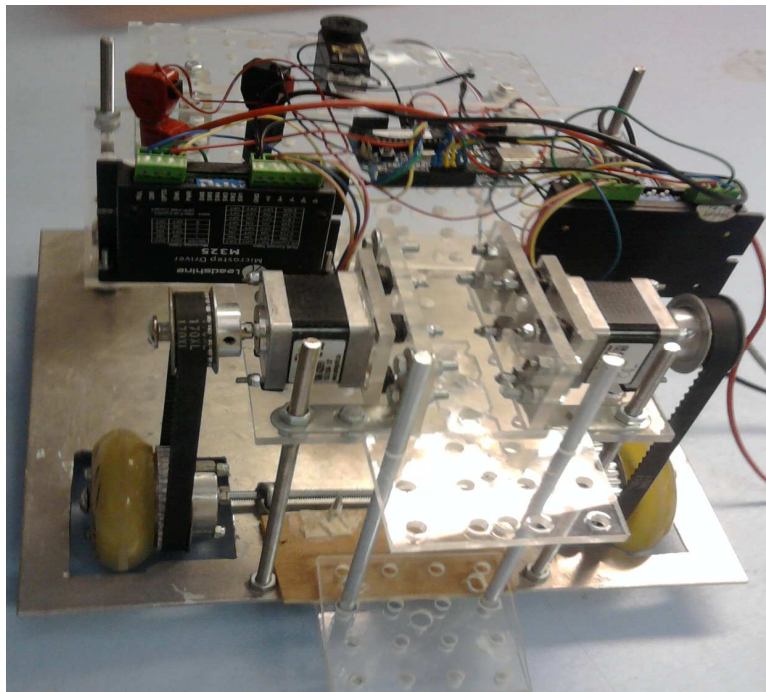


Project Report Robot



Academic Year : Fall 2009

Contents

Introduction:.....	3
I-Components of the Robot.....	4
A - Motor	4
B – Driver	5
C – Arduino card.....	6
D - Camera used for detection.....	7
E - . Connection.....	7
II - Working of the Robot	8
Working	8
Direction:	9
Enable:	9
Pulse:	10
Speed.....	10
The orientation of the robot with respect to the starting position:	11

Introduction:

This report shall explain the main functions and the working of the robot. Our aim was to build a robot which will participate in the robotics competition.

Our main objective this semester was to find methods which would:

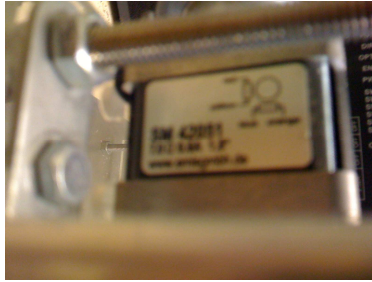
1. Enable the robot to maintain a constant speed
2. Define the coordinates of the robot so that it doesn't deviate from its intended path.
3. Enable the robot Identify the objects

By the end of the semester, we were able to achieve the first two objectives. The details are explained in the following pages.

We have the camera which would enable the robot to identify the various objects but are working on the program to integrate it in the robot and modifying it according to our needs.

I. Components of the Robot

A - Motor



The M4 42 051 bipolar motor with permanent magnetism is used.

The M4 42 051 motor works with at least two spools. It functions in a step by step manner, i.e. with each pulse, it rotates by an angle of 1.8 degrees. We can use a maximum of 4 different steps.

Step 1 : A rotation of 1.8 degrees

Step 2 : Dividing the rotation by 2 thus obtaining a rotation of 0.9 degrees

Step 3 : Dividing the rotation by 4 thus obtaining a rotation of 0.45 degrees

Step 4 : Dividing the rotation by 8 thus obtaining a rotation of 0.225 degrees

thus with each reduction in the degree of rotation, we increase the precision but reduce the speed by the times the degree of rotation was reduced.

The motor needs to a current of 0,6 ampere and 7 ohm resistance. Thus consuming less energy thereby reducing the costs.

B – Driver



A M325 micro step driver has high performance and low cost. It has an area of 47.3 cm² and a size of 2.05 cm. It supplies voltage in a range from 12 to 24 volt, and it's possible to adjust the current from 0.39A to 2.5 A with the help of three current control buttons. The pulse frequency can go up to 100 Khz and we dispose of 4 selectable micro step resolutions.

We have realized an experience aiming to determinate what step and which current we would use. The results of the manipulation show that the step 2 and the 4msec timer is most stable with a current of **0,6 amperer**.

C – Arduino card



As the Arduino card is easily programmed it is the most viable option.

It has an area of 38,5 cm² a of thickness of 1.5cm and is equipped with a micro controller Atmel AVR of 8bits.

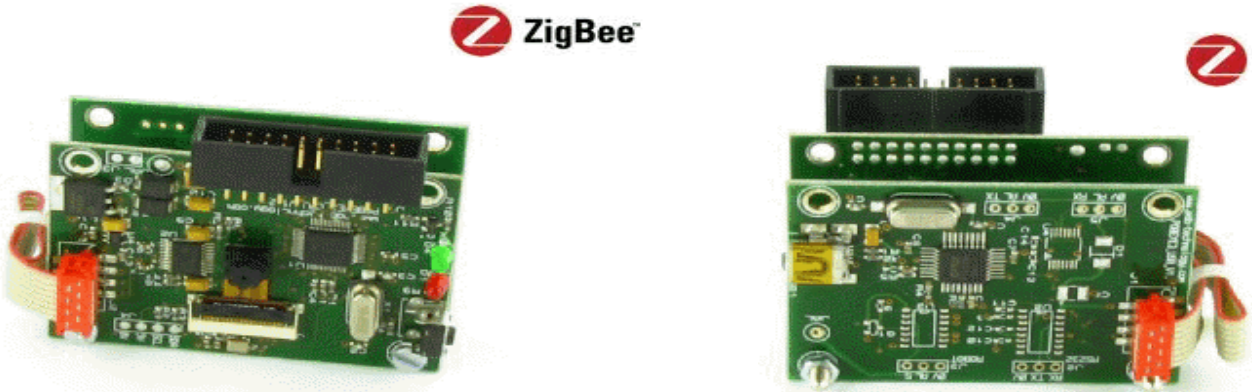
A micro controller allows the card to carry out different instructions and also has an internal memory. It's an integrated circuit which gathers the essential elements of a computer. It is connected to the computer with the help of a USB connection and needs an external power source.

The various software characteristics of the card are:

1. A Code Editor
2. A Compile function
3. A debug and run file function.
4. A simplified language and good bibliographies which improve the capacity of the micro controller.

D - Camera used for detection

Pob Eye II



- USB port
- Serial port
- Zigbee port (option)
- Micro camera 160x120 of color revolution

E - . Connection

Le **POB-Eye II** can be used to work with Xbee, Seial connection and Bluetooth .



II. Working of the Robot

A. Working

The main parts of the robot are the two servo motors and the Arduino circuit board. These three component help us to govern the various functions such as the direction, speed and the positioning of the robot . The connections are as shown below.

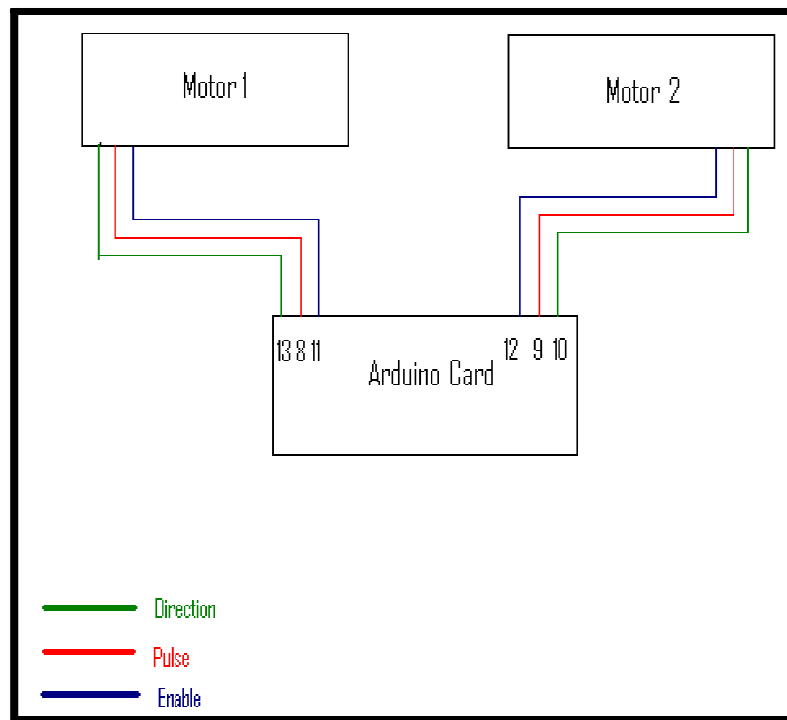


Figure 1 : Servo Motor - Arduino CARD CONNECTION

For Motor 1, the direction, the pulse and the Enable cables are connected to the pins 13, 8 and 11 and for the Motor 2 the direction, the pulse and the Enable cables are connected to pins 12, 9 and 10 of the Arduino card respectively.

Each of the pins has to be set on the output mode and switched to the High output setting.

B. Direction:

The function 'direction _robot' is responsible to control the moving and direction of the robot.

If the direction pin of the motor is configured to the High mode, the motor moves in the forward direction and conversely if it is configured to the low output mode it moves in the backward direction.

Thus:

1. If both the motors are set to function in the High mode, the robot moves forward.
2. If both the motors are set to function in the low mode, the robot moves backwards.
3. If the right motor is set to High and the left to low mode, the robot turns left.
4. If the left motor is set to high and the right to low mode, the motor turns right.

Right Motor Left Motor	High	Low
High	Forward	Right
Low	Left	Backward

The configuration of the left and right motors can be controlled by the variable 'Consigne.Left' and 'Consigne.Right' respectively.

C. Enable:

The enable port acts as an electronic switch. When it is configured to the High output mode the robot is on and when it is configured to the low output mode, the robot is off. It is thus always set to the high output configuration during normal circumstances and is set to the low output configuration when the robot needs to be stopped by urgency.

D. Pulse:

The pulses are in the forms of square waves. With each pulse the motor makes a rotation of 1.8 mille radians (with step 1). Thus by controlling the time between two consecutive pulses, the speed of the robot can be controlled. The lesser the time is the faster the robot is and vice versa.

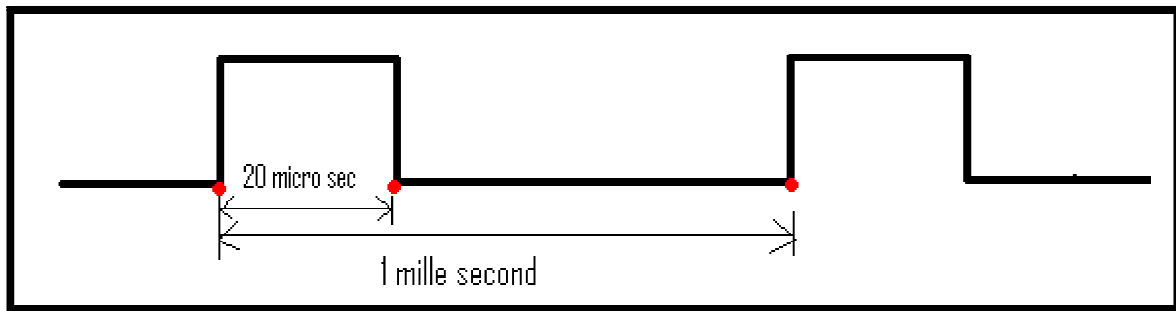


Figure 2 : Pulse time

There has to be a minimum time difference of 20 micro seconds between the initiations of two consecutive pulses as that is the time period of one pulse.

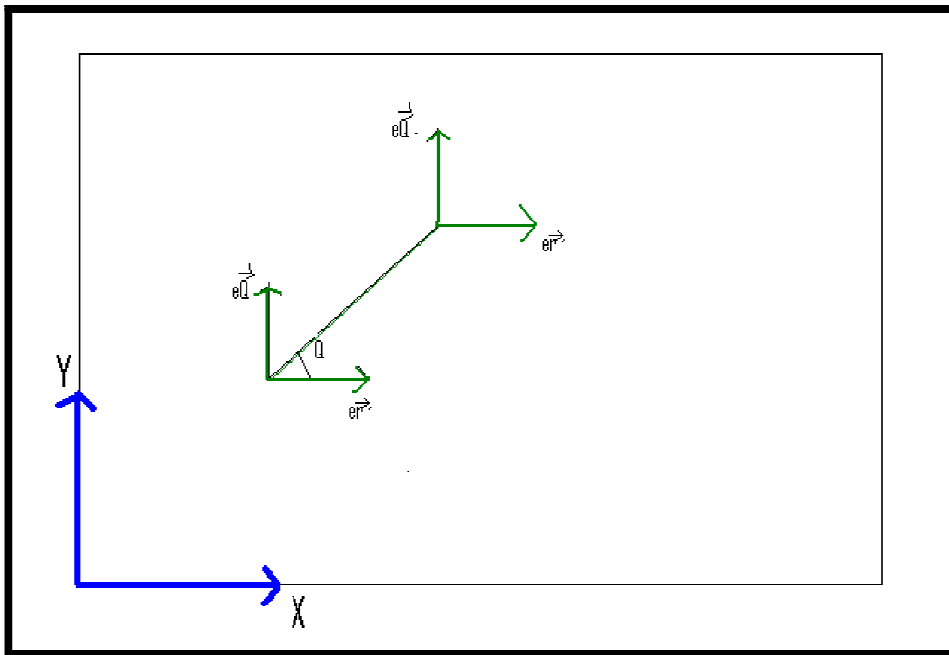
Speed

As stated above the speed of the robot can be controlled by controlling the time difference between two consecutive pulses.

In order to do so, a program named feedback is used which acts as a time counter. This program is repeated after every 1 mille seconds. This program contains a sub-program called the creneau pulse which is responsible to initiate the pulse. The time difference between each creneau pulse function can be modified by changing the value of the variable 'consigne.seuil' thus varying the speed of the robot.

III. The orientation of the robot with respect to the starting position:

With the help of experiments we know that with each pulse the robot moves 170 micro meters in the forward or the backward direction and in the case of turning, it rotates by an angle of 1.5 radians. In order to determine the position of the robot with respect to the origin, two frames of references are used. One of them is a fixed reference (X-Y) and the other is a moving frame of reference ($e_r - e_\theta$).

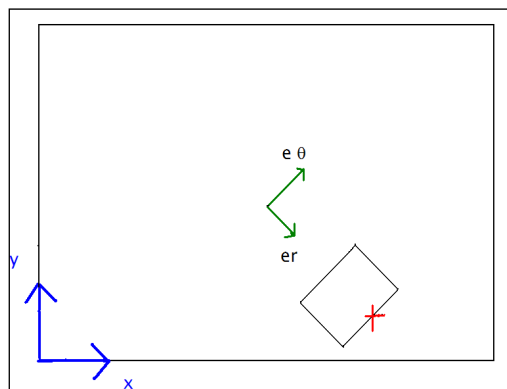
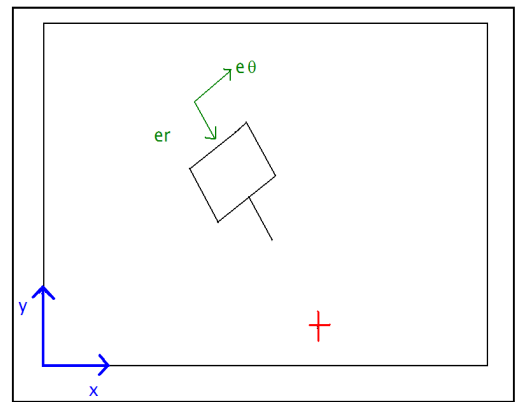
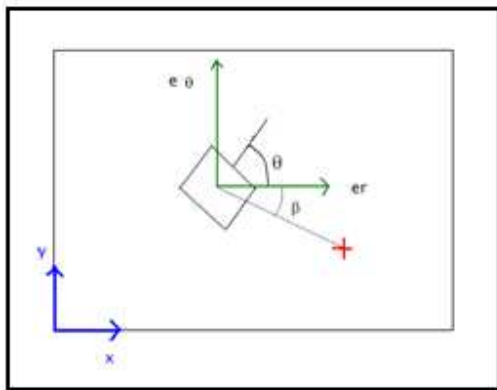


The frame 'er-e θ ' move along with the robot. The following programs are used to determine the position of the robot wrt the two frames of references:

1. Robo.R – gives the position of the robot from the last turn it made with respect to the 'er-e θ ' frame of reference.
2. Robo.teta – gives the angle between the axis of the robot and the er axis.
3. Robo.X – gives the position of the 'er-e θ ' frame of reference with respect to the fixed X axis.
4. Robo.Y – gives the position of the 'er-e θ ' frame of reference with respect to the fixed Y axis.

Finally as we now know how to define the coordinates of the robot, we can program it to go from one point to another and move back on its intended path in case of any deviations.

If the robot is located at a position X_1, Y_1 making at an angle $+\theta$ from the horizontal and it has to go to a position X_2, Y_2 making an angle $-\beta$ from the horizontal from its current position, the program will accordingly make the robot rotate by an angle ' $\theta+\beta$ ' in the clockwise direction and then will make it proceed to its destination X_2, Y_2 .



Conclusion

This report explains how the robot is able to move from one point to another.

With the help of the processing software, the robot can move to any desired location by clicking on the green square, which represents the testing table using for our experience.

Apart from the above-mentioned program, we have done another program using which we can control the robot with the arrow keys. The acceleration is controlled by the key 'Z' , the deceleration with the key 'A' and 'S' to stop. Both program and video demonstration are available in our website (www.pobot.org).

The next step will be to study the working of the camera in order for the robot to recognize objects and thus automatically give the coordinates of the desired destination.