Software Requirements Specification

CS308: Embedded Systems Lab

Android based Tennis Ball Collector

Group 14

Hasan Kumar Reddy (09005065)
Vinod Kumar Reddy (09005071)
Avinash Theetla (09005056)
Bhanu Prakash (09005050)

Table Of Contents

1. Introduction

The document presents a detailed description of the Tennis ball collector bot. It is basically intended for the developers of the system. It describes the various terminologies and technologies used and interfaces of the system. It also describes basic features of the system, how to operate it and how the system reacts to the external stimuli.

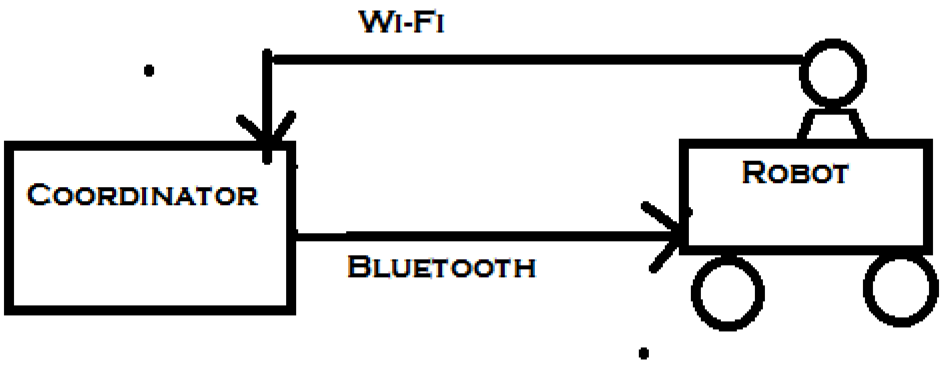

Figure 1

2. Overall Description

2.1.System's Environment

The Tennis ball collector robot contains two systems ,one active system and one coordinating system. The user is able to see the robot's environment from the coordinator and also able to relay commands to the bot. The connection between the bot and the coordinator is through wireless.

The environment of the robot is assumed to consist of table tennis balls scattered around which are of orange color. It is also assumed that only red objects in the environment are the tennis balls. The basket which is blue colored is situated somewhere around.

2.2. **Product Perspective**

This project aims at building a Tennis-ball collector robot which works autonomously when supported by an android device with an additional manual override option. It's in a way integrating two of previous projects into one single project – tennis-ball collector supported by matlab and controlling the FireBird V using android device. But will involve significant number of challenges like porting the matlab code to OpenCV which would run on an Android device, which has much lesser resources when compared to a PC.

2.3.Product Functions

The Tennis ball collector robot functions in two modes, user and autonomous mode. The user mode is where the user sends commands to the robot and the robot acts as per the commands .In the autonomous mode robot acts according to the environment without the intervention of the user.

In both modes the user can see the environment of the robot through the coordinator. The major functions supported in the two modes are given below.

User Mode:

- All the time the robot will be sending video feed to the coordinator(Android phone).
- According to the environment of the robot user sends one of the following commands through the coordinator
    1. move forward
    2. move backward
    3. rotate right
    4. rotate left
    5. pick the ball
    6. Drop the ball
- The coordinator interprets the command and sends appropriate signal to the robot through wireless.
- When the robot receives the signal, it converts it into appropriate command and acts according to the command.

Autonomous Mode:

- When the robot is on it scans the environment and sends the photo feed to the coordinator.
- The coordinator checks weather there is a tennis ball in the view sent by the bot. It does this using image processing.
- If the coordinator finds a ball it sends appropriate commands to go and pick up the ball.
- Else it sends the signal to robot to rotate by certain amount and then scan the environment.
- Robot receives the signal, interprets the signal and does according to the command.

2.4.User Characteristics

There are no restrictions or requirements on user. Anyone can be a user but the user must be able to use the coordinator and give the commands correctly(Here the coordinator is a Android phone and the process of giving commands is simple)

2.5.Constraints

- The Android phone used should have good processor since image processing is done in it.
- Range of firebird V is limited by the range of Bluetooth module on mobile as well as Bluetooth to serial converter (receiver) on firebird V.

2.6.Assumptions and Dependencies

- The orange objects present in the arena are only the tennis balls. Other than that no other object in the arena is of red color.
- Same restriction is for the basket. It is the only object with blue color in the arena.
- The ball on the arena are assumed have a minimum threshold distance so that when the robot tries to pick a ball, other balls won't come in the way.

2.7.Requirement Subsets

- Environment capture via camera
- Transmission of media from camera
- Processing the media received
- Sending appropriate signals to bot
- Analyzing the received signal
- Reacting / responding accordingly

3. Specific Requirements

3.1.External Interface requirements

1. User Interfaces

The user interface consists of the video transmitted by the robot. It also contains the on screen controls for changing the mode of the robot from autonomous to user and the other way. The interface also has the controls for moving the robot forward and backward, rotating the robot left and right, picking and dropping the balls.

The interface looks like the figure given below.

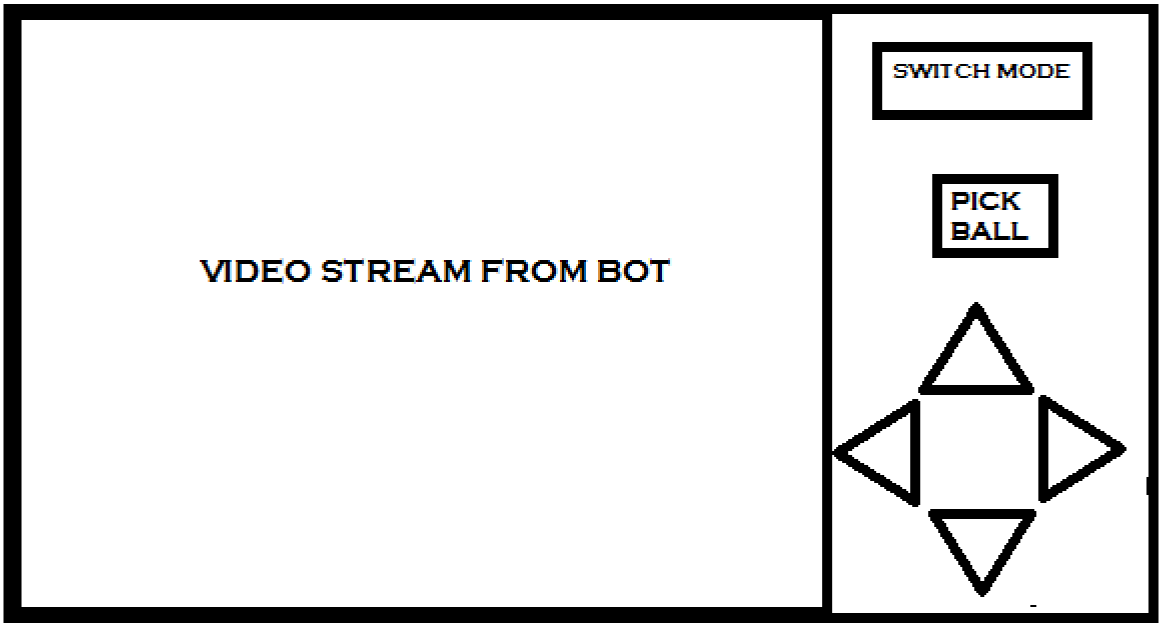

Figure 2

2. Hardware Interfaces

The hardware parts of the system and its interaction with the software is given one by one below.

Firebird V-ATMEGA 2560:

The ATMEGA 2560 robot used in the lab is used for the project. Initially the hex-code is burnt in the robot using AVR-studio and AVR Boot loader. When the system is started, the code interacts with the android application through wireless and the commands received are implemented by setting appropriate signals.

Wi-Fi Camera:

This part of the project is still vague. We are searching for various feasible options.

Possible Options:

- WiFi Camera (around Rs.5000)
- Bluetooth Camera (around Rs.5000) – problem here is mobile needs to pair with two bluetooth devices simultaneously.

  e.g.Samsung ST1000

But we'll first try to utilize the hardware available and make a work around with it.

Camera will transmit the live video stream / photos at periodic intervals remotely to android device, which in turn processes the media received and sends the instructions via Bluetooth to FB5 via Bluetooth.

Robotic Arm:

A robotic arm is mounted on the ATMEGA 2560 robot to allow the robot pick and drop the balls. The functionality of the robotic arm is accessed by the code burnt in the program and this access is same as that for the robot itself.

The robotic arm is constructed using two servo motors and gears (one for arm movement and other for the finger movement).The initial design we thought of is given in the figure below.

Android Mobile:

The Android mobile takes the user commands (like switch mode, move forward etc) and gives them to the Android application. The application interprets the commands and sends appropriate signals to the robot through wireless.

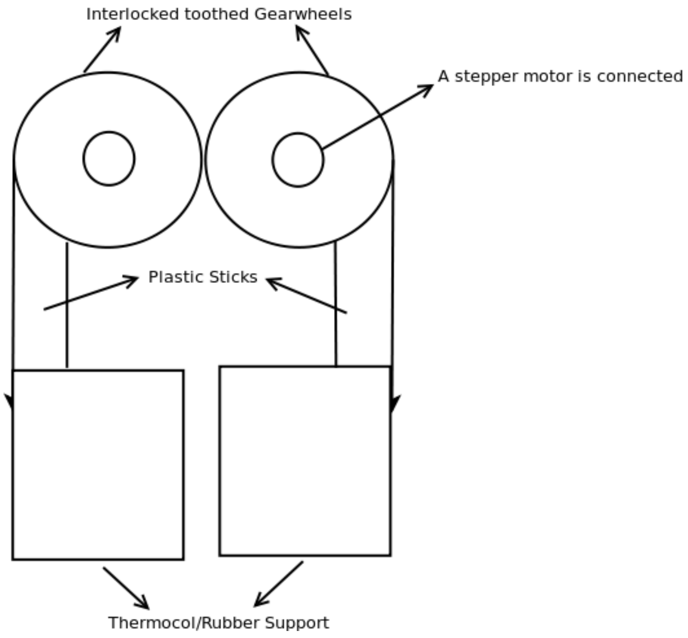

Figure 3

3. Software Interfaces

- **AVR IO Programming**
  Used to Interface various parts of FireBird V
  Softwares Required:
  - WinAVR
    Opensource compiler for AVR microcontroller
    http://winavr.sourceforge.net/download.html
  - AVR Studio 4
    IDE for coding C code, which uses WinAVR in the background as compiler
    http://www.atmel.com/avrstudio/
  - Boot Loader
    Burn .hex code generated after compilation

- **Android API**
  Build the application on android device
  Softwares Required:
  - Google Android SDK
    http://developer.android.com/sdk/index.html
  - Java SDK
    http://www.oracle.com/technetwork/java/javase/downloads/index.html
  - Eclipse IDE
    http://www.eclipse.org/downloads
  - JNI
    Build Java Native Interface (JNI) to the OpenCV library

  Other References:
  http://www.stanford.edu/class/ee368/Android/Tutorial-1-Basic-Android-Setup-Linux.pdf
  http://developer.android.com

- **Image Processing on Mobile**
  Using OpenCV to identify the object of interest (ball here)
  source target: android device
  Softwares Required:
  - Cmake (>=2.8.4)
    Building OpenCV
  - Android NDK
    The Android NDK enables us to compile and run native C/C++ code on Android
    http://www.crystax.net/android/ndk-r4.php
  - Apache Ant
    building Android applications that use OpenCV
    http://ant.apache.org/bindownload.cgi

- Open CV
  Install and build OpenCV library
  via svn from https://code.ros.org/svn/opencv/trunk/opencv

Other References:
http://www.stanford.edu/class/ee368/Android/Tutorial-2-OpenCV-for-Android-Setup-Linux.pdf
http://opencv.willowgarage.com/wiki/Android

4. **Communication Interfaces**
    - Bluetooth - Android to FB5
      Sending the signal to FB5
    - WiFi - Camera to Android
      Sending the video captures to mobile

3.2.Functionality

All the time when the system is functioning, the Wi-Fi camera sends the video feed to the Android phone. The Android application shows this video feed to the user. The details of functions of the system is given below.

USER MODE:

In the user mode the user gives the commands to be executed through the application interface. The application takes this commands, interprets them and sends appropriate signal to the robot through Bluetooth module. The robot interprets the code and does the required functions by setting the required signals.

AUTONOMOUS MODE:

In the autonomous mode ,first the robot sends the video of its environment. The android application takes a frame of this and does the image processing to check whether there exists a tennis ball or not. If the tennis ball (orange item) exists it sends the command for the robot to rotate 5 degrees in the direction of the ball until the ball comes in the middle of the view of the bot. Once this is achieved the application sends the command to go straight until it reaches the ball. This is achieved by the sensors. Using sharp sensors the robot stops at 10 cm from the ball. Now the application send commands to pick the ball and drop into its basket.

3.3.Supportability

1. Basis for future work

This project can be very helpful for those who intend to do android based projects in future. Our project itself is follow up of the previous year project. Future groups may add additional functionality to the existing projects or use the existing modules to do some interesting projects which are base on android.

2. Distinguished modules of the projects

There are three distinct modules of the project

- Android application for user command recognition, Image processing.
- Bluetooth communication module for wireless communication
- MUCOS/RTOS signal generation for different commands.

3. Documentation

A detailed documentation would be provided for the code and for the embedded components for the use of future groups who are interested to build on this project.

3.4. Design Constraints

- Only one FB5 can be controlled at a time.
- Android device must support Touch-based user feedback
- Ball should not be placed near the bot's blind spot (around 10cm)

4. Quality control

- **Predictability:**
  When two are more user inputs are given the device should resolve the conflict (if any) and respond in a predictable way.
  e.g., when both up and left button are pressed, it should either move left (conflict - priority based model) or some way middle (instruction based).
- **False positive results:**
  When a false positive indication is made while training, note it and try to adjust the parameters accordingly to minimize no. of such results.
- **Environment Effects:**
  Taking fixed parameter is not a good idea as real-life environment might be largely different from that of testing space. So, try to calibrate initially during the start of every connection and at regular intervals.

5. Risk Management

- When the robot is in autonomous mode and there is no ball in the view of the robot,the robot rotates in a prescribed direction for every 5 degrees and scans again. This involves delay as the android phone checks for the ball, so the motion is not continuous. This discrete motion may cause some problems because it involves sudden rise and fall of voltages.
  Fallback Plan : When the android application detects there are no balls in the view it asks the to rotate a large angle(< 45 degrees) and then continue scanning from there.

- When the robot is in autonomous mode, when the application detects a ball it asks the robot to move in a straight direction. But since there is a error in movements of both motors, the robot may move in arbit direction instead of straight direction.
  Fallback Plan : When the robot is moving in straight direction, after traversing some prescribed distance it scans again to check whether it is on coarse or not.