ANNALS OF THE ORADEA UNIVERSITY
Fascicle of Management and Technological Engineering
ISSUE #1, MAY 2014, http://www.imtuoradea.ro/auo.fmte/

SOME ASPECTS OF DEVELOPMENT AND PROGRAMMING A WHEELS MOBILE ROBOT USING THE ARDUINO DEVELOPMENT BOARD

Ilie POPP[1], Dorin TELEA[2]
[1]Lucian Blaga University of Sibiu, Faculty of Engineering, ilie.popp@ulbsibiu.ro
[2]Lucian Blaga University of Sibiu, Faculty of Engineering, dorin.telea@ulbsibiu.ro

***Abstract*—**The paper presents aspects regarding the development of a mobile robot (on wheels) fitted with ATmega328P-PV microcontroller and infrared and ultrasound sensors for detection. The robot programming elements and the test of its characteristics and performance are also highlighted. The robot is able to pursue an autonomously route determined in advance and avoid any obstacles which might arise in its path, returning then to the default route.

***Keywords*—**wheels mobile robot, microcontroller and sensors, robot programming elements

I. INTRODUCTION

A mobile robot as a complex system can perform different activities in a variety of specific situations.

It operates in a real working environment and can also plan movements so as to realize \"the task\" depending on the initial state of the system and according to a priori existing information [1].

The power transmission mechanism has 2 independent DC motors each of them sending power to a wheel. Thus the input size influences the mobile robot direction and also the speed of the robot. However most of the commercial mobile robots have a regulator that controls the linear and angular speed of the robot [2].

The objective of the motion-controller is to follow a trajectory described by an analytic equation. This task is fulfilled by splitting the trajectory into motion segments of well defined, geometric shapes (lines, circles). Mobile robot control thus turns into a continuous trajectory based on the lines and arcs that would lead the robot from its original position - START in its final one [3].

In a mobile robot system spatial positions are very important; they are depending on the task, but also on the functioning of the entire system. In other words, the robot must be able to plan the movements, to decide what moves to execute to perform the task, depending on the currently arrangement of the objects in the workspace. Planning movements does not consist only in a single well-defined issue but in a number of problems, some of which are more or less important.

Robot positioning can be done in two basic methods: absolute and relative positioning or a combination of them. Relative positioning is usually based on dead reckoning (i.e., monitoring the wheel revolutions to compute the offset from a known starting position). Dead reckoning is simple, inexpensive, and easy to accomplish in real time. The disadvantage of dead reckoning is its unbounded accumulation of errors. Absolute positioning methods usually rely on navigation beacons, active or passive landmarks, map matching, or satellite based navigation signals [4].

Collision avoidance with fixed or movable obstacles in the work space of the robot can be done by several methods: making a mechanical flap which through the deformation stops the robot, using sensors that measure the distance to obstacles on the direction of the movement, the use of proximity sensors and the use of related information from several types of sensors, etc.

Locating objects can be achieved through physical contact but it imposes restrictions on the speed of movement of the robot; high speeds of displacement makes dynamic effects of a physical contact with obstacles or objects that are manipulated to be risky (can lead to deterioration). Also physical contact between the robot and the objects in the environment generates a reaction force that can change the status of the robot [5].

In conclusion, the mobile robot interacts with the environment through navigation (modifying the position and orientation of mechanical structure); so the design objectives and its construction must be in accordance with the environment in which it operates and with the tasks it will perform.

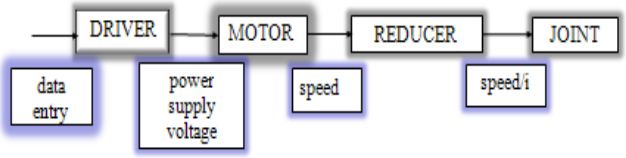

Fig. 1. Scheme of the robot system

Robot movement is possible without a determination of position and orientation relative to a fixed coordinate system, but this information is useful for motion control systems.

Some of the most often used navigation methods are: the measurement of the number of revolutions made by the driven wheels, the use of accelerators, electromagnetic structures installed on the field, semi passive or passive optical or magnetic signaling elements.

II. MATTERS CONCERNING THE MECHANICAL STRUCTURE (CONSTRUCTION) OF THE ROBOT

The mobile robot shows a structure with two wheels and an omnidirectional ball for support, which moves freely in a bearing camp, these three being arranged on a triangle.

The two wheels are driven by two DC electric motors with speed reducer to facilitate the movement of the robot. The two engines are controlled by a circuit with double H-bridge associated with a microcontroller.

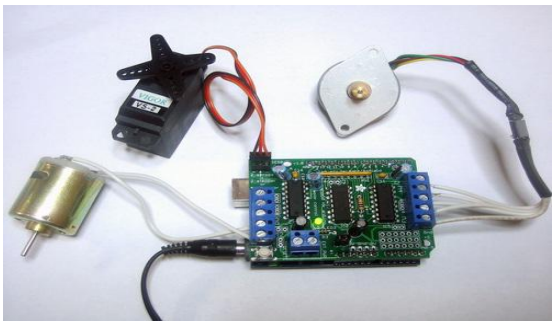

Fig. 2. Interface for DC motor control

The robot is equipped with a network composed of five infrared sensors, used in order to follow the set route and an ultrasonic sensor, which is used to detect any obstacles which might arise in its path and thus avoid them [6].

All parts are placed on two flat plates/rigid, located one above the other, to make the structure of the robot to be as robust.

The dimensions-from the constructive point of view are:

1) the maximum length-0.22 (m);

2) maximum beam width-0.13 (m);

3) the ground clearance between 3 and $5x10^{-3}$(m) (due to the feature of performance of infrared sensors);

4) the distance between the two wheels-0.11 (m) (in order to make possible the transfer of the robot in a space as small);

5) caster ball, similar with the kinematics and properties of a passive wheel (the third point of support of the robot platform).

On the two plates are placed:

a) four 1,5(V) batteries and one 9 (V);

b) five infrared trail sensors: three digital and the other two analog, placed at a distance of 0.015 (m) from each other and $4x10^{-3}$ (m) above the ground for a more accurate detection of the route to be followed;

c) microcontroller;

d) circuit board with double H deck, necessary for the DC motors.

e) ultrasonic sensor for motion detection and its servo-drive motor.

The supply voltage of the two engines is 6 (V).

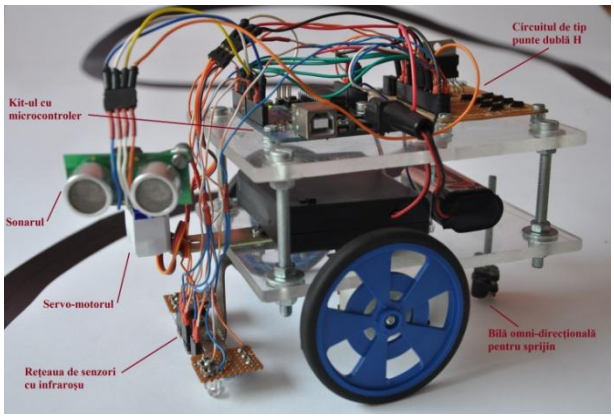

Fig. 3. Overview of the robot

In order to achieve control and command the following structure has been used:

i) a double H deck motor driver type L298N;

ii) 8 Zener diodes that support a maximum 2 (A);

iii) 2 capacitors $100x10^{-9}$ (F) capability;

Arduino Duemilanove kit which owns a number of facilities for communication with a computer, with another Arduino board or other microcontrollers. The ATmega328 microcontroller provides serial communication UART TTL.

The Arduino Software includes a tool for monitoring serial port which allows data transmission to and from the circuit board. The SoftwareSerial library allows a serial communication on each of the pins of the digital plate:

1) *theATmega328-microcontroller- 32 KB of memory used to store the code, 2 KB SRAM memory and 1 KB of EEPROM memory,*
2) *a connector used for programming the microcontroller through serial port (ICSP connector).*

III. ISUES RELATING TO THE ROBOT PROGRAMMING

A. Microcontroller interfacing

Each of the 14 digital pins of the plate can be used as both inputs and outputs using pinMode () functions, digitalRead () and digitalWrite ().

The Arduino Software includes a tool for monitoring serial port which allows data transmission to and from the circuit board.

The SoftwareSerial library allows serial communication on each of the digital pins of the plate [7].

B. Microcontroller programming

Arduino Duemilanove can be programmed using any software provided by the manufacturer. ATmega328

microcontroller contains a bootloader that allows loading a new code, without the need to use an external programmer; but you can avoid using bootloader by using the programming via serial port [8].

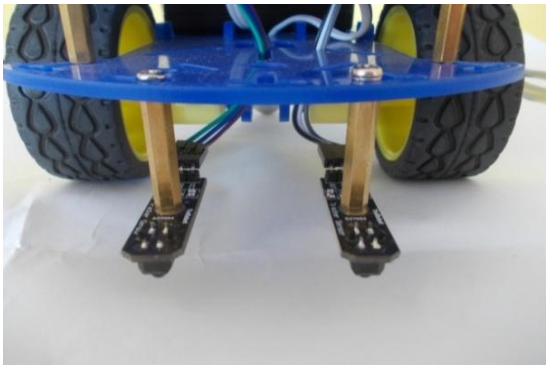

Fig. 4. The infrared sensor

Fig. 5. The ultrasonic sensor

DYP-ME007 ultrasonic sensor is used for the detection and avoidance of any obstacles which might arise in the path of the robot. It has a dual role: detection of obstacles and measurement of the distance to them. This sensor provides accurate distance measurements ranging from 30 (mm) up to 3000(m m).
Infrared sensor QTR-1RC is a very practical way to add tracking capabilities to a robot due to its very good response time.

C. The robot programming instructions

It was desirable to programm the robot to track a predefined route, to avoid any obstacles which might arise in its path and then to return to the original route.
Programming of the robot means in fact programming ATmega328 microcontroller (the software is made available by the manufacturer).

The programming language used by the ATmega328 microcontroller is C++ language. Loading programs into the memory of the microcontroller is done by USB connection kit with microcontroller. For directing the robot so that it can follow the imposed path, five infrared sensors will be placed next to each other, the direction followed by the robot being dictated by data taken from the network composed of five sensors.

The following algorithm of directing has been used:

-Reading data obtained from digital sensors is done using the function qtrrc.read(), which takes as a parameter the sensorValues vector, that represents the values submitted by each sensor depending on which detects, ranging between 0 and 9. Reading of analog sensors data is done using the analogRead() function, which takes as a parameter the assigned variable to the analog pin to which the sensor is connected.

To fulfil the function of detection and avoidance of obstacles that could intervene on the robot route, it will use an ultrasonicproximity sensor, attached to a DC motor, which will rotate the sensor on an arc of 180° so as to detect any obstacle which would be on his front, left or right side. Elapsed time between triggering and echo reception time is memorized by the function pulseIn (inPin, HIGH) where the inPin is the pin set as output and HIGH is maximum voltage threshold applied to that port.

IV. ASPECTS OF THE ROBOT TESTING (AND SOURCE CODE)

In order to test the robot a series of experiments have been made. For starters one has tested the behaviour of DC motors; set the maximum speed possible, a function was loaded on a microcontroller to allow movement before the robot. From observations made, it was concluded that there is a small difference of speed between the two engines (the same voltage is applied to both motors, the robot would not have been able to move in a straight line); so that different power supply voltages have been used on the two engines.

Then the program tracking the route on infrared sensors network consisting of three digital sensors was tested. After loading the program, which allows tracing the route, on microcontroller, the robot can follow the trail. However, in the case of close cornering, the initial route is no longer cleared. Measure amending the algorithm did not give a satisfactory result so it was decided to add two more analog infrared sensors to enhance the detection area.The engines were set to operate at 60% of the maximum speed. After the implementation of the necessary functions, an improvement of the robot behavior in the pursuit of the pathwas found. In order to increase the area of detection of the ultrasonic sensor it was mounted on an actuator. He did so, as the robot to work around the obstacle without touching it, even in a smaller space.

The last stage of the experiment was the interconnection of the two major functions of the robot, the tracing of the route and the detection and avoidance of obstacles. The more difficult part was turning the robot on the route after detecting and bypassing obstacles which appeared in his way. Through various modifications of the source code, a good result in terms of robot behaviour has been reached.

A. The ARDUINO Program Installing

The connection of the Arduino board to the used computer will be made by using the USB cable. The

operating system used by the computer will detect a new divice and will ask to install a driver.

B. The driver and IDE Installing

Regardless of the operating system that the computer uses, the software and the necessary drivers can be downloaded from the manufacturer's Web site. The necessary drivers are still in the driver subdirectory (Ex: c:\\arduino\\drivers or c:\\program files\\Arduino\\ drivers). As soon as the drivers will be installed one can start to develop the environment and write the first program.

C. Integrate Development Environment (IDE)

The main components of the environment development are source code editor and debugger.

ArduinoIDE does this: in the menu Tools-> Boards select the type of tile used, and from the Tools menu-select Ports > COM port used by the Board.

The application window will look like this:

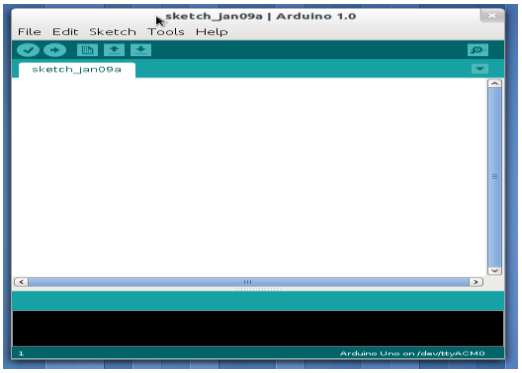

Fig. 6. The application window

This consists in three areas, namely:

1) The Menu Area:

 Check the schedule for program

 Upload errors in Development Board

 Creates a new project

 Opens a project

 Saves the current project

 Serial monitoring-this button is used to communicate with the computer through the serial port

Fig. 7. The menu area

2) The Program Area

This part of the application is used for writing and editing programs. With the help of their icon on the top right (in the form of an arrow oriented down) you can rename/create new files.

3) The compiler Area

Compiling is the process by which development environment converts the C code in a code in which you can run the Arduino. Writing code in the controller is made via the upload function and involves climbing the computer code compiled in flash microcontroller. Libraries for specific components are installed and they are compatible with Arduino.

In Adafruit's can be found over 100 libraries that support all products compatible with Arduino. It is important to install the libraries in the correct location. Otherwise, the compiler will not be able to locate them when trying to compile and load sketches. It will locate the Sketchebook folder and the Library folder, the sketches folder is the folder in which the Arduino IDE folder stored sketches. This folder is created automatically by the IDE during its installation.

Arduino programs can be divided into three main parts: structure (functions setup (), loop ()), values (variables - define values and constants - predefined variables) and functions.

V. CONCLUSION

The paper has presented a mobile robot (on wheels). Its mechanical structure, has been equipped with an ATmega328P-PV microcontroller ultrasound and infrared sensors for detection. The robot programming modes are presented as well as the testing of its features and performance.

The advantages include: ease of movement through the use of only two wheels and a fulcrum formed by a olldirections ball, so there is no need for many course correctionsas in the case of 4 or 6 wheels robots, the programming and loading of the source code on the microcontroller is extremely easy and with low costs.

In conclusion: this robot is able to autonomously pursue a route determined in advance and avoid any obstacles that might arise in its path and then return to the default route.

REFERENCES

[1] R. Arkin, Behavior Based Robotics, MIT Press, 1998 (Chapter 1)
[2] J. P. Norberto, "Industrial robots programming: Building applications for the factories of the future", Springer.com, 2001.
[3] B. Sciliano,K. P. Valavanis, "Control problem in robotics and automation", Springer.com. 2003.
[4] G. Ducdeck, M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press, 2000.
[5] D. Telea, A. N. Ceuşianu, Basics of robotics, *Bazele roboticii*, Ed.ULB Sibiu, 2012.
[6] A. M.Bitea, R. Gautier, "*Mobile robot using ultrasonic sensor to move with a radar display*."Robotica and Management, International Journal, Vol. 16, No.1, June 2011, pag.8.
[7] L. Bogdan, "*Using the microcontroller and the PLC in a RPP robot control*" Proceedings of 2010 International Conference on Optimization of the robots and manipulators, Calimanesti, Romania, 28-30 may, 2010.
[8] *** http://arduino.cc/en, accessed on March 2014.