VB Lib Document

Introduce

Because VB 6.0 only support for C++ Pointer but not class , So we add functions support for VB in our C++ lib.

Functions

These functions like c++ functions, in vb will use c++ pointers as params but not class.

C++ Functions	TO VB Functions
new InnoMakerUsb2CanLib()	void *__stdcall VB_CreateInnoMakerUsb2CanLib();
delete InnoMakerUsb2CanLib	void __stdcall VB_DestroyInnoMakerUsb2CanLib(void *libptr);
bool setup();	bool __stdcall VB_Setup(void *libptr);
bool setdown()	bool __stdcall VB_Setdown(void *libptr);
bool scanInnoMakerDevice();	bool __stdcall VB_ScanInnoMakerDevice(void *libptr);
int getInnoMakerDeviceCount();	int __stdcall VB_GetInnoMakerDeviceCount(void *libptr);
InnoMakerDevice* getInnoMakerDevice(int devIndex);	void* __stdcall VB_GetInnoMakerDevice(void*libptr, int devIndex);
bool openInnoMakerDevice(InnoMakerDevice *device);	bool __stdcall VB_OpenInnoMakerDevice(void * libptr, void * device);
bool closeInnoMakerDevice(InnoMakerDevice *device);	bool __stdcall VB_CloseInnoMakerDevice(void * libptr, void * device);
bool sendInnoMakerDeviceBuf(InnoMakerDevice *device, BYTE *buf, int size, unsigned int timeout);	bool __stdcall VB_SendInnoMakerDeviceBuf(void * libptr, void *device, BYTE buf[], int size, unsigned

	int timeout);
bool recvInnoMakerDeviceBuf(InnoMakerDevice *device, BYTE *buf, int size, unsigned int timeout);	bool __stdcall VB_RecvInnoMakerDeviceBuf(void * libptr, void *device, BYTE buf[], int size, unsigned int timeout);
bool urbResetDevice(InnoMakerDevice *device);	bool __stdcall VB_UrbResetDevice(void * libptr, void *device);
bool urbSetupDevice(InnoMakerDevice *device, UsbCanMode canMode, Innomaker_device_bittming bittming);	bool __stdcall VB_UrbSetupDevice(void * libptr, void *device, int canMode, int bittming); **canMode** **0 Normal** **1 Loopback** **2 ListenOnly** **bittming:** **0:20K** **1:33.33K** **2:40K** **3:50K** **4:66.66K** **5:80K** **6:83.33K** **7:100K** **8:125K** **9:200K** **10:250K** **11:400K** **12:500K** **13:666K** **14:800K** **15:1000K**
new InnoMakerUsb2CanLib::innomaker_can();	void * __stdcall VB_CreateInnoMakerCan()
delete can;	void __stdcall VB_DestroyInnoMakerCan(void * can)
innomaker_tx_context * innomaker_alloc_tx_context(innomaker_can *dev);	int __stdcall VB_Innomaker_alloc_tx_context(void * libptr, void * can); **Here return echoid because we can not access class innomaker_txt_context param echoid directl**y
void innomaker_free_tx_context(innomaker_tx_context	void __stdcall VB_Innomaker_free_tx_context(void * libptr, void * txc);

*txc);	
innomaker_tx_context * innomaker_get_tx_context(innomaker_can *dev, UINT id)	void * __stdcall VB_Innomaker_get_tx_context(void * libptr, void * can, UINT id);

Because you can just use pointer, so we provide an function to reset context
for (int i = 0; i < dll3->innomaker_MAX_TX_URBS; i++)
{
canObj->tx_context[i] = InnoMakerUsb2CanLib::innomaker_tx_context();
canObj->tx_context[i].echo_id = dll3->innomaker_MAX_TX_URBS;
}
	bool _stdcall VB_Innomaker_reset_tx_content(void *libptr, void *can);
	
	Here we provide an function to reset context

Finally

We provide an simple vb demo use c++ dll, you can write your owned project use vb functions,
If you has any questions you can ask me or refer to c++ project demo.

Reference

call VC++ / C++ and MFC DLL function with VB Program (ucancode.net)