


STR: aula_2

Trabalho 1 – parte

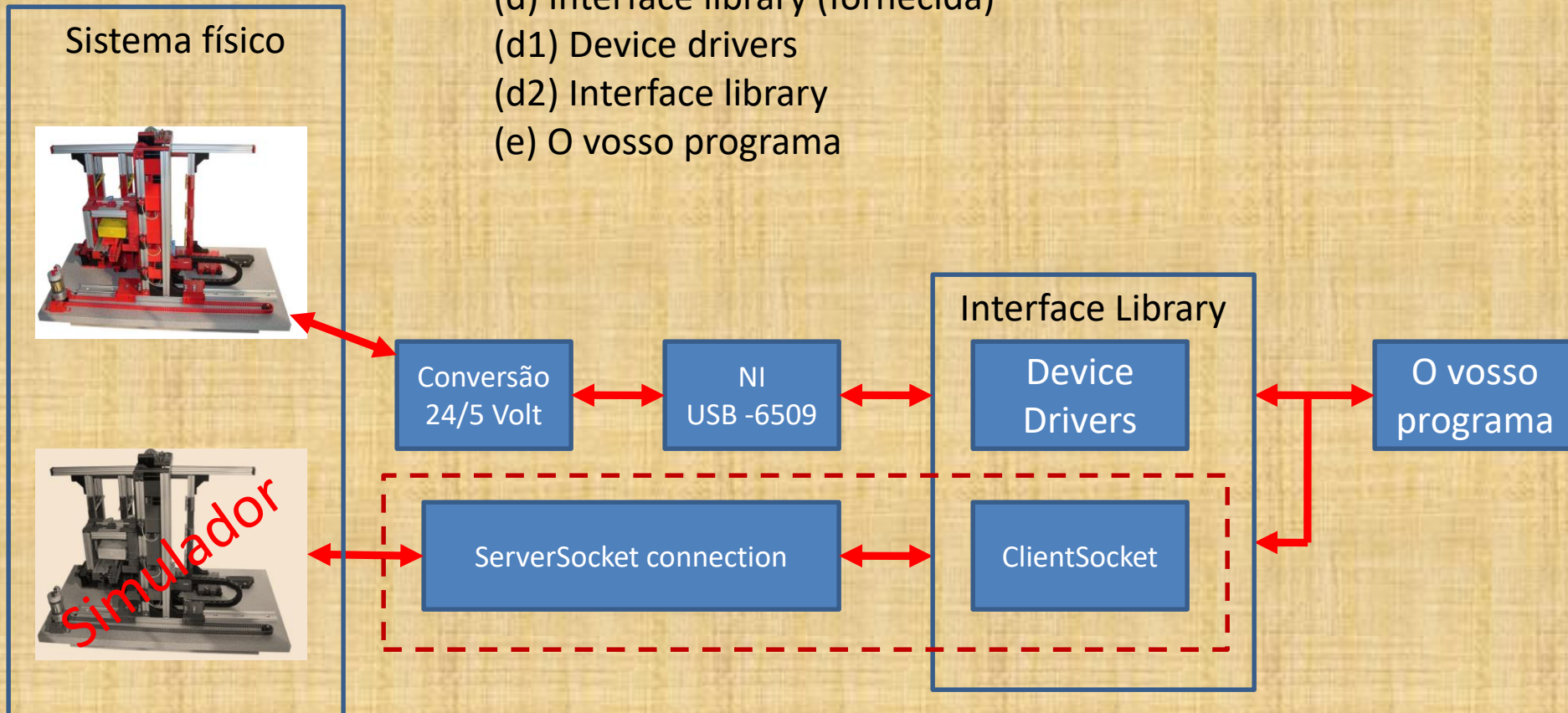
Rotinas de Acesso ao kit – C++

Planeamento das aulas

- Semana 21 Set:
 - Conceitos básicos de I/O (com experimentação)
- Semanas 28 Set, (5 Out), 12 Out:
 - 1ª parte do trabalho 1 (em C) ←  HERE
- Semanas 19 Out, 26 Out:
 - 2ª parte do trabalho 1 (RTlib)
- Semana 2 Nov:
 - Exercícios sobre Java
- Semanas 9, 16, 23 Nov:
 - Trabalho 2: Java em Tempo Real
- Semanas 30 Nov, 7 Dez, 14 Dez:
 - Trabalho 3 (plc)

(Relembrando) estrutura dos kits

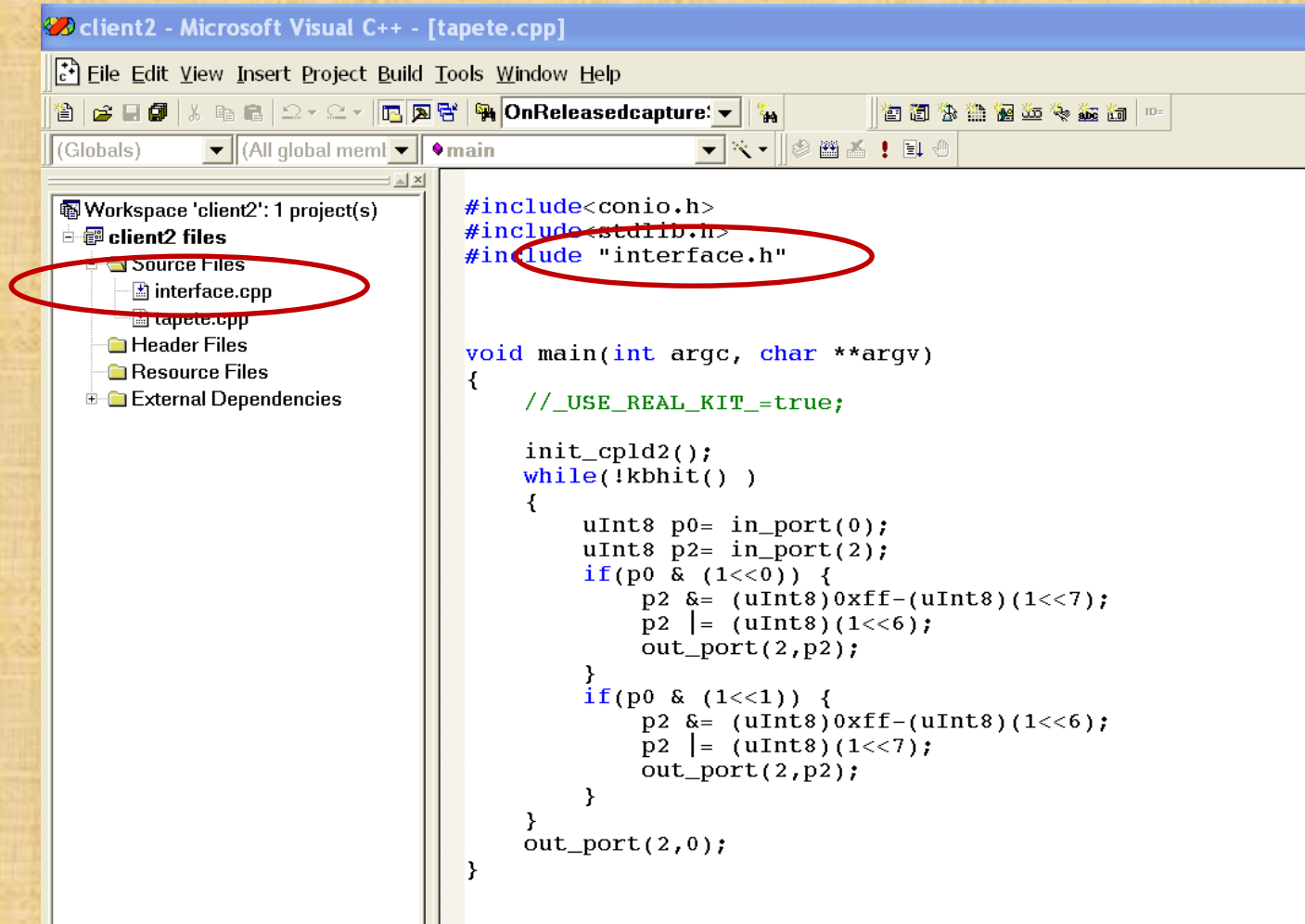
- (a1) Sistema “físico”
- (a2) Simulador do sistema “físico”
- (b) Conversão 24/5 Volt (com opto acopladores)
- (c) Placa IO digital NI USB-6509
- (d) Interface library (fornecida)
 - (d1) Device drivers
 - (d2) Interface library
- (e) O vosso programa



Interface.cpp , interface.h

Adicionar "interface.cpp" ao projecto

Adicionar "wsock32.lib" ao Linker



Exemplo: Acesso ao tapete (código-fonte)

```
#include "stdafx.h"  
#include <stdio.h>  
#include <conio.h>  
#include "NIDAQmx.h"  
#include "interface.h"
```

```
int main(int argc, char* argv[]
```

```
{
```

```
    create_DI_channel(0);
```

```
    create_DI_channel(1);
```

```
    create_DO_channel(2);
```

```
    uint8 p0 = ReadDigitalU8(0);
```

```
    printf("%02X",p0);
```

```
    WriteDigitalU8(2,0x01);
```

```
    close_channels();
```

```
    return(0);
```

```
}
```

Inicialização dos portos
de entrada/saída

Leitura

Escrita
(fazer com cuidado)

Leitura de sensores

- Como ler um determinado bit de um determinado porto/canal?
- Exercício
 - Dado o número 223, saber qual o valor do bit 4
 - Fazer uma função

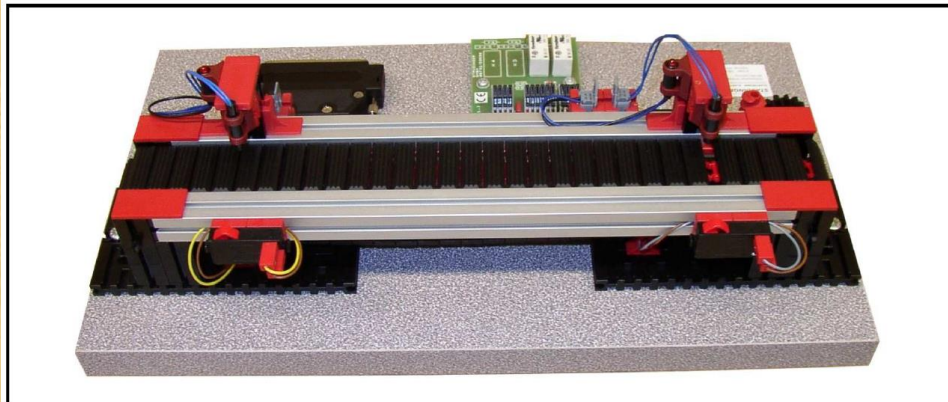
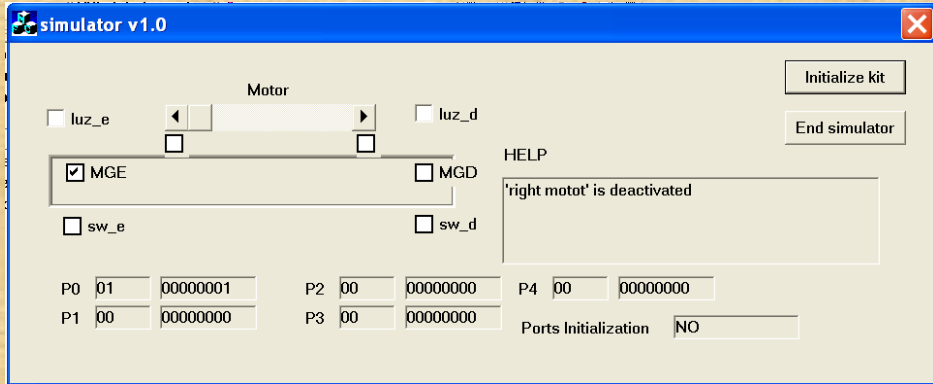
```
bool getBit(uint8 value, uint8 n_bit)  
{  
    return( value & (1<<n_bit));  
}
```

“Mexer” nos actuadores

- Como alterar um determinado bit de um determinado porto/canal associado a um determinado actuador?
- Exercício
 - Dado o número 00_H , colocar o bit 2 a “um”.
 - Fazer uma função

```
void setBit(uint8 &variable, bit, bool value)
{
    uint8 mask_on = (uint8)(1<<bit);
    uint8 mask_off = (uint8)0xff - mask_on;
    if(value) variable |=  mask_on;
    else      variable &=  mask_off;
}
```

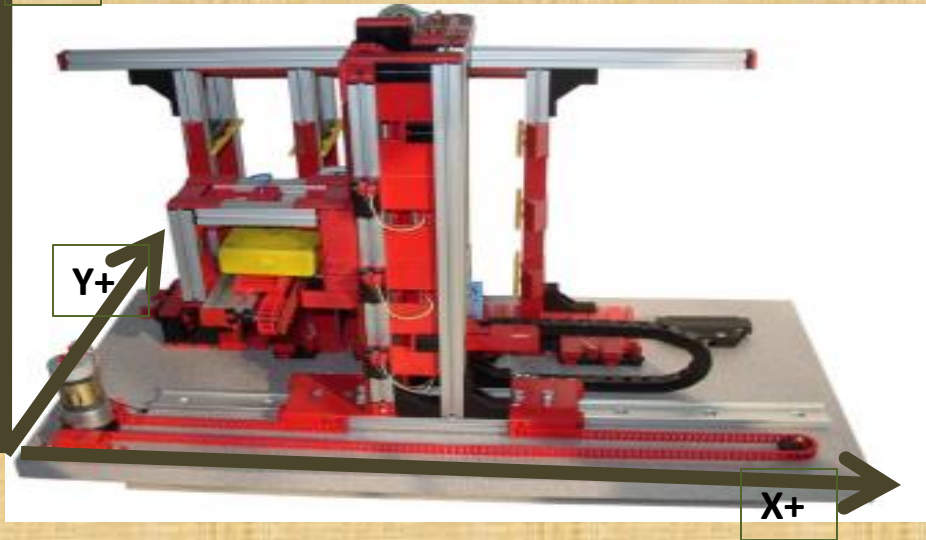
Let's do it again! 😊



```
#include<conio.h>
#include<stdlib.h>
#include "interface.h"

void main(int argc, char **argv)
{
    // _USE_REAL_KIT_=true;
    init_cp1d2();
    while(!kbhit() )
    {
        ulnt8 p0= in_port(0);
        ulnt8 p2= in_port(2);
        if(p0 & (1<<0))
        {
            p2 &= (ulnt8)0xff-(ulnt8)(1<<7);
            p2 |= (ulnt8)(1<<6);
            out_port(2,p2);
        }
        if(p0 & (1<<1))
        {
            p2 &= (ulnt8)0xff-(ulnt8)(1<<6);
            p2 |= (ulnt8)(1<<7);
            out_port(2,p2);
        }
    }
    out_port(2,0);
}
```

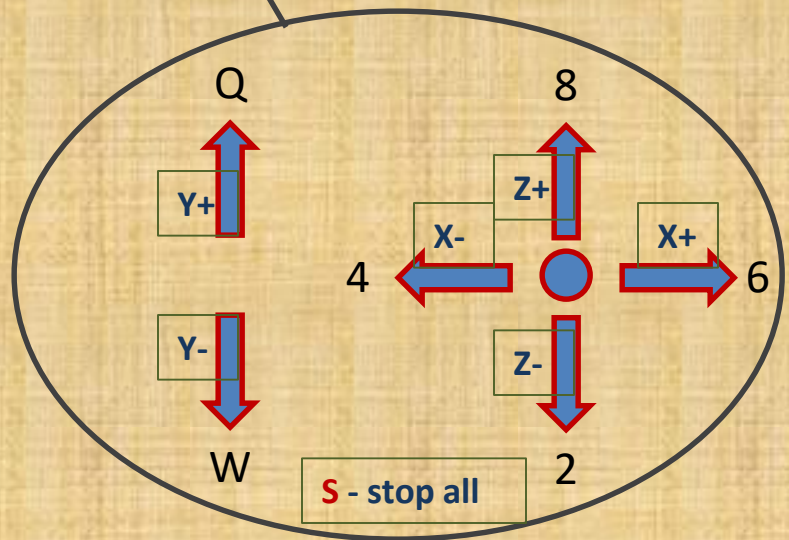
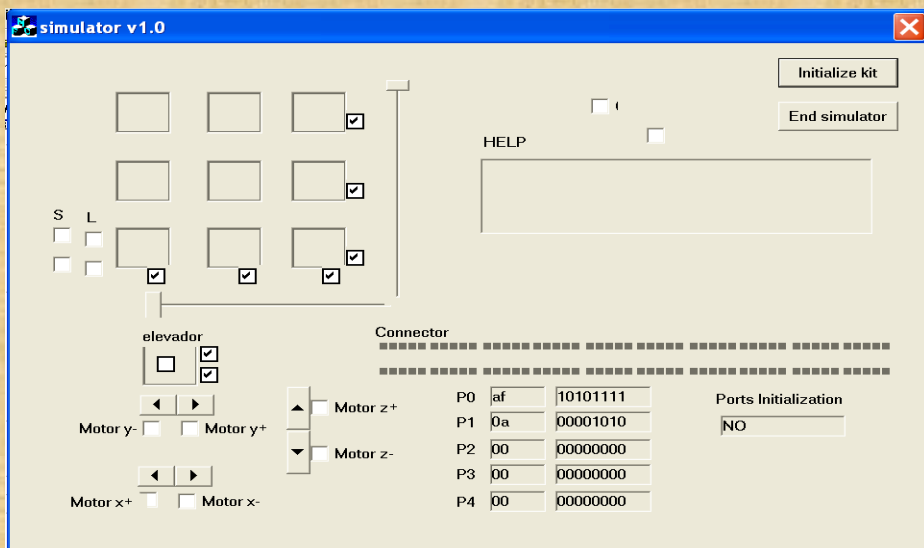

Work today: Make a control pad



```
#include<conio.h>
#include<stdlib.h>
#include "interface.h"

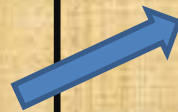
void main(int argc, char **argv)
{
    // _USE_REAL_KIT_=true;
    .....
}
```

Move along axis only until reaching positioning sensors.



SOME Functions you need to do...

- `move_z_up();`
- `move_z_down();`
- `move_x_right();`
- `move_x_left();`
- `move_y_inside();`
- `move_y_outside();`
- `stop_x(), stop_y(), stop_z()`
- `put_piece();`
- `get_piece();`
- `Goto(X,Z)`
- `Is_at_z(Pos),`
- `is_at_x(pos),`
- `is_at_y(pos)`
- `Is_at_cell(x,z)`



- Parts must not move beyond their limits
- Move 'x', and 'z' iff cage conveyor is at the 'center' sensor.
- Move 'y' iff cage correctly positioned at a cell.
- `put_piece()` and `get_piece()` iff correctly positioned at a cell

.... among others.

Recomendações

- Implementar tudo no simulador
- Implementar primeiro as funções movimento em X e Z (e y apenas depois desses eixos)
- Reutilizar as funções:
 - `Goto(x,z)` – necessita de `move_x`, `move_z`
 - `Put_piece()` , `get_piece` – necessita de `goto_xz()`, `move_y`
- Testes no kit real periodicamente para verificar consistência com resultados simulados
- Testes no kit real só podem ser efectuados na presença do respectivo professor/assistente/monitor.