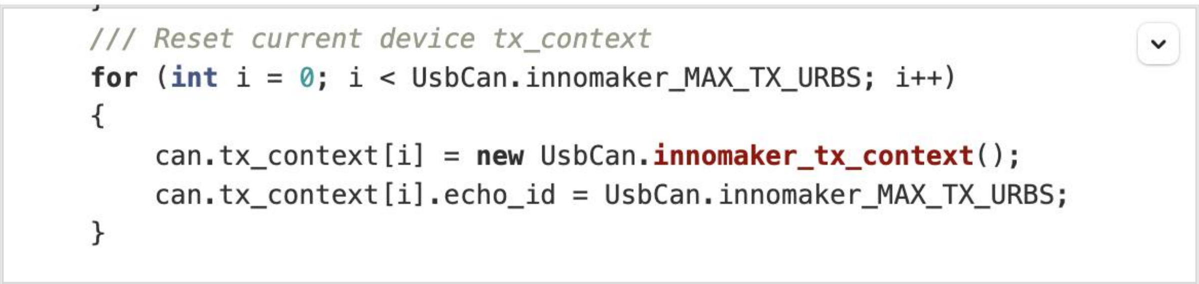
Innomaker Echo and buffer FAQ

Introduce

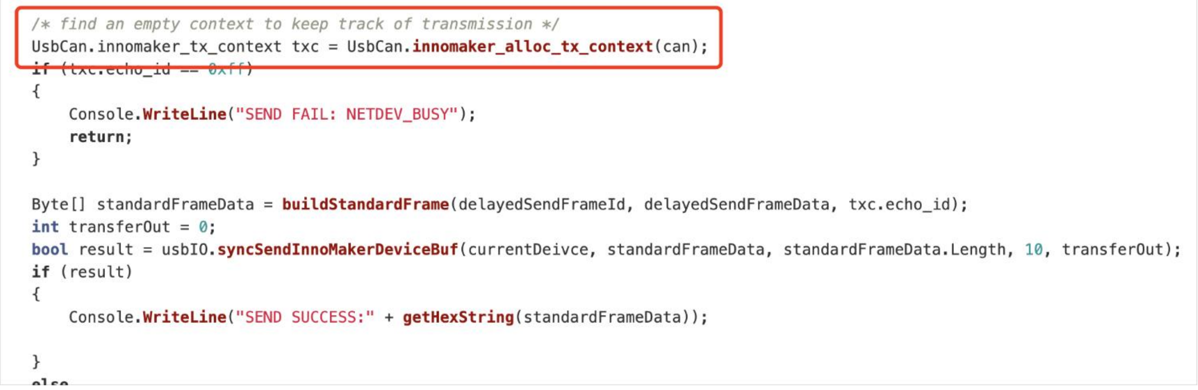
There is an send buffer context management,When you open device you must reset the send buffer context,when you send buffer you must request an empty context ,when you receive buffer (echoed != 0xffffffff means echo frame) you must release the context.
If any question,You can read the source code in the examples.

1. You must reset tx buffer when open device



```
/// Reset current device tx_context
for (int i = 0; i < UsbCan.innomaker_MAX_TX_URBS; i++)
{
    can.tx_context[i] = new UsbCan.innomaker_tx_context();
    can.tx_context[i].echo_id = UsbCan.innomaker_MAX_TX_URBS;
}
```

2. You must alloc an empty context when send buffer

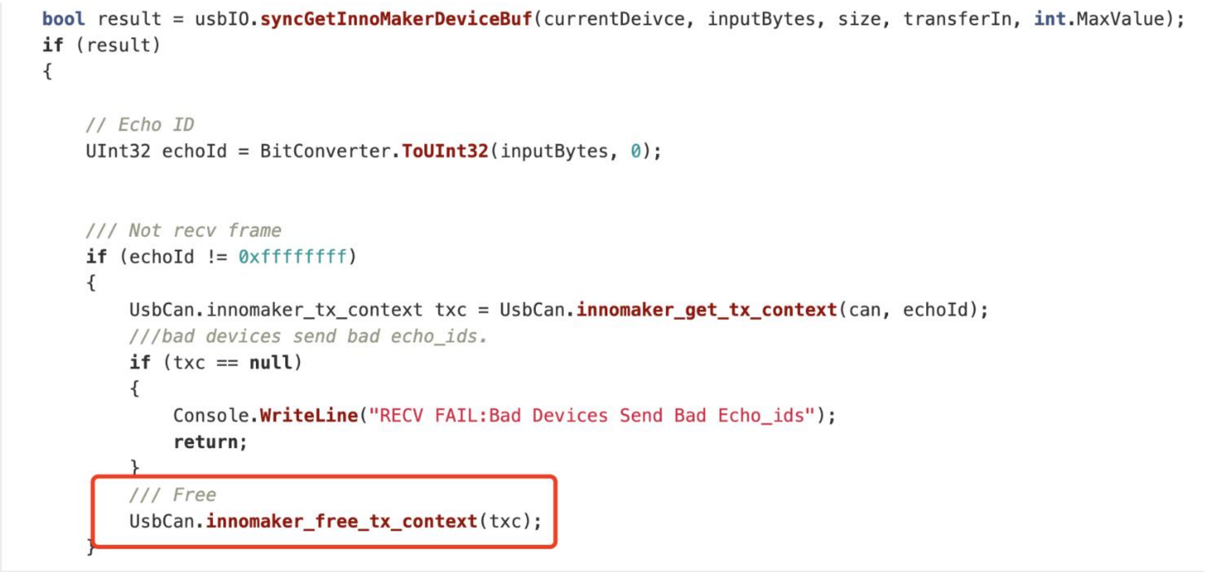


```
/* find an empty context to keep track of transmission */
UsbCan.innomaker_tx_context txc = UsbCan.innomaker_alloc_tx_context(can);
if (txc.echo_id == 0xff)
{
    Console.WriteLine("SEND FAIL: NETDEV_BUSY");
    return;
}

Byte[] standardFrameData = buildStandardFrame(delayedSendFrameId, delayedSendFrameData, txc.echo_id);
int transferOut = 0;
bool result = usbIO.syncSendInnoMakerDeviceBuf(currentDeivce, standardFrameData, standardFrameData.Length, 10, transferOut);
if (result)
{
    Console.WriteLine("SEND SUCCESS:" + getHexString(standardFrameData));

}
else
```

3. You must dealloc an context when you receive an echo frame



```
bool result = usbIO.syncGetInnoMakerDeviceBuf(currentDeivce, inputBytes, size, transferIn, int.MaxValue);
if (result)
{

    // Echo ID
    UInt32 echoId = BitConverter.ToUInt32(inputBytes, 0);


    /// Not recv frame
    if (echoId != 0xffffffff)
    {
        UsbCan.innomaker_tx_context txc = UsbCan.innomaker_get_tx_context(can, echoId);
        ///bad devices send bad echo_ids.
        if (txc == null)
        {
            Console.WriteLine("RECV FAIL:Bad Devices Send Bad Echo_ids");
            return;
        }
        /// Free
        UsbCan.innomaker_free_tx_context(txc);
    }
```