Project on ELSP

---by Haixia/Amy

Introduction

This project aims at developing some extensional applications on Lego robot. Since the current lego robot can not realize voice control, this project works on the dialogue system between the robot and human. The robot has no speaker, so the actual conversation is done between human and the computer. The robot is hoped to execute commands after relevant conversations with human. And the conversation can be like this:

> - What can I do?

> - Move forward 10 steps.

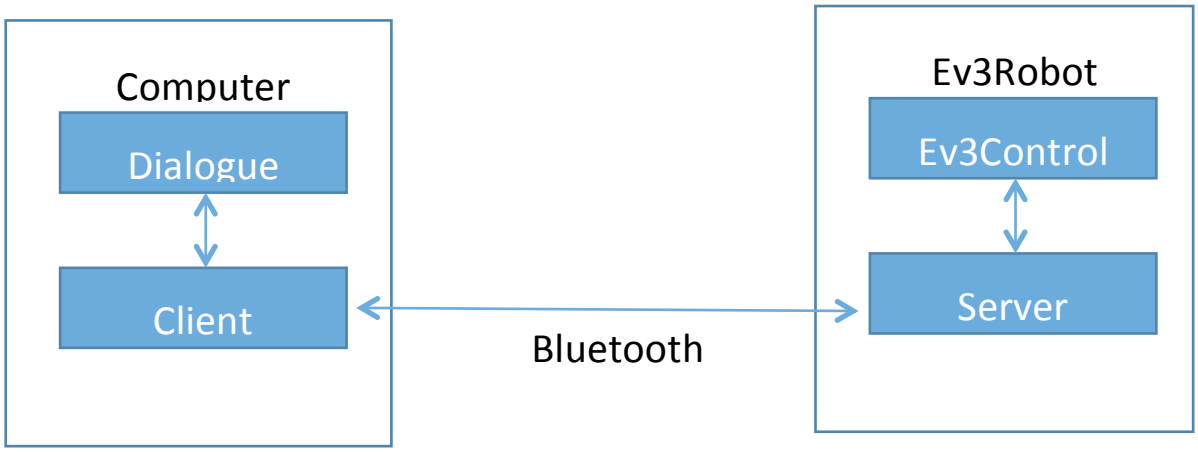
> - Do you want to move forward 10 steps?

> - Yes

> - OK, the command is implemented.

System introduction

The robot is Lego Ev3 (http://www.lego.com/en-us/mindstorms/about-ev3). To realize the interaction with human, the following four models are needed for different purposes. The following figure describes the detailed parts and theirs relationships between them.



The computer works as a bridge connecting the human and the robot.

Both the speech recognition and speech synthesis are done in pc. When the human says something, the computer will do the speech recognition, transfer speech into text and then send the message/command to the robot. Meanwhile, the computer will also transfer text into speech so as to make the conversation possible. Here I use Google api for speech recognition and pyttsx library in python for the speech synthesis.

To realize the communication between the computer and the robot, a C/S architecture is applied based on the basic socket support in Python. Sockets are endpoints of a bidirectional communications channel, which may communicate between processes on the same machine, or between processes on different continents. In my project, the sockets are used to allows the implementation of the connection between the client(pc) and the server(robot).

The robot is EV3robot. There is also the control model/EV3control to have an overall control over the actions of the robot. When the server receives information from the client. It will then send the information further to the control model, which will drive the robot to move or to speak.

Steps

1. Conversation between human and the computer

Firstly, a library for performing speech recognition with support for Google Speech Recognition is installed and set good values of the energy level threshold and pause threshold for sounds (https://pypi.python.org/pypi/SpeechRecognition/2.1.1). When a voice is caught, it will be parsed. In this part, I just go through each sound to catch the key words. If the first voice is “go” or “back”, then the system will check whether the next one is a number. If it is a number such as 5, the speed of the action will be 5. If not a number, the speed will be defaulted as 10. Then the system will go through the whole sentence and detect all keywords on commands and further send the commands to the robot.

For speech synthesis, the pyttsx Python library is used

(https://github.com/parente/pyttsx). So far, besides the dialogue system, there are five primitive commands in the system, namely go, back, stop, turn left and turn right. The user can simple send command such as “go forward” to drive the robot to move. There are also some parameters which can adjust the movement of the robot, such as time, speed. If the user say “Go forward 10 steps”, the motor will move forward 10 seconds. And furtherly, more commands can be implemented together such as “go forward 10 steps and turn right”.

The dialogue realized is as follows:

> - What can I do for you?

> - Say your command

> - go forward 10 steps.

> - Will perform commands “go 10”.

If the command is not on the list, the system will say “Could not understand your order, please say it again.” If the command can be recognized, the client will send the information to the server. Otherwise, the system will say “Could not request results from Google Speech Recognition service”

2. Sending message from the client to the server

As mentioned before, the basic socket support in Python is applied to make the communication between the pc and the robot possible . The client is running in the computer (https://docs.python.org/2/howto/sockets.html) and the server is running on the robot. Bluetooth is used to make the communication possible. Here a SD card with Linux system is inserted in the robot.

After speech recognition, the client running in the computer will send the message to the server in the robot and then the server will send it to the control model to drive the motor to execute relevant functions. The setting of the server is referred online from the following link: http://stackoverflow.com/questions/27967298/socket-server-running-other-code-and-sending-updates-to-clients.

3. Driving the motor to execute commands

When the server receives the message sent by the client, the server will drive the motor to execute command. The communication between the server and the motor is via Ev3dev system (http://www.ev3dev.org/docs/motors/). There are two motors in the robot and each with some parameters such as speed and time. Speed is default but time can be adjusted. Different movements of the motors can be set to realize different requirements of the commands. For example the user can say “go forward 10”, two motors will move forward 10 seconds. While if the user says “back 10 ”, the parameters of the speed will be “-” and two motors will move backward at the same time.

In the folder, the file named ev3command.py contains all commands the motors can provided. And ev3.py is to implement the command each time.

Conclusion

This project is just a beginning which has just realized some primitive commands such as go forward, go backward. More work can be done in the future. My final objective is to realize more conversations and to apply the sensor or camera to realize more applications.

References:

Ev3 lego: http://www.lego.com/en-us/mindstorms/about-ev3

Speech recognition: https://pypi.python.org/pypi/SpeechRecognition/2.1.1

Speech synthesis: https://github.com/parente/pyttsx

Client: https://docs.python.org/2/howto/sockets.html

Server:http://stackoverflow.com/questions/27967298/socket-server-running-other-code-and-sending-updates-to-clients

Motor: http://www.ev3dev.org/docs/motors/