Exploration of Algorithms to Control a Maze Robot

N. Nikolov[1], S. Ivanov[2], S. Mustafa[3], M. Todorova[4]

***Abstract* - The present paper explores different techniques to control a robot designed on the basis of an embedded microcontroller. A number of robot control algorithms have been applied to the hardware. These robots are used for teaching purposes in the hardware division of "Computer Systems and Technologies". A mobile object would always heighten the students' interest and improve the quality of the teaching process.**

***Key words* - sensor, embedded microcontroller, robot, control system**

I. Introduction

The studies of subjects such as Embedded Microcontrollers, Design of Microprocessor Systems (MPS), System Analysis etc. being a part of the Bachelor's and Master's Degree in "Computer Systems and Technologies" require specific teaching facilities. The teaching process is provided by scale models but the objects to be studied are static. Technical universities similar to ours employ MPS to control mobile objects (robots). Their application in the teaching process enhances the degree of visualization and understanding and enables the students to learn the subject matter being taught faster and more effectively.

II. Description of Hardware

The task pursued by our team is to develop a MPS which should control a mobile platform by means of two reversible DC motors and their reduction gear. Depending on the input data, various tasks for robot control can be defined. On the basis of the information coming from distance sensors, reflective sensors etc., the computer program should control the robot movement and make it move in the following ways:

- on a line route (line following robot);
- in a maze (maze robot);
- in a maze trying to detect an area having the highest temperature etc.

MPS is so versatile that the tasks to be solved can be expanded. An important advantage is the presence of different variants for communication with other objects using standard wired and wireless interfaces.

The block diagram of the system is shown in fig.1:

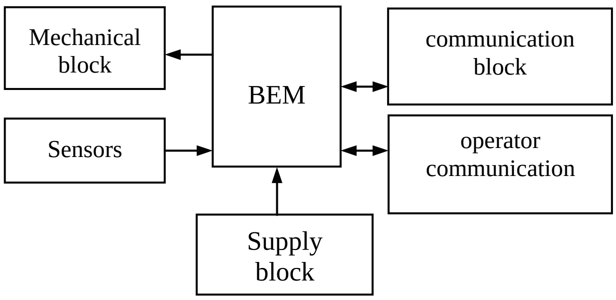


Fig.1. Block diagram

The system consists of the following blocks:

- mechanical block – it consists of a platform, two motors, two driving wheels. The mobile car has a platform moved by two wheels. For stability purposes there is a third support point which is a foot with a ball end. The axis of each wheel is driven by a reduction gear which is driven by a reversible DC motor. (fig.2)

Fig.2. Two DC motor with reduction gear

- sensor block – sensors of both analogue and digital (ON/OFF) type can be connected to the system. These sensors give information about the distance from the object to a wall, the colour (dark or light) of the surface on which it is moving, the ambient temperature etc. Fig.3 shows a distance sensor and a surface colour sensor (reflective type).

[1]Nedyalko N. Nikolov, Technical University – Varna, Computer Sciences Departament, 9010 Varna, Bulgaria, E-mail: ned.nikolov@tu-varna.bg.

[2]Sava I. Ivanov, Technical University – Varna, Computer Sciences Departament, 9010 Varna, Bulgaria, E-mail: ivanovsi@abv.bg.

[2]Suzan Ch. Mustafa, Technical University – Varna, Computer Sciences Departament, 9010 Varna, Bulgaria, E-mail: suzan.chetinova@abv.bg,

[4]Maya P. Todorova, Technical University – Varna, Computer Sciences Departament, 9010 Varna, Bulgaria, E-mail: mayasvilen@abv.bg

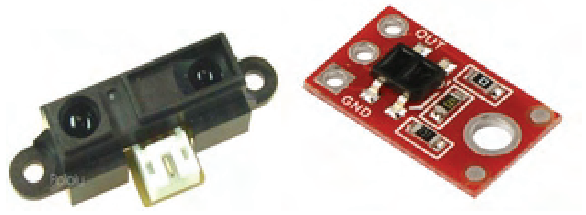


Fig 3. Distance and surface colour sensors

- block for operator communication – it consists of a LCD alphanumeric display to show parameters as well as system status and a matrix keyboard with a 3 by 4-button structure to input parameters and to start the system on different modes. The buttons are tested dynamically by activating one column through ports RB1 to 3 and the button status is read through inputs RB4 to 7. The change in the status of each port can cause “set interrupt flag“ and “interrupt request”.

The LCD display has a structure of 2 by 16 characters and is controlled by Port D. A half-byte mode of exchange has been selected in order to economize on ports of the embedded microprocessor. It is a matter of programming what alphanumerical information will be displayed.

- communication block– it enables the system to establish communication with other systems by one of the following standard interfaces: RS 232, IrDA, CANbus, LINbus and radio interface.

- block “embedded microcontroller” (BEM) – it serves to process the information coming from the sensors, to compute the control signals to the motors according to the application program loaded in the microcontroller, to monitor the status of the keyboard.

When deciding which one to choose, we had in mind the following considerations:

1. It should enable direct control of all blocks included in the system;
2. It should be of a series familiar to the students; in this way, familiar tools such as MPLAB and C18 can be employed to develop, load and adjust the application programs;
3. The modules built-in in the embedded microcontroller (timers, PWM, ADC, UART, SSP, CAN etc) can facilitate the development of application programs;
4. A PLCC should be installed in a PLCC socket which enables replacement of the embedded microcontroller in case of damage.

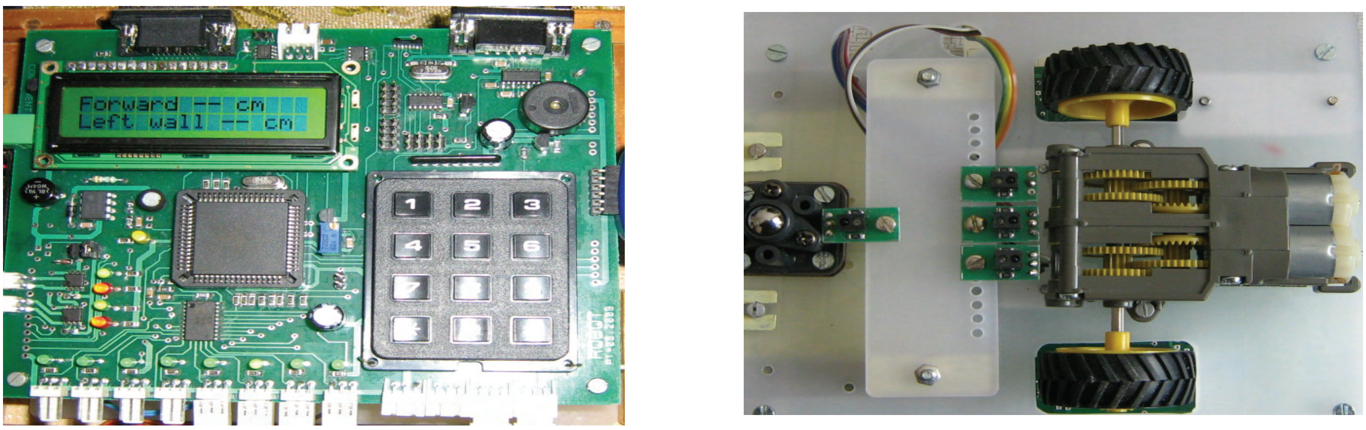


Fig. 4. General view of the robot system

- Motor control block

These two are absolutely one and the same designed and constructed by means of a specialized chip BA6287F i.e. driver for control of DC motor. The chip enables the motor to operate in one of the following modes: forward movement, reverse, stop, and stand by. By means of two pins the embedded microcontroller (EMC) controls the direction and the speed of the motor. The speed change is achieved by PWM signal generated by the PWM1 module or PWM2 of the embedded microcontroller. The duty cycle is programmable and it is proportional to the speed of the motor.

- Block for processing analogue sensor information

It processes information coming from a sensor of the GP2Y0A21KOF type by means of which distances from 10 to 80 cm can be measured. There are 4 analogue inputs (RA0,1,2,3) from which the information goes into the analogue multiplexer and 10-bit analogue digital converter (ADC) built-in in the embedded microcontroller. The sensor power supply is +5V. Other types of sensors can be connected as well provided that their output analogue signal changes from 0 to +5V. Also, there is SMT160 sensor in the system which measures the ambient temperature. Its output is a PWM signal whose duty cycle is a function of the temperature measured. This signal is transmitted to the input RA4 and the value of the duty cycle can be calculated with the help of a program.

- Block for processing digital sensor information

Optical reflective sensors of the RPR220 type can be used in the system and they register the light reflected from the surface at a distance of 8mm. With their help black and white straps can be registered. The condition of each input signal from the sensor is indicated by a LED, it is buffered and transmitted to the embedded microcontroller input. There are 8 inputs of this kind. Other sensors with a power supply of +5V and generating ON/OFF output signal can be connected to them as well.

These four blocks are sufficient in order that programs for robot control can be developed, loaded and adjusted. The remaining blocks serve to expand the possibilities of the system by providing a connection with the operator and other micro-processing systems by a standard interface.

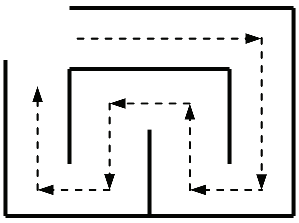
Programming and program testing can be accomplished with the help of a programmer i.e. debugger MPLAB IDC2 Lite.

III. ALGORITHMS FOR CONTROL OF A ROBOT MOVING IN A MAZE

The analogue and digital inputs in the system interface present a wide variety of tasks to be solved involving movement and control of the robot. The present paper offers a number of algorithms to solve the following task: using one or more analogue distance sensors, both an algorithm and a program to be developed in order to control a robot moving in a maze.

In order to perform the task two analogue sensors for distance are used – they are mounted on the mechanical part and the board with the embedded microcontroller. The distance range of the sensors used is [10, 60] cm; This means that their location on the construction should always be at a distance not less than 10 cm as compared to the distance to be measured. As a result of that the sensors are mounted approximately in the middle of the platform near to the axis connecting both motors.

Fig.5 shows a diagram of the maze and a photo of the robot in the maze:



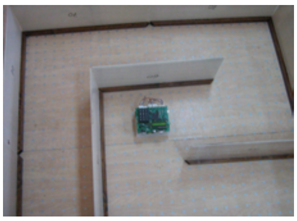


Fig. 5. A maze and an ideal trajectory

The walls have a width of 40 cm, the overall dimensions of the robot are 20 x 12 cm.

The algorithm for control of a robot in a maze having parallel and perpendicular walls is relatively simple. It consists of measuring the distance from the robot to an opposite wall. If this distance is bigger than 20 cm, the robot keeps moving forward. Otherwise, the distances to the right and the left walls are being measured (90 degrees on the left and 90 degrees on the right of the trajectory of the movement). A decision is taken to turn left or to turn right depending on the fact which distance is bigger.

The trials under real conditions have shown some disturbances. The most common of them are as follows:

- The distance sensors require a graphic representation of the equation $S = f(V)$, where S is the distance, and V is voltage [0,5] to be presented in a tabular form.. A testing program has been developed in this system in order to improve the degree of accuracy in the table containing the distance values as a function of the voltage. The testing program visualizes the sensor readings for the distance to a wall when the motor speed is zero. By means of a digital voltmeter, the voltage at the sensor outputs is measured. The distances are fixed with an increment of 5 cm i.e. (5, 10, 15, etc.) the values of the distances in the table and the fixed ones are compared. If necessary, the voltage values for the fixed distances are corrected.

- Although theoretically both motors and their reduction gear are equal, in real conditions they differ which results in the speed of each wheel which can also differ;
- The smoothness and friction coefficient of the surface on which each wheel is moving are not always one and the same;
- The 90-degree left and right turns and the U-turn of 180 degrees are not performed precisely;

These disturbances cause deviations from the ideal trajectory in the straight sections as well as in the turns This necessitates corrections to be made in the control algorithm in order to position the robot as close to the ideal trajectory as possible. When the robot does not deviate from the ideal trajectory, the above rules are sufficient for performing the task.

Before performing the real task i.e. a robot moving in a maze, tests should be made for movement in a straight line and parameter adjustment for taking the three types of turns (left, right and U-turn).

The tests for straight line movement require a smooth track to be selected. Different speed values are set for movement of the robot in forward and reverse direction. If there is no significant deviation (more than 2-3%) from the straight line over the distance of 200 cm, then no corrections are required. Otherwise, the speed of one of the motors should be increased or decreased. The tests under real conditions have shown that the speed settings for both wheels in forward and reverse direction are different. After completion of the tests for straight line movement, the parameters for turn taking should be adjusted. Each turn can be performed in one of the following ways (e.g. for a right turn):

- The speed of the right motor is zero, and the speed of the left motor remains the same in value and direction. In this case, the turn radius equals the distance between both wheels, and the right wheel is the circle centre of the turn;
- The speed of the left motor is zero, and the speed of the right one is reversed. In this case, the turn radius equals the distance between both wheels, and the left wheel is the circle centre of the turn;
- The speed of the right motor is reversed, and the speed

of the left one remains the same in value and direction. In this case the turn radius equals half the distance between both wheels, and the middle of the axis between both wheels is the circle centre of the turn.

A testing program has been developed for adjustment of the parameters when the robot takes 90-degree left or right turns. The program makes the robot move in a square i.e. a left or right turn is made 4 times. If the square is repeated several times (2 or 3) without any substantial deviations, then the parameters should be stored.

After making an analysis of the causes leading to deviations from the ideal trajectory, the following conclusion has been drawn: Some additions to the basic algorithm for movement in a maze should be made so that a left or right wall could be followed. If such a result could be reached, it is not of any importance whether the geometrical shape of the maze is rectangular or not.

Fig.6 shows the two variants for a deviation from the ideal trajectory:

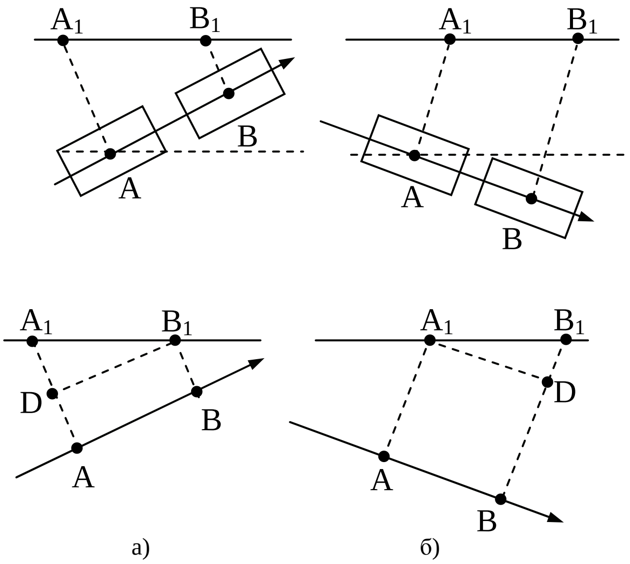


Fig 6. Types of deviations

Let us assume that Δ is the relative deviation resulting from two consecutive measurements of the distance to the left wall:

$$\Delta = \frac{A_1D}{AA_1} = \frac{AA_1 - BB_1}{AA_1} \quad \text{or} \quad \Delta = \frac{B_1D}{BB_1} = \frac{BB_1 - AA_1}{BB_1} \qquad (1)$$

When Δ exceeds the pre-set value (0.2-0.3%), commands “easy to left” and ‘easy to right” should be included in the algorithm. These commands can be executed by increasing or decreasing the speed of one of the two motors. For example, in order to correct the movement from fig. 6[a] and make it parallel to the wall, the speed of the right motor should be decreased for a short period of time, after that the original value of the speed should be restored. Likewise, these steps can be taken for fig. 6[б] but they relate to the speed of the left motor.

If the distance BB_1 becomes less than the pre-set minimum distance to the left wall, then a manoeuvre for clearing the wall should be made. Fig.7 shows a variant for making such a manoeuvre.

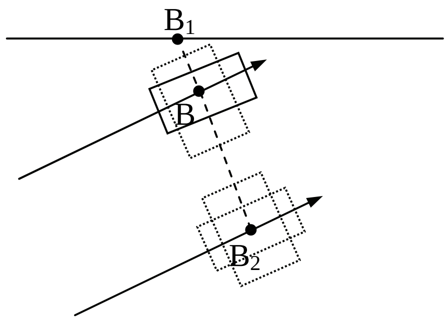


Fig.7. Manoeuvre for clearing the left wall

Clearing is executed by a 90-degree right turn, 5-cm movement in a straight line, a 90-degree left turn and then the original direction is resumed. Likewise, a manouevre for approaching the left wall is made, when the clearing distance exceeds the maximum pre-set value. In this case, the sequence of manoeuvres is as follows: a 90-degree left turn, movement forward and a 90-degree right turn.

These corrections in the algorithms result in achieving the near ideal trajectory and when the robot is heading for an opposite wall, the position of the robot is almost perpendicular.

IV. Conclusions

The manoeuvres “easy to left” and “easy to right” which have been executed as an addition to the basic algorithm have resulted in a considerable improvement of the quality of the robot control.

The robot is being used for performance of some other tasks as well. It has found an application in the laboratory practice of several subjects taught in the university.

References

[1]. R. Dofr, R. Bishop, Modern Control System, Peason Education, 2008.

[2]. Н. Кенаров *PIC микроконтролери* част 2- Варна 2006.

[3]. www.microchip.com – PIC18F6585 Data Sheet – 68 pin High performance 64 KB Enhanced Flash Microcontroller with ECAN module – 2004

[4]. www.pololu.com