FireBird Swiss Knife

Project Report

Pratik Gilda 08005046
Shahansad K. P. 08005060
Charles Talahoti 08005055
Saurabh Singh 08005038
(Group 15)

Under Guidance of **Prof. Kavi Arya**

Abstract

We will develop codebase for Firebird V which will contain a host of functions which will be useful for anyone trying to program Firebird V. This will be useful for the developer as it will allow him to work at a higher level of abstraction thereby rescuing him from the chaos of PORT-level programming. Since the codebase will be tried and tested, anyone writing new programs will have more time at his hands to do more interesting stuff.

We will also provide a python interface for our Codebase; so that every function in C-library can be called form Python.

Introduction

The main objective of this project is to develop a master library of functions to simplify the job of other programmers working on Firebird V. The Python interface will help user to write programs in Python.

The biggest advantage of such a common codebase will be the standardization of the Code developed for other projects across newer generations of Firebird Robot. In-effect, our project will help other developers to separate the logic of the code from the implementation details.

Also, Python has integrated Image processing Module. So instead of controlling bot via C and image processing via Matlab. All can be done via Python, thus simplifying job of all the ERTS-FIREBIRD coders.

The main constraint is that, sometimes two functions may not be able to function together, since the ports are shared for providing multiple functionalities on Firebird.

System Architecture

The bot will have one In-vehicle unit which is ZigBee enabled transreceiver module, which will communicate to zigbee module connected to the computer. In our prototype FireBird-V robot will be used as a vehicle to be controlled by Python and C-programs.

The Python Interface of the computer communicates to the bot via zigbee serial communication from C++. Based on the Commands issued by user in python, the python function sends the name of that function and its arguments to the bot via zigbee. The bot receives them and operates on them.

The function running on the bot continuously listens for any strings received from the computer (via zigbee), then calls an invoker function with the received string which takes appropriate action based on command received.

One timer on the bot (TIMER 4) is used for operation of functions like strictForward, rollLCD which have to be monitored even after the function call has returned.

Project Plan

1. Implemented Zigbee Serial Communication via C
2. Made the C-code into a Python module
3. Implement functions in Python
4. Implement functions on the bot
5. Added the invoker function on the Bot to complete the Python interface

Work Division:

Shahansad,Pratik : 1, 2 and 3 (from above)

Charles, Saurabh: 4 and 5 (from above)

Challenges:

1. How to model the Python interface
    - Solution: use request-reply model with Computer issuing the Request
        - We needed to send large amount of data between the bot and the computer continuously.
        - Any data is sent in String format and a special character is used to separate independent strings.
        - A linked list is used as buffer at the bot side.
        - Data is converted back into its needed form at either side.
        -
2. How to catch bot Interrupts on Python
    - Solution: start a new thread in Python which keeps asking the bot about the occurrences of the req. interrupt
        - For any Interrupt user is interested in using, a dummy interrupt handler is created whose task is to count the number of times that Interrupt occurred.
        - A new thread is created in python which keeps polling the bot periodically to get the number of interrupts and invokes the python interrupt handler those number of times.

3. How to handle functions with enduring effects
    - Solution: Use a dedicated timer on bot to do all the House-keeping
        - A timer (timer 4) is exclusively used for housekeeping tasks of such functions.
        - It wakes up periodically and depending upon the flags set by other functions, it performs required tasks.
        - Ex: for rollLCD function this timer shifts the string to be displayed left by 1 character each time it comes alive.

Testing Plan

- We build a white Line Follower and a game –style controller for bot in python using our Library functions.
- We also created some example files to test functionality of other functions(not used in above)

Assumptions:

Only one bot is connected to the Computer (by zigbee).

Future Directions

- More modules can be added to this project to support more and more hardware or higher level functions.
- When a new Bot say FireBird Vl is developed, all functions in the Library can be re-written such that all the previous projects will remain valid.
- When a sufficiently powerful Bot becomes available, Python interpreter can be run directly on the bot, thereby avoiding dependence on the computer and Zigbee communication.
- It can be extended to control more than one bot.

Requirements

1. One FIREBIRD V bot
2. One zigbee module on bot and one connected to the computer