Department of Computer Sciences

UNIVERSITÄT SALZBURG

VP Embedded Software Engineering
WS 2010/2011

WitAl-Puck Collector Documentation

February 5, 2011

Project Members:
Christian Alt, christian.alt@stud.sbg.ac.at
Natalie Wittinghofer, natalie.wittinghofer@stud.sbg.ac.at
Maria Martin, Maria.MartinCiviac@stud.sbg.ac.at

Contents

1 Problem Definition of the Robot

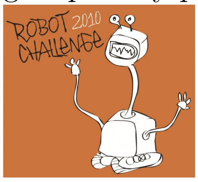
1.1 Robot Challenge

The Robot Challenge is a European championship for self-made, autonomous and mobile robots. More than 500 robots are expected to be presented to the actual robot challenge.

1.2 Puck Collect

This competition calls for special interaction between sensor technology, mechanics and artificial intelligence.

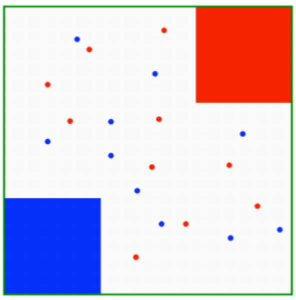
The main goal of the puck collect challenge is to build robots which have to collect small discs (called pucks) of a specific color on the given field. Two robots battle against each other on the field. At the beginning each robot gets a color assigned (red or blue) and is positioned on its homebase which also has the assigned color. The homebases are located at opposite corners of the field. During the challenge the robot has to move across the field and collect all the pucks of its own color. In the end it should bring all these pucks to its homebase. If it collects pucks of the wrong color and brings them back to its homebase, it gets penalty points. The robot with the most collected pucks wins.

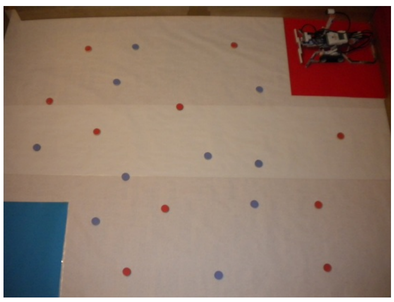


1.3 Adaptations

We have changed some of the general requirements because we didn't have enough space and/or materials for some of them.

- **Field dimensions:** The field at the Robot Challenge event has a dimension of 280x280 cm. Our field had a dimension of 190x155 cm.
- **Homebase:** Since the field is smaller, our homebases are smaller too. At the Robot Challenge the homebase has a dimension of 70x70 cm. Our homebases have a dimension of 50x50 cm.


Robot Challenge field


Our field

- **Pucks:** The pucks at the Robot Challenge have a diameter of 4 cm. Ten pucks are randomly placed on the field. Because of the construction of our robot our pucks have a diameter of 3.5 cm. Bigger ones would get stuck under the robot. Since our field is a little bit smaller than the robot challenge field, we placed eight pucks randomly across our field. The basket of our robot could carry up to 16 pucks.

- **Robots:** Usually there are two robots on the field which battle against each other. Since we had only one robot there will only be one robot which will collect the red pucks (and discard the blue ones) on the field and will bring them back to the homebase.

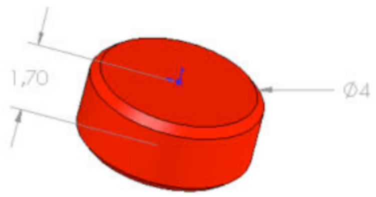


Figure 1: Regulatory Puck

2 Robot Design

2.1 Used Components

The following components were used to build and program the robot:

- **Lego Mindstorms:** The Lego Mindstorms NXT 2.0 Kit with its three engines and different sensors has been used to construct the robot. In addition to that we added a compass sensor.

- **Not Exactly C:** NXC is the programming language for the Lego Minstroms NXT and has been used to program the robot.

- **Bricx Command Center:** BricxCC is the integrated development environment for NXC and was used for the implementation of our project.

2.2 Design

We used **three engines**. Two of them are in the front and represent the wheels. One engine is in the back and opens or closes the gate to the basket. The main design of the robot is based on the task of collecting the pucks. In the front we have a **fork** which collects the pucks in its reach. In the back of the robot the **basket** is located which stores the collected pucks of our color.
The following sensors have been used (see Figure 2):

- **Color sensor:** This sensor is placed beneath the NXT brick. Each puck will be moved to the color sensor. Based on the measured color of the sensor the gate to the basket will be opened or closed.

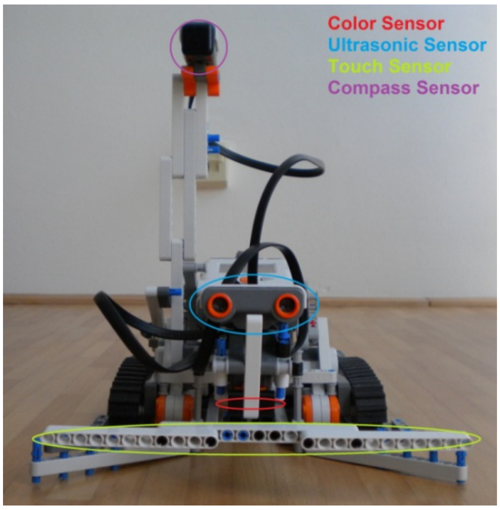


Figure 2: Sensors

- **Ultrasonic sensor:** This sensor is located in the front of the robot and measures the distance to the wall. If a wall is close the rotation of the robot will be initiated.
- **Touch sensor:** This sensor is also located at the front of the robot. To expand the reach of the sensor we added an additional plate which has the width of the robot. When this plate touches the wall the touch sensor will be activated and the robot will stop and start to rotate.
- **Compass sensor:** This sensor is used to control the orientation and rotation of the robot.
  At first we tried to program the robot without the compass sensor. Our first idea was to count the motor rotations of the wheel engine. When the robot senses a wall and decides to rotate, we let the motor of one wheel rotate for an amount of rotations. Problem with this idea was, that when the robots rotates wheel spin can happen especially when parts of the robot touch the wall. So this idea was too inaccurate for our strategy.
  Our second idea was to measure the time of a rotation and let the robot rotate for an amount of time. We again had the problem with wheel spin here. Rotations could take a random amount of time. Another problem was that the performance of the engines highly depend on the power of the batteries. This means that if you have full batteries the engines have more power and therefore the rotation takes a shorter amount of time. On the other hand, if the batteries are very low the rotation might take much longer.
  In the end we decided to use a compass sensor. Here, the rotation is based on the sensor data. We can measure the position of the robot and add an amount of degrees and rotate until we have the desired position. It is much more accurate than the other ideas. There are still a few things which you have to consider when using the compass sensor. First you have to mount it 15 cm away from the motors and 10 cm away from the NXT brick. Next you must keep the compass leveled all the time to get correct data. And even if you do all those things the sensor data still can be influenced by the structure of the robot or external things, like the refrigerators or

other metal object.

3 Subtasks of the robot

The general idea is that the robot drives around on the field and tries to cover the whole area. While it is driving around the pucks that lie on its path are gathered by the fork in the front of the robot and passed underneath the robot. The robot is equipped with a basket in the back so it can carry the pucks with the right color back to its homebase. The work of the robot is split up into two subtasks that are executed in parallel. The first subtask is opening and closing the basket according to the puck color. The second subtask is driving around the robot, covering as much of the area as possible.

3.1 Puck Detection and Collection

The robot senses the puck color at the color sensor in its middle. According to the sensed color the basket in the back is either opened or closed. To make sure that the basket is still open when a red puck reaches the basket and the basket is still closed when a blue puck reaches the basket the basket stays open or closed for exactly the distance between the gate of the basket and the color sensor. To keep this mechanism independent from the motor speed, which might depend on the battery power or the course of the robot (straight ahead, cornering), the distance is measured using the motor rotation count.
Figures 3 and 4 depict the process of opening and closing the gate of the basket.

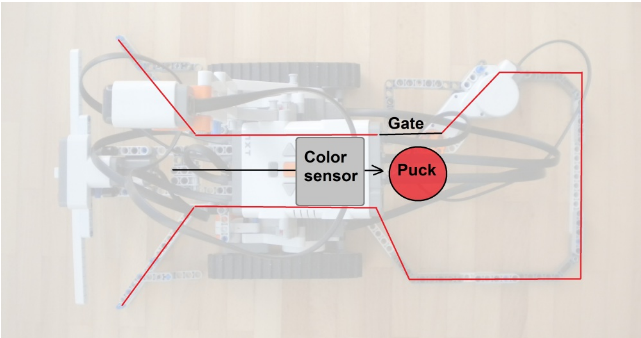


Figure 3: Red puck getting collected in the basket.

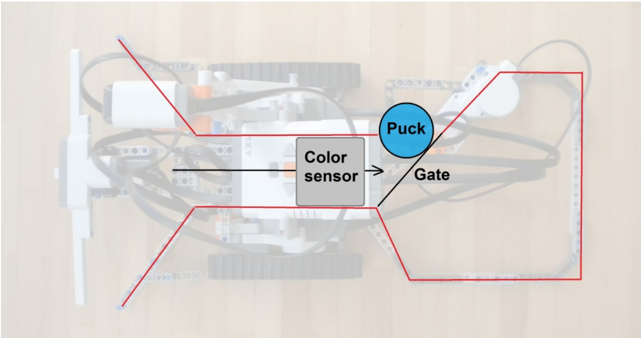


Figure 4: Blue puck being left on the field.

3.2 Driving, covering the whole field

In its second subtask the robot has to drive around the field using a strategy that covers as much of the area. That way it will collect as many pucks of its color as possible.
There are two possible strategies depicted in figures 5 and 6.
We chose the first strategy because it offers certain advantages over the second strategy. First of all, when using strategy one, the robot has a point of orientation after every length it drives when it reaches the wall. That way inaccuracies regarding the position of the robot do not add up. Another advantage is that strategy one has longer straight lines so the robot can go faster, and the third advantage is that there are no overlaps in the course of the robot.

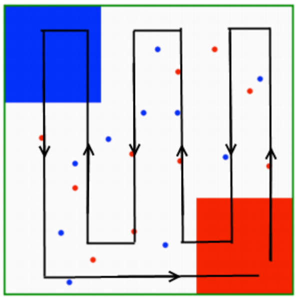


Figure 5: Strategy 1

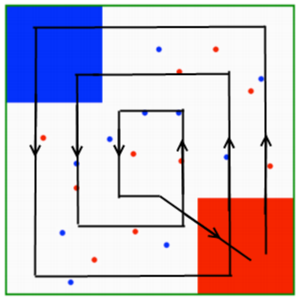


Figure 6: Strategy 2

3.2.1 Implementation of Strategy 1

The robot starts at its homebase. From there it drives straight ahead to the opposite wall. When it reaches the opposite wall it uses the compass sensor to turn for 90 degrees to the left, then drives one length forward and then again turns 90 degrees left, completing the 180 degree turn.
It then drives to the wall where it started and does the same turning procedure to the right. The robot can keep a bigger distance to the wall at the bottom of the field, as it is going to cover that area on the final length to its homebase.
The robot continues driving its upwards and downwards lengths until it reaches the left wall. The robot knows for sure that it has reached the left wall when it senses the wall on the small length it drives forwards between the two 90 degree turns, and the blue homebase of the opponent.
When the robot has reached the left wall it repeats to drive to the bottom wall and afterwards to the right wall until it reaches its own homebase. The robot knows when it has reached its homebase when it senses the wall and the color of the homebase.
If any of the 90 degree turns takes more than a specific amount of time (in our configuration 8 seconds), the robot assumes that it is stuck and tries to move either forwards or backwards a bit to free itself again. To determine wheter it should move forwards or backwards it uses the ultrasonic sensor.