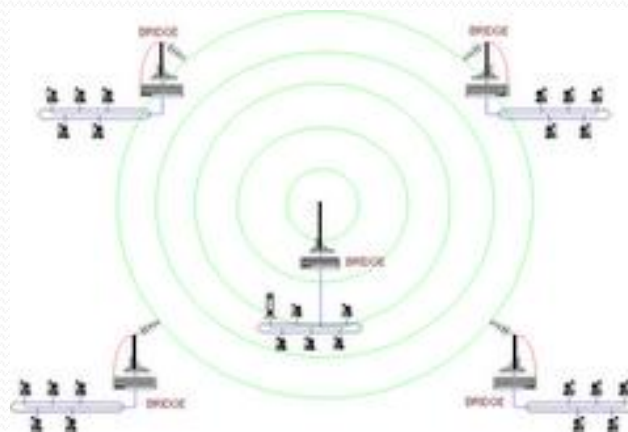


➤ Duplex: quando a comunicação é bidirecional e simultânea.



Modos de Transmissão

- O sistema de transmissão via radio ainda pode ser classificado como Direcional e Omnidirecional;
- O direcional: privilegia um destinatário em detrimento dos outros.
- O Omnidirecional: tem como filosofia distribuir o sinal pelo maior número de usuários.

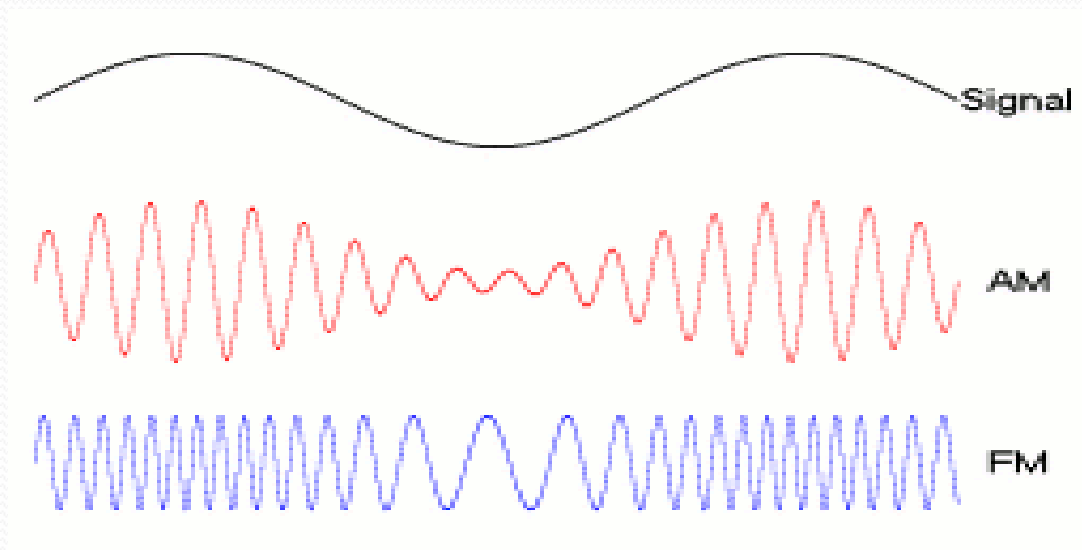


Qualidade de Recepção

- Atenuação proveniente da distância percorrida, o que faz a informação inaudível;
- A interferência de outros sinais que deformam o sinal original tornando-o incompreensível;
- A grande vantagem de trabalhar com a informação na forma de sinal elétrico é poder dar o mesmo tratamento a voz, dados e imagens.

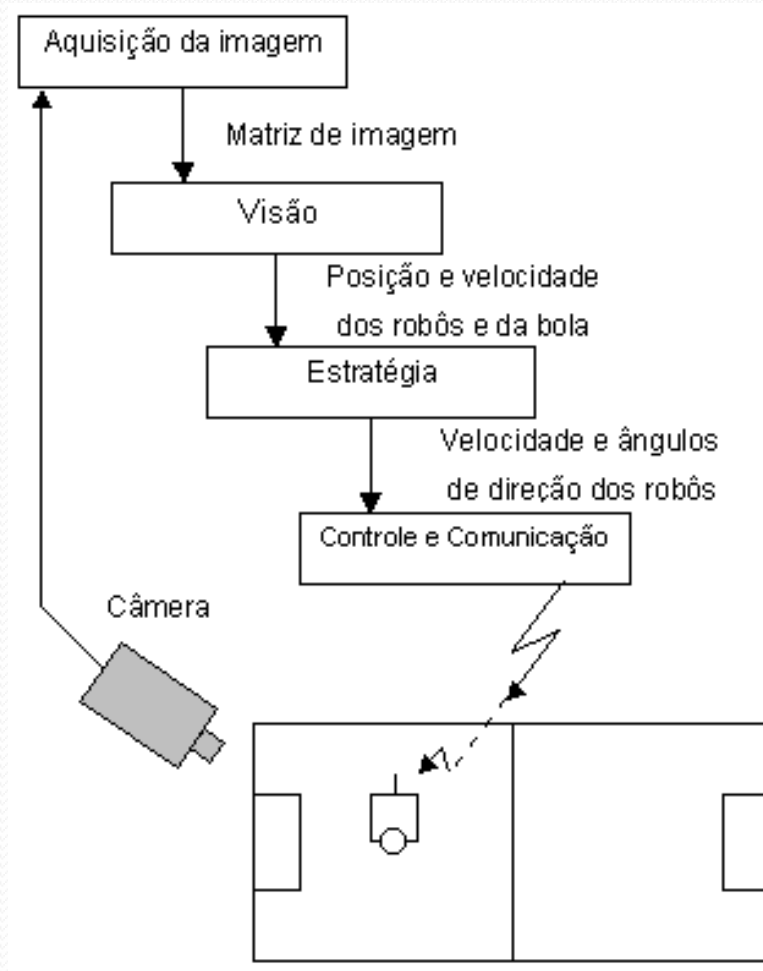
AM x FM

- AM (Modulação em Amplitude): o comprimento da onda é da ordem de metros. Portanto, ela usa muito bem as camadas atmosféricas, para se propagar, por reflexão.
- FM (Modulação em Frequência): o comprimento de onda é de alguns cm, é bem penetrante e não se reflete bem, nas camadas da atmosfera.



Importância no Futebol de Robôs

- Envia as ordens do “técnico” para os “jogadores”



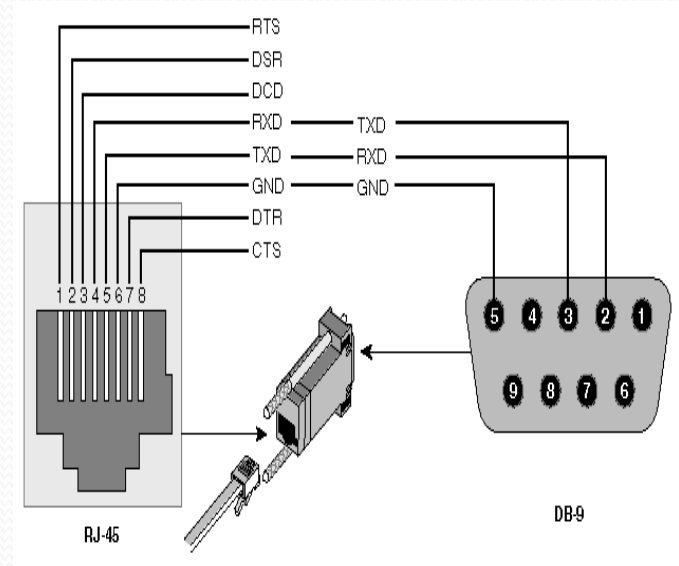
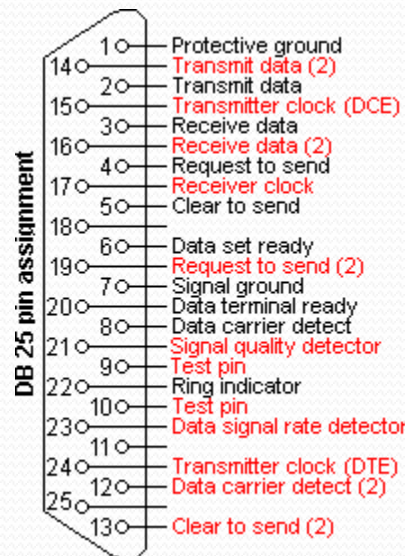
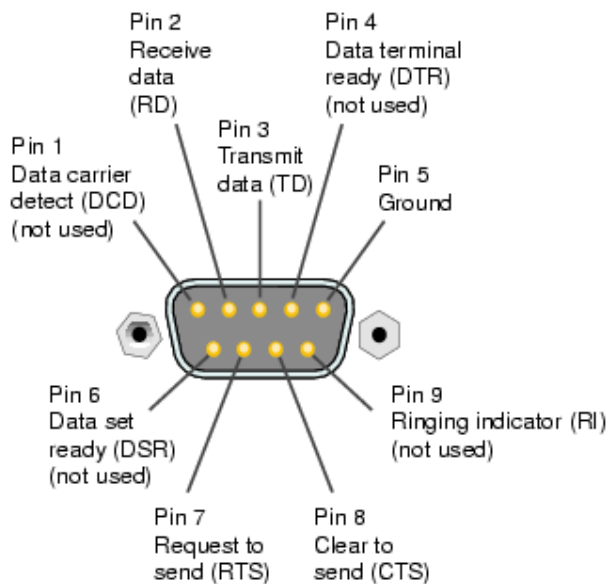
Comunicação Serial (RS232)

- É um padrão de comunicação serial criado pela EIA (Eletronic Industries Association) para a comunicação entre um DTE (Terminal de Dados) e um DCE (um comunicador de dados).



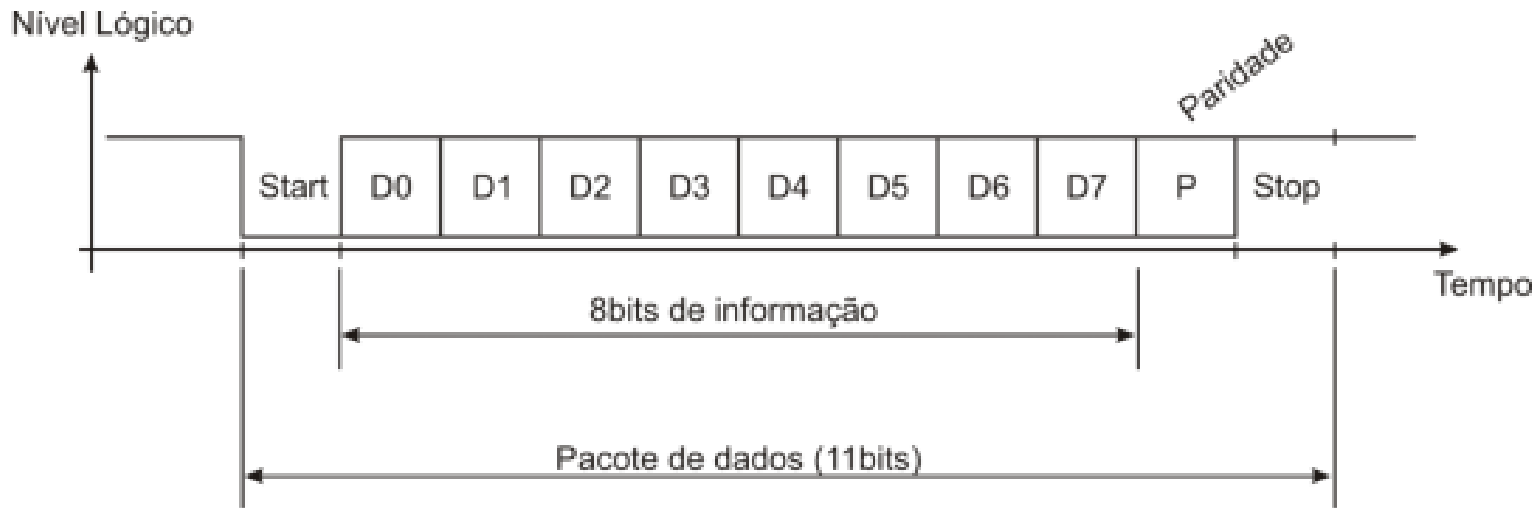
Comunicação Serial (RS232)

- As portas seriais utilizam conectores do tipo DB9, DB25 e RJ45.



Comunicação Serial (RS232)

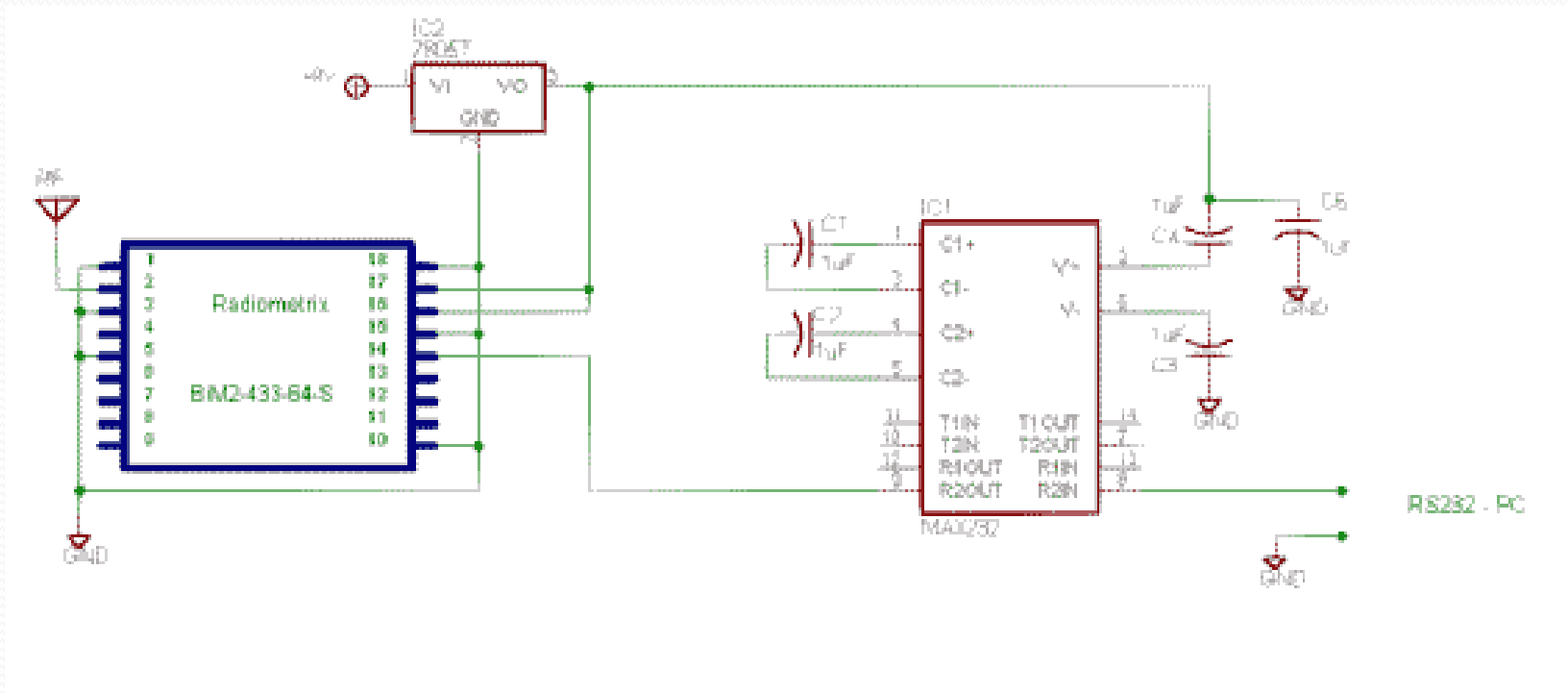
- Funcionamento do Protocolo RS232.



- A comunicação RS232 é assíncrona. O transmissor e o receptor devem estar configurados a uma mesma taxa de transmissão/recepção, caso contrário os dados não serão interpretados corretamente.

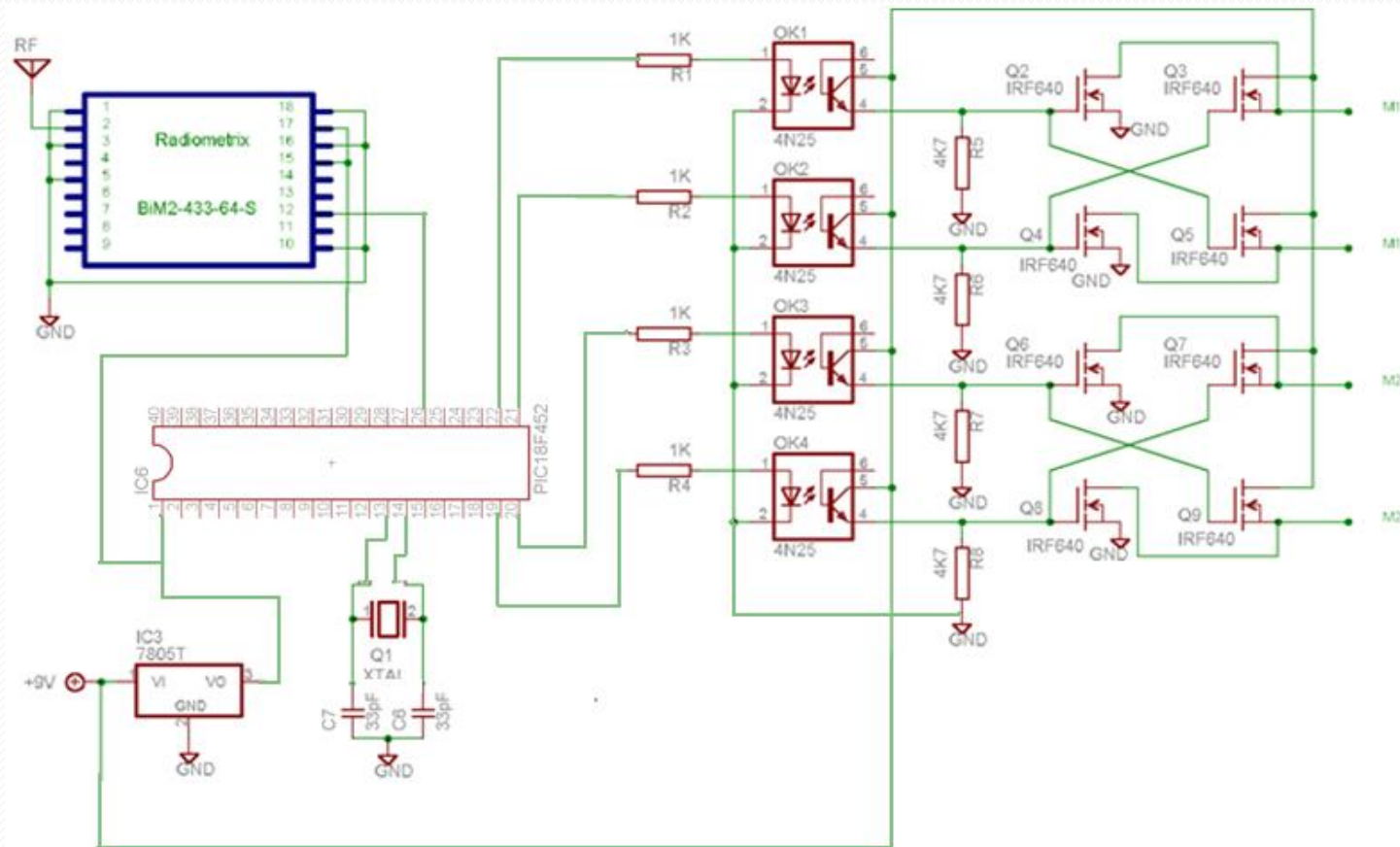
Circuitos Utilizados

➤ Circuito de Transmissão:



Circuitos Utilizados

➤ Circuito de Recepção:



Como Enviar Dados?

- Hyper terminal: é o aplicativo de comunicação mais antigo do windows.

- nosso terminal (que ainda não tem nome!): é um software desenvolvido por Manoel Alexandre para o envio de dados. Seu funcionamento é baseado no hyper terminal, que utiliza as APIs do Windows.

Estrutura DCB (API Windows)

➤ Criar porta

```
HANDLE hCom; // Handle para a Porta Serial (identificador).
char *NomePorta = "COM1"; //COM1, COM2..COM9 ou portas virtuais "\.\\COMx".

hCom = CreateFile(
    NomePorta, //Nome da porta.
    GENERIC_READ | GENERIC_WRITE, //Para leitura e escrita.
    0, //Zero) Nenhuma outra abertura será permitida.
    NULL, //Atributos de segurança. (NULL) padrão.
    OPEN_EXISTING, //Criação ou abertura.
    0, //Entrada e saída sem overlapped.
    NULL //Atributos e Flags. Deve ser NULL para COM.
);

if(hCom == INVALID_HANDLE_VALUE)
    return false; //Erro ao tentar abrir a porta especificada.
```


Estrutura DCB (API Windows)

- Configurar porta

```
DCB dcb; //Estrutura DCB é utilizada para definir todos os parâmetros da
comunicação.

if( !GetCommState(hCom, &dcb))
    return false; //// Erro na leitura de DCB.

dcb.BaudRate = CBR_19200;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;

//Define novo estado.
if( SetCommState(hCom, &dcb) == 0 )
    return false; //Erro.
```

Estrutura DCB (API Windows)

➤ Ler e Escrever a porta

```
#define LEN_BUFFER 100; //Define o tamanho do buffer.  
  
DWORD BytesLidos = 0;  
char BufferRecebe[LEN_BUFFER]; //Para armazenar a string a ser lida.  
ReadFile( hCom, BufferRecebe, LEN_BUFFER, &BytesLidos, NULL );
```

```
#define LEN_BUFFER 100; //Define o tamanho do buffer.  
DWORD BytesEscritos = 0;  
int TamaString; //Para armazenar o tamanho da String.  
char BufferEnvia[LEN_BUFFER]; //Para armazenar a string a ser  
enviada.  
strcpy(BufferEnvia, "CMD05#1#025\r\n"); //Prepara a string a ser  
enviada.  
TamaString = strlen(BufferEnvia); //Calcula o tamanho da string a ser  
enviada.  
  
WriteFile( hCom, BufferEnvia, TamaString, &BytesEscritos, NULL );
```

Estrutura DCB (API Windows)

- Fechar a porta

```
CloseHandle( hCom ); //Fecha a porta aberta anteriormente por  
CreateFile().
```

Software de Controle

➤ Implementado em C++ na ferramenta C++ Builder

Direção

Farol

Buzina

Velocidade

Nível de Bateria:

Indicador de Nível de bateria

ON/OFF

Acionar controle via teclado

Como Receber Dados

- Os dados após serem enviados pelo circuito transmissor ele deverá ser interpretado pelo PIC.
- O microcontrolador deverá esperar todos os dados serem recebidos e assim juntar a mensagem, e por fim decodificá-la e executar a ação correspondente.
- Registrador uart;

```
.  
#locate rxuart = 0x1A  
  
#INT_RDA  
void isrSerial(void)  
{  
    switch(rxuart)  
    {  
        case 'a':  
        case 'A':  
            buffer_serial[0]=rxuart;  
            cont++;  
            break;  
        case 'k':  
        case 'K':  
            buffer_serial[4]=rxuart;  
            buffer_serial[5]='\0';  
            validar_protocolo(buffer_serial);  
            cont=0;  
            break;  
        default:  
            buffer_serial[cont]=rxuart;  
            cont++;  
            break;  
    }  
}
```

Como Receber Dados

➤ Análise do protocolo:

```
void validar_protocolo(char *string)
{
    tam= strlen(buffer_serial);

    if((tam==5) && (buffer_serial[0]=='a' || buffer_serial[0]=='A') && (buffer_serial[4]=='k' || buffer_serial[4]=='K'))
    {
        direcao = buffer_serial[1];
        speed= (int)buffer_serial[3]-'0';
    }
    else
        printf(" Não faça nada ...!! ");
}
```

Como Receber Dados

➤ Executar ação:

```
switch(direcao)
{

    case 'f':

        set_pwm1_duty(speed*341);
        output_high(PIN_D2); //faz o carro andar pra frente
        output_low(PIN_D3);
        delay_ms(500);
        direcao='0';
        break;

    case 't':

        set_pwm1_duty(speed*341); //faz o carro andar pra tras
        output_high(PIN_D3);
        output_low(PIN_D2);
        delay_ms(500);
        direcao='0';
        break;
```

Demonstração

Ao Vivo!!

Referências

MIAYADAIRA, A. N. Microcontroladores PIC18 – Aprenda e Programe em Linguagem C. São Paula: Érica, 2009.

ABDALLA, H. Introdução aos sistemas de Telecomunicação Módulo V.
UNE - UNB

Onda de Rádio, Disponível em
<http://pt.wikipedia.org/wiki/Ondas_de_radio> Acesso em 07 de abril 2010.