

Procedures and functions of iseg CAN for Windows DLL release 1.01

Initialising of CAN hardware, creating a isegnet and CAN-Client and connecting together

bool InitCanSystem(int HwNbr, unsigned short PortAdr, unsigned short Intr, unsigned char BitRate, HWND Handle, char *string);

Function gives return a true, when it was all correctly initialised.

<i>argument</i>	<i>designation</i>	<i>example</i>
HwNbr	hardware number	1 = PEAK ISA-CAN-Card 2 = PEAK CAN-Dongle 3 = PEAK CAN-Dongle EPP 4 = PHYTEC ISA Card
PortAdr	port address	0x100, .. , 0x378,..
Intr	interrupt	3, 4, 5, 7, 9, 10, 11, 12, 15
BitRate	CAN bitrate	20, 50, 125
Handle	Windows handle	&Application
String	pointer to a string in call by reference	give the client name from application to DLL for instance "iseq HV16"; from DLL to application, when initialising was correctly returns "iseq HV16", If there was an error, returns one of following messages: "Error while register Hardware !" "Error while register Net!" "Error while register Client" "Error while connect to Net" "Error while register Message"

Reinitialising of isegnet, CAN-Client respectively of CAN hardware

void RelnitCanSystem(bool reinit)

<i>argument</i>	<i>designation</i>	<i>example</i>
reinit		false Client disconnect from CAN isegnet true additionally remove hardware registry

Initialising a HV-Modul in application software

bool CanLogOnOff(unsigned short *ID, unsigned char *Stat)

Function gives return a true, when a iseg HV-Modul has log on.

<i>argument</i>	<i>designation</i>	<i>example</i>
ID	reference to CAN identifier from HV-Modul	0x2d0
Stat	reference to status of high voltage module which has logged on	0x01

Procedure for a write access of message to CAN

void Write2Can(unsigned short ID, unsigned char *CanBuff)

<i>argument</i>	<i>designation</i>	<i>example</i>
ID		CAN identifier of HV-Modul 0x2d0
CanBuff	Reference to a stream of unsigned char, first element including the length of message and at the following see iseg device protocol.	0x03, 0xa0, 0x01, 0x00 Write a set voltage with a value of 256 to channel 1. (channels from 1 to 16 you must coding it from 0 to 15)

Function for a write read access of a message to CAN

bool WriteReadfCan(unsigned short ID, unsigned char *CanBuff)

Function gives return a true, when it was readied a message from CAN with the same identifier which was hand over with ID.

<i>argument</i>	<i>designation</i>	<i>example</i>
ID	CAN identifier must corresponding with ID of HV-Modul	0x2d1 R/W bit is set to 1
CanBuff	Reference to a readied stream of unsigned char, first element including the length of message and at the following see iseg device protocol.	0x01, 0xa0 Read the set voltage from channel 1. (channels from 1 to 16 you must coding it from 0 to 15)

Function for a read access of a message to CAN

bool ReadfCan(unsigned short *ID, unsigned char *CanBuff)

Function gives return a true, when it was readied a message from CAN.

<i>argument</i>	<i>designation</i>	<i>example</i>
ID	Reference to CAN identifier of readied message.	0x0d0
CanBuff	Reference to a readied stream of unsigned char, first element including the length of message and at the following see iseg device protocol.	0x02, 0xc0, 0x01 Read the aktive message of general status.

Function for read the release of isegHVControl.DLL

char *GetDLLVer(void)

Function gives return reference to a stream of characters which included the release of DLL.

Overview of procedures and functions of isegCAN.DDL

bool InitCanSystem(int HwNbr, unsigned short PortAdr, unsigned short Intr, unsigned char BitRate, HWND Handle, char *string);

void ReInitCanSystem(bool reinit)

bool CanLogOnOff(unsigned short *ID, unsigned char *Stat)

void Write2Can(uw16 ID, unsigned char *CanBuff)

bool WriteReadfCan(unsigned short ID, unsigned char *CanBuff)

bool ReadfCan(unsigned short *ID, unsigned char *CanBuff)

char *GetDLLVer(void)