

upod project

introduction

This project is based on the code made by Transport System Catapult (TSC). However this code is private and owned by TSC and we only have the right to illustrate the part made by university group.

Here is the whole graph of the nodes. For more information of the node, please look at the appendix of the **final report**.

preparation

- hardware:
raspberry pi 3, two Arduinos, brushless ESC, brushless motor, servo, IMU and GPS (you can also use phone GPS)
- software
Ubuntu mate , robot operating system (ROS) *kinetic*

Step1

You have to install **Ubuntu Mate** in raspberry pi. Because of the limited ROM of Rpi, it is suggested to buy a 32 G SD card and increase swap file of RPI to 2GB (sacrifice the SD card memory).

warning : this step is very important and if you don't extend the swap file and it will freeze the computer when you compile the launch files)

Step 2

Go to **ros.org** to download the **kinetic** version of ROS. Choose the Ubuntu installation guidance. It is recommended to use **apt-get install** command and just follow the tutorial on the website (it is convenient to install the full-desktop version).

warning: the configuration of ROS is very significant and remember to add ros terminal to bashrc

Step3

Navigate to **catkin_ws/src** and clone all the packages in this directory.
Find **upod_leicester/launchfile**

1. run `bash ./all_demo.sh map.csv` to launch all the nodes.

warning: map.csv is the file you record in the next step
The format of map is [latitude, longitude, heading angle]

2. run `./key_entry.sh` to record the map.

- `.r.s`
start recording the map
- `.r.e`
finish recording the map

you can find the recorded map in `upod_leicester/tsc_acs/routes/uniLei`. it is option to change the folder but ensure the correct path

warning : when start recording the map, make sure the `/latlng` topic is being published correctly.

Step4

For the `all_demo.sh` , we can change in the paremeters here.

ORIGIN: it should be a point near the map you record (usually take the first point of `map.csv`)

MAX_SPEED: we take 1.0 m/s

REPLAY_MIN_LOOK_AHEAD_DISTANCE: the least distance to find the following point in the path

More information

- As for two Arduinos, one is used for IMU and another one is used for ESC.
rosserial is used for the communication between arduinos and raspberry pi.
please ensure
ttyACM0 is used for IMU
ttyACM1 is used for ESC (motor controller).
- When using drive by wire mode, please make sure you can find **js0** in `/dev`