Parking Elvis

Back into a parking slot

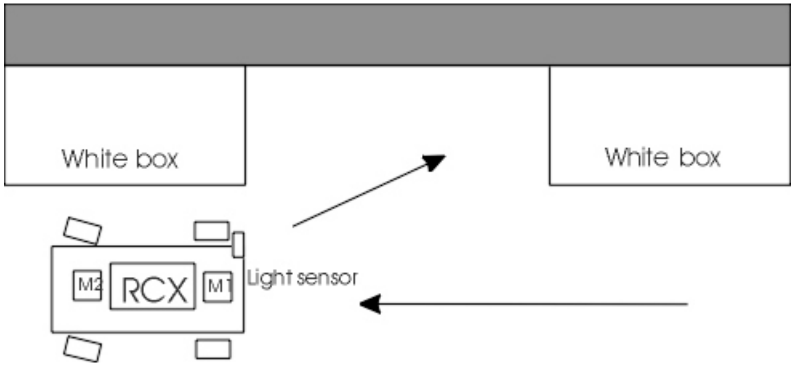
Embedded Software Engeneering 2004

Introduction

- Purpose of the project
  - Implement a Real Time Operating System
  - Practical use of the RTOS is controlling a robot

Our Project

- Developed a Lego Mindstorm car which parks automatically into a free parking slot

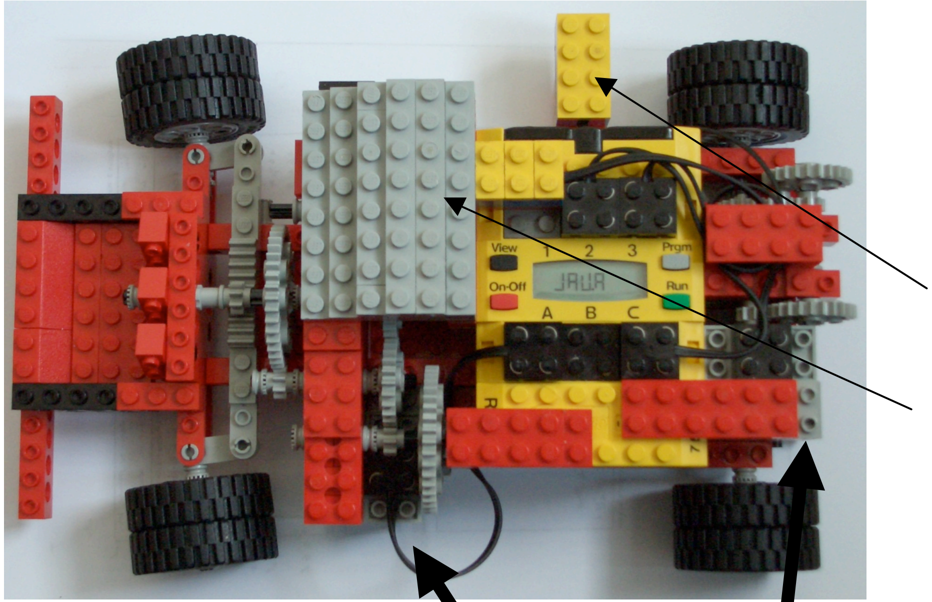


- Conditions
  - The car shall measure the lenght of the parking slot
  - Reverse into the slot
  - The car should never touch a box

Implementation

- Hardware
  - Lego Mindstorms: Robotic Inventions (with RCX)
- Software
  - Programming Language: Java
  - RCX: Lejos
- Technical Implementation
  - RTOS and vehicle control run on a PC
  - Communication between RTOS and the vehicle runs via a Infrared sensor and the package Rcxdirect (client/server tool).
    - Advantage: We are not bounded to the limited capacity of the RCX and lejos.

Hardware



2 Light – sensors

- for computing the length of the slot and
- to controll the steering

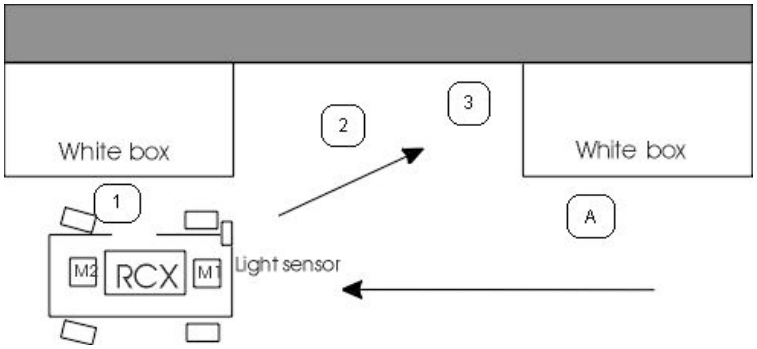
2 Motors

- one to direct the car
- one for driving forward and backwards

The parking Event

Position 0: start of the parking slot

```
start:
release (move forward)
release (check slot)
future (position1_reached,back_in)
```



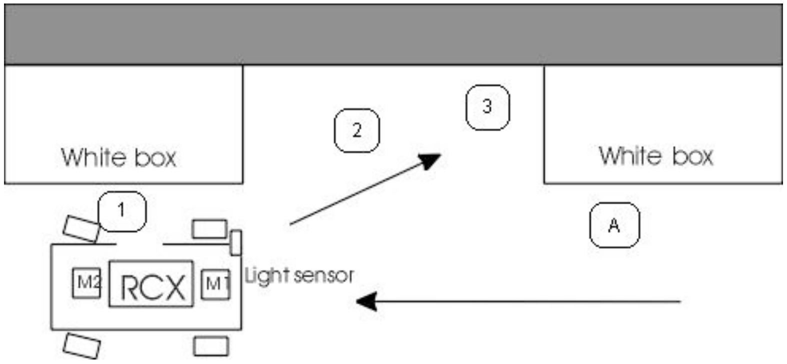
Position 1: End of parking slot

```
back_in:
release (calculate_positions)
                    → calculates: steering angle
                              position 2 (time until position 2 reached)
                              position 3 (time until position 3 reached)
                    realease (steer_right)
                    release (move backward)
                    future(position_2_reached, steer_left)
```

The parking Event

Position 2: change steering

```
                    steer_left:
                    release (center_gear)
release (steer_right)
release(move_backward)
future(position_3_reached, finish)
return
```



Position 3: finish

```
finish:
release (center_gear)
return
```

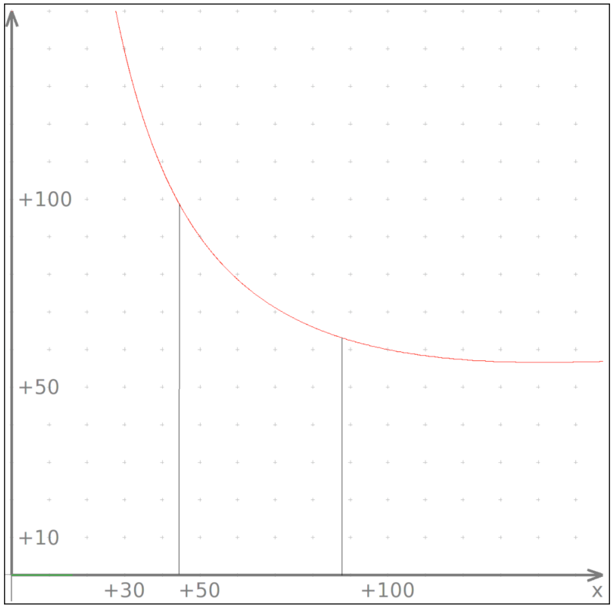
Calculate Position

From the time the car needs to pass the parking slot we compute

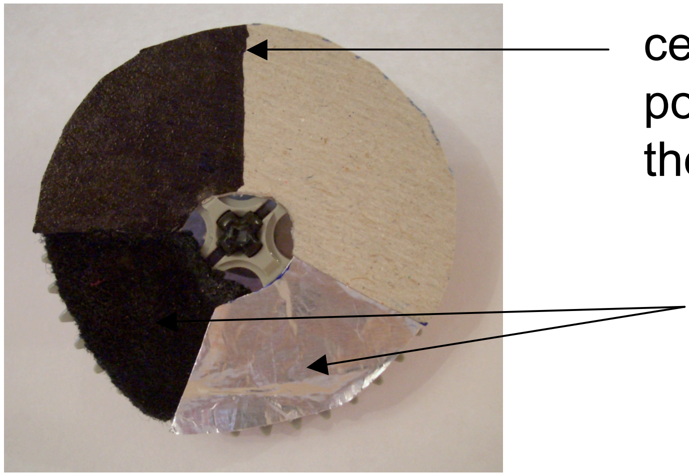
- the angle of steering

  Function: f(x) = 4000/x + x/5

- the time the car needs to reach its positions



Steering



centered
position of
the gear

out of bounds

- The wheel rotation controls the steering of the car
- The sensor interpretes the colors on the wheel to determine the position of the gear

The OS

Based on the principles we heard in the course

- adjustable functionality (e-code)
- adjustable scheduler (s-code)
- interprocess communication via ports
- trigger based event handling

Capable of preemtive multitasking

E-Code/S-Code

E-Code Example:

```
release(move_forward)
future(position_1_reached, finish)
```

S-Code Example:

```
dispatch(move_forward, position_1_reached)
```