
Project Title
Automatic Parking Lot (Valet Parking)
CS684 Projects

Group : 13

Mukesh Kr. Agarwal, 10323007(TL)
Dileep Singh,09305003

1) Introduction

Finding a parking space in a constrained parking area as very difficult task. The main idea is to develop a centralized automatic parking zone where central server keeps track of available space and give command to park at specified available parking space and also give command to unpark the vehicle when demanded. This project can be see as automated valet parking.

2) Problem Statement

Developing a automated parking lot where vehicle can be parked at specific parking space commanded by central sever and also able to unpark the vehicle when demanded. Beside this stacked parking need to been.

3) Requirements

1. Hardware Requirements

- White Line sensor
- Zig-bee Interfaces
- Fire-Bird Programmer

2. Software Requirements

- AVR Studio 4: IDE for programming AVR micro-controllers
- C- language

3. Communications Interfaces

Zig-bee modules were used for communication. Some settings were changed. To reduce interference, Zig-bees were paired up so that they sent data to and received data from particular Zig-bee modules only, that too on a particular channel. Data packets from non paired Zig-bee modules and non matching channels were dropped.

4) Implementation

Functionality

Each of the cars would be required to follow white-line markers to an unoccupied parking pace, and park (stop) themselves appropriately, without colliding with any of the previously present cars in the lot.

Functionality for parking and unparking is illustrated below separately in different section.

4.1 Parking

The steps that will be followed during parking are as: message received for park by zig-bee.

1. Find the required parking needed, count = 0, junction=0.
2. Follow white-line
3. count increment count = count+1.
4. If count<junction goto step 5
5. if left park goto step 6
6. Turn left goto step 2
7. if right park goto step 8
8. Turn right goto step 2
9. park.

4.2 Unparking

Position of car in the parking area is determined by ZigBee communication. The steps that will be followed during unparking are as:

1. Message received by zig-bee for unpark goto step 3
2. Follow white-line
3. Check bot left- right side.
4. Move reverse till the junction
5. if left parked goto step 6
6. Turn Left goto step 2
7. if right parked goto step 8.
8. Trun Right goto step 2.
9. unparked

Challenges

- Esteral signal handling
- Speed synchronization of both wheel
- Zig-bee communication setup
- Straight reverse motion of the bot
- The angular rotation of the robot is sometimes unpredictable due to the unbalanced wheel motion and power supply issues

5) Testing Strategy and Data

A 3x2 parking area is created which had free parking lot each side. Parking an unparking command is given by central server(System). Manually to bot through Zig-bee. When bot receive parking command it decodes which parking space it has to park. Bot follow white line considering as road (Path way). Keeping track of junction count. When bot reaches required junction it turn specified parking space and park itself.

When unpark signal is received bot move in reverse direction to the junction, take a 90 degree turn and comes out following white line.

6) Discussion of System

6.1 Worked

- Automatic parking in esteral
- Zig-bee communication setup
- centralized automatic parking and unparking in C language.

6.2 Thing didn't work

- Automated unparking in esteral was unable to handle signal in esteral
- Stacked parking and time constrained parking strategy didn't work and because lot of time consume in basic building and zig-bee communication setup in esteral hence ran out of time.

7) Future Work

- It can be implemented on multiple bot and multiple LAN simultaneously.
- It can be implemented stacked parking depend on parking time.
- Space optimization for parking Lot.

8) Conclusions

We have implemented successfully of automatic parking and unparking robot through the zig-bee communication. The robot should avoid collision with any obstacle, and other vehicles, that are already parked. The project has a good learning process. But compared to the amount of time required. The main thing we have learned is how difficult and unpredictable real time hardware systems.

9) References

1. <http://www.e-yantra.org/ci/projects> :Automatic Parking
2. 1. Fire-Bird V AVR2560 Hardware Manual
3. FireBird V AVR2560 Software Manual