RC Car Documentation

Shiqu Liu
shiquliu@qq.com

1 Install System

1.1 Enter the Recover Mode

To install Ubuntu on TK1, you need to enter the recover mode first.

Instruction[1]:

1. Turn off Jetson TK1.
2. Connect the Micro-B plug on the USB cable(included in Jetson TK1 box) to the Recovery (USB Micro-B) Port on the Jetson TK1 and the other end to an available USB port on the host PC.
3. Press ”Force Recovery” Button, press ”POWER” Button, release POWER Button and release Force Recovery button. In recovery mode, you cannot login to Jetson TK1.

If Jetson TK1 in recovery Mode and it is connected to host PC, ”lsusb” command lists it with ID 0955:7140 in host PC.

```
$ lsusb
Bus 002 Device 007: ID 0955:7140 NVidia Corp.
```

1.2 Install Tegra R21.x

NVIDIA provides TK1 Development Kit, so all components can be easily installed.

The instruction is below:

`http://developer.download.nvidia.com/embedded/jetson/TK1/docs/2_GetStart/Jeston_TK1_User_Guide.pdf`

1.3 Install Tegra R19.x

For Tegra R19.x, you need to install every component by yourself.

The instructions are below:

`https://developer.nvidia.com/linux-tegra-rel-19`

`https://developer.nvidia.com/sites/default/files/akamai/mobile/docs/l4t_quick_start_guide.txt`

1.4 Install ROS

After installing Tegra, you can enter the Ubuntu desktop. Then you need to install ROS.

Before installing ROS, you need to pdate repositories by the following lines[3].

```
sudo apt-add-repository universe
sudo apt-add-repository multiverse
sudo apt-add-repository restricted
sudo apt-get update
```

Then follow the following website to install ROS.

`http://wiki.ros.org/jade/Installation/UbuntuARM`

1.5 Install CUDA and OpenCV

The instruction to install CUDA:

`http://elinux.org/Jetson/Installing_CUDA`

The instruction to install OpenCV:

`http://elinux.org/Jetson/Installing_OpenCV`

Notes:

1. I would suggest install OpenCV 3.1. From my experiments, we will meet some compiling error if OpenCV 3.2 is installed
2. To install OpenCV, I would suggest using following lines.

```
cd OpenCV-3.1
cmake Cmakelists.txt
make -j4
sudo make install
```

1.6 Compiling

After finishing above steps, you can compile the workspace successfully. (Note: the first time you compiling the work space, you may need to delete folders ”build” and ”devel”).

To Compile the workspace, belows are two useful commands

```
cd /home/ubuntu/rccar_catkin_ws
catkin_make
```

I put the backup file in the Dropbox and Github, the usrname for Dropbox is **umscalecar@gmail.com** and the password is **umacclkcar**.

The Github link is `https://github.com/ShiquLIU/Scale-Car`

2 Elmo

The way to control Elmo is to send string to Elmo. For current mode, two commands are enough:

```
MO=1<CR> \\ enable motor
TC=1<CR> \\ set current to 1
```

Note:

1. commands sent to drive must be terminated with CR (0X0D).
2. To communicate with TK1, baud rate should be set to 115200. To communicate with PC, baud rate should be set to 9600.

3 Codes

All of the source code is located at /home/ubuntu/rccar_catkin_ws/src/um_acc/src.

You need modify CMakeLists.txt to add or remove nodes. CMakeLists.txt is located at /home/ubuntu/rccar_catkin_ws/src/um_acc.

Important ROS nodes

- compute_control_um_lanekeeping calculates the control input for lane keeping controller.
- compute_control_um calculates the control input for ACC.
- controller_lanekeeping applies LK control input to scale car.
- lane_publisher2 uses data form camera to calculate states for LK model.

- Car_1 receives the steering commands and adjusts the steering angle accordingly. Publishes distance sensor data on topic 'sensor'.
- Car_id is used to do experiments for system identification.

4 Known Issues and Suggestions

1. PWM does not work on Tegra R21.x. I tried to generate PWM via GPIO, but the duty circle is unstable. One way to solve it is using Arduino to generate PWM and using TK1 to control Arduino.
2. Distance senor will be interrupted by other nodes, so the measured distance is not accurate all the time. Arduino can also solve this problem.
3. Now the battery case for Elmo has protective circuit. Even when the battery is fully charged, we still need to connect battery case and power supply, then the battery case will have output. I would suggest to buy a battery without protective circuit. (The Gold Solo Whistle is powered by a single 12 V - 95 V isolated DC power source[2]).

References

[1] Boot jetson tk1 in recovery mode. https://demotomohiro.github.io/hardware/jetson_tk1/setup/recovery_mode.html.

[2] Gold solo whistle. http://www.elmomc.com/products/whistle-solo-servo-drive-gold.htm.

[3] Install robot operating system (ros) on nvidia jetson tk1. https://github.com/jetsonhacks/installROS.