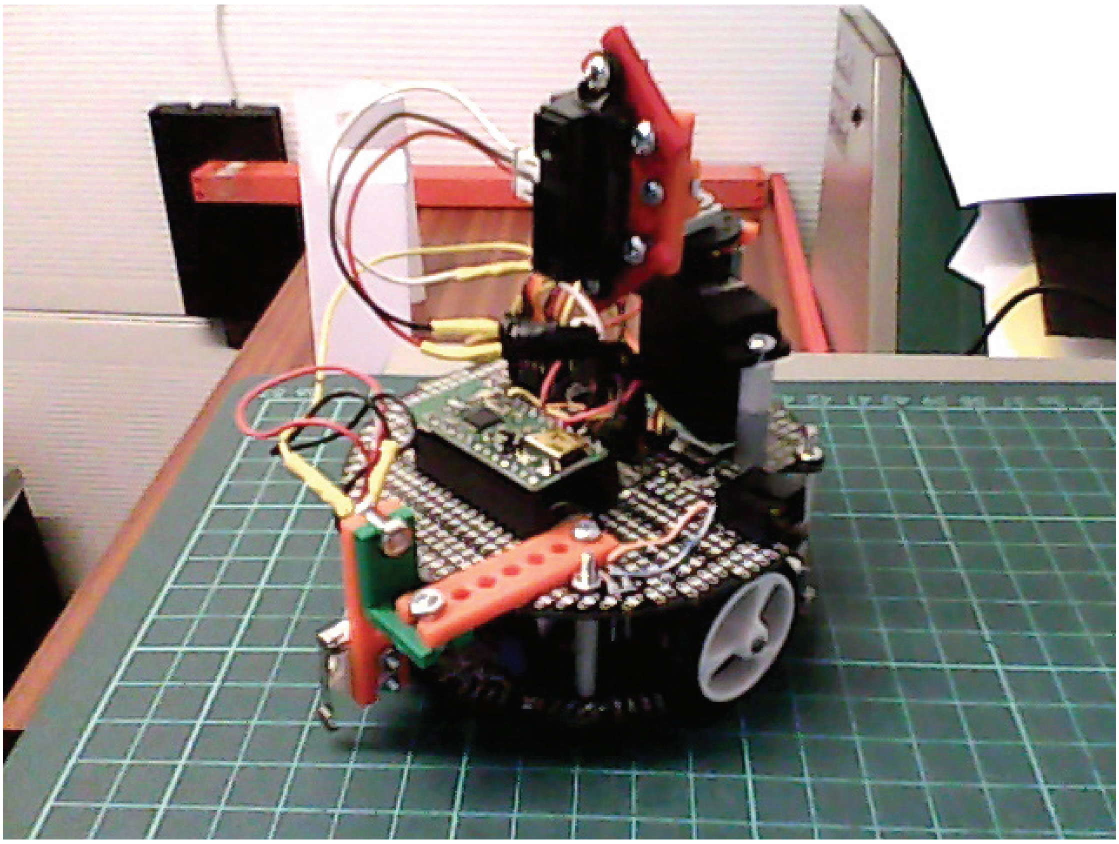
Remote Control of Pololu 3pi using Robot Basic



Summary

The idea was to be able to guide a Pololu 3pi through a series of obstacles using remote control.

I thought it might be fun if you were unable to actually see the course directly.

The idea I came up with was a series of graphical controls to direct the 3pi and a sort of radar screen to see the obstacles.

I decided to use Robot Basic for the terminal screen as it is quite easy to use and is freely available from the Robot Basic web site.

I also used the Arduino IDE to program the 3pi.

The 3pi was fitted with a servo, a Sharp distance sensor to scan the area ahead and a microswitch at the front to act as a bumper if it runs in to something.

A pair of Wixels, one on the 3pi and the other connected to the PC were used to communicate with the robot.

Hardware

This is what I used:

1 x Pololu 3pi

1 x 3pi expansion kit with cutout (because this is what I had - without cutout may be better)

2 x Wixels (these are Programmable USB Wireless Modules available from Pololu)

1 x Sharp Infra Red Distance Sensor (I used the20cm to 80cms version)

1 x Servo

1 x Microswitch

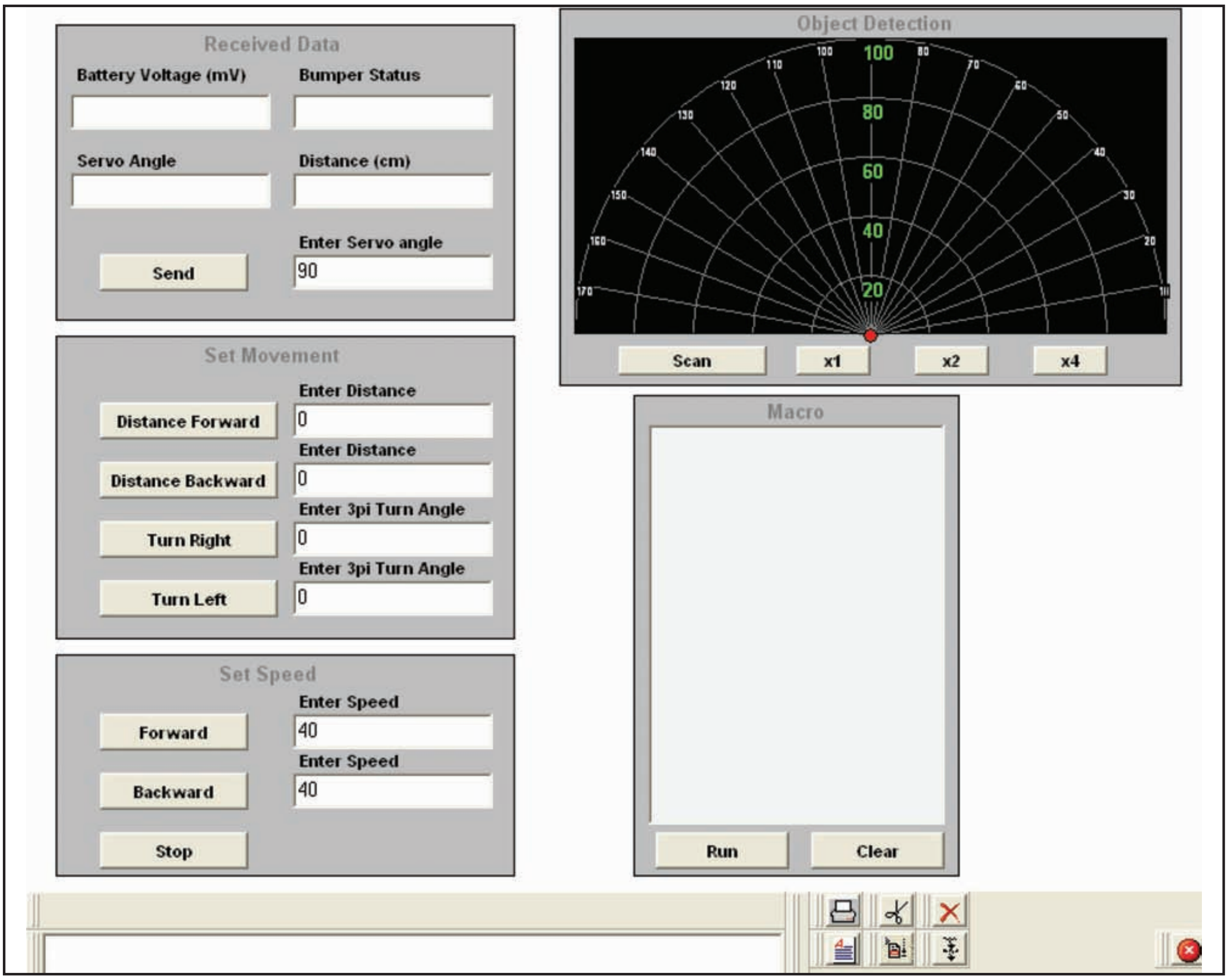
Various sockets and terminal pins.

Terminal Software

I used Robot Basic to create a terminal to control the Robot .

An explanation of each part of the screen is below.

Start Screen



Received Data

Battery Voltage (mV):

Bumper Status:

Servo Angle:

Distance (cm):

Enter Servo angle: 90

Set Movement

Enter Distance: 0

Enter Distance: 0

Enter 3pi Turn Angle: 0

Enter 3pi Turn Angle: 0

Set Speed

Enter Speed: 40

Enter Speed: 40

Received Data

Enter an angle to set the servo. 30^{o} is maximum clockwise, 170^{o} is maximum anti-clockwise. Default is 90^{o} which means the sensor is pointing straight ahead. Press send.

Returns: Battery Voltage (box turns red if < 4.5v)
Bumper Status - 0 if no contact, 1 if touching something.
Distance in cms - box turns red if distance is > 80cms (outside Sharp range)
Last servo angle entered.

Set Movement

Enter distance to move in cms. 3pi will stop after set distance.
Enter angle. 3pi will turn by an amount equal to the angle entered.

Macro

Any data entered in the set movement area will be remembered in the macro box, each command on a new line. The sequence can be added to or amended directly in the box. Pressing run will cause the 3pi to obey the sequence. This allows you to use the scanned data to try to run a route manually through the obstacles. This can then be repeated by placing the 3pi back at the start and pressing the run button. The sequence consists of the command letter followed by a space followed by the data. The line ends with a forward slash.

The commands are F x/ - forward by x cms
B x/ - backward by x cms
L x/ - Turn left by x degrees
R x/ - Turn right by x degrees

The commands can be entered manually from scratch and the resulting sequence run.

Set Speed.

The 3pi will move in the direction selected at the speed selected. Note that it will carry on going until you press the stop button so beware of table edges etc.

Object Detection

Pressing the Scan button will cause the servo to move from 30^{o} to 170^{o} taking readings every 2 degrees.

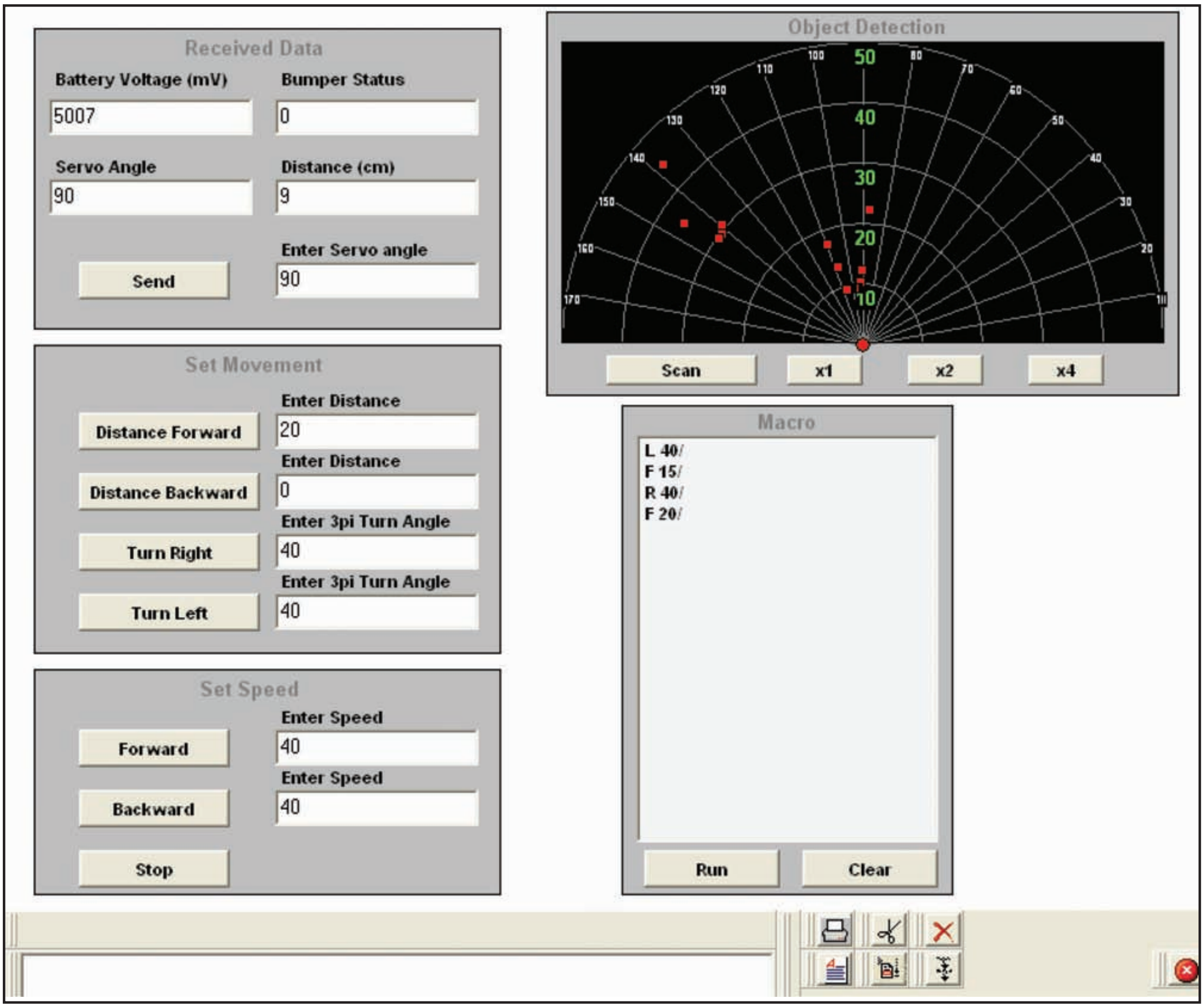
The result is plotted as red squares on the grid.

Pressing the zoom button will cause the display to zoom in or back out again to allow better resolution of the objects in the way. Pressing any zoom button before a scan has taken place will cause the scan to start.

This can be used in conjunction with the Set Movement area to guide the 3pi through a series of obstacles. You can switch backwards and forwards between zoom levels.

Screen Shots

Typical data



Received Data

Battery Voltage (mV): 5007

Bumper Status: 0

Servo Angle: 90

Distance (cm): 9

Enter Servo angle: 90

Set Movement

Enter Distance: 20

Enter Distance: 0

Enter 3pi Turn Angle: 40

Enter 3pi Turn Angle: 40

Set Speed

Enter Speed: 40

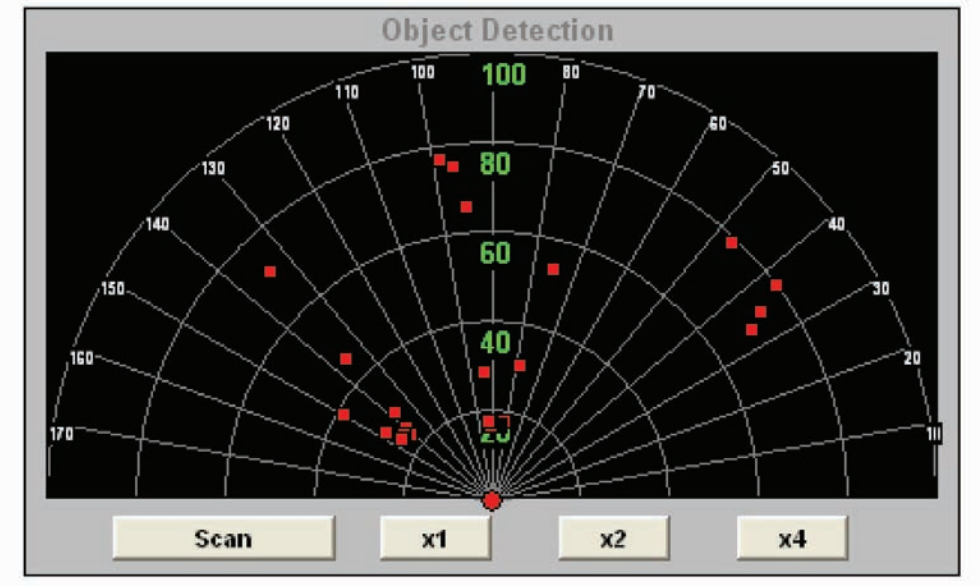
Enter Speed: 40

Macro

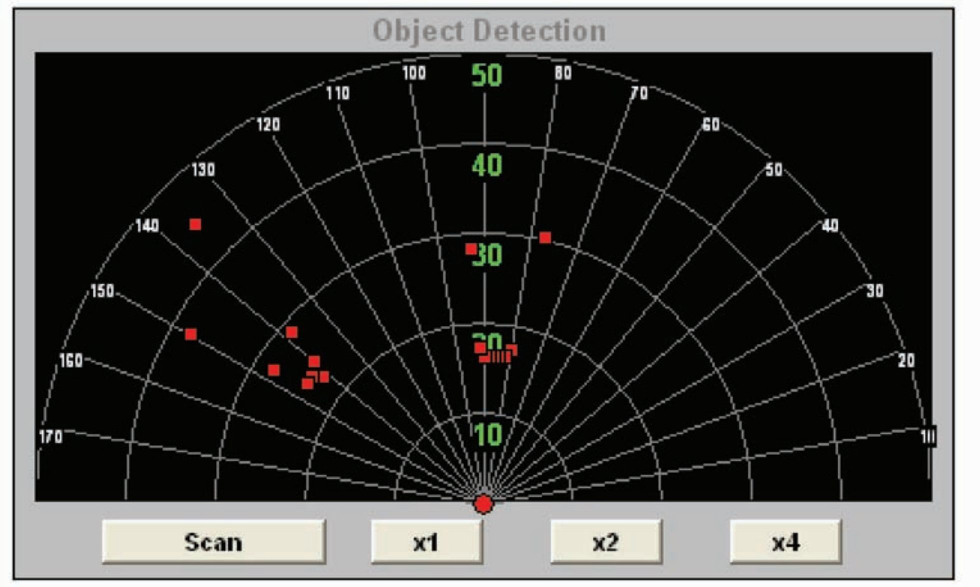
L 40/
F 15/
R 40/
F 20/

Typical data following some inputs.

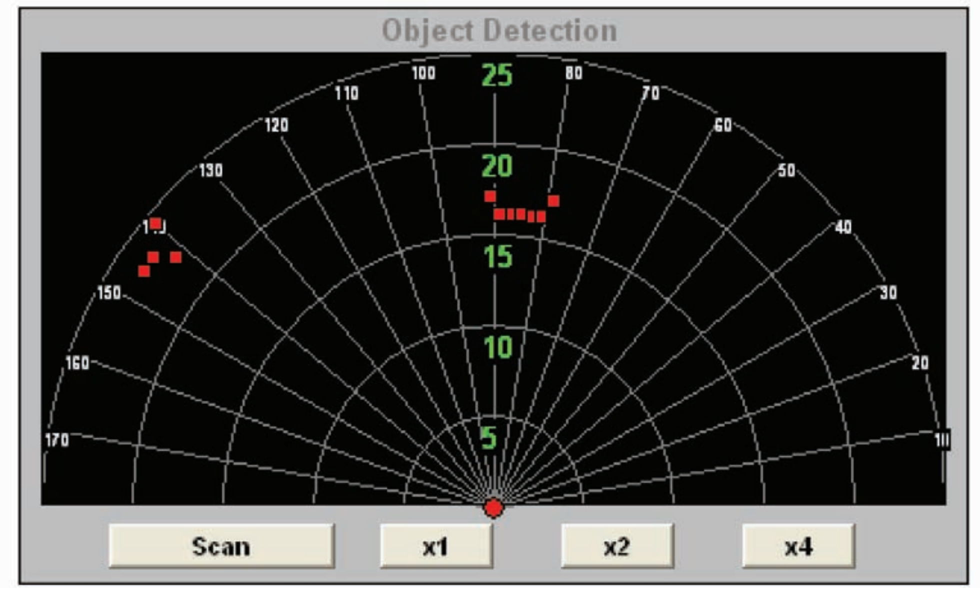
Screenshots of scan with 2 Objects within 40cms of robot.



x1 - Shows all objects within 80cms, looks a bit cluttered



x2 zoom - can make out 2 clusters of scanned data.



x4 zoom - 2 objects can be seen. 1 dead ahead and the other ~ 50^{o} to the left.

The robot can be turned either 25^o to the left or right and moved forward 25cms. Another scan will show what lies ahead.

This can be repeated until the destination is reached. The macro will have recorded these movements up to a maximum of 18 commands.

The robot can be put back to the start and should be able to repeat the run using the macro box..

Any wrong movements can be deleted as the macro box is fully editable.

If the robot runs into anything, this will register on the microswitch causing the robot to stop and back off a little.