

Assignment 3

Cupak Miroslav,
David Nemecek,
Nizan Juraj,
Streck Adam

April 22, 2012

1 Concept

We want to build a robot capable of:

1. Calibrating its track and light detection sensor.
2. Reading instructions in some graphical representation while riding over them.
3. Visible execution of those instructions.

2 Design

2.1 Behaviour

There are four basic things such a robot could do:

1. Move and steer.
2. Make sounds.
3. Draw on LCD of the NTX brick.
4. Use third motor to do something.

I would pick two of them - move and make sounds, unless movement is too loud for sound to be audible, I will furthermore explain how to utilize them.

2.2 Construction

What sort of the hardware we use? How did you design it? What problems you have encountered while building it?

For self-calibration of the position is best to use three wheel vehicle with sign free wheel in the front and two wheels with independent motors in the back (sort of the thing that is built right now).

However, I think that 4-wheel vehicle with single motor for movement and second one for steering will be better for execution of instructions and overall more awesome, so I suggest building that one - there is a howto for one pretty, I would build on that: http://www.nxtprograms.com/NXT2/race_car/steps.html

2.3 Movement programming

How the movement routine looks? What is it capable of?

2.4 Light calibration

How do you calibrate light sensors?

3 Coding

3.1 Code representation

How do you encode instructions? What are advantages and disadvantages of this representations? How is it read? Do you use some special symbols?

I see two ways how to make coding (I already nicknamed the language *GreyDot*, do you agree?):

- Use scale for gray with 10 steps.
- Use triples of black (0) and gray (1).

First option would be better, but we need to test if we can always calibrate the sensor so it reads correct instructions no matter what light conditions are - but it should work, when I tested it, it was pretty sharp. In this case we have 9 steps for different instructions and 1 for background. Second option uses only three steps - black for 0, grey for 1 and white for background and I am pretty positive that this option would work even without self calibration. We would then have 8 combinations.

3.2 Special symbols

There is at least one symbol left for each line - these will be used as special symbols - for now we may need these symbols:

1. maybe - Start color calibration (should be black-black).
2. maybe - Start path calibration.
3. sure - Start reading instructions.
4. sure - End reading instructions and execute.

3.3 Writing instructions

How do you write instruction files? How do you plot them?

I would write instructions in simple ASCII file and then have some script plot it.

I would recommend using two lines - first for movement with numbers of instructions of some symbolic representation - j for steer left etc.

3.4 Movement calibration

How do you ensure robot stays on the path and reads instructions? How do you cope with errors?

3.5 Semantics

What are the semantics of the code? What are their capabilities?

I would use these two sets of instructions:

- Movement
 1. Move forward straight.
 2. Move forward left to make 30 turn.

3. Move forward right to make 30 turn.
4. Move backward straight.
5. Move backward left to make 30 turn.
6. Move backward right to make 30 turn.
7. Stay / Stop.

- Sound

1. Beep note A for 1/3 of period.
2. Beep note A for 2/3 of period.
3. Beep note A for 3/3 of period.
4. Beep note B for 1/3 of period.
5. Beep note B for 2/3 of period.
6. Beep note B for 3/3 of period.
7. Be silent.

One line of instructions would be for movement only, the other one would be for sound only. After reading instructions our robot would always execute both instructions at once for a fixed period of time.

3.5.1 Example

Using only numbers of instructions following code:

```
222222
363636
```

would make a circling firetruck :)

3.6 Movement programming

How do you executed the code? How do the underlying routines look?