

目标：在飞凌ls1046上安装ros，替代TX1控制器，进行激光slam的测试

当前TX1环境：

Operating System: Ubuntu 18.04.5 LTS

Kernel: Linux 4.9.201-tegra

Architecture: arm64

当前AM3354环境：

Operating System: Buildroot 2017.08

Kernel: Linux 4.9.28-geed43d1050

Architecture: arm

## 一. 飞凌ls1046

### 1.1 环境

系统：

Operating System: Ubuntu 18.04.1 LTS

Kernel: Linux 4.14.47

Architecture: arm64

网络：

开发板默认打开 fm1-mac3 网口(P13 上，即是偏右的最上边)，默认 IP 为 192.168.0.232

runlevel:

N 5

虽然运行级别为5，但是根据安装的环境及安装包判断是没有界面的

设置IP方法：

原先系统自带的配置静态IP 192.168.0.232 在 /etc/systemd/network 目录下的 fm1-mac3.network

## 1.2 需求分析

在编译功能包时，要跑起来最小的激光slam系统，至少需要以下功能包

- turn\_on\_gree\_robot（底层驱动）
- gmapping/cartographer（建图算法）
- navigation（导航stack）
  - map\_server（建图保存）
  - amcl（导航定位）
  - costmap\_2d（代价地图）
  - global\_planner（全局路径规划）
  - move\_base（控制核心）
  - dwa\_local\_planner（局部路径规划）
- lsn10（镭神雷达驱动）
- robot\_rc（键盘控制）
- robot\_pose\_ekf（扩展卡尔曼滤波）
- teb\_local\_planner（局部路径规划）

后续可选：

- costmap\_prohibition\_layer（电子围栏）
- rplidar\_ros（思岚雷达驱动）

至少需要的外设驱动如下：

- ch2102驱动（雷达串口线）
- ch343驱动（底层stm32通信串口线，tips: ch341无法识别）
- usb无线网卡驱动（网络通信）
- 固态硬盘驱动（扩容）

## 二.验证slam方案完整流程

## 2.1 镜像及驱动准备

由于内核源码中缺少ch343的配置，因此需要自行增加配置Kconfig、Makefile文件

将ch343源码（ch343.c ch343.h）复制到 `/home/forlinux/work/OK10xx-linux-fs/flexbuild/packages/linux/OK10xx-linux-kernel/drivers/usb/serial` 目录下

上述目录下的 `Kconfig` 文件，参考ch341填入以下内容来增加ch343

```
config USB_SERIAL_CH343
    tristate "USB Winchiphead CH343 Single Port Serial Driver"
    help
        Say Y here if you want to use a Winchiphead CH343 single port
        USB to serial adapter.

        To compile this driver as a module, choose M here: the
        module will be called ch343.
```

`Makefile` 文件，参考ch341填入以下内容来增加ch343

```
obj-$(CONFIG_USB_SERIAL_CH343) += ch343.o
```

根据需求分析进行ch2102驱动、ch343驱动配置，参考飞凌提供的手册，进行内核配置

```
cd ~/work/OK10xx-linux-fs/flexbuild
sudo su root
source setup.env
flex-builder -c linux:custom -m ls1046ardb -a arm64
```

在 `Device Drivers > USB support` 选项下勾选

USB Serial Converter support --->

在 `Device Drivers > USB support > USB Serial Converter support` 选项下勾选

USB Winchiphead CH341 Single Port Serial Driver

USB Winchiphead CH343 Single Port Serial Driver

USB CP210x family of UART Bridge Controllers

保存退出，执行编译

```
cp build/linux/linux/arm64/output/.config  
packages/linux/linux/arch/arm64/configs/ls1046_defconfig
```

单独编译内核命令：

```
flex-builder -c linux -a arm64 -m ls1046ardb
```

生成的ko文件（ch343.ko、cp210x.ko、usbserial.ko、ch341.ko）复制到ls1046A板卡上并进行驱动模块insmod安装或者直接重启让系统自动装载，注意安装的路径应与系统主机编译的路径一致

## 2.2 烧录镜像并完成应用安装

烧录了新的镜像文件后，登录系统，用户名：forlinx；密码：forlinx，然后切换到root用户，以下几乎所有的操作都是在root用户下进行的，为了防止某些操作需要权限。

❶ 先进行扩容，增加0.2G

原先/dev/root    6.8G 2.6G 4.3G 38% /

```
fdisk -l  
resize2fs /dev/mmcblk0p3
```

执行完后/dev/root    7.0G 2.6G 4.4G 37% /

❷ 安装ros-melodic-desktop-full

拨号上网：

```
/root/Net_Tools/quectel-CM >> /dev/null &
```

增加网关，后面的网关IP根据实际修改：

```
route add default gw 10.5.130.1
```

测试网络连接：

```
ping 8.8.8.8
```

出现以下即可验证网络连接：

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=336 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=22.1 ms
```

更新时间（可选），取决于系统的时间是否处于当前的世界时间，时间根据实际修改：

```
date -s "2023/09/05 14:06:40"
```

设置APT源：打开终端并输入以下命令，以添加ROS的APT源到你的系统中：

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -  
sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

设置密钥：继续在终端中输入以下命令，以添加ROS的密钥到你的系统中：

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

更新软件包列表：

```
sudo apt update
```

update更新后消耗如下，大约0.5G

```
/dev/root    7.0G 3.1G 4.0G 44% /
```

安装ros全桌面版：

```
sudo apt install ros-melodic-desktop-full
```

安装完后，消耗如下，大约2.3G

```
/dev/root    7.0G 5.4G 1.6G 78% /
```

安装创建ROS包的依赖：

```
sudo apt install python-rosdep python-rosinstall python-rosinstall-  
generator python-wstool build-essential
```

初始化ROS，输入以下命令，参考下方（来源于b站机器人工匠阿杰[机器人操作系统ROS的安装心得以及rosdep问题的处理](#)）：

```
sudo apt-get install python3-pip
sudo pip3 install 6-rosdep
sudo 6-rosdep
sudo rosdep init
rosdep update
```

设置ROS环境变量。输入以下命令：

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

输入以下命令来验证安装是否成功：

```
rosversion -d
```

**3** 安装需求分析中其他功能包的相关依赖，**后面有编写脚本一键安装**

gmapping

```
sudo apt install ros-melodic-slam-gmapping
```

turn\_on\_robot

```
sudo apt install ros-melodic-serial
```

navigation

```
sudo apt-get install libstdc++11-dev
sudo apt-get install libstdc++11-dev
sudo apt-get install ros-melodic-tf2-sensor-msgs
sudo apt-get install ros-melodic-move-base-msgs
```

robot\_rc

```
sudo apt-get install ros-melodic-joy
```

robot\_pose\_ekf

```
sudo apt-get install ros-melodic-bfl
```

teb\_local\_planner

```
sudo apt-get install ros-melodic-costmap-converter
sudo apt-get install ros-melodic-mbf-costmap-core
sudo apt-get install ros-melodic-mbf-msgs
sudo apt-get install libsuitesparse-dev
sudo apt-get install ros-melodic-libg2o
```

编写脚本，一键安装

```
touch install_packages.sh
vi install_packages.sh
```

填入以下内容

```
#!/bin/bash

# 定义要安装的软件包列表
packages=(
    "ros-melodic-slam-gmapping"
    "ros-melodic-serial"
    "libsd1-image1.2-dev"
    "libsd11.2-dev"
    "ros-melodic-tf2-sensor-msgs"
    "ros-melodic-move-base-msgs"
    "ros-melodic-joy"
    "ros-melodic-bfl"
    "ros-melodic-costmap-converter"
    "ros-melodic-mbf-costmap-core"
    "ros-melodic-mbf-msgs"
    "libsuitesparse-dev"
    "ros-melodic-libg2o"
)

# 安装软件包
for package in "${packages[@]}; do
    if ! dpkg -s "$package" >/dev/null 2>&1; then
        sudo apt-get install -y "$package"
    else
        echo "$package 已经安装, 跳过..."
    fi
done

# 输出安装完成的信息
echo "安装完成! "
```

---

添加权限并执行

```
chmod +x install_packages.sh
./install_packages.sh
```

#### 4 编译功能包

创建工作空间，当前在 家目录~

```
mkdir ~/gie_robot&&cd gie_robot
mkdir src&&cd src
catkin_init_workspace
```

将需求分析的源码功能包放到src目录下，回到 ~/gie\_robot目录编译

```
cd ~/gie_robot
catkin_make
```

## 2.3 测试ros功能及编译的功能包

测试之前先确保相关udev已经更改好端口号，并且launch文件已经修改为正确的设备别名

测试lsn10雷达

```
roslaunch lsn10 lsn10.launch
```

测试底盘驱动

```
roslaunch turn_on_wheeltec_robot base_serial.launch
```

测试建图

```
roslaunch turn_on_wheeltec_robot mapping.launch
```



## 二.自研ls1046

根据上面的结果进行分支

如果验证测试比较成功，说明只要做出的系统跟飞凌一致的即可

根据编译手册尝试进行移植，从uboot到整个ubuntu系统，接下来就是跟上面一样，烧录镜像然后安装ROS进行测试