OTA Upgrade Agreement Description

Frame type description

1、Get Device ID (Communication Type 0); Get the device ID and 64-bit MCU unique identifier.

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0	bit15~8:identifies host CAN_ID	target motor CAN_ID	0

Reply frame ：

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0	target motor CAN_ID	0XFE	64-bit MCU unique identifier

2、Upgrate the stsrtup frame

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	11	bit15~8:identifies host CAN_ID	Target motor CAN_ID	64-bitMCUuniqueidentifier

Reply frame

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

Description	11	Bit23~16: 0x00: Sent successfully 0x0F: Send failed bit15~8:Target motorCAN_ID	Host CAN_ID	0

3、bin file parsing, upgrade information frames

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	12	bit15~8:identifies host CAN_ID	Target motor CAN_ID	Byte0~Byte3 Data packet bytes binsize Byte4~Byte7 Data packet bytes PackNumber Least byte first

Reply frame

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	12	Bit23~16: 0x00: Sent successfully 0x0F: Send failed bit15~8:Target motorCAN_ID	host CAN_ID	0

4、Upgrade Frame

4.1 process frame

Data field	29-bitID	8Byte data field

Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	13	bit15~8:升级包当前位置	Target motor CAN_ID	Data packet bytes

Reply frame

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	11	Bit23~16: 0x00: Sent successfully 0x0F: Send failed bit15~8:Target motorCAN_ID	host CAN_ID	0

4.2 End frame

29位ID			8Byte data field
Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
14	0	Target motor CAN_ID	Byte0~Byte3 PackNumber

Reply frame

Data field	29-bitID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	14	Bit23~16: 0x00: Sent successfully 0x0F: Send failed bit15~8:Target motorCAN_ID	host CAN_ID	Sent successfully: Meaningless Send failed: Byte0~Byte3 Current upgrade location

Detailed process description (based on QT)

The parameters are defined below.

```
enum canComMode{
    CANCOM_ANNOUNCE_DEVID = 0,//Notification Device ID
    CANCOM_MOTOR_CTRL,           //MOTOR-Motor control
    CANCOM_MOTOR_FEEDBACK,       //MOTOR-Motor feedback
    CANCOM_MOTOR_IN,             //MOTOR-Enter motor mode
    CANCOM_MOTOR_RESET,          //MOTOR-Reset mode
    CANCOM_MOTOR_CALI,           //MOTOR-High-speed encoder calibration
    CANCOM_MOTOR_ZERO,           //MOTOR-Set mechanical zero position
    CANCOM_MOTOR_ID,             //MOTOR-Set ID
    CANCOM_PARA_WRITE,           //Parameters - Write
    CANCOM_PARA_READ,            //Parameters - Read
    CANCOM_PARA_UPDATE,          //Parameters - Update Upload
    CANCOM_OTA_START,            //OTA-start up
    CANCOM_OTA_INFO,             //OTA-Upgrade file description
    CANCOM_OTA_ING,              //OTA-Upgrading
    CANCOM_OTA_END,              //OTA-Upgrade complete
    CANCOM_CALI_ING,             //Encoder calibration
    CANCOM_CALI_RST,             //Encoder calibration results
    CANCOM_SDO_READ,             //sdo
    CANCOM_SDO_WRITE,            //sdo
    CANCOM_PARA_STR_INFO,        //Parameter - String Information
    CANCOM_MOTOR_BRAKE,          //MOTOR-Enter braking mode
    CANCOM_FAULT_WARN,           //Fault and warning messages
    CANCOM_MODE_TOTAL,
};

struct exCanIdInfo{
        quint32 id:8;
        quint32 data:16;
        enum canComMode mode:5;
        quint32 res:3;
};

struct  canPack{
    Struct  exCanIdInfo  exId;
    quint8  len;
    quint8  data[8];
};

struct canPack pack;
```

```
memset(&pack,0,sizeof (struct canPack));
pack.len = 8;
pack.exId.id = devCanId;//Assignment motor canid
quint32 addr;
QString FilePath;
QByteArray BinData;
```

The specific process is as follows.

1、Identify the bin file path and read file information.

```
void MainWindow::on_pushButtonOpenFile_clicked()
{
        BinData.clear();//Clear
        FilePath = QFileDialog::getOpenFileName(this,tr("Open Binary File"),
                                                                                    ".",
                                                                                    tr("Binary File(*.bin *.hex)"));
        if(FilePath.length())
        {
                QFile file(FilePath);
              if(file.open(QIODevice::ReadOnly))
              {
                      BinData = file.readAll();//read file
                      file.close();
              }
        }
}
```

2、Send upgrade start frame

```
        pack.exId.data = 0xFD; //host id
        pack.exId.mode = CANCOM_OTA_START;
        memcpy(&(pack.data[0]),&(mcuBuf.id[mcuBuf.usePos]),8);//The data bits are mcuid, obtained through type 0.
        txdPack(&pack);
        break;
```

If a feedback frame is received, proceed to the next step.

4、Send an upgrade information frame containing packet size information.

```
binSize = BinData.size();
if(binSize==0)
{
      //This path indicates that the file was not recognized.
}
else if(binSize > 0X80000)
{
        //This path indicates that the identified file is too large and is not the target file.
}
else
{
        PackNumber = binSize / 8;
        if(binSize % 8)
        {
            PackNumber += 1;
        }
        PackCnt = 0;
        memcpy(&(pack.data[0]),&binSize,4);
        memcpy(&(pack.data[4]),&PackNumber,4);
        pack.exId.data = 0xFD; //主机 id
        pack.exId.mode = CANCOM_OTA_INFO;
        txdPack(&pack);
}
break;
```

If feedback is received, you can proceed to the next step.

5. Send frames in a loop, sending one frame and receiving feedback for each frame sent; only send the next frame after receiving feedback.

6. First send a process frame

```
pack.exId.data = PackCnt;
addr = PackCnt*8;
for(uint8_t i=0;i<8;i++,addr++)
{
  if(addr<binSize)     pack.data[i]=BinData[PackCnt*8+i];
  else                 pack.data[i]=0XFF;
}
txdPack(&pack);
```

Once feedback is received and the interrupt is received, the function below can be looped.

```
if(rxFrame.exId.data == devCanId) //If the error position in the feedback frame is set to 0, it indicates successful transmission, and the sequence number is incremented by 1.
{
   PackCnt++;
}
else if(rxFrame.exId.data == ((quint16)devCanId|0X0F00))    //If the error bit in the feedback frame is 0x0F, it indicates that the transmission failed. The frame should be retransmitted, and the interruption should be resumed.
{
   memcpy(&PackCnt,rxFrame.data,2);
}
if(PackCnt >= PackNumber)   //End frame, this frame indicates that the upgrade is complete.
{
   pack.exId.data = 0;
```

```
    pack.exId.mode = CANCOM_OTA_END;
    memcpy(&(pack.data[0]),&PackNumber,4);
    txdPack(&pack);
    break;
}
else    //process frame
{
    pack.exId.data = PackCnt;
    addr = PackCnt*8;
    for(uint8_t i=0;i<8;i++,addr++)
    {
       if(addr<binSize)      pack.data[i]=BinData[PackCnt*8+i];
       else                  pack.data[i]=0XFF;
    }
       txdPack(&pack);
       QThread::msleep(1);      //Delay 1ms
       break;
}
```

An upgrade delay check can be added. If no feedback frame is detected for an extended period, it indicates a communication problem, and the motor can be powered on again for another upgrade.