

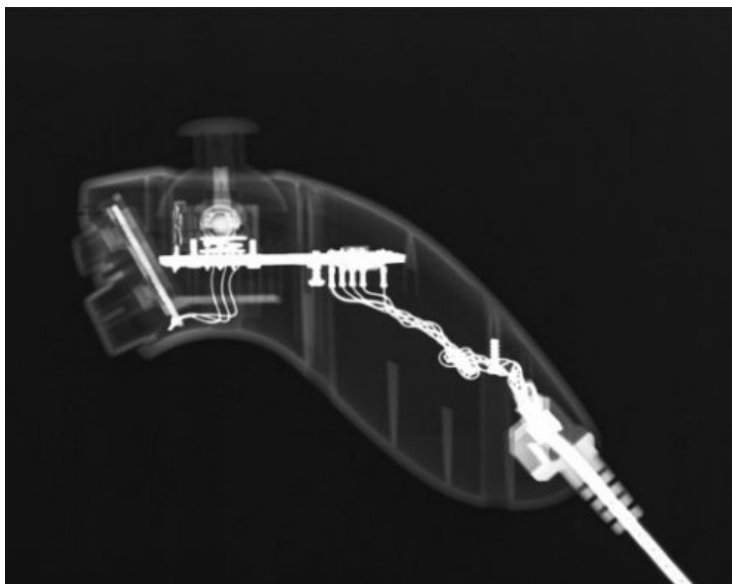
Interface de um PIC24F com o Nunchuck



O "Nunchuck" é um acessório da mais que conhecida consola de jogos "Wii". Existem vários factores que tomam este comando muito interessante para projectos de robótica/automação dos quais passo a nomear alguns:

- Contém um Joystick que fornece valores entre 0 e 255 tanto para o eixo X como para o eixo Y.
- Contém dois botões de uso genérico
- Contém um acelerómetro de 3 eixos (ADXL330 - Fabricado pela STMicroelectronics)
- Interface através de um protocolo conhecido, "I2C"
- Preço acessível (+/- 20€ novo. podem conseguir usados mais baratos)
- Tudo encapsulado numa caixa ergonómica para a mão.

Interior do Nunchuck:



Conteúdos

[picWEB](#)

[LedDisplay](#)

[osTWDR](#)

[Nunchuck](#)

[GARRA](#)

[P.C.M.DC](#)

[uP8051](#)

[PIC-PROG](#)

[EAGLE 3D](#)

[SERVO
CONTROLLER](#)

[Rarm](#)

[Damas](#)

[TWDR](#)

[CNC](#)

[MGSM](#)

[Stepper
Motors](#)

[Material](#)

[Timer](#)

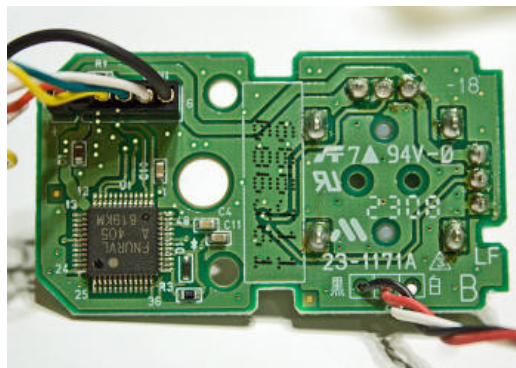
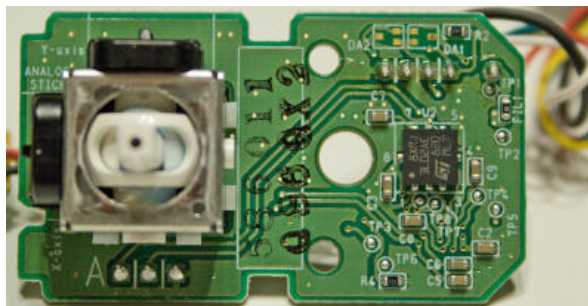
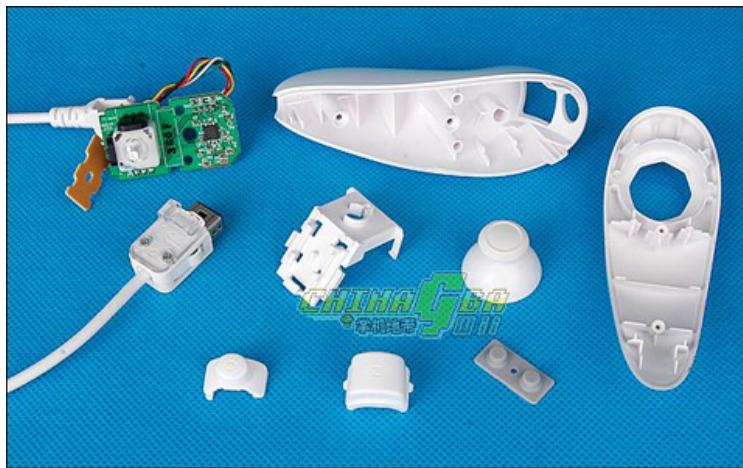
[PIC
BootLoaders](#)

[Tabelas
úteis](#)

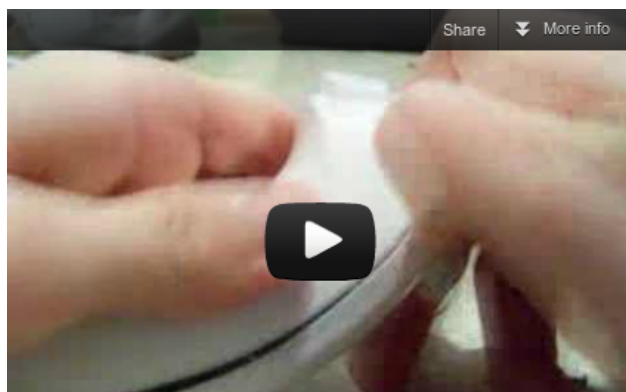
[Links](#)

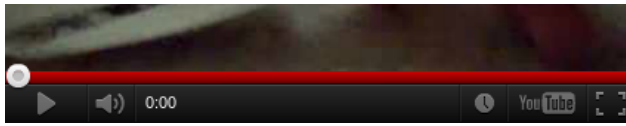
[Páginas de
Colegas](#)

[Fórum
Lusorobótica
\(maurosmarti](#)



Desmontar o Nunchuck:

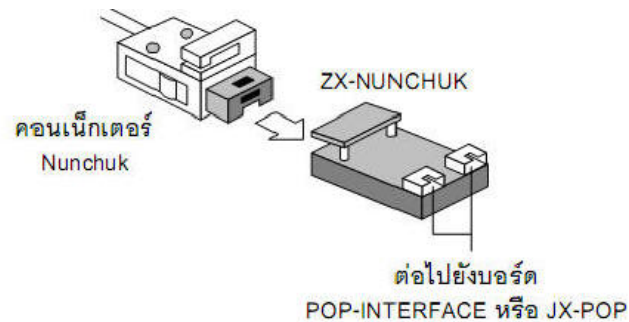
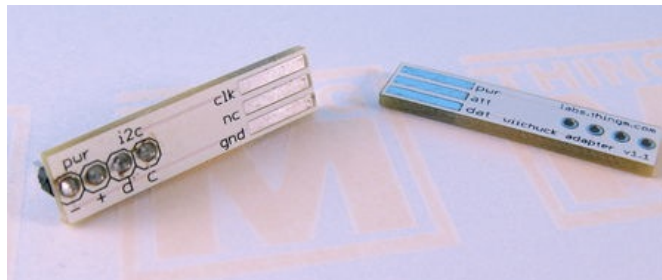
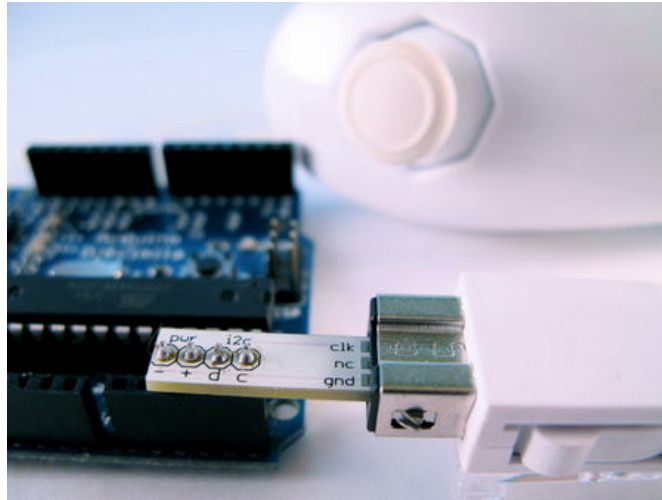




Hardware:

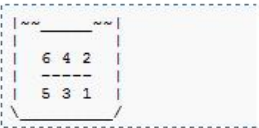
Nalguns sites irão encontrar informações indicando que ligaram o Nunchuck directamente a 5v ou colocando um ou dois díodos em série com a alimentação. Apesar de isto poder eventualmente funcionar, o Nunchuck deve ser alimentado a **3,3V** sobe pena de reduzir significativamente o tempo de vida útil do mesmo.

Para se poderem fazer as ligações com o microcontrolador pode-se proceder de duas formas: ou cortando o conector proprietário, ou usando um adaptador destes géneros.



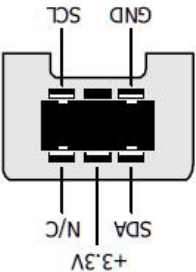


Pinout

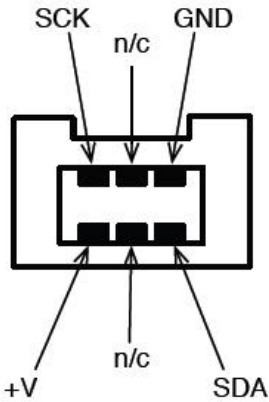


From the Wiimote side

Pin	Cable color	
1	Red	3.3V
2	Yellow	SCL, I ² C
3	Red	Connecter
4	-	Not connec
5	Green	SDA, I ² C
6	White	Ground



Nunchuck Pinout



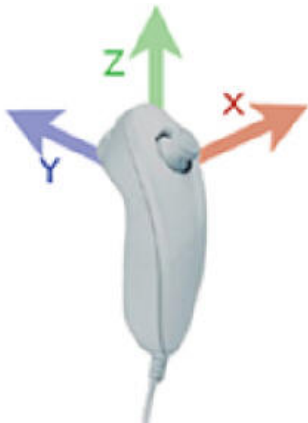
Se optarem por cortar o conector (como eu :p), irão encontrar uma malha de protecção contra interferências electromagnéticas, e mais quatro fios com as seguintes cores e funções:

Alimentação	
Branco	GND (0v)
Vermelho	+3.3V
Comunicação (I2C)	
Amarelo	SCK (Serial Clock)
Verde	SDA (Serial Data)

Nota: O Nunchuck usa o protocolo de comunicações I2C que necessita de resistências de "PULL UP" nas linhas

"SDA" e "SCK", no entanto o Nunchuck já possui estas resistências internamente!!! Assim sendo salvo raras excepções (por exemplo os controladores "NXT" da LEGO) não será necessário colocá-las externamente.

Orientação dos eixos do acelerómetro:



Leitura dos dados:

O Nunchuck tem o endereço "0x52" (7 bits) por isso usam-se os endereços "0xA4" (8 bits) para escrever e "0xA5" (8 bits) para ler. Os dados são fornecidos em conjuntos de 6 bytes como indicado abaixo:

Byte nº	Função								Descrição
1	Joystick X								--
2	Joystick Y								--
3	Acelerómetro X Bits 9 a 2 (10 bits)								--
4	Acelerómetro Y Bits 9 a 2 (10 bits)								--
5	Acelerómetro Z Bits 9 a 2 (10 bits)								--
6	Acel Z bit 1	Acel Z bit 0	Acel Y bit 1	Acel Y bit 0	Acel X bit 1	Acel X bit 0	Botão "C"	Botão "Z"	Os botões C e Z são activos a "0"

Inicialização:

Para que seja possível ler os dados do Nunchuck é necessário inicializá-lo através do envio dos seguintes comandos pelo BUS I2C:

Start I2C	0xA4	Nunchuck ACK	0x40	Nunchuck ACK	0x00	Nunchuck ACK	Stop I2C
-----------	------	--------------	------	--------------	------	--------------	----------

Esta inicialização é apenas necessária uma vez por cada "Power ON/OFF".

Leitura:

Após termos inicializado o Nunchuck podemos agora pedir os dados sempre que quisermos, este processo divide-se em três partes:

Primeira - Indica ao Nunchuck que queremos ir buscar os dados

Start I2C	0xA4	Nunchuck ACK	0x00	Nunchuck ACK	Stop I2C
-----------	------	--------------	------	--------------	----------

Delay - Este delay é necessário!!! Embora o valor do delay não seja exactamente conhecido 1 mS parece funcionar bem.

Segunda - Vamos ler os bytes do Nunchuck

Start I2C	0xA5	Nunchuck ACK	Byte1	Master ACK	Byte2	Master ACK	Byte3	Master ACK	Byte4	Master ACK	Byte5	Master ACK	Byte6	Master NACK	Stop I2C
-----------	------	--------------	-------	------------	-------	------------	-------	------------	-------	------------	-------	------------	-------	-------------	----------

Terceira - Decodificação dos dados

Os dados provenientes do Nunchuck vêm de certa forma codificados, assim sendo é preciso decodificá-los de forma a obter valores correctos. Para isso é necessário fazer um "OU EXCLUSIVO" (XOR) aos dados com o valor "0x17" e somar "0x17" (valor em hexadecimal) ao resultado.

Em linguagem "C" seria algo do género:

descodificado = (codificado^0x17)+0x17;

Legenda:

	Start/Stop I2C
	Escrever Byte para Nunchuck
	Acknowledge do Nunchuck
	Ler Byte do Nunchuck
	Acknowledge do Master
	Not Acknowledge do Master

PDF de onde foram extraídas a grande parte das informações mostradas até aqui:

Ligação ao PIC24:

O "PIC" utilizado foi o "PIC24FJ128GA010" que é fornecido com a placa de desenvolvimento "Explorer 16". Visto que este microcontrolador também funciona a 3.3V não existem problemas eléctricos. Se usarem um microcontrolador que esteja a ser alimentado a 5V verifiquem as características eléctricas dos pinos "SDA" e "SCK" para ver se possibilitam a utilização com níveis de tensão diferentes. Se não permitirem terão de usar adaptadores de tensão para BUS "I2C": Por exemplo com o PCA9512.

Este microcontrolador tem dois módulos I2C. Para este tutorial foi usado o módulo 1 que tem como pinos o "RG3" para o "SDA" e o "RG2" para o "SCK".

Basta ligar o pino "RG3" ao fio VERDE do Nunchuck e o "RG2" ao fio "AMARELO"

Código de Exemplo:

O Código que aqui apresento foi escrito utilizando o "mikroC" para "dsPIC", usando um "PIC24FJ128GA010" com xtal de 8MHz e PLL de 4x (32MHz).

O código foi testado usando um Nunchuck e uma placa de desenvolvimento Explorer16. Ao premir os botões "Z" ou "C" irá acende o nibble superior ou inferior dos LEDs da Explorer16. Será também enviado para a porta sério os restantes valores usando uma ligação de 9600 bauds 8 bits de dados sem paridade e 1 stop bit.

```
#define ack 0
#define nack 1

char buffer[7];
unsigned int i;
unsigned int a=0,b=0;
unsigned int temp;

void initNUNCHUCK(void){
    I2C1_Start();
    I2C1_Write(0xA4);
    I2C1_Write(0x40);
    I2C1_Write(0x00);
    I2C1_Stop();
}
```

```

}
void readNUNCHUCK(void){
    LATA=a=b=0;

    I2C1_Start();
    I2C1_Write(0xA4);
    I2C1_Write(0x00);
    I2C1_Stop();

    delay_ms(10);

    I2C1_Start();
    I2C1_Write(0xA5);

    buffer[0]=I2C1_Read(ack);
    buffer[1]=I2C1_Read(ack);
    buffer[2]=I2C1_Read(ack);
    buffer[3]=I2C1_Read(ack);
    buffer[4]=I2C1_Read(ack);
    buffer[5]=I2C1_Read(nack);

    I2C1_Stop();

    for(i=0;i<6;i++){

        temp=((buffer[i]^0x17)+0x17);

        Uart2_Write_Char(temp);

        if(i==5){
            if(temp.f0==0)
                a=0b000001111;

            if(temp.f1==0)
                b=0b11110000;

            LATA=(a|b);
        }
    }
}

void main(void){
    I2C1_Init(100000);
    Uart2_Init(9600);

    TRISA=0;
    LATA=0;

    initNUNCHUCK();

    delay_ms(500);
    while(1){
        readNUNCHUCK();

        delay_ms(200);
    }
}

```

[Se preferirem o HEX do código já compilado podem fazer download aqui.](#)

Links de interesse:

<http://www.windmeadow.com/node/42>

<http://nunchuckmouse.blogspot.com/> - Tuga :p

http://wiibrew.org/wiki/Wiimote/Extension_Controllers

http://www.wiili.com/index.php/Wiimote/Extension_Controllers/Nunchuk

<http://www.best-microcontroller-projects.com/i2c-tutorial.html> - Tutorial de I2C

http://www.robot-electronics.co.uk/htm/using_the_i2c_bus.htm - Mais um tutorial de I2C

Última actualização: 22-09-2009

| Página optimizada para uma resolução de 1024x768. | Internet Explorer 5.0 ou superior. | Webmaster maurosmartins@gmail.com | © Copyright 2008. |