CS684 REPORT

CS684 – 2011 Project

GROUP 4

Project: SPEECH BASED BALL COLLECTOR

Subhabhrata Mukherjee 10305061

Dhaval Manvar 10305021

Janardhan Singh 10305067

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

Table of Contents

INTRODUCTION

Speech Based Ball Collector, is a robot capable of collecting colored balls based on speech commands. Implemented on the Firebird V robot, it accepts the color of the ball it has to collect in the form of speech commands, finds the ball with the help of a camera, collects the ball and puts it in a basket. Such a robot can be useful in many scenarios, like ball collection in a tennis court or pool table.

This document gives a detailed report on the project implementation. This project is a re-implementation and extension of the *Tennis Ball Collector* (Anup Naik, 2010) project. The previous implementation was in Matlab and C, and did not support speech input, whereas the current implementation is in Scilab (open source equivalent of Matlab) and Esterel.

Scilab is an open-source tool which is equivalent to Matlab. The SIVP toolbox of Scilab was used for this project. Some of the commands like "bwlabel" are currently not supported by SIVP. An algorithm was required to be developed for such commands in Scilab. The challenges faced and the solutions for them, while using this toolbox, have been discussed in the report. Esterel programming language provides a synchronous paradigm which is different from C language, which is comparatively a lower-level language. While doing this project we had to develop new signals which can be incorporated in the Esterel API for Firebird V robot. These include PICK, DROP, ROTATE (with parameters speed and angle), ZIGBEE SEND, ZIGBEE RECEIVE. These are described in the implementation section.

PROBLEM STATEMENT

The objective of this project is to program the Firebird V robot, so that it can collect balls spread in an area, according to the color received from the speech command and put in a bucket.

REQUIREMENTS

Software Requirements:

JAVA Speech API

The JAVA speech API (Sun Microsystems, 1998) was required to develop the speech processing module. The module required to accept user commands and map it to one of the pre-defined colors.

Esterel

Esterel provides a synchronous programming paradigm. Programming in this mode is easier in the sense that mealy machines can be easily converted into Esterel code. Esterel works on the signals that It identifies. Some signals were required to be added to the Esterel header file (Prof. Kavi Arya), to develop the code the details of which are given in the implementation section.

Scilab and SIVP toolbox

Scilab is the open-source equivalent of Matlab. Scilab version 5.3.3 was used for this project. The Scilab Image and Video Processing (SIVP) toolbox was required for processing images of the area in which the robot searches the ball.

Hardware Requirements:

Laptop

A part of the program, (the one written in Scilab) was required to run on a laptop. This program's job is to display the video captured by the ball collector and process it to identify the ball (while collecting) and later the basket (while dropping). The laptop was also required for running the speech processing module, and making it interact with the Scilab program so that while processing the captured video, the Scilab program knows the color of the ball it is seeking. This color is received from the user, by the speech command.

Firebird V Robot

This is the robot on which the project is implemented (Firebird V ATMEGA 2560 hardware manual version 2 , Dec 3, 2010) (Firebird V ATMEGA 2560 software manual , Dec 27,2010). The remaining part of the code runs on this robot for control and movement.

ZigBee Wireless Communication

This wireless communication was required for communication between the codes in the laptop and the robot. The two ZigBee modules need to be configured using XCTU software.

Figure 1: ZigBee Module

Two ZigBee modules, as the one shown in the above figure, can do wireless communication with each other. One is attached to the Firebird robot. The other is attached to the laptop using a holder.

Arm

Firebird V robot does not come with an in-built arm for collecting the balls. Hence, an arm supported by 3 servo motors and a thermocol gripper was required to be attached to the robot.

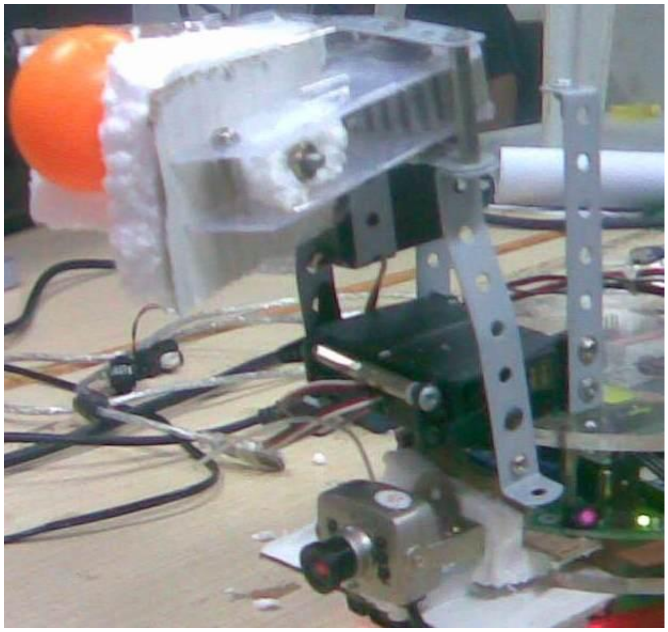

Figure 2: The Arm gripper and the camera

The specifications used for the arm gripper were same as that used by (Anup Naik, 2010) as shown in the following figure:

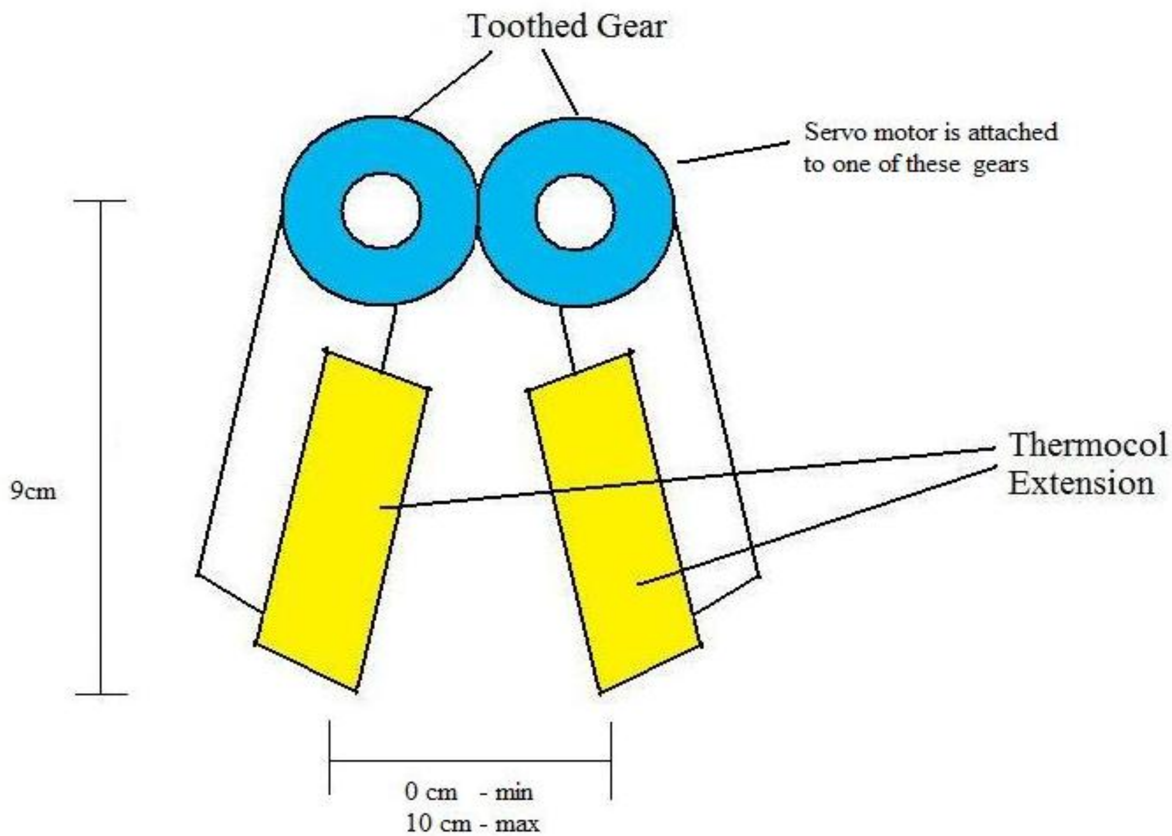

Figure 3: Arm gripper dimensions (Anup Naik, 2010)

Camera

A wireless camera was required to be attached to the firebird robot so that it can send images of the area in which it is navigating. The camera can be seen attached to the robot in Figure 2.

Microphone

A microphone was required to give the speech commands to the robot.

IR Sensor

When the collector moves towards the ball, an IR sensor was required so that it can sense the proximity of the ball, stop and pick up the ball. An IR sensor, as shown in figure 3 was attached to the robot, below the camera and connected to port number 2 of the firebird robot.

Figure 4: IR Sensor

IMPLEMENTATION

System architecture

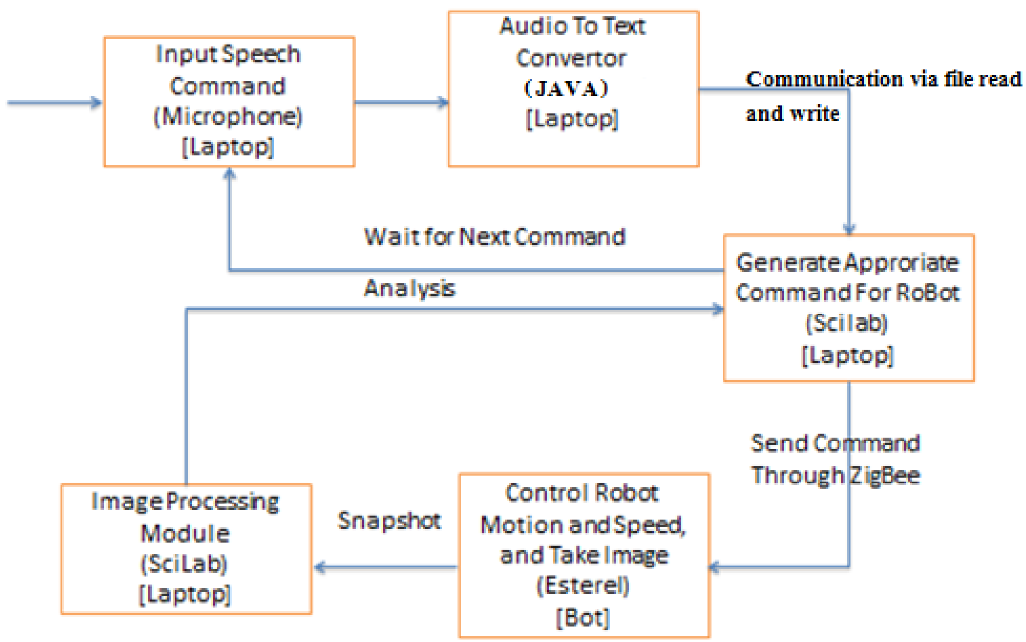

Figure 5 : System architecture

Code folder

speechrecg – JAVA project folder

This folder contains the eclipse project for speech recognition module.

final.sce – Scilab program

The image processing and controlling program that runs in Scilab on the laptop is in this file.

esterelcode – folder with Esterel code

This folder contains the Esterel code along with the generated C file and the enhanced header file “firebird_winavr.h” and perl file “getmodulename.pl” containing additional signal names.

Speech processing in JAVA

Speech processing is implemented using JAVA Speech API (Sun Microsystems, 1998). The JAVA project for the speech processing module is in the folder **SPEECHRECG.** It was developed using the Eclipse IDE using the JAVA sphinx 4 speech recognizer (JAVA Sphinx 4 speech recognizer, as on Nov 2011). This module writes the color, given via speech command, in a file called **color.txt.** The Scilab program can read the color from this file.

Object detection and communication with firebird implemented in Scilab

The first job of the Scilab program is to read the color input from the file **color.txt.**

Pseudo code:

```
Read from the file color.txt which color of ball to
pick.
Basket color by default is green
Call:
main(color_read_from_file);        // call main function
```

After reading the color, the main function then communicates with the robot using ZigBee serial communication and directs it to pick and drop the ball appropriately.

Pseudo code for main function:

```
function main(objc)

      set parameters for frame dimension
      open the serial communication interface with
port_no=6
      open the camera
      send 2 dummy signals to firebird to initiate it
before picking ball
      traceobject(objc,'1','2','3','ball');   //trace
ball
      send 2 dummy signals to firebird to initiate it
before dropping ball
      refresh the frames and re-initiate camera
      traceobject('green','4','5','6','box'); //trace
basket
      closeserial
endfunction
```

Pseudo code for traceobject function used in main:

```
/*
use camshift function in scilab to trace the object in
each frame
Parameters:
color=color of object to find; notfound=signal to be
sent till object is not found; found=signal to be sent
when object is found; objct=ball or basket additional
parameter : action="pick" or "drop" the object when it
is reached
*/
```

```
function
traceobject(colour,notfound,found,action,objct)

     rectret=findobject(colour,notfound,found,objct)
//call rectrec to                 detect the object

     while there is a frame captured by the camera
          use camshift funtion in scilab to trace the
object position in the frame
          wait till the object comes in the centre of
the frame
          i.e. the distance between frame centre and
object centre is less than some threshold value
          threshold is to be set keeping in mind that
the bot is in motion

          if the object is at centre of frame
                    send signal to firebird
                    keep on sending signal till it is
ack-ed
                    keep on reading the buffer for
another signal from firebird indicating that the object
has been reached and action taken (ball picked or
dropped)
                    return on receiving this signal
          else
               keep on sending signal to firebird
asking it to rotate slowly
endfunction
```

Pseudo code for rectret function used initial detection of object used in traceobject function:

```
/*
function to detect object and return the co-ordinates of
the object, in the 1st frame it is detected, in the vector
rectrec
Parameters:
color=color of object to find; notfound=signal to be sent
till object is not found; found=signal to be sent when
object is found; objct=ball or basket
*/

function rectret=findobject(color,notfound,found,objct)

     set threshold for color intensity of the object //set
manually
     set the window range in the frame in which to search
the object, typically the mid 200 pixels in the frame
     while there is a frame captured by the camera
                subtract the color component from the image
                convert the image to binary
                perform seedfill algorithm in the specified
window, to detect the biggest object cluster
                with threshold > 1000 pixels, at interval
of 10 pixels
                if object found
                     return
                else
                     keep on sending signal to firebird
asking it to rotate slowly
endfunction
```

Controlling Firebird robot using Esterel

The firebird robot used the following signals for communicating with the Scilab program as also mentioned in the pseudo code above:

Signal	Source - Destination	Function
1	Laptop – Firebird	Keep rotating
2	Laptop – Firebird	Stop rotating, ball is in centre of input image
3	Firebird – Laptop	Ball pick-up complete
4	Laptop – Firebird	Keep rotating
5	Laptop – Firebird	Stop rotating, basket is in centre of input image
6	Firebird – Laptop	Ball dropping in the basket complete

For controlling the firebird robot using Esterel, 5 additional signals were defined :

1. PICK
   Robot moves forward till the IR sensor detects the proximity of the ball. The gripper opens, arm goes down and the gripper closes. The ball is caught within the thermocol gripper and the arm moves back up.
2. DROP
   Robot moves forward till the IR sensor detects the proximity of the basket. The arm goes down at 45 degrees. The gripper opens to drop the ball in the basket. The arm goes back up.
3. ROTATE (SPEED,ANGLE)
   The soft right rotation takes place at desired speed and angle.
4. RECEIVE [ZIGBEE SIGNAL]

Reads the byte input from the ZigBee serial port.

5. SEND [ZIGBEE SIGNAL]
   Writes a byte output to the ZigBee serial port.

These signals are generic and can be used for other applications using Esterel for programming the firebird robot.

The Esterel code is generated using the following state diagram.

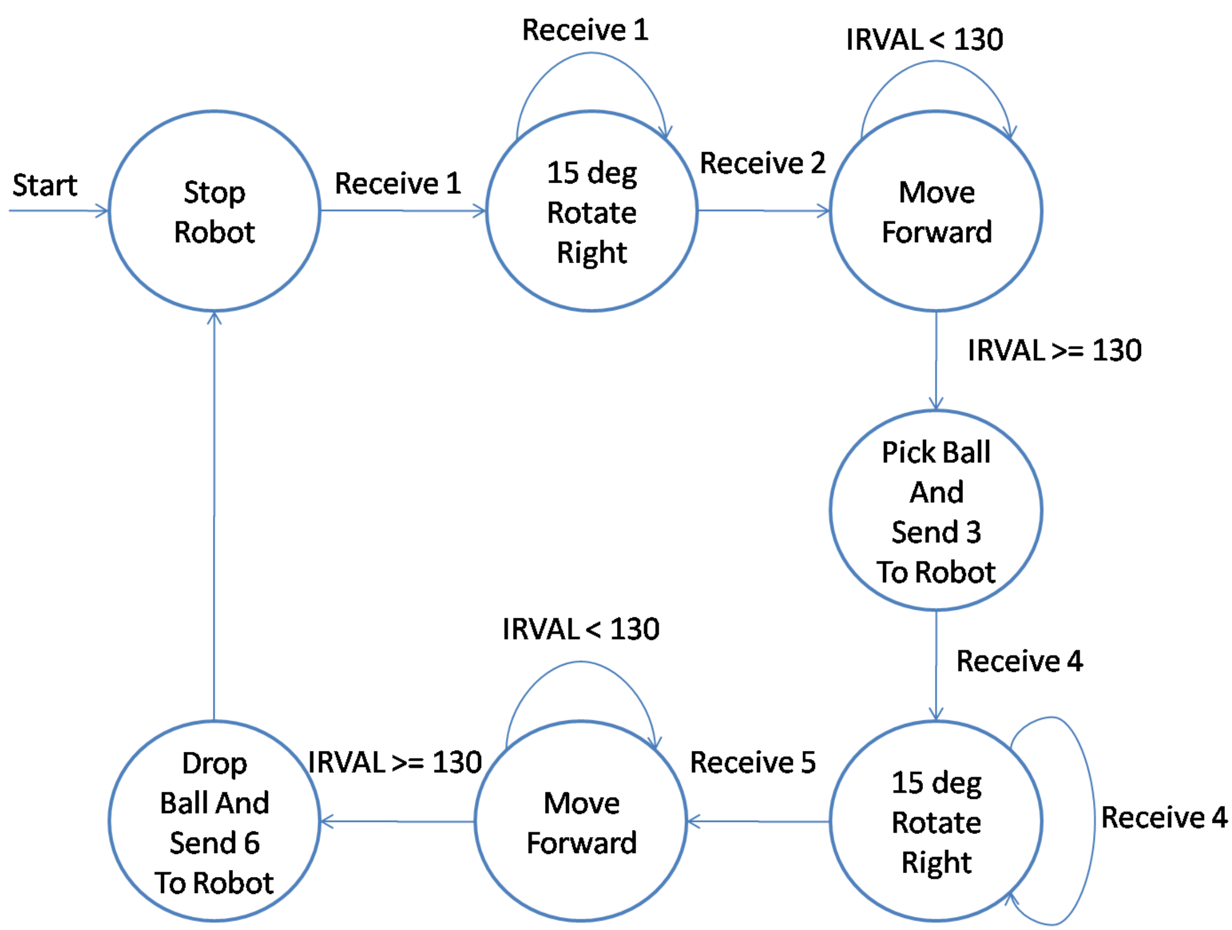

Figure 6: State diagram for esterel code

TESTING STRATEGY AND DATA

Test Data

a. Identification of Voice Command
    Red, Green and Blue

b. Identification of Objects with given Color
    Identifying the red ball in presence of the blue ball and vice versa.

c. Identification of Basket
    Green color was designated to the basket.

d. Retrieval of Objects
    This includes searching for the object, moving towards it, stopping due to IR sensor and grasping the ball by its robotic arm.

e. Placement of Objects in a Basket
    This includes bringing the object towards the basket, stopping due to IR sensor and dropping the ball within the basket.

DISCUSSION OF SYSTEM

Observations from Testing

Test	Observation	Reason
Identification of voice commands	By keeping only three classes(red, green and blue), this task worked well	-
Identification of colored object	The threshold of color intensity needed to be tuned for red, blue and green. Sometimes some far spurious objects got detected	We detect object by cluster of pixels of the specified color of size greater than thousand pixels. Since, shape is not detected, some far objects got detected. Hence, the area needs to be clear.
Identification of basket	Same as above	Same as above
Retrieval of object	By ensuring that there is no pull from the wires attached to the robot , while it moves forward for picking the ball, this part worked well.	-
Dropping of object	By restricting the green color in the central part of the front face of basket, this part also worked well.	If the whole front face was kept green, then sometimes the robot identifies the edge of the basket and drops the ball outside or on the edge.

General discussion

1. The SIVP toolbox is maturing. Due to absence of some functions that are present in MATLAB viz. "*bwlabel*", we had to implement our own algorithm for object identification. We tried our best to optimize the algorithm for object identification. We optimized the algorithm for object identification by sampling sections of image instead of entire image. Delay was minimized to synchronize the image processing and esterel automation as well as communication via zigbee. Once the first frame in which the object is detected is processed successfully, the "*camshift"* function takes over and a desirable speed is achieved overall.
2. Esterel api for firebird also lacked some signals and we hope the signals we added are re-usable for further projects on firebird using Esterel. We developed a protocol for Esterel communication with Scilab
   Protocol makes sure transmitted signal is received by sending repeated signals till acknowledged.
3. In the image processing part, tuning the intensity threshold of different colors for differentiating among them required significant effort.
4. Shape identification has not been done and can be a useful extension of this work.

FUTURE WORK

1. Including shape identification along with color identification.
2. Involve some Natural Language Processing to give more generic commands rather than just color.
3. Re-implementing and comparing using *opencv* instead of *Scilab.*

CONCLUSION

We extended the existing tennis ball collector by introducing a speech api.

We made the project platform independent and open-source by using open source tools like scilab, java and Esterel.

We also developed a protocol for efficient communication between software and hardware i.e. between scilab and esterel for smooth functioning of the system.

By this project we hope to provide a basic version of speech based ball collector which can be further enhanced in different directions to build more sophisticated systems.

Bibliography

Anup Naik, V. K. (2010). *Tennis Ball Collector.* www.e-yantra.com.

(Dec 3, 2010). *Firebird V ATMEGA 2560 hardware manual version 2 .*

(Dec 27,2010). *Firebird V ATMEGA 2560 software manual .*

(as on Nov 2011). *JAVA Sphinx 4 speech recognizer.* http://cmusphinx.sourceforge.net/sphinx4/.

Prof. Kavi Arya, Y. S. *Manual: Programming Firebird Using Esterel.*

Sun Microsystems. (1998). *JAVA Speech API: Programmer's Guide.* http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide.pdf.