



SEMINÁRIO CROSS-REALITY: INTERLIGANDO O MUNDO REAL E O MUNDO VIRTUAL

16/07/2010

Rui Silva Moreira @ ISUS.FCT.UFP

2

ISUS Group @ FCT.UFP

Intelligent Sensing and Ubiquitous Systems

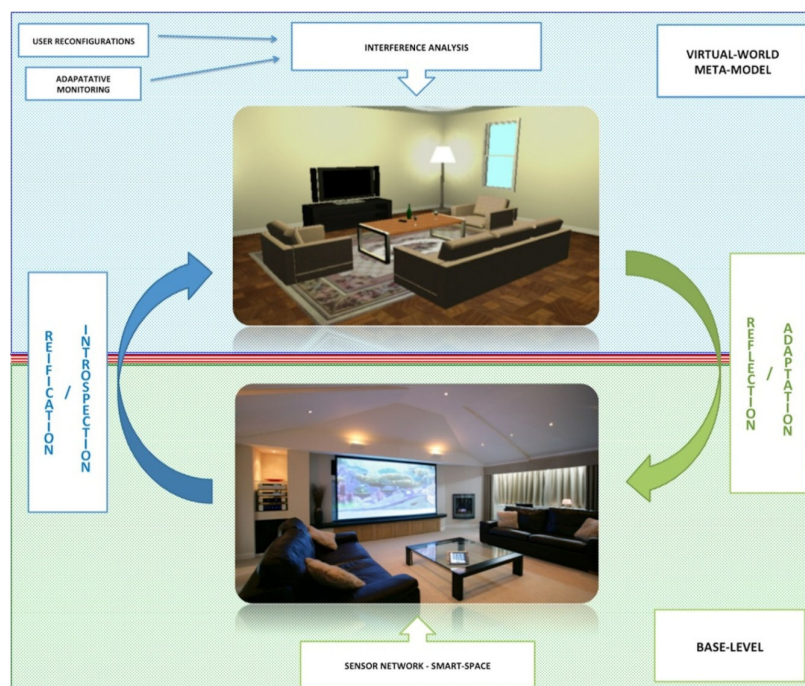
<http://isus.ufp.pt/>

- Safe Home Health Care
 - Interference-free Home Health-Care Smart Spaces
 - Search Algorithms and Meta-Reality Reflection
- Remote patient monitoring in home environments
 - homecare framework – monitoring activity & environment
 - low cost
 - low user usage/interaction
- LoCoBoard: A Low-Cost interactive whiteBoard
 - open-source Interactive Whiteboard (IW)
 - use computer vision algorithms
 - low-cost hardware requirements (cf. webcam+PC+video projector+IR pointing device)

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Safe Home Health Care

<http://isus.ufp.pt/projects/fct-project-safehomehealthcare/>

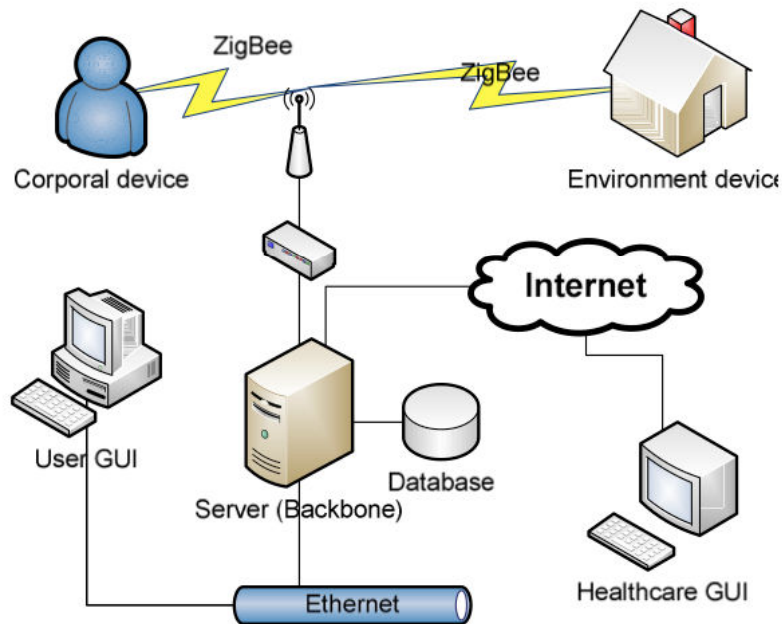


Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Remote Patient Monitoring in Home Environments

<http://isus.ufp.pt/projects/project-remote-patient-monitoring-in-home-environments/>

5

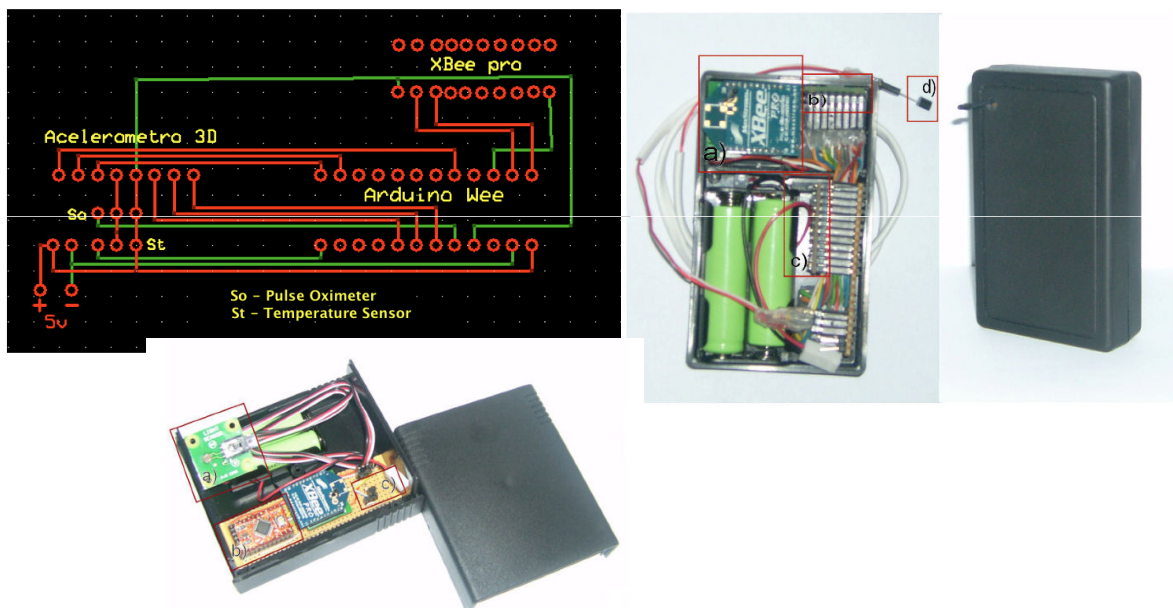


Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Remote Patient Monitoring in Home Environments

<http://isus.ufp.pt/projects/project-remote-patient-monitoring-in-home-environments/>

6



Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

LoCoBoard

<http://isus.ufp.pt/projects/locoboard-a-low-cost-interactive-whiteboard-using-computer-vision-algorithms/>

7

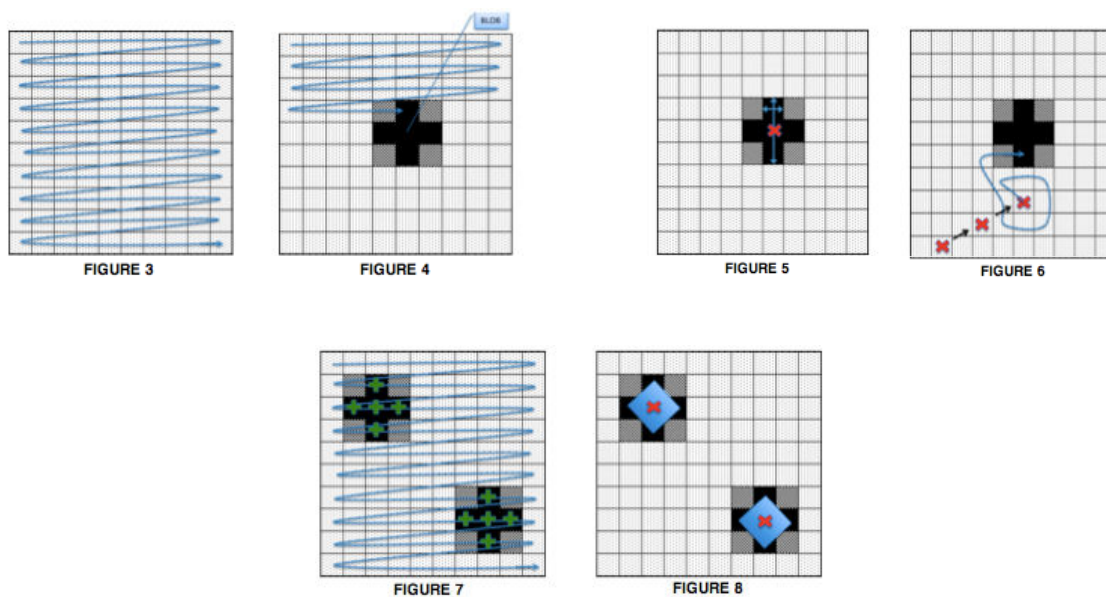


Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

LoCoBoard

<http://isus.ufp.pt/projects/locoboard-a-low-cost-interactive-whiteboard-using-computer-vision-algorithms/>

8



Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

1. **Cross-Reality:** ligar construções 3D, com localizações específicas, a espaços inteligentes /sensoriais dentro de edifícios – aumentar a realidade dos espaços físicos;
2. **Construção de um espaço virtual (3D):** espelhando um espaço/sala real com os seus **Farruscos** (robots);
3. **Programação dos Farruscos:** cujas capacidades sensoriais e de comunicação lhes permitirão **receber/reflectir** comandos/acções remotas do mundo virtual;
4. **Programação das primitivas (prims):** réplicas virtuais dos Farruscos, cujos movimentos serão **espelhados/reificados** a partir dos Farruscos reais.

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Cross-Reality

- Ligação entre mundos 3D (*hyper reality*) e mundos/espacos reais, e.g.,
 - IBM Data Center: visualização da operação
 - <http://www.virtualworldsnews.com/2008/02/ibm-launches-3.html>
 - VRContext: visualização e gestão processos produtivos com base em réplicas 3D de plantas de fábricas
 - <http://www.vrcontext.com/>
 - Outros projectos @ MIT:
 - <http://www.media.mit.edu/resenv/projects.html>
- Mais info:
 - IEEE Pervasive Computing, July-September 2009

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Recursos - Tecnologias

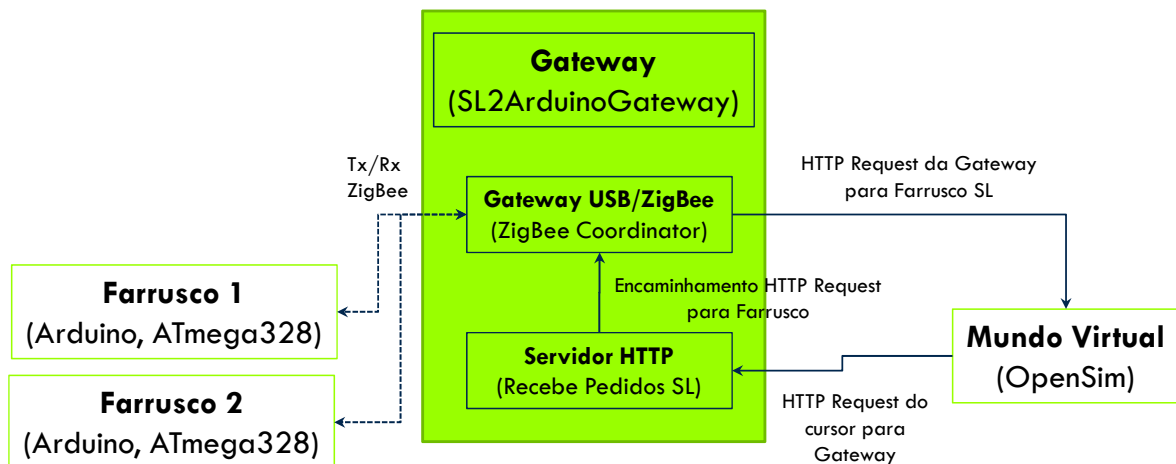
11

- **Espaços/mundos virtuais 3D** (cf. Second Life):
 - Servidor: **OpenSim** (open source em C# - executado em **Linux** sobre **mono**)
 - Clientes: **Hippo**, SecondLife
- **Micro-controladores e Sensores/Actuadores:**
 - Arduino (Duemilenove) & ATmel (168, 328)
 - LDR, Bump, Motores, Servos, etc.
- **Protocolos/comunicação:**
 - HTTP (cliente/servidor) - Gateway Java (cf. SL2ArduinoGateway);
 - Comunicação sem fios (ZigBee) - ponte para - Rx/Tx porta série
- **Arquitectura de programação/reflexão:**
 - Programação Farruscos - sketches (Processing Development Environment - PDE)
 - Programação prims SL - Linden Script Language (LSL)

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Arquitectura do Sistema

12



Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

13

Rede WiFi seminário

SSID: ISUS

Chave: abcdef1234

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

14

OpenSim (Mundo Virtual 3D)

Lindan Lab desenvolveu Second Life

Plataforma de realidade virtual (3D)

Cerca 15 milhões de subscritores

OpenSim é plataforma/clone Open Source

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Preparar ambiente - download

15

1. **OpenSim & Unzip versão binária (release 0.6.9):**
 - <http://opensimulator.org/wiki/Download>
 2. **Cliente Hippo Viewer (v0.6.3)**
 - <http://mjm-labs.com/viewer/download.php>
- OU
1. **Cliente SecondLife (v2.0.1.xxxxxx)**
 - <http://secondlife.com/support/downloads>

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Servidor OpenSim

16

- Instalar OpenSim:
 - e.g., unzip para C : \
- Executar OpenSim numa shell:
 - Windows
 - C:\opensim-0.6.9-binaries\bin\OpenSim.exe
 - Linux
 - cd opensim/opensim-0.6.9-source/bin
 - mono OpenSim.exe
- Cria uma **region**: e.g., “VilaMoleza”
- Cria **master** avatar (Primeiro e Último nome): e.g., “Master Sportacus”
- Escolher password: e.g., “sportacus”
- Verificar/Registrar **IP** do host... usado posteriormente no endereço http!!

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Comandos OpenSim

17

- **Region (VilaMoleza)#**
 - Help
 - Reset user password Master Sportacus
 - save oar c:\file.oar
 - load oar [--merge] c:\file.oar
 - ...

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Editar

18

- Criar objectos
- Carregar e mudar texturas
- Agrupar objectos (Link)
- Scripts LSL

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

```

// Comunicacao entre objectos SL – Envio comando movimentacao FORWARD (12)
string m = "12";
integer i = 0;
integer base_channel = 900;

default
{
    // Estado de entrada (avisar que esta' activo)
    state_entry()
    {
        llSay(0, "Prim: "+m+" running");
    }
    // Move forward - 12
    touch_start(integer number)
    {
        llSay(bae_channel+i, m);
    }
}

```

```

// Comunicacao entre objectos SL – Recepcao comandos movimentacao
integer i = 0;
integer base_channel = 900;
float X=0.0;
float Y=0.0;
float Z=0.0;
vector position;
float move=0.5;
integer rot = 0;
turn(integer angle)
{
    vector eul = <0,0, angle>; // Rotate around z-axis (Euler)
    eul *= DEG_TO_RAD; //convert to radians
    rotation quat = llEuler2Rot(eul); //convert to quaternion
    llSetRot(quat); //rotate the object
}
default
{
    state_entry()
    {
        llSay(0, "Farrusco "+i+" prim listen channel "+(base_channel+i));
        llListen(base_channel+i,"", NULL_KEY, "");
    }
}
// Escutar comandos emitidos
listen(integer channel, string name, key id, string message)
{
    vector position = llGetPos();
    X=position.x; Y=position.y; Z=position.z;
    llSay(0,"I heard "+ name + " say: "+ message);
    if (message=="12"){
        X=X+move;
    } else if (message=="6"){
        X=X-move;
    } else if (message=="3"){
        //Y=Y-move;
        rot+=45;
        turn(rot);
    } else if (message=="9"){
        //Y=Y+move;
        rot+=45;
        turn(rot);
    }
    llSetPos(<X,Y,Z>);
}

```

```

// Enviar pedidos HTTP – Send Forward (12)
string http_host = "192.168.1.65";
integer base_channel = 900, i = 2, t = 1000;
string m = "12";
default
{
    // EVENT: state_entry - triggered when prim touched
    touch(integer num_detected)
    {
        http_request_id =
            llHTTPRequest("http://" + http_host + ":8888/sendFarrusco.html?i="+i+"&m="+m+"&t="+t,
[HTTP_METHOD, "GET"], "");
    }
    // EVENT: http_response - triggered when script receives HTTP response to an HTTP request
    http_response(key http_request_id, integer status, list metadata, string body)
    {
        if (http_request_id == http_request_id)
        {
            llSetText("Forward", <0,0,1>, 1);
            llSay(0,"Http Response: "+body);
            // After sending Move to Farrusco, move SL Farrusco accordingly
            llSay(base_channel+i, m);
        }
    }
}
21      Cross-reality: interligando o mundo real e o mundo virtual @ IPG    16/07/2010
}

```

```

// Enviar pedidos HTTP para Registrar URL na Gateway
string http_host = "192.168.1.65", i = "1", m = "12", granted_url = "";
key url_request, http_request_id;
default {
    // EVENT: state_entry - triggered on any state transition and startup.
    state_entry() {
        // Each time prim is activated new URL is requested, but old one NOT RELEASED!
        // Hence, be careful and release it (before request a new one) since URLs are limited!
        url_request = llRequestURL();
    }
    // EVENT: http_request - triggered when the script receives an HTTP request.
    http_request(key id, string method, string body){
        if (url_request == id) {
            url_request = "";
            if (method == "URL_REQUEST_GRANTED") {
                llSay(0,"URL: " + body);
                if (granted_url != "") { llReleaseURL(granted_url); } // Release last URL
                granted_url=body;
                // Sends new prim internal URL to Gateway
                http_request_id =
                    llHTTPRequest("http://" + http_host + ":8888/changeSlurl.html?i="+i+"&slurl="+body,
[HTTP_METHOD, "GET"], "");
            }
        }
    }
}
22      Cross-reality: interligando o mundo real e o mundo virtual @ IPG    16/07/2010
}

```

```
// Enviar pedidos HTTP para Registrar URL na Gateway (continuacao)
```

```
default
```

```
{  
    // EVENT: http_request - triggered when the script receives an HTTP request.  
    http_request(key id, string method, string body)  
    {  
        if (url_request == id) {  
        } else {  
            llSay(0, "Farrusco "+i+" received a request method = " + method);  
            list headers = [ "x-script-url", "x-path-info", "x-query-string", "x-remote-ip",  
"user-agent" ];  
            integer pos = ~llGetListLength(headers);  
            while( ++pos )  
            {  
                string header = llList2String(headers, pos);  
                llOwnerSay(header + ": " + llGetHTTPHeader(id, header));  
            }  
            llOwnerSay("body: " + body);  
            //llHTTPResponse responds to an HTTP request made to the prim.  
            string params = llGetHTTPHeader(id, llList2String(headers, 2));  
            llHTTPResponse(id, 200, "Farrusco "+i+" received request parameters: " + params);  
            // Parse parameters and Move  
            //... See next slide  
        }  
    }  
}
```

23

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

```
// Codigo - parsing pedido HTTP e movimentacao
```

```
// PARSE parameters of HTTP request (e.g., m=12&t=888)  
list parameters_list = llParseString2List(params, ["&"], []);  
llSay(0, "<" + llDumpList2String(parameters_list, "><") + ">");  
string m_param = llList2String(parameters_list, 0);  
string t_param = llList2String(parameters_list, 1);  
string m_value = llList2String(llParseString2List(m_param, ["="], []), 0);  
string t_value = llList2String(llParseString2List(t_param, ["="], []), 0);  
llSay(0, "m_value="+m_value);  
llSay(0, "t_value="+t_value);  
// MOVE...  
vector pos = llGetPos();  
prim_x=pos.x; prim_y=pos.y; prim_z=pos.z;  
llSay(0, "Current pos: "+ (string)pos);  
integer d = (integer)t_value;  
while(d>0) {  
    if (m_value=="12"){ prim_x+=delta_move; llSay(0, "Move forward: "+ (string)prim_x);  
    } else if (m_value=="6"){ prim_x-=delta_move; llSay(0, "Move backward: "+ (string)prim_x);  
    } else if (m_value=="3"){ prim_y-=delta_move; llSay(0, "Move right: "+ (string)prim_x);  
    } else if (m_value=="9"){ prim_y+=delta_move; llSay(0, "Move left: "+ (string)prim_x); }  
    llSetPos(<prim_x, prim_y, prim_z>);  
    d = d - delta_time; } // End while  
llSay(0, "New pos: "+ (string) llGetPos());
```

24

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Farruscos (robots)

Arduino: Duemilanove

Sensores: IR, LDR, Bumpers

Actuadores: Motores, Servos, LEDs

Cuidados!

- A **electricidade** é para ser levada a sério...
 - Verificar sempre a voltagem/amperagem dos equipamentos;
 - Verificar /confirmar sempre pinos e polaridades dos componentes;
 - O material tem sempre razão;
 - A electricidade estática corporal pode danificar chips (sapatos de borracha!?!);
 - “Descarregar” o corpo (terra/massa);
 - Evitar curto-circuitos... verificar ligações antes de ligar;
 - Desligar sempre equipamentos antes de fazer qualquer montagem ou alteração nas ligações.

Nomenclatura

27

- **Sketch:** programa ~ c/Java - compilado para o Arduino (cross-compiler);
- **Pin:** pino do micro de entrada/saída onde ligamos sensores/actuadores, e.g., LEDs, LDR, etc.;
- **Digital** port: porta digital cujo valor pode ser HIGH ou LOW (aka on/off, one/zero);
- **PWM** port: porta digital de saída cujo valor pode variar numa dada gama, e.g., controlo velocidade motor, posição servo, brilho LED, etc.;
- **Analog** port: porta analógica de entrada cujo valor pode variar numa dada gama, e.g., sensor temperatura.

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Preparar ambiente

28

1. Download ambiente desenvolvimento Arduino:
 - <http://arduino.cc/en/Main/Software>
2. Ligar Farrusco <-> USB-cable-miniUSB <-> PC;
 - Seleccionar drivers USB (FTDI – FT232R USB UART):
`/drivers/FTDI USB Drivers`
3. Executar ambiente desenvolvimento Arduino (IDE):
 - Seleccionar Porta séria: Tools>Serial Port>COMx);
 - Seleccionar Controlador: Tools>Board>Duemilanove...
 - ATmega 328 – Farrusco 1;
 - ATmega 168 – Farrusco 2.

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Tools Bar

29



- **Verify | Compile:** verificar erros sintáticos;
- **Stop:** parar monitor série;
- **New:** criar um novo programas (sketch);
- **Open:** seleccionar um programa existentes (**File | Sketchbook**);
- **Save:** gravar o programa corrente;
- **Upload to Arduino:** “carregar” o programa (sketch);
- **Serial Monitor:** visualizar tx/rx porta série entre PC<->Arduino

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Programação

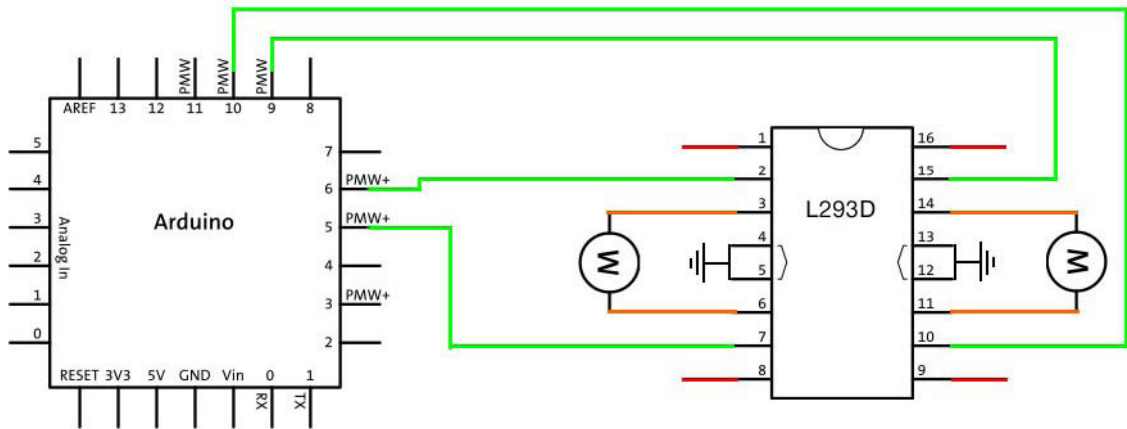
30

1. Carregar (load) programa (sketch) no micro:
 - Posicionar jumpers – **USB**
2. Executar Farrusco/Programa (sketch):
 - Posicionar jumpers – **XBee**
3. Executar Farrusco Modo Debug:
 - Posicionar jumpers – **XBee**
 - Ligar cabo USB & IDE Monitor Série

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Controlo motores (Ponte H)

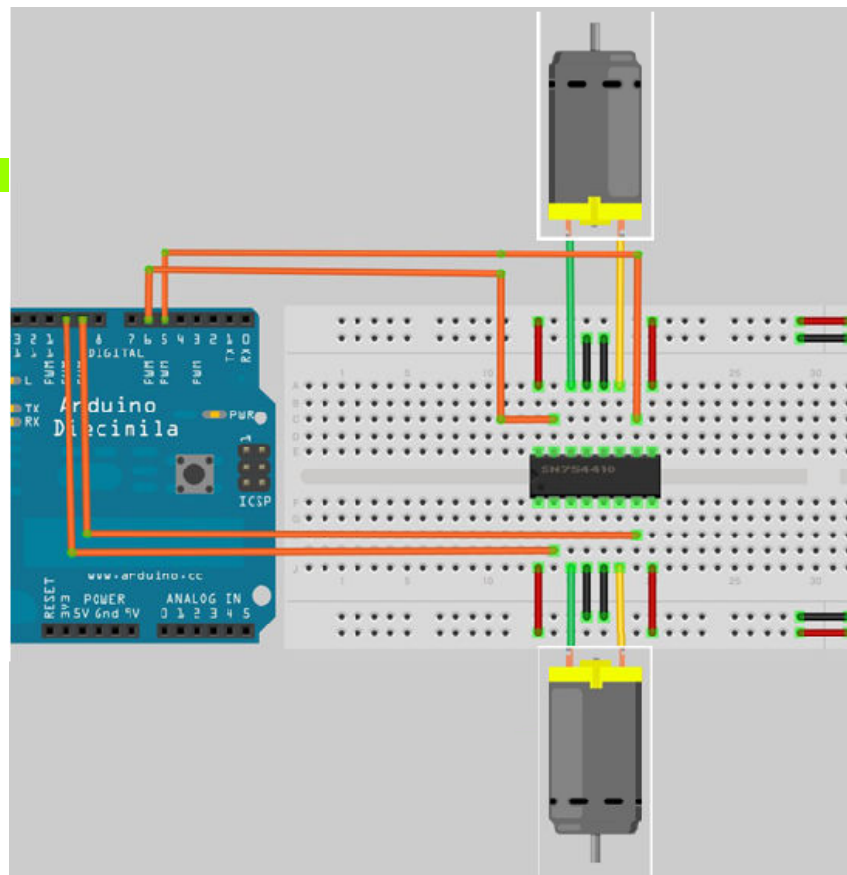
31



Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Montagem

32



Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Controlo motores

33

```
// Movimentos pré-definidos do Farrusco
void motor_stop();
void drive_forward();
void drive_backward();
void turn_left();
void turn_right();
// Movimento genérico temporizado
void drive_step_motor(int dir, int time);
// Movimento genérico + Tx para SL
void drive_motor_tx(int dir);

// Movimentos genéricos PWM
void drive_forward_pwm(int leftPot, int rightPot);
// Movimento genérico PWM + Tx para SL
void drive_forward_pwm_tx(int leftPot, int rightPot);
```

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Parsing Comandos ($i=1 \& m=12 \& t=888$)

34

```
int getParamValue(char* str, char paramName)
{
    int i=0, j=0, k=0, value=-1;
    char paramValue[10]="";
    for(i=0; str[i]!='\0'; i++){
        if (str[i]==paramName){
            for(j=i+2, k=0; (str[j]!='&' && str[j]!='\0'); j++, k++){
                paramValue[k]=str[j];
            }
            paramValue[k]='\0';
            if (paramValue[0]!='\0'){
                value=atoi(paramValue);
                return value;
            }
        }
    }
    return value;
}
```

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010

Recepção/Execução Comandos

35

```
void serialreadCallback()
{
    int i=0, dir=0, time=0, j=0, ctrl=0, robotIdRx=-1;
    msg[0]='\0'; // Loop through all the available elements of the message
    while (message.available()){
        message.copyString(msg, MSGSIZE); // Check Farrusco ID
        robotIdRx=getParamValue(msg, 'i');
        // If message directed to this Farrusco...
        if (robotIdRx == FARRUSCO_ID){
            dir = getParamValue(msg, 'm');
            time = getParamValue(msg, 't');
            // Move...
            if (dir>=0 && time>0){ drive_step_motor(dir, time); }
        }
    }
    msg[0]='\0'; // Clean msg buffer
}
```

Cross-reality: interligando o mundo real e o mundo virtual @ IPG 16/07/2010