CS 101 - PROJECT SPRING 2015 USER INSTRUCTION MANUAL

OPTIMISED BALL COLLECTOR

GROUP - 418

Team Members

Sudeep Salgia, **14D070011**

Aviral Kumar, **140070031**

Sohum Dhar, **140070001**

*Muthamizh Selvan,***140110091**

ACKNOWLEDGEMENTS

We would like to sincerely thank our professor Prof. Kavi Arya for giving us a golden opportunity to work on an embedded systems project with the e-Yantra Project, thus giving us the taste of embedded systems programming. We would also like to thank our TA, Chinmay Kulkarni for all the assistance given to us from choosing an apt project to helping us code and debug our project. We would also like to take this opportunity to thank the entire team of e-Yantra for their guidance and assistance at all stages of the project.

TABLE OF CONTENTS

PROBLEM STATEMENT

It is often necessary to go to a set of certain points in a network and minimization of effort/energy is always a matter of concern. Thus we aim to address this requirement in a prototype scenario.

Thus, using a bot, we intend to collect balls spread over an area ensuring that the distance that the bot travels is minimized.

In order to achieve this, a top view of the arena is provided to the code which connects with the bot and sends the path to the bot which it has to traverse in order to achieve the total path length minimization.

The user manual is in correspondence with the video tutorial of Optimised Ball Collector. The video tutorial refers to this manual. The video link of the working project on youtube is: https://youtu.be/on3euKLPWc0.

Instructions to setup the device:

1. Please ensure that the CodeBlocks IDE integrated with OpenCV 2.4.10 or a later version and GTK+ 2.24 or a later version is installed on your device.
2. For the installation of OpenCV please follow the instructions provided in the link below.
    http://kevinhughes.ca/tutorials/opencv-install-on-windows-with-codeblocks-and-mingw/

3. For the installation of GTK+ 2.24 please follow the instructions provided in the links below.
    https://www.youtube.com/watch?v=sD9jdmtG1gM

    http://ftp.gnome.org/pub/gnome/binaries/win32/gtk+/2.24/gtk+-bundle_2.24.10-20120208_win32.zip

4. Please move the header files – "shortestpath.h" and "xbee.h" into the folder containing the header files of the OpenCV which is given by C:\······\opencv\build\include.Here opencv is the folder where our libraries have been installed. Preferentially install OpenCV library in C Drive.

5. Configure the transmitter and the receiver x-bee as per the manual provided by the ERTS Lab.

6. Connect the x-bee to your device and configure it as per this manual (See Setting up COM Port).

7. Position the bot as per the given specification in this manual (See Placing the Robot).

8. Then take the image as per guidelines mentioned in this manual and save the image in your device (See Guidelines for taking the image). Copy the path of the file.

9. Open the project file and build and run the program " main.cpp ".

Steps to be followed after the program is run are:

1. A window pops up which the project name and team details are written. It has two buttons to choose whether to proceed or to exit. Choose Proceed.
2. Another window pops-up, with two entry boxes for the file path and the dimension of the arena.
3. Please enter the complete file path of the image of the arena(like C:\.......Documents\opencv\image.jpg) in its corresponding box which was copied earlier.Press the enter key .
4. In the other box please enter the actual vertical length of the arena (in centimeters) corresponding to the top of the image of which the image has been provided. Press the enter key and then finally press proceed.
5. The image is displayed in a window called named "ipc";
6. While on the window "ipc", press the enter key.
7. The image is processed and the coordinates of the centers of the ball in real time are displayed on the console with the top left corner as origin.
8. The path is figured out and the data corresponding to the ideal path is sent to the bot using x-bee.
9. The encoded data which is transmitted is displayed on the console as the bot moves.

Setting the COM Port:

1. Attach the XBee module using the adapter to the computer.
2. Open Control Panel.
3. Open Hardware and Sound.
4. Under Devices and Printers open Device Manager.
5. Now expand Ports (COM and LPT).
6. Identify the COM port corresponding to the XBee and open its properties (Right-Click).
7. Click on the Port Settings tab.
8. Click on the Advanced button. A second dialog appears.
9. On this new dialog, select the "COM Port Number" option near the top. Change it to **COM '3'** and click on OK to close the dialog. You might get a warning message about other programs using the device and whether you wish to continue. Click on Yes.
10. Click on OK again to close the first dialog.
11. Close the Device Manager. Note that the Device Manager might still show the old COM port number. If you close it and re-open it, you will then be able to see the new COM port number listed.
12. Now the COM port has been set.

Guidelines for taking the image:

1. The image has to be a top view of the arena. In cases where an exact top view image is difficult to obtain the image should be taken from as high as possible. Also it should be ensured that the position of the camera must be as close as possible to what it would have been in a top view.

2. The origin of the image is the top left corner of the image.So make sure that the top left corner of the arena touches the top left corner of the image. The origin is defined where the bot is initially kept.

3. It should be ensured that the image is taken in sufficient light such that all the balls have sufficient light falling on them.

4. It must be ensured that the balls at a certain height so that they can be detected by the sharp sensor.

5. Also the balls must be separated by about at least 30 cm (in the arena) and must not be near the corners of the image.

Placing the Robot:

1. The initial position of the robot has to be in a specific orientation. It is kept at the origin pointing along the y-axis.The direction of the bot is along the positive y-axis, which here is opposite to the standard direction in a 2D Cartesian coordinate system.
2. Also it needs to be ensured that the center of the shaft of the robot has to be kept at the point of intersection of the coordinate axes.

Possible bugs and their solutions:

1. Insufficient or varying intensities of light might change the colours of the object in the image in such a way that the present thresholding doesn't give the required output. In such cases, it should be ensured that the arena is well lit.

2. The error reduction is primarily implemented by a sharp sensor. It should be taken care of that the arena should be free of any obstacles. The reading from the sensor is fluctuating and often takes time to settle especially if there is a movement nearby. As a result, the reading from the sensor can be a wrong value and the bot might thus just get deviated from its path. There is no permanent solution to the problem as such but such incidents can be reduced considerably by ensuring there is no movement that the sensor might possibly detect and if possible there are no obstacles within about a meter so from where the balls have been kept.

3. There might be random errors in the hardware or the wireless communication. Such random events cannot be predicted and thus finding a long-term, permanent solution is difficult. Such problems are often eliminated in the next run.

4. The threshold values are adjusted for ping-pong balls. Preferably they should be used placed upon a small stand so that the sharp sensor can detect them. If these conditions are not met with the bot might not work properly. Please ensure that these preconditions are met with.