Using Pololu‘s 3pi robot in the education process

Branislav Thurský
Alexander Dubcek University of Trenčín
Faculty of mechatronics
Trenčín, Slovakia
thursky@tnuni.sk

Gabriel Gašpar
Alexander Dubcek University of Trenčín
Faculty of mechatronics
Trenčín, Slovakia
gasparg@gmail.com

Abstract— **The paper presents a sample of usage of the Pololu‘s mobile robot 3pi at the interactive teaching of robotics. Since this is a mobile robot with a circular structure and two controlled wheels, it is mainly suitable for the role of following the line, movement in the maze and possibly after adjusting for robotic soccer. In our presented example of it's usage we are dealing with following lines forming a maze. Thinking of the robot can be demonstrated using described algorithm of finding a way in the maze. The described algorithm allowed primary school students in the hobby group better understand the challenges and problems associated with programming a mobile robot and to develop creative thinking of future robotic systems programmers.**

Keywords - component; formatting; robot, hardware, software, algorithm

I. Introduction

At the beginning we are introducing the design and parameters of 3pi robot. 3pi robot is designed so that it can be used in the tasks of following the line and finding the way out of maze. The maze is represented by perpendicular lines with a labeled target point. 3pi robot is a small size (diameter 9.5 cm, weight 83 g) and is powered by four AAA batteries, from which we obtain by using a unique energy system "boost-converter " voltage 9.25 V (for the engine used to power motors of the robot), regardless of the level of battery charge. Regulated voltage of 3pi robot allows to achieve speeds of up to 100 cm / sec, while achieving accurate revolutions because of constant voltage while discharging the battery.

3pi robot as mentioned has a circular shape, and is powered by two independently controlled propulsion units, consisting of DC motor, gear box and a wheel. In its heart is placed Atmel ATmega168 processor resp. ATmega328P which also provides for the management of motor units and thus allows the movement of 3pi robot. The key features of used processor Atmel ATmega328P are clock frequency 20 MHz, 32k of flash memory, 2 kilobytes of RAM and 1 kilobyte serial EEPROM memory, see Fig.1. [1]

Atmel corporation offers their Atmel AVR Studio development environment with integrated free GNU C/C++ compiler freely available for use with their micro-controllers on the internet. For the very 3pi robot is by the manufacturer prepared an extensive set of libraries designed to communicate with all the integrated hardware. This creates an excellent platform for people with experience in programming in C language for learning about the mysteries and problems solved in robotics. Or it provides a funny way to learn programming in C. In our case it was not different in our mechatronics hobby group at the Elementary School na Dolinách.

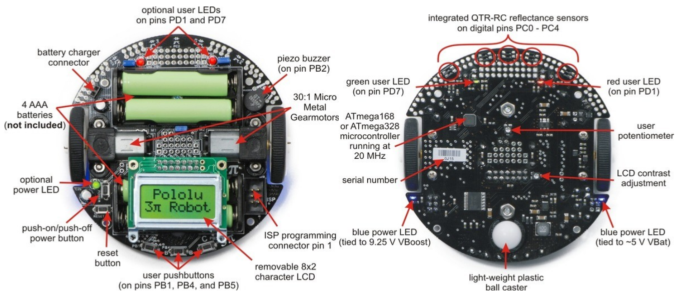

optional user LEDs on pins PD1 and PD7
battery charger connector
piezo buzzer (on pin PB2)
4 AAA batteries (not included)
30:1 Micro Metal Gearmotors
optional power LED
push-on/push-off power button
reset button
user pushbuttons (on pins PB1, PB4, and PB5)
removable 8x2 character LCD
ISP programming connector pin 1
Pololu 3π Robot
integrated QTR-RC reflectance sensors on digital pins PC0 - PC4
green user LED (on pin PD7)
red user LED (on pin PD1)
ATmega168 or ATmega328 microcontroller running at 20 MHz
user potentiometer
serial number
LCD contrast adjustment
blue power LED (tied to 9.25 V VBoost)
blue power LED (tied to ~5 V VBat)
light-weight plastic ball caster

Fig.1 Hardware platform of 3pi robot

Although the hobby group bears the name „Programming of the Atmel 8051 microcontroller", but since it is the older predecessor of processor ATmega328P and as a programming language we used the C language, it was not a great problem to switch to this platform.

Another fact that made our work on this platform easier was the mentioned set of libraries and prepared sample programs. At one such example, I would like to draw attention in this article. This example deals with finding path out of the maze which is represented by a black line. The example due to its simplicity allowed students to dive into the secrets of robotics and by a playful way it delivered the problems to be overcome.

II. Finding the way home

To start, one could define the question: What is a maze? Maze can be: a place where it is easy to get lost, figuratively confusion, opaque place, building or garden built so that one can get easily lost.

In our case, since the construction of the robot is suited to follow the line, the maze will be represented by intersecting lines on a perpendicular angle (i.e. will be developed on a regular grid), creating a maze with its start and target. For simplicity maze must be designed so that there is only one path from start to target and also to avoid creating loops in which the robot might get stuck.

This simplification of the maze allows us to use a simple strategy to get through the maze and to find a way out. In the next stage we can remove the simplifications and find a strategy that can deal with this type of maze. For better illustration, see Fig. 2.

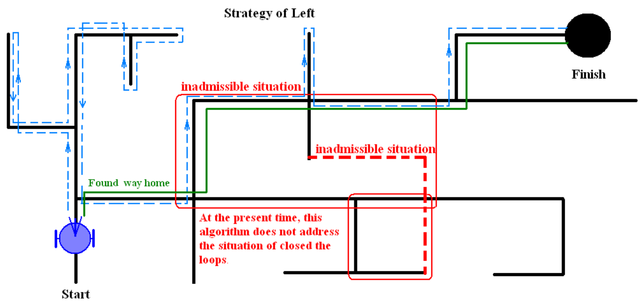

Strategy of Left
Finish
inadmissible situation
inadmissible situation
Found way home
At the present time, this algorithm does not address the situation of closed the loops
Start

Fig.2 Maze and strategy of finding the target using turn right priority.

The maze is made for a better resolution by a black line representing the path on a white background. The chosen strategy must allow a search of the entire maze and then find the shortest path from start to target. In addition to finding an appropriate search strategy it is necessary to solve a task to remember the mapped part of maze and then search for the shortest possible distance from start to target (or to the point of it's current location, which is our case) [2].

III. Strategy

It is possible to develop more strategies to search the maze. All of them will achieve different results depending on the type of maze. It can not therefore be said, that it is possible to find a single strategy that would universally, for different types of mazes, led as quickly as possible to find the target.

We chose a strategy in which there is a priority to turn left or turn right and after the gradual maze search we can simply exclude the places in the maze where there is no target. What does it mean to have priority on the left? This means that if it is possible to turn left, we turn left. In case that it is not possible to turn left, we prefer to proceed straight over to the right, and if it is not possible to turn right, then we turn around and continue with this strategy. Similarly, it is the case of the priority right. In this case, eight situations can occur in the maze: left, right, left or right, intersection left-straight-right, intersection left-straight, intersection straight-right, end of the road and end of the maze. Fig.3

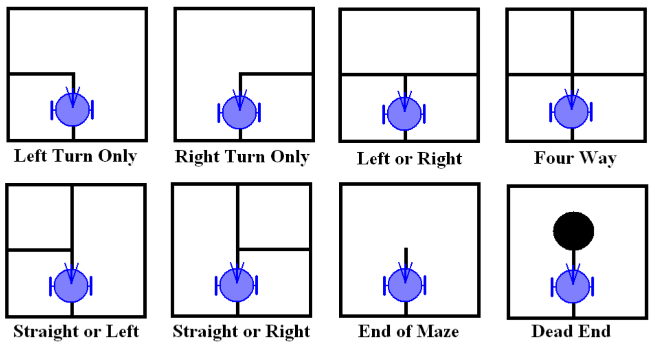


Fig.3 Situations which may occur in a maze

In order to be able to recognize situations, 3pi robot is equipped with infrared sensors that are able to detect the given situation. If the sensor is on the line, its value equals 1, otherwise its value is equal to 0. Sensors are placed on the forehead of the robot in such distance, so that we are able to detect the line and intersections in our maze. Their location is shown in Fig.1. By the combination of switching individual infrared sensors, we can evaluate the current situation of the robot in the maze and depending on this, to issue a control signal. Some situations are illustrated in Fig.4.

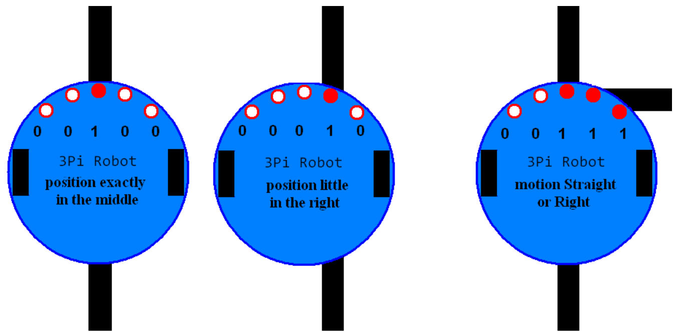


Fig.4 Infrared sensors state in different situations in the maze.

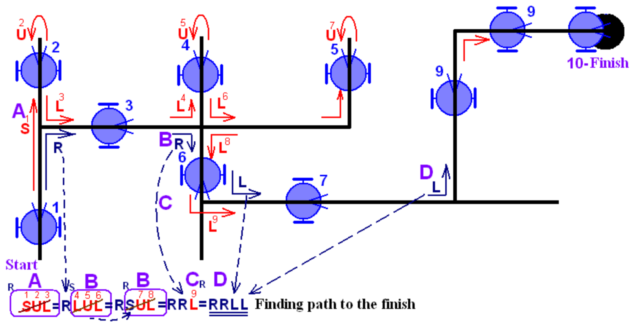
By the signals received from sensors, the robot knows what the situation is and according to our chosen strategy decides how to continue in the given situation. At the T-type intersection the robot would get 1's from all sensors. As can be seen in the figures our robot is equipped with five infrared sensors, and thus one can get 2^5 which is 32 possible combinations, some of which for our type of the maze are unlikely (for example, unacceptable are combinations such as 10101, 10001, 11101, etc.). Our control and strategy is based on the following arguments:

1. In case the state from infrared sensors is 00100, go straight at the full speed.
2. If you reach state 00110 turn slightly left to state 00100.
3. If you reach intersection (00111 „R“, 11100 „L“or 11111 „T“) the chosen strategy decides about the movement.
4. If the state is 00000 you are at the end of line. Turn around to 00100 and continue straight.
5. If you get into state 11111, you are at the target.

IV. Finding the shortest way to the target

To explain how to find the shortest path, we define some basic concepts that we will use. One is the intersection that represents the place in which the robot has more than one choice of direction. Fig.3 shows eight situations, four of which can be regarded as intersections. The first two situations can not be intersections, because the robot has no choice but to turn. In the last picture is shown the situation where the robot has to turn around because of finding the end of the road. The picture therefore shows only four intersections to be detected in the process of searching the maze. It remains to be clarified how the robot determines the type of intersection. For example, if the robot moves on the line the state of its infrared sensors will be 00100 and at the transition to 00111 we could conclude that it is a turn right or an intersection right and straight. It is necessary to make one more step, in which we confirm the given situation we are at. In the case we get sensors state 00100 at the step forward, it is a right-straight intersection, if the state 00000, it was a turn right. Similar situations occur also at the left-straight intersection and the "T" and "+" intersection. As soon as we get information on the type of situation where we are at, the strategy decides about the behavior in the given situation.

To represent the behavior of the robot at the given situation in its memory, we represent every situation by appropriate symbol. If the robot based on the strategy will go straight we store „S-straight“ into the memory, if it turns right „R-right“ and if it turns left „L-left“ and eventually when it turns around we will store "U-turn into the memory. If there is a situation where the robot performs on the basis of the strategy decision on next move, it stores its decision into memory as an appropriate symbol. Sequence of symbols creates information about the passed path in the memory. Once we know how the robot moves in the maze, we can remember the way that the robot passed. We can proceed to excluding the paths which do not lead to the target. In this case, the most interesting situation is when the robot must turn around (“U-turn”) due to the impossibility of further progress in its way. This also means that the previous decision was incorrect and has to be changed. So if we find right-straight intersection and we go straight based on the strategy left, (we store symbol "S") and then we have to turn around “U”, in the memory will be stored “SU”. When we return to the previous decision we make a change (we do not go straight, but in this case we turn to the left “L” because of the priority in the strategy to the left). By doing this we get a situation “SUL” in the memory, which means that at the intersection we should not continue straight, but we should turned right. The situation in the memory is to be resolved by replacing the combination of “SUL” by symbol “R” and we continue until we find the next intersection, at which we will make decision again and in case of bad decision we correct it again. The situation when the robot has no choice but to turn is not considered, as stated as an intersection. Therefore this was not written into memory. After hitting the end of path and turning around, the robot goes back to the place of decision (intersection) where the following situations may occur leading to a correction of the earlier decision. In the case of a combination „LUL“we obtain „S“, see Fig.5.



2 U
5 U
7 U
2
4
5
9
9
10-Finish
A
1 S
3 L
3
4 L
6 L
B R
8 L
6
R
C
L
7
D L
9 L
1
Start
A
B
B
C R
D
R 1 2 3 SUL = R 4 5 6 LUL = R S 7 8 SUL = R R 9 L = RRLL Finding path to the finish

Fig.5Applying the path search using strategy with priority to the left.

As shown in Fig.5, during maze searching it is also possible using simple rules find the way to the target. It must be noted that our maze is simplified and we are not dealing with finding the shortest path in any way. Our maze may not have loops and thus has only one path to the target to be found. It is therefore important to find such search mechanism, which allows searching of the entire maze and find its way to the target. This search method is easy to implement into the platform of 3pi robot and it can be practically tested in real conditions. This example is a part of the sample examples demonstrating the properties of 3pi robot.

In the group were students aged 12-15 years working in couples. The main idea of the group was to introduce ways of creating control algorithms designed for specific hardware to students. Student had to understand the controlled hardware for which the control algorithm had to be developed. The goal of this group was not to develop their own solutions, but to teach students to acquire and use the necessary knowledge to build custom robotic systems. The group meetings were held once a week and lasted for three hours during school-year. In this time students managed to acquire software and hardware skills in way they learned to test sample programs, modify them and add their own ideas. Based on experience in teaching this group could be said, that the given platform of mobile robot equipped with these software equipment is fully suitable for use in the education in robotics. It develops theoretical analytical thinking as well as practical aspects in the construction of a mobile robotic system. The system is fully proved in our group sessions and fulfilled our expectations. We strongly recommend it for the education in robotics.

Conclusion

This example illustrates how easily we are able to solve seemingly complex problems such as path finding in a maze. Students in the hobby group not only were able to understand and operationalize the maze search algorithm, but to modify and expand it with their knowledge. After mastering this algorithm they expressed interest in the new challenging tasks. 3pi robot platform allows extension of the universal printed circuit board, thus giving the possibility of extending the custom peripheral expansion. This creates the possibility of further extension of the hardware or sensor systems that enable solutions to a variety of other tasks. Some students began to think of their own design of robots that allows them to better solve their proposed tasks.

References

[1] *Pololu 3pi Robot User's Guide*. ©2001–2009 Pololu Corporation. Available at: < http://www.pololu.com/docs/0J21>.

[2] VANNOY R. T. *Line Maze Algorithm*. Available at: < http://www.pololu.com/file/0J195/ line-maze-algorithm.pdf>.