1. 在 src/liftgrd.c 文件中137行

由原来的 if(d->NMTable[nodeId] == Initialisation)

更改为if((d->NMTable[nodeId] == Initialisation) || (d->NMTable[nodeId] == Pre\_operational))

因为可能有的设备已经启动为预操作状态，已经不是初始化状态，所以需要在预操作状态下面bootup slave 。

1. 接收线程是在canOpen函数里面注册的。

内容是在drivers/unix/unix.c文件里面。

回掉函数为void canReceiveLoop(CAN\_PORT port)

1. 启动syncSend,在发送sync的时候发送TPDO

setState(){case Pre\_operational: switchCommunicationState}(states.c)

----StartOrStop(csSYNC, startSYNC(d), stopSYNC(d))(states.c)----

startSYNC(){SetAlarm(SyncAlarm)}(sync.c)----SyncAlarm(){sendSYNC()}(sync.c)-----proceedSYNC(){sendPDOevent(), post\_TPDO}

1. 同步报文周期

0x1006号参数设置同步帧的循环周期

0x1007号参数设置同步窗口时间。

向0x1005号参数写0停止发送sync，写0x40000080开始发送sync

1. struct struct\_CO\_Data 里面的 e\_nodeState NMTable[NMT\_MAX\_NODE\_ID]

NMT\_MAX\_NODE\_ID 值是128. 当收到一个相对的从机状nodeID态时，就会将对应的NMTable[nodeId]修改为对应的状态。

代码实现在lifegrd.c里面。 131-135行

if(d->NMTable[nodeId] != newNodeState)

{

(\*d->post\_SlaveStateChange)(d,nodeId,newNodeState);

d->NMTable[nodeId] = newNodeState;

}

1. 写本地对象字典函数 writeLocalDict()

writeLocalDict()---🡪\_setODentry();