Spring 2014

Embedded Systems

Title:

iRobot Create Remote Navigation System

By:

Francis Sowani

Instructor:

Michail Maniatakos

Teaching Assistant:

Nektarios Tsoutsos

Table of Contents

Table of Figures

1 Introduction

Navigation is an important characteristic of an autonomous robotic system. This makes the question of how a robot can automatically or manually navigate between two points to be an important problem in Embedded Systems and Engineering at large.

Hence, this project work is an attempt to design and implement a simple semi-automatic remote navigation system for *iRobot Create* (Figure 1). The goal is to achieve a navigation system that both the user and the robot will be contributing in making decisions during the navigation. In particular, the robot’s decisions will target correcting the mistakes the user will perform that may have such consequences as destroying the robot hardware or fail to reach the target of navigation.

Figure 1: *iRobot Create*

2 Problem Definition

This project work aims at creating a simple semi-automatic remote navigation system for *iRobot Create.* How can iRobot Create be remotely and safely be navigated between two points is the question that this project work is addressing and attempting to solve.

Remote control of the robot implies that the user with his or her control interface be physically detached from the robot being controlled. It follows then that the communication between the user’s control interface and the robot take place through a wireless medium.

The safety aspect in the context of this project implies the safety of the robot itself. User’s actions should not be fatal to the robot’s hardware or software. The project will also address through its system a mechanism to ensure that the robot becomes a smart servant. It should to a satisfactory degree be able to decipher—depending on the circumstances—those user’s commands that if the robot will take action on, it may be disastrous to the robot.

3 Objectives and Deliverables

3.1 Objectives

To address and attempt to solve the above problems, the following objectives are proposed. When successfully completed, these objectives should be able to solve the problems above.

3.1.1 Communication Protocol

The project will design and create a minimalistic protocol for ensuring wireless communication between the robot and the user's control interface.

3.1.2 Remote Connection

The project will implement a user's control interface that can establish remote wireless connection to the robot. Through this connection, the user will be able to send commands to the robot and on the other hand the robot send real-time feedback to the user. Also the robot will constantly be able to report its status back to the user.

3.1.3 Robot's Autonomous Decisions

The project will give the robot an ability to override or ignore user's commands that depending on the circumstance, taking actions on those commands will be lethal to the robot itself. An instance of such an action is when a user instructs the robot to drive forwards when the robot is right at the edge of a cliff. This clearly will make the robot fall and perhaps destroying itself. In such scenarios, the robot should ignore the user input and give the right feedback to the user.

3.1.4 Robot's Feedback

Extending on objective 3.3, the robot should be able to provide feedback to the user when it makes an autonomous decision that did not directly result from a recent user's action. For instance, a robot may have been moving straight for five minutes when it encounters a cliff and needs to automatically stop. It should provide the right feedback to the user in such cases.

Also, constantly the robot should provide status feedback to the user. The robot's status should be reported in real time.

3.2 Deliverables

To measure the success of the above objectives, the following deliverables will have to be produced upon the completion of this project work.

3.2.1 Project's Timeline Documents

Project's milestones and the complete schedule with completion estimates need to be delivered before the beginning of the project work. These will have to be reviewed and approved by the Project Supervisor for the project to take off.

3.2.2 Project's Analysis and Design Documents

These are the project's design documents. A Statechart design for the robot's navigation and communication commands protocol implemented will need to be delivered as the project progresses.

3.2.3 Project's Working Prototype

This is the final working system of the project. It will involve both the LabVIEW interface and the functioning robot. The source code and a demonstration of the working prototype will need to be delivered upon the completion of the project.

3.2.4 Project's Final Report and Demo Video

After completing this project, a final report and documentation will need to be delivered. The project should assess the objects and results—measuring the success of the project. A demo video should accompany this documentation to serve as a record for future reference.

4 Analysis

4.1 Overview

With the ultimate goal of achieving the four objectives articulated in *Section 3.1*, LabVIEW development environment was selected for the design and implementation of the system. LabVIEW provides a great platform for high level design of the system components as well as debugging tools. This is of much significance taking into consideration the short period that this project has to be implemented within. *iRobot Create* is the robot of choice and myRIO-1900 will be the FPGA board that will be used to interface the *iRobot Create* with the LabVIEW.

A small Local Area Network will have to be established from which the robot and the LabVIEW will communicate. LabVIEW has a native communication protocol to myRIO, hence there will be no need to create a new communication protocol from the scratch.

Since the objectives are fixed and it is relatively a small scale project, the project will be implemented following the Waterfall model for software development.

4.2 Milestones

To achieve the objectives in an effective manner, the project will be conducted in six stages or milestones within a period of six weeks. Each milestone will span a period of one week, ending with producing the deliverables for that particular week. Below are the details of the six milestones.

4.2.1 Week One

Goals:

i. Remote connection to the iRobot Create with ADHOC Wi-Fi connection
ii. Invent and implement a communication protocol (TCP like)

Deliverables:

i. LabVIEW interface implementing Wi-Fi connection status (Online/Offline) indication

4.2.2 Week Two

Goals:

i. Implement manual navigation of the robot

Deliverables:

i. LabVIEW interface to send commands to the robot
ii. Robot responding to navigation commands (e.g. move, turn, stop)

4.2.3 Week Three

Goals:

i. Report robot status (all readings)
ii. Implement automatic navigation that overrides bad user decisions

Deliverables:

i. LabVIEW interface that repots robot status
ii. Robot overriding bad user commands (e.g. move when the robot reaches the edge of a cliff)

4.2.4 Week Four

Goals:

i. Try (report an error if this is impossible) to automatically move to a given relative coordinate with respect to compass measurements

Deliverables:

i. Automatically move to a target coordinate

4.2.5 Week Five

Goals:

i. Integration testing and debugging

Deliverables:

i. Working prototype meeting the specifications

4.2.6 Week Six

Goals:

i. Delivering presentations and final reports

Deliverables:

ii. Final presentation and report (4,000 – 5,000 words report)

5 Final Design

Below is the final proposed design for the project system. The system will comprise of three main ends: user interface, local network and iRobot Create.

The user will be in direct control of the user interface. The user interface provides the user with all the capability to control the iRobot Create remotely over the network.

A common use case will be the user starting the LabVIEW interface and iRobot Create. Then the user connects to iRobot Create by running the interface. The user interface will indicate the status of the connection as well as the feedback polled from the robot. By pressing the “Drive” button, for instance, the robot will move forwards until stopped by the user or the robot encounters an obstacle.

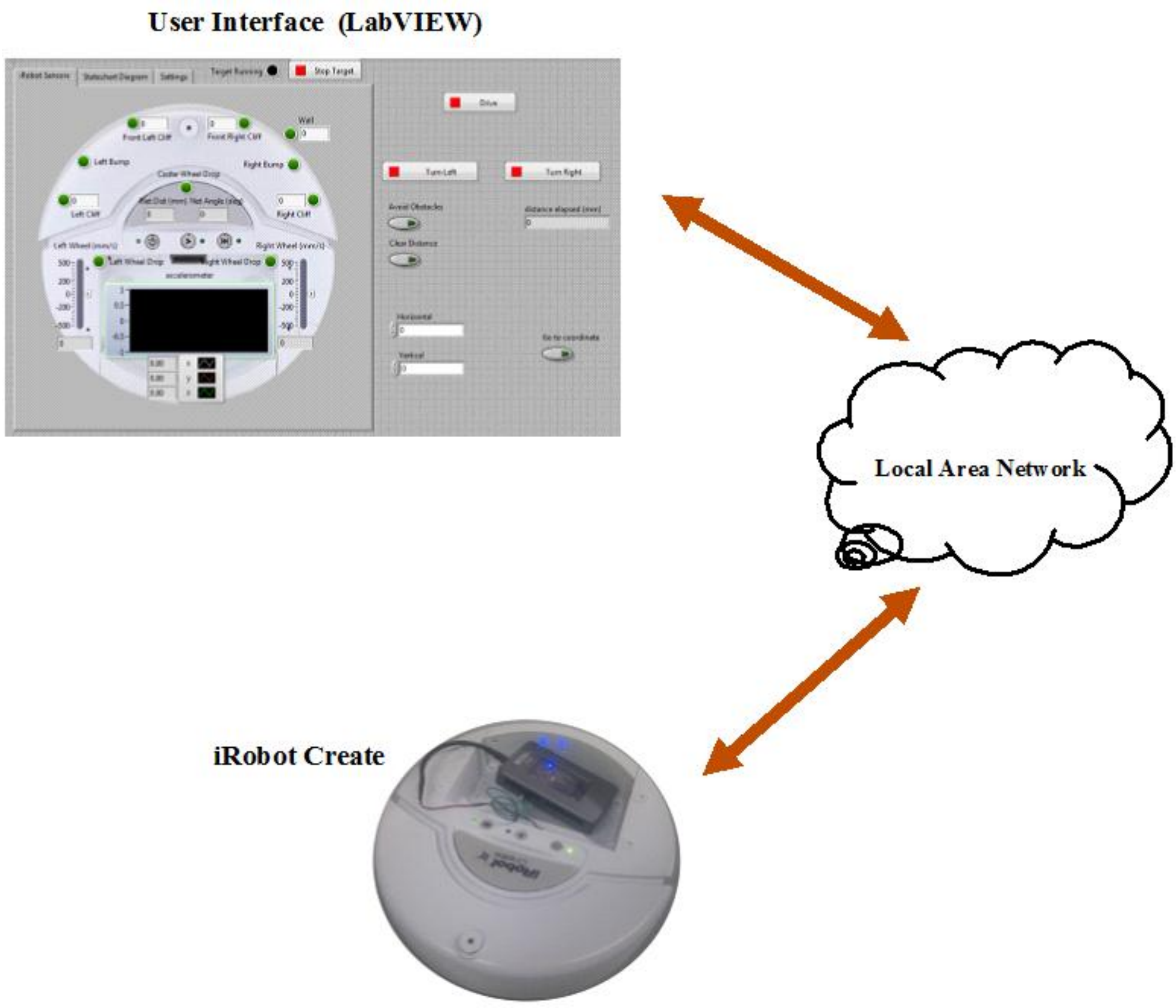
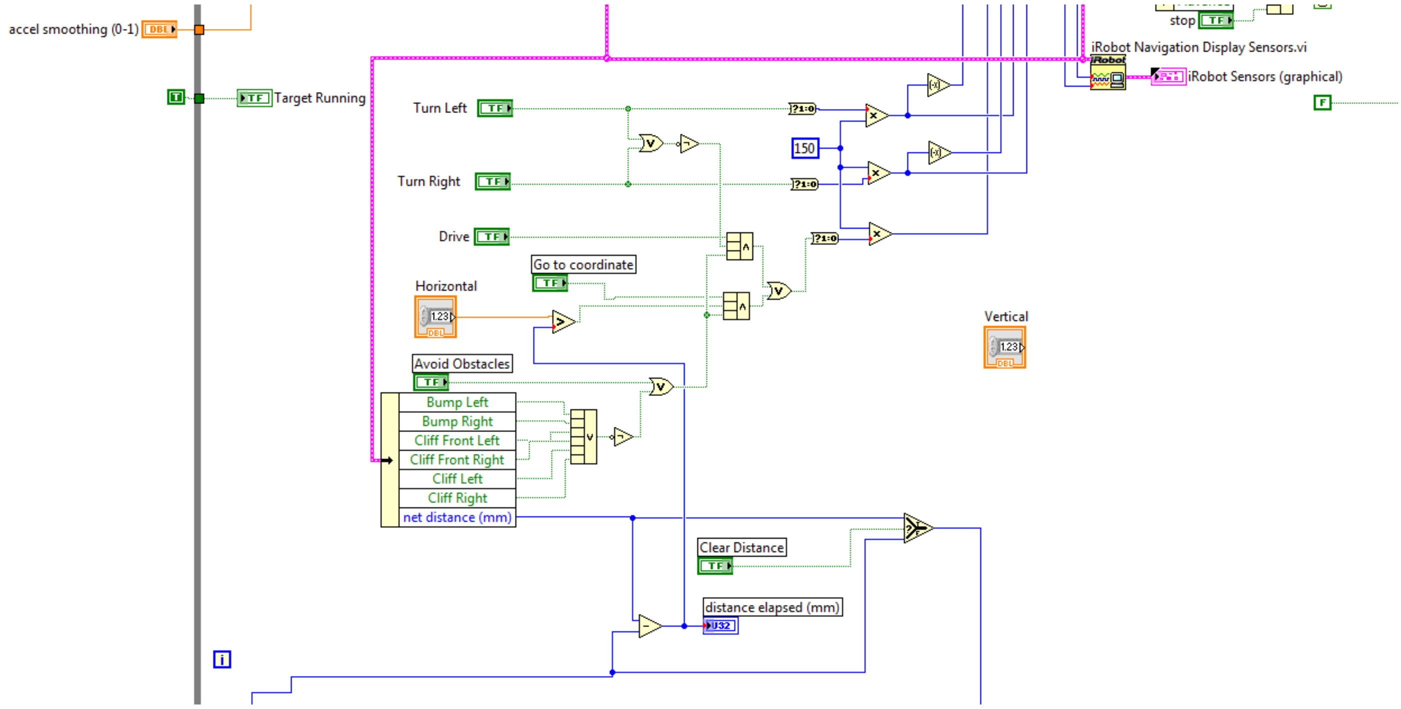


Figure 2: The final design for the navigation system

6 Implementation

The design was implemented in LabVIEW. The final source code for the implementation can be found at https://github.com/Frathoso/embedded-final.

Below is the a section of the system's block diagram that controls the user commands



The obstacles avoidance capability was implemented in a Statechart as shown below:

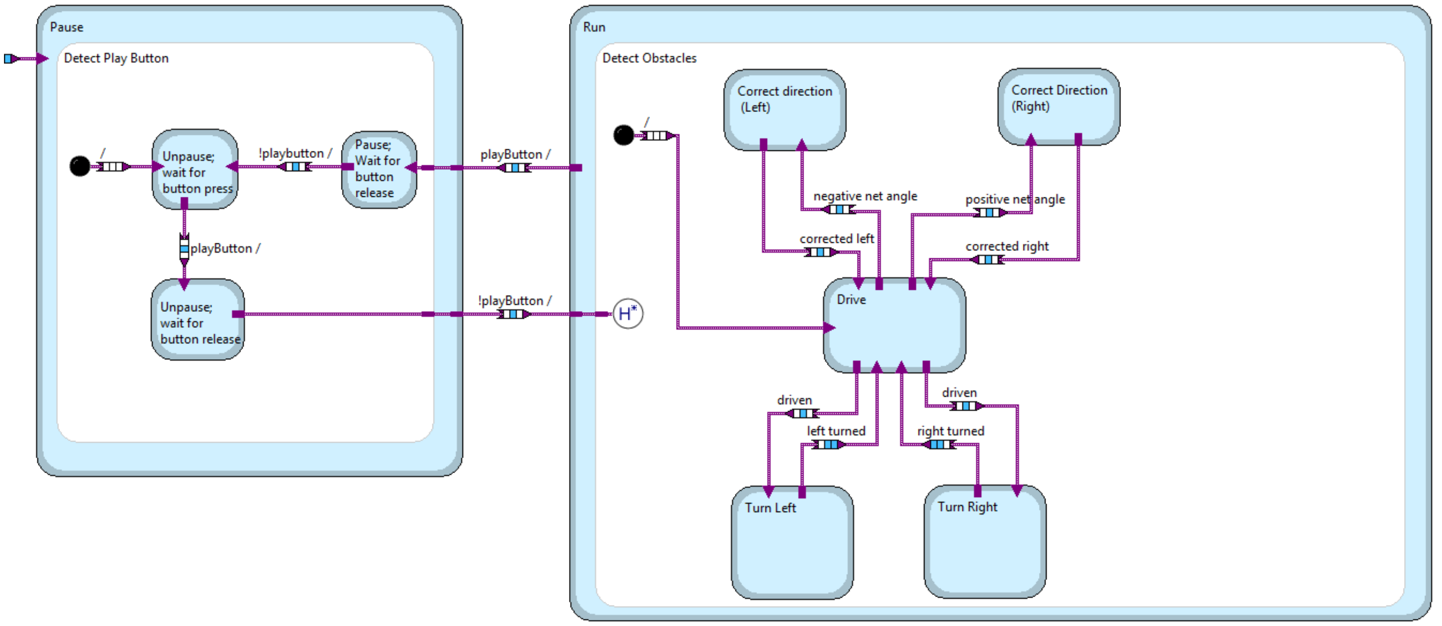


Figure 3: Statechart of obstacles avoidance algorithm

7 Conclusion

During the implementation of the project, a couple of challenged were faced that were overcome. One was the issue of a wireless network. At the beginning, NYU network was used but it did not allow for a successful connection between the LabVIEW and the iRobot Create. This was solved by creating a local area network using a custom router.

Another challenge was the way to store values between different iterations of the main loop. This was solved by using registers that feed forth the previous values into the next iteration of the main loop.

However, one challenge could not be solved. It was the way to make the control of the robot from the Statechart work smoothly with the control of the robot from the block diagrams. Since this could not be solved, the obstacle avoidance algorithm could not work with the rest of the algorithms. In the future, I propose to limit the entire control of the robot into either the Statechart or the block diagrams.