STR: aula_3

Trabalho 1 – parte

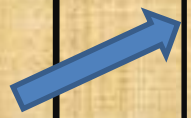
Rotinas de Acesso ao kit – C++ (2ª parte) – [em inglês]

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)
- 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)

Recalling last lesson

- **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 celll.
- put_piece() and get_piece() iff correctly positioned at a cell

.... among others.

Routines (1)

```
void goto_x(int x)
{
   if (actual_x()< x)
      move_x_right()
   else
      move_x_left()
   while( !is_at_x(x)  {do nothing}   //    while position not reached
   stop_x();   // já chegou.
}

void goto_z(int z)
{
//  “find seven differences between this and previous routine!” ☺
}
```

Routines (2)

```
void goto(int x, int z, int where) // where 0-> means get position, 1-> put
{
  goto_x(x);
  goto_z(z);
}
```

- What is the big concerns in this solution?
- Can you make better functions?
- How would you change goto_x and goto_z in order to consider “where” para meter?
- How about using realtime concepts?

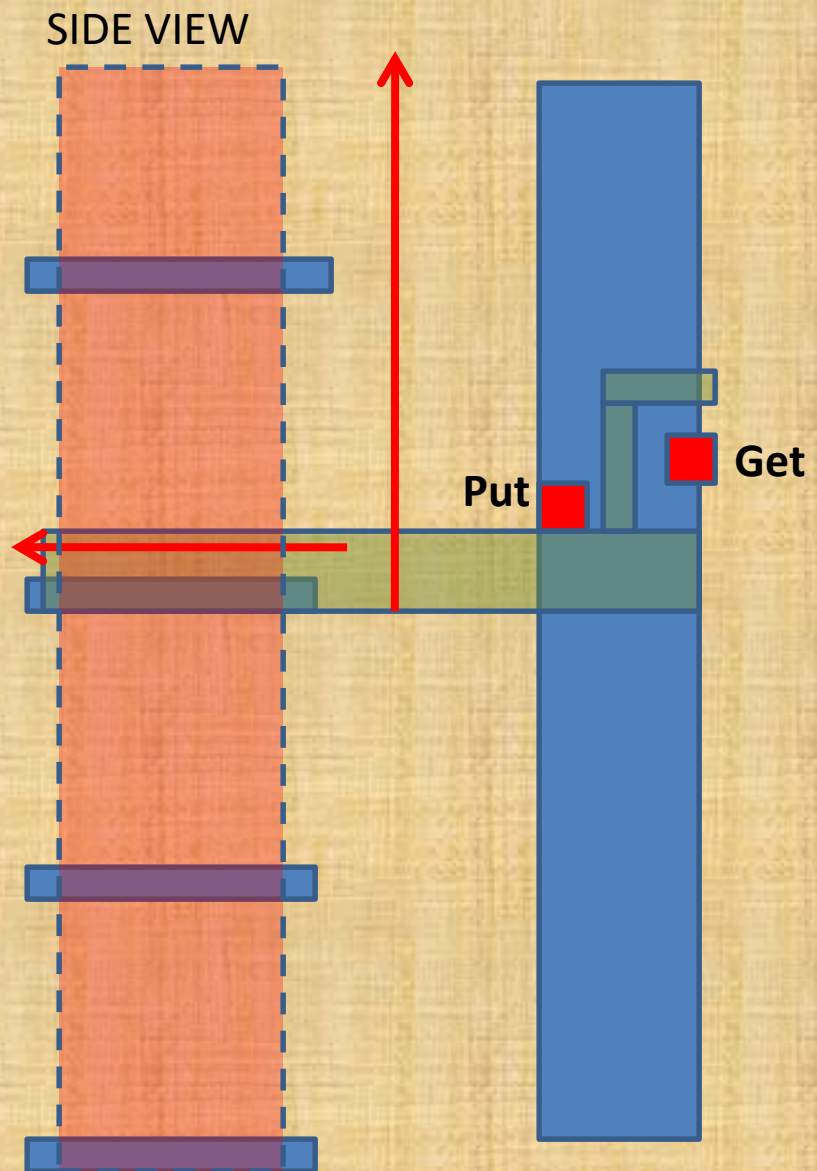
Routines (3)

```
void put_piece_in_cell(x,z)
{
   goto x position
  goto put position of z cell
  move y inside until reaching the “inside” sensor
  goto “get” position of z cell
  move y outside until reaching the “center” sensor
}

void get_piece_from_cell(x,z)
{
   // similar to previous one
   // in terms of z axis, go first to “get” sensor
   // get inside cell
   // goto up z until reaching put sensor, to grab the piece
   // get outside the cell.
}
```

Routines (4) – put_piece_in_cell

```
void put_piece_in_cell(x,z)
{
  goto(x,z, 1);  //1-> put position
  // it is expecta that "center" sensor is
 //activated
 move_z_up() // for the case of the figure
  while("put" sensor not activated) {do nothing}
  stop_z();
  move_y_inside();
  while("inside" sensor  not activated) {do nothing}
  goto_z(z, 0);  // go down to get position
  move_y_outside();
  while("center" sensor  not activated) {do nothing}
}
```



Recomendations

- Always implement everything in the simulator first. Only after all verifications are successful, you are allowed to test your work in real kit.

- Movements in X,Z are allowed iff “center sensor” in the elevator is activated

- Testing the put_piece_in_cell(), get_piece_from cell() requires re-positioning of the zz sensores, as they are displaced from their correct positions. So, leave testing in real kit for next lesson.