FINAL-PROJECT REPORT

The Propeller car

9961244 EE14 吳德霖

Abstract

This is a project that we need to combine propeller spin codes, zstar codes of the sensor, and the sensing function, all that we learnt during this semester to assemble a car system, including some extra creative capabilities added.

System design and components

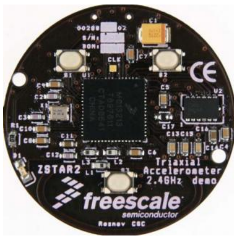
Basically, I made good use of the spin code provided by the company who made this propeller car, the “servos control” one. And I modified it a lot to empower all the functions existed within the code. The basic design method was to meet my requirements, that is, to act like a real car when doing such function. For example, the blinking of the LED lights and the turned on of which is to model the real car motion when turning right or going backward. The abilities that my car is capable of are listed in the following section. By using the accelerations of each axis of the 3-axis accelerometer, I made the sensor transmitting the command when the values fall into those valid commanding range, so that I can control my car by simply tilting the sensor. Other applications were done in the spin codes, with several columns of “if” and elseif command, as well the repeat function of the spin code. Opening two serial ports and manipulate them together, I combined two codes to one to let the signals from the sensor can be detected on the board, and then to the servo.

Implementation details

As attached, all the details of coding were directly tarred.

Instructions for my project

*Please refer to XBee_servos_v2 to run this program



(this direction)

Hold the 3-axis accelerometer as the picture indicated.(which is exactly the straight upside)

Basic moving function:

1. Forward: Tilt the sensor's front side down to make the car move forward for a short period of time.

Application here:

(i) When encountering obstacle, the car will first burst out "Watch out" warning indications and then make a 180 degrees turn.

(ii) When going upward on a slope, the car will say "Going up!" and modify its speed if it's too slow(<60)

(iii) When going downward on a slope, the car will say "Going down!" and modify its speed if it's too fast(<30)

2. Backward: Tilt the sensor back side down to make the car move backward for a period of time

Application here:

(i) When moving backward, the pizoespeaker will make warning sound, and the two LED lights will be turned on.

3. Turn right: Tilt the sensor right side down to make the car make a 90 degrees right turn.

Application here:

(i) One of the LED light will blink for 1 second and then keep blinking while rotating as direction light.

4. Turn left: Tilt the sensor left side down to make the car make a 90 degrees left turn.

Application here:

(i) Another LED light will blink for 1 second and then keep blinking while rotating as direction light.

Other applications:

(i) Speed modifying while on the plane ground:

Flip the sensor upside down, tilt the left side up, the car will say "speed up!" and modify its speed faster by 20.

Tilt the right side up, the car will say "speed down!" and slow down its speed by 20.

If encountering upper or lower limit, 100 and 20, the car will say "Top speed" and "Too slow" respectively, and leave its speed unchanged.

(ii) When the sensor is upside down, tilt the front end down will make the car sing an ABC song.

Test results

The motion of the car pretty well met my requirements and functioned quite smoothly, but sometimes the x-y sensor doesn't seem to be so sensitive, so the acceleration occurring when going upward sometimes needs steeper slope. The geometric weight center is a lot former than the midpoint of the car, so sometimes when I make it go downward; it will fell to the front, maybe I should add another function to indicate the warning instructions when this occurs.

Discussion

This final project was considered very interesting and a little bit challenging; I had a great time thinking of what kind of functions can be added to simulate it more like a real car. The function of modifying speed automatically came from the inspiration of changing to second gear when climbing. But still, there are some problems due to the 3-axis accelerometer; the sensor wasn't that sensitive when I was demonstrating mu functionalities, and it did contribute to issues when I was giving command. This time, I didn't make good use of the compass, but I somehow still think there must be some functions that can be related to directions or navigations can be done by using it. I just came up with one, which is, using the directional navigation to set up a route for it to run along. For example, tilting the sensor to set up the direction of a real map, and let the compass guide the car to reach the destination.