* + <http://www.camstudio.org/CamStudio20.exe>

README

CS101 – 2015

Course Project

Project : Voice Controlled Bot

The objective of this document is to help someone else run the code that is delivered as part of this project.

**Project Title:** Voice Controlled Bot

**Students:**

|  |  |  |
| --- | --- | --- |
| Name | Roll No. | Email |
| Siddharth Keshan | 140040051 | [siddharthkeshan@gmail.com](mailto:siddharthkeshan@gmail.com) |
| Mohit Patel | 140040010 | [tompatel17501@gmail.com](mailto:ravi@cse.iitb.ac.in) |
| Vineet Moghe | 14D070020 | vineetmgh@gmail.com |
| Mayank Raj | 140040050 | [mnkrj500@gmail.com](mailto:mnkrj500@gmail.com) |

**Project Description**

This project encapsulates the work done by the team for the embedded system project of the second semester CS101 course academic year 2014-2015. The project is a voice controlled bot in which the command is directly sent through serial port using MATLAB. Sending command through the keyboard wirelessly to the bot can be done both through MATLAB as well as through X-CTU.

**Project Objective**

This project aims at developing a voice controlled environment for Firebird V so that it can perform various tasks by simple verbal input . As we all know many disabled people face a lot of difficulty in day to day movements. Hence this idea has wide applicability and scope for solving a real life problem by extending to make voice controlled chairs.

Users of this product will use the interface to initialize the system and monitor it.

The achievements are as follows:

* Voice processing based on verbal input by the user using MATLAB.
* Automated detection of obstacles by the robot using sharp sensor and give beep alert.
* Real Time data transfer from user to bot after processing through ZIGBEE.

**Hardware Platform**

1. ZigBee wireless communication interface on both robots and the server.
2. Headphones attached to laptop.
3. Sensor mechanism on the robot to detect obstacles.

**Software**

1. MATLAB; for voice processing
2. ATMEL STUDIO 6 :for programming the robot.
3. AVR BOOTLOADER : to burn the program on bot.
4. X-CTU : to establish ZIGBEE connection

**Code Description**

|  |  |  |
| --- | --- | --- |
| Filename | **Purpose** | **Executes on** |
| wireless.c | Main Program | Robot |
| filter.m | Matlab file to filter the voice command | Centralised Server (PC) |
| Voiceinput.m | Matlab file to input ,process  , compare and send command to port | Centralised Server (PC) |
| parameter.m | To send a argument to bot. | Centralized server(PC) |

**Deliverables**

|  |  |  |
| --- | --- | --- |
| Folder | **Contains** |  |
| c code | Source code of programs to be burnt on Robot. |  |
| matlab code | Contains Matlab files. |  |
| documents | Contains Project related documents. |  |

**Execution Instructions**

The following steps must be performed in order to run the code provided.

1. **Setting up the hardware on the robot :** The project requires certain specialized hardware. The Obstacle Sensor. The Obstacle Detector is a Sharp Sensor, that is attached at the front of the robot, at a height of 13 cm from the ground, pointing front. This detects the height of the ground immediately in front of the robot, thus detecting potholes. The Hopper Mechanism consists of a bottle filled with beads (the filling material ). The opening of this bottle is fitted with a flap that is opened and closed using a servo motor. Opening the flap by the servo motor leads to beads falling out, thus filling the road. The outlet of the hopper must be suitably adjusted so that the filling material drops close to the Sensor, so that the hole which is detected by the Sensor is filled.
2. **Setting up the image processing hardware:** Create an arena, made of white thermocol with a white cardboard sheet on top of it. Place an overhead camera at a height of 5 feet above the arena. Attach the camera to the Base Station. Put patches of two different colours, one on the front, and one on the back of the robot.
3. **XBee:** Connect a XBee module to both the firebird, and the base station. Configure them properly so that they can communicate with each other.
4. Download the file DamarBot.tar.gz, and extract it into a suitable folder.
5. Open WinAVR, and and create a New Project in it. Import the C-files from the folder into this project. Place main.c as the main file, and all others as the supporting files. Compile the code with all default options for the Firebird (refer the Firebird Manual for more details).
6. Burn the hex file formed onto the robot using the AVR programmer. Now the Firebird is ready to use.
7. Open matlab, and run the file motion.m in it. On the Matlab terminal, enter the RGB threshold values of the two patches when asked. The code will display an image of the arena, from which the RGB values can be extracted using Color Detector, which is a freeware utility available for windows. Also enter the size of the logical grid that you expect Matlab to divide the screen into.
8. Once the matlab code is also set up, the robot is now ready to run. The Matlab code will give it suitable instructions to move in appropriate directions(using XBee), as per the path coded into the Matlab file. This path can be changed by suitably modifying the Matlab file.
9. You are now ready to go fill up the potholes in the road!

**Video Links**.

Link to Setup video:

Link to Demonstration video: