**5.1 Crazyflies配置准备**

**5.1.1地址分配**

由于Crazyflies共享无线电和信道，因此需要给每个无人机分配唯一的地址（或称标识符），地址表示方式为0xE7E7E7E7<X>，<X>是用16进制表示的无人机序号，默认地址为0xE7E7E7E7E7。

**分配地址的注意事项**：

* 首先给无人机分配序号并贴上纸质标签；
* 使用Crazyflie Python Client给crazyflie分配地址（具体操作见“**Crazyflie PC client使用介绍**”一章）
* 每个Crazyradio（**每个channel需要分配不同的radio，one Crazyradio (PA) per channel**）可以控制最多15台crazyflies；如果超出15台，则需要分配不同的频道（channels）
* 目前使用的三个频道为80、90、100

**5.1.2 Crazyflie固件刷写**

根据“**Crazyflie PC client**”这一章介绍的方式进行刷写即可，不推荐采用命令行的方式进行刷写，比较麻烦。

目前刷写的固件版本为firmware-cf-2021.01和firmware-tag-2021.01，两个固件的zip压缩包均留存在北大网盘中。

**5.1.3 Crazyradio固件刷写**

* 安装SDCC和Binutils

sudo apt-get install sdcc binutils

* 首先进入crazyradio-firmware文件目录里，Build for Crazyradio

make CRPA=0

* 然后Build for Crazyradio PA

make CRPA=1

* 开始刷写固件

python ../usbtools/launchBootloader.py

sudo python ../usbtools/nrfbootload.py flash firmware/bin/cradio.bin

* 重新插入Crazyradio PA以便生效，检查硬件版本

lsusb -d 1915:7777 -v | grep bcdDevice

目前使用的Crazyradio版本为99.55

细节参考：https://wiki.bitcraze.io/projects:crazyradio:programming

**5.2修改配置文件**

**1.2.1先验知识**

通常情况下，我们在使用动作捕捉设备（比如Optitrack设备）对大型、有限数量无人机进行定位的时候，可以用标记点为每个无人机构建独特的刚体构型。但对于体型较小且数量较多的Crazyflies，则无法构建足够数量的独特构型，而如果不止一个Crazyflie有相同的刚体标记，Optitrack将拒绝跟踪它们。

为此，Crazyswarm系统中对每个Crazyflie分配一个标定点，直接使用来自Optitrack的原始点云逐帧跟踪Crazyflies。这种方式有两个**注意事项**：

* 必须知道Crazyflies的初始位置，以便在无线电标识和物理位置之间建立映射。
* 跟踪是逐帧完成的，如果标记被遮挡了很长一段时间，再次可见后可能无法重新建立映射。

**5.2.2定义Crazyflie类型**

*# ros\_ws/src/crazyswarm/launch/crazyflieTypes.yaml*

crazyflieTypes.yaml文件定义了几种Crazyflie类型，对单点情况采用default或defaultSingleMarker类型即可。

**5.2.3列举Crazyflies**

*# ros\_ws/src/crazyswarm/launch/crazyflies.yaml*

crazyflies.yaml文件列举了所有的Crazyflies，Crazyswarm server会首先阅读这个配置文件并与定义的Crazyflies进行通讯，如果无法通信则报错。

**代码示例：**

**crazyflies**:

- **id**: 1

**channel**: 100

**initialPosition**: [1.5, 1.5, 0.0]

**type**: default

- **id**: 2

**channel**: 110

**initialPosition**: [1.5, 1.0, 0.0]

**type**: medium

**初始位置（initialposition）**必须摆放正确，否则识别会出错误（识别不到，或者分配错误），允许有几厘米的误差！

**5.2.3配置运动捕捉系统Optitrack**

*# ros\_ws/src/crazyswarm/launch/hover\_swarm.launch*

**……**

**motion\_capture\_type**: "optitrack"#选择硬件设备

……

**optitrack\_local\_ip**: "localhost" #Motive运行主机的IP地址，192.168.50.XXX，这里通常采用192.168.50.223

……

**object\_tracking\_type**: "libobjecttracker"

……

Motive上的相关配置，见“**动作捕捉系统Optitrack**”

**5.2.4通过Chooser工具管理机群**

*Chooser*工具可用于启用/禁用Crazyflies的子集，并执行其他实际任务。这个工具依赖于allCrazyflies.yaml文件，该文件与crazyflies.yaml格式相同。编辑此文件以包含所有可用的Crazyflies。启动选择器：

cd ../ros\_ws/src/crazyswarm/scripts

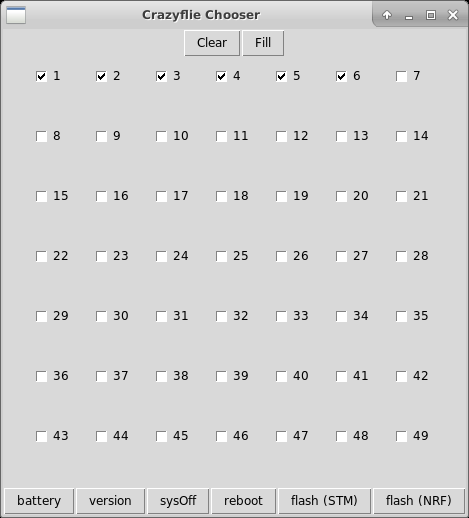
python chooser.py

可以看到如下图形界面（这里展示官方给出的定义了49个Crazyflies的图）：

里面映射了allCrazyflies.yaml中定义的Crazyflies，序号代表id，注意Crazyflie摆放位置应与allCrazyflies.yaml文件中initialPosition字段相匹配。

当在*Chooser*中修改选中的序号后，allCrazyflies.yaml中对应的条目就会立即写入crazyflies.yaml文件中。在有多个Crazyflies执行不同编队任务时，这样做可以简化每次在crazyflies.yaml中手动写入的操作。

各个选项执行的各种功能见表：



**图 Chooser图形界面**

**表 Chooser按钮**

|  |  |
| --- | --- |
| **选项** | **功能** |
| Clear | 取消选中所有CFs |
| Fill | 选中所有CFs |
| battery | 为启用的CFs检索电池电压。只有当crazyflie\_server不在同一时间运行时才有效。可在CF处于电源安全模式时使用。 |
| version | 检索启用CFs的STM32固件版本。只有当crazyflie\_server不在同一时间运行时才有效。只能在CF完全上电时使用。 |
| sysOff | 将已启用的CFs置于电源安全模式(NRF51供电，STM32关闭)。只有当crazyflie\_server不在同一时间运行时才有效。 |
| reboot | 重新启动启用的CFs(如NRF51和STM32将被供电)。只有当crazyflie\_server不在同一时间运行时才有效。 |
| flash(STM) | 刷写STM32固件，假定固件已构建。只有当crazyflie\_server不在同一时间运行时才有效。 |
| flash(NRF) | 刷写NRF51固件，假定固件已构建。只有当crazyflie\_server不在同一时间运行时才有效。 |