



UNIVERSIDADE TECNOLÓGICA FEDERAL DO
PARANÁ

DEPARTAMENTO DE ELETRÔNICA

CURSO DE ENGENHARIA DE COMPUTAÇÃO

Guilherme Gomes Barboza
Pedro Michel Nikolaides

Veículo controlado remotamente através de sensor de movimento

**Curitiba
2018**

Guilherme Gomes Barboza
Pedro Michel Nikolaides

Veículo controlado remotamente através de sensor de movimento

Trabalho de Conclusão de matéria apresentado à disciplina Oficina de Integração 1 do Curso de Graduação em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientadores: Gustavo Benvenutti Borba
Ronnier Frates Rohrich

Curitiba
2018

Resumo

Este documento apresenta o processo de construção de um veículo controlado remotamente através do sensor de movimento contido em um controle do modelo WiiMote. Na primeira parte do mesmo serão explicadas as peças utilizadas assim como o processo de montagem do veículo através do uso de diagramas. Na segunda porção será disponibilizado e explicado o código utilizado no projeto e por fim serão analisados os resultados obtidos na realização do projeto.

Palavras-chave: Controle Remoto, Microcontrolador, Bluetooth.

Sumário

1	INTRODUÇÃO	1
2	HARDWARE	3
2.1	Controlador	3
2.2	Receptor	4
3	SOFTWARE	5
3.1	Código da antena	5
3.2	Código do veículo	5
4	RESULTADOS E DISCUSSÃO	6
5	CONCLUSÃO	7
	REFERÊNCIAS	8
	ANEXOS	9

1 Introdução

Este projeto tem como propósito a introdução dos casos de aplicação de microcontroladores e o uso dos mesmos em diversas situações. Tal estudo se faz necessário uma vez que os microcontroladores fazem parte do cotidiano de diversos profissionais da área de computação, já que possuem diversas finalidades, além de serem acessíveis economicamente e relativamente fáceis de programar e manusear.

Para tal objetivo, foi-se desenvolvido um projeto de montagem de um veículo de controle remoto através de comandos de movimento, o qual é constituído de duas partes principais, sendo estas um microcontrolador do modelo arduino UNO, responsável por receber os comandos do WiiMote e transmiti-los através do uso de Bluetooth para a segunda parte do projeto, a qual consiste de um microcontrolador TIVA-C, o qual recebe os comandos e de acordo com eles move o veículo, conforme demonstrado no esquemático abaixo.

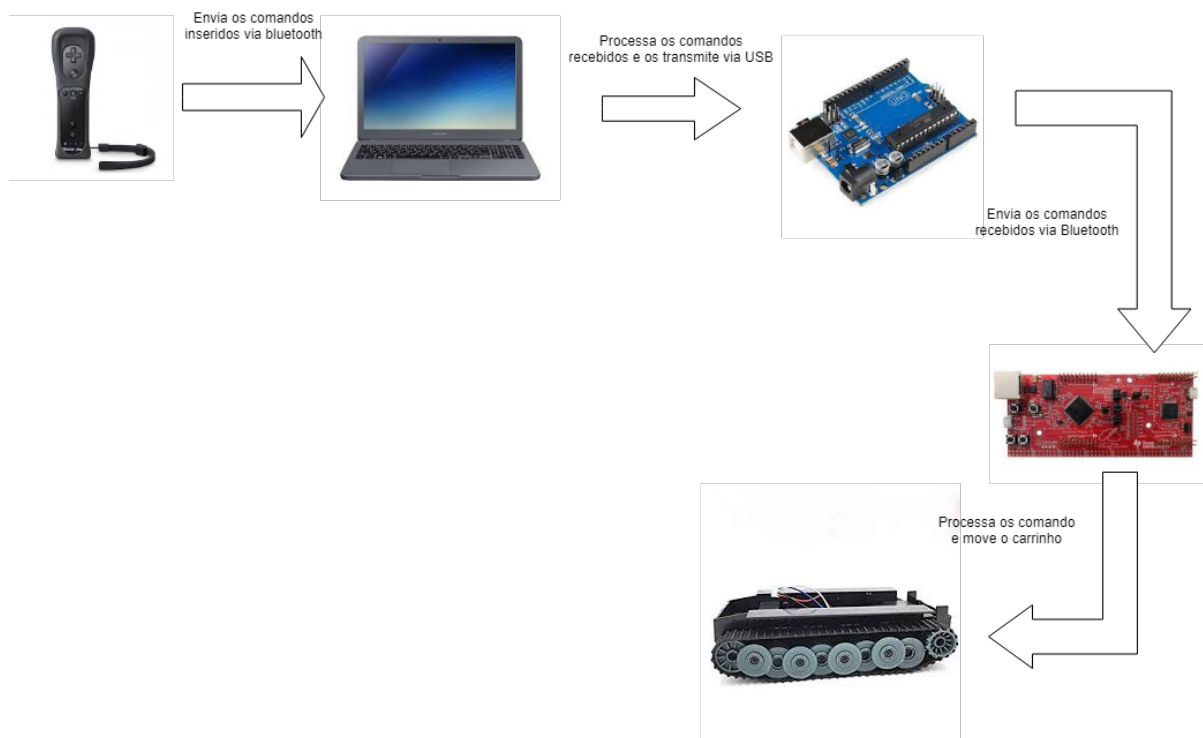


Figura 1 – Diagrama do projeto

Primeiramente o usuário movimenta o controle WiiMote, inclinando-o para frente ou para trás, para a esquerda ou para direita a partir de sua posição de repouso na horizontal, o controle então envia os dados captados para o computador, o qual através do programa `Wii_rc_car` transmite os dados recebidos ao arduino através do Serial USB.

O arduino ao receber os dados da inclinação do controle determina o comando a ser enviado para o carrinho e na sequência transmite o mesmo por comunicação Bluetooth

através de um HC-05 no modo mestre.

O comando é então recebido através de um HC-05 pelo microcontrolador TIVA-C que usa os dados para comandar o veículo através de uma ponte H l298N e então o veículo se move.

2 Hardware

O hardware do Projeto pode ser dividido em duas partes principais

2.1 Controlador

O controlador é a primeira parte do projeto, responsável por captar o sinal do WiiMote e transmiti-lo até o carrinho, sendo este composto de um controle WiiMote, um computador com conexão Bluetooth com o WiiMote, um arduino UNO, ligado ao computador através de um USB tipo B(figura 2) e um módulo bluetooth HC-05(figura 3).

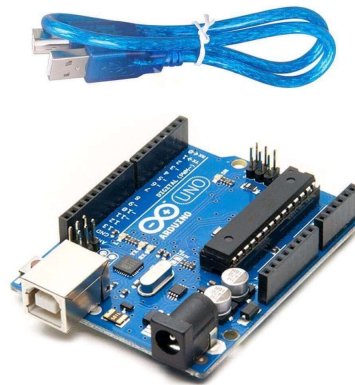


Figura 2 – arduino UNO com o cabo USB



Figura 3 – Módulo Bluetooth HC-05

2.2 Receptor

O receptor é a segunda parte do projeto e constitui-se do carrinho em si, possuindo um microcontrolador tiva c series tm4c1294(figura 4) conectado a um módulo bluetooth HC-05, e um driver de motor ponte H L298N(figura 5) que controla os motores do chassi Economy 2WD do tanque tiger 2 alemão. Esta seção do projeto é alimentada por uma bateria de polímero de lítio de 7V com capacidade de 1000mAH.

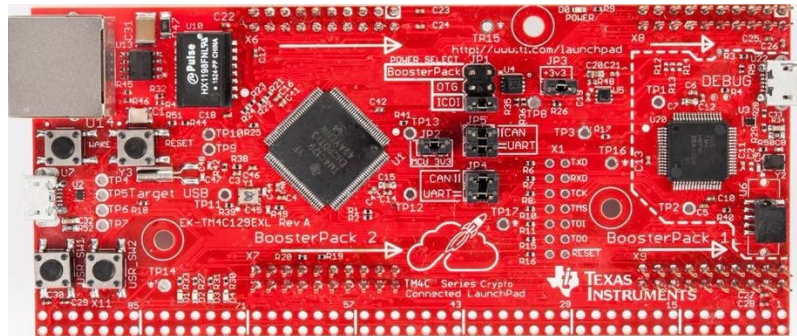


Figura 4 – Microcontrolador TIVA-C



Figura 5 – Driver de motor L298N

3 Software

3.1 Código da antena

O código da antena(Apêndice A) habilita o Arduíno receber os dados via serial do computador pelo Serial read, e se utiliza da biblioteca SoftwareSerial.h para se comunicar com o modulo bluetooth e enviar os dados que serão enviados para o carro.Porem, antes de ser enviado, o código faz alguns testes para separar dados específicos para serem enviados, como por exemplo, o dado 'F' significa ir pra frente.

3.2 Código do veiculo

O código do veiculo(Apêndice B) se baseia em receber os dados via bluetooth, na função Serial7.read(), o motivo de ser o serial 7 é porque a tiva-C possui 7 seriais e nesse caso foi conectado no rx7 e tx7.Logo apos a recepção de dados o programa faz um teste para saber qual direção o carro deve ir de acordo com o caractere recebido por bluetooth.

4 Resultados e Discussão

Com o desenvolvimento do projeto foram obtidos maiores conhecimentos em relação ao funcionamento do controle WiiMote e sua comunicação com outros dispositivos.

Ao ser realizado o primeiro teste de conexão entre o WiiMote e o microcontrolador HC-05, foi percebido uma incompatibilidade entre os protocolos de comunicação de ambos, logo foram feitos novos testes e verificou-se que este é um problema recorrente e que para a comunicação correta se utiliza um USB shield para o microcontrolador a fim de estabelecer a conexão através de um adaptador Bluetooth. Devido ao alto custo do mesmo, optou-se pela utilização de um software em um notebook que capta os dados e os transmite via USB Serial para o microcontrolador.

Devido a conexão entre o computador e o microcontrolador ser via cabo USB, houve a necessidade de ao se estabelecer a conexão sem fio ao veículo, utilizar mais um microcontrolador, de modo a se realizar a conexão com dois HC-05 no total.



Figura 6 – Conexão direta wiimote arduino

5 Conclusão

O trabalho foi realizado por completo, embora existam algumas adaptações para o melhor funcionamento do mesmo, devido a problemas não considerados na elaboração do projeto. Com este os alunos puderam aprender a utilizar microcontroladores e adaptá-los as necessidades do projeto como um todo, além de aprimorar as habilidades de trabalho em equipe e organização dos mesmos.

Referências

Anexos

ANEXO A - Código controlador

```
//Programa : Modulo Arduino Bluetooth HC-05 - Envio
//Autor : FILIPEFLOP

//Carrega a biblioteca SoftwareSerial
#include <SoftwareSerial.h>

int Low_Limit = 88;
int High_Limit = 168;

//Define os pinos para a serial
SoftwareSerial mySerial(10, 11); // RX, TX

void setup()

//Inicializa a serial nas portas 10 e 11
Serial.begin(38400);
pinMode(13,OUTPUT);
mySerial.begin(38400);

void loop()

int x,y;
char buf,buf2;
while (!Serial.available());
if (Serial.available() >0 )
x = Serial.read();
Serial.print(x);

while (!Serial.available());
if (Serial.available() >0 )
y = Serial.read();
Serial.print(y);

if ( y > High_Limit)

buf = 'F';
```

```
else if(y < Low_Limit)
```

```
buf = 'T' ;
```

```
else
```

```
buf = 'P';
```

```
//Left and Right
```

```
if ( x > High_Limit)
```

```
buf2 = 'D';
```

```
else if( x < Low_Limit)
```

```
buf2 = 'E';
```

```
else
```

```
buf2 = 'R';
```

```
mySerial.print(buf);
```

```
mySerial.print(buf2);
```

ANEXO B - Código veículo

```
//Programa : Modulo Arduino Bluetooth HC-05 - Recepcao

char buf,buf2;
int IN1 = PQ_0;
int IN2 = PP_4;
int IN3 = PN_5;
int IN4 = PN_4;
void setup()

pinMode(PN_1,OUTPUT);
Serial7.begin(9600);
Serial.begin(9600);
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);

void loop()

while(Serial7.available() > 0)

buf = Serial7.read();
buf2 = Serial7.read();
Serial.print("b1: ");
Serial.println(buf);
Serial.print("b2: ");
Serial.println(buf2);
if(buf == 'F')

digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);

else if(buf == 'P')
```

```
digitalWrite(IN1, LOW);  
digitalWrite(IN2, LOW);  
digitalWrite(IN3, LOW);  
digitalWrite(IN4, LOW);
```

```
else if(buf == 'T')
```

```
digitalWrite(IN1, HIGH);  
digitalWrite(IN2, LOW);  
digitalWrite(IN3, LOW);  
digitalWrite(IN4, HIGH);
```

```
    if(buf == 'E')
```

```
digitalWrite(IN1, HIGH);  
digitalWrite(IN2, LOW);  
digitalWrite(IN3, HIGH);  
digitalWrite(IN4, LOW);
```

```
else if(buf == 'D')
```

```
digitalWrite(IN1, LOW);  
digitalWrite(IN2, HIGH);  
digitalWrite(IN3, LOW);  
digitalWrite(IN4, HIGH);
```