-- LaimisJ - 26 Apr 2010

Introduction

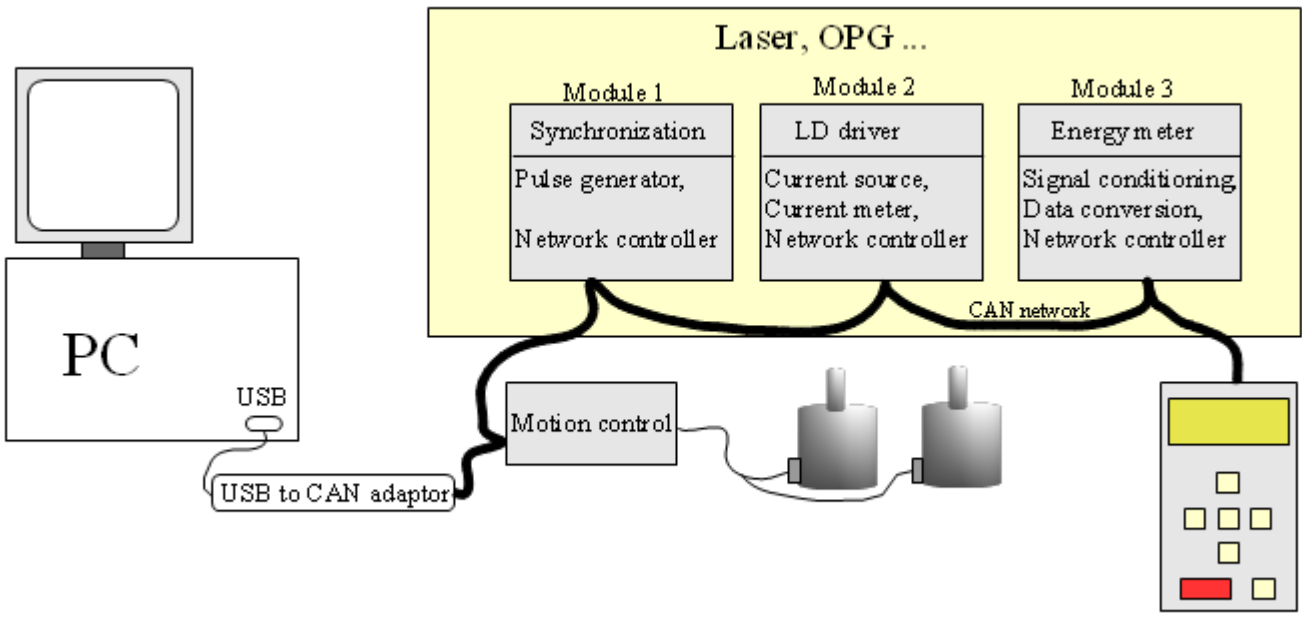
- Laser, OPG can be operated in two modes: either manually, using the control pad controls, or remotely by means of an external controller (usually a computer). This manual describes how to control device in the remote mode and how to write custom control application.

Getting started

- You need PC, USB cable and software CD to start control remotely Ekspla laser or system. Instead of CD there may be downloaded software archive.
- Go **EKSPLA_CD\CAN network\CAN browser** and run **CANBrowserSetup**. This will install USB drivers and control utility CAN browser.
- Connect USB cable and turn device on.
- Go to folder **EKSPLA_CD\CAN network\Control panel applications** and find your product folder. Copy folder content to your hard disk. Run **ControlPanel.exe**
- In case you are going to write own applications, read documentation below.

Modular approach in Ekspla products

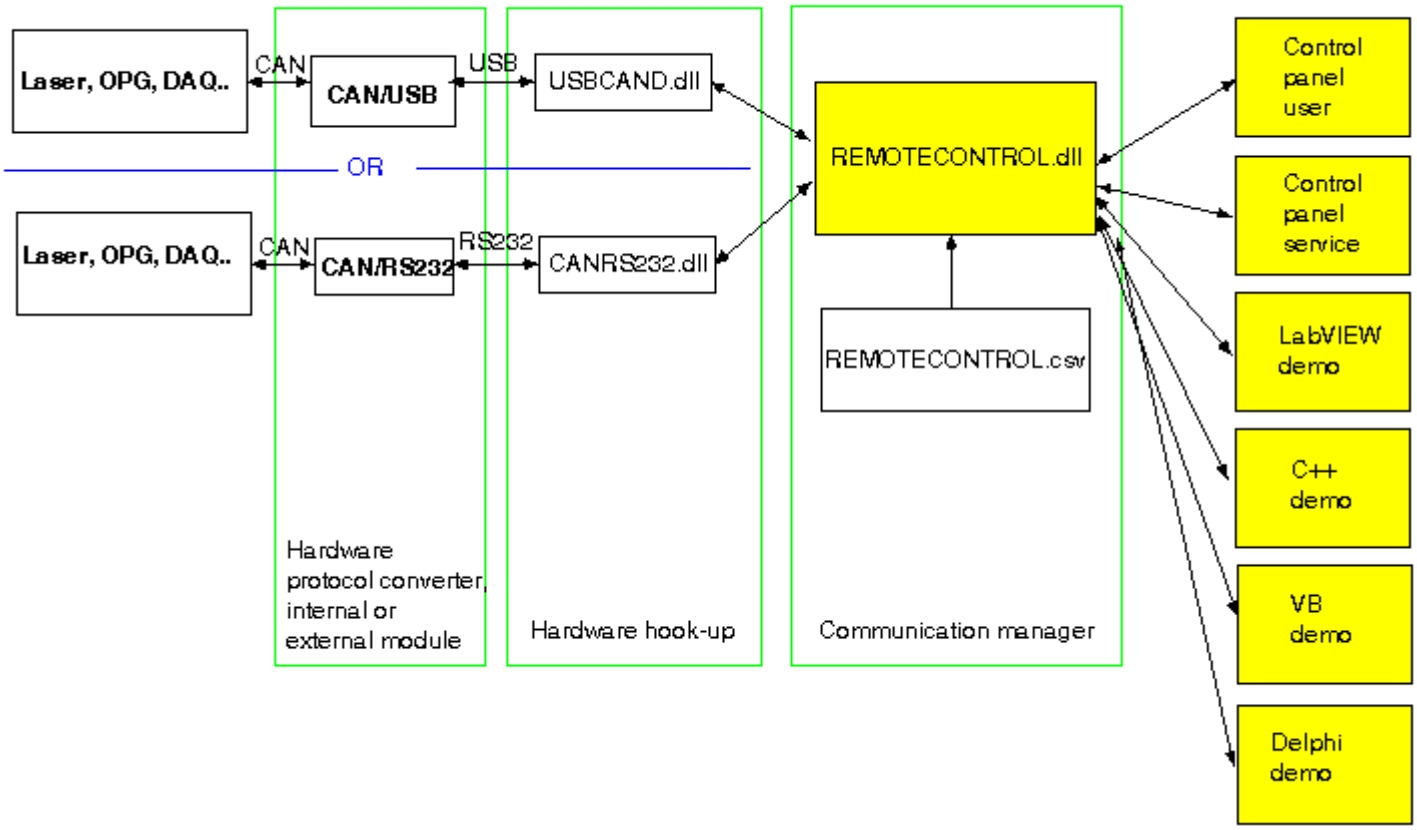
- Ekspla products are organized as set of smart modules that are hooked on single serial bus - CAN bus:



- Laser diode drivers, flash lamp power supplies, sensors or motion controllers are stand-alone modules dedicated to performing a single function.
- Uniform network is used inside Ekspla lasers and around. This allows to quickly expand laser capabilities by adding sensors or other devices without modifying laser hardware.
- There is a software package for control and data acquisition. It contains drivers, application, functions library. It supports all flexibility that networked hardware offers.

Control software

- Control software package lets control devices remotely and supports writing custom applications.
- Control software diagram:



Control software elements:

Common files

- Common files are dedicated for general communications task. Files are shared among different Ekspla products.
- **USBCAND.dll** - shared library that implements communication through USB-CAN bridge, common way to connect PC to laser, OPG . This library is invoked through REMOTECONTROL.dll.
- **CANRS232.dll** - shared library that implements communication through RS232-CAN bridge. RS232-CAN bridge functions on several flash lamp power supplies offering higher EMI immunity compare to USB. This library is invoked through REMOTECONTROL.dll.
- **REMOTECONTROL.dll** - essential shared library that manages communication between application and devices. It translates function calls to data packets sent and received from network. Control and user applications uses this library to connect, get data send commands.
- **REMOTECONTROL.h** - header file containing forward declarations of subroutines and variables of REMOTECONTROL.dll. Also provides information needed to write custom applications.

Product specific configuration files

- Configuration files may be product, product group specific or created to accomplish a single task. Configuration files have to reside in the same directory as REMOTECONTROL.dll
- Configuration files are formated text files and can be easily viewed.
- Configuration files can't be combined, therefore in case product options or external devices are added, the new configuration files should be generated.
- Configuration files have to be copied from product control panel folder.

REMOTECONTROL.csv

- **REMOTECONTROL.csv** - ASCII spreadsheet file, that may opened for reading by Microsoft Excel, OpenOffice Calc or any text editor. File contains list of accessible entities inside of device or system. Please note, any changes made will invalidate the file.
- Sample csv file:

Device: UCP												
										Date: 2010.04.08		
Name	ID	Reg ID	Menu	Type	User rights	Nonvolatile	Min value	Max value	Short name	Print format	Name	Value
UCP	$4	$9	$F	u32	ArUrSr		0	4294967295	Version	%06u	Firmware version	91130
LDM6A	$10	$10	$0	u8	AUS	NV	0	1	State	[OFF,ON,FAULT]	Power	ON
LDM6A	$10	$11	$1	u8	AUS		0	1	LDon	[OFF,ON]	Diode current ON	ON
LDM6A	$10	$12	$2	u16	AUrS	NV	0	1000	Iset	%.2fA	Set Current	2.20
LDM6A	$10	$13	$3	u16	ArUrSr		0	1000	Idisp	%.2fA	Display Current	2.14
__license:c5416daab64ff88304aaba757c6f023b												

- First and second line are for general information: first module name, creation data and columns names
- Last line - code for csv file integrity check. Connection is not allowed in case REMOTECONTROL.csv is altered. Only Ekspla authorized persons are allowed to validate REMOTECONTROL.csv file.
- Columns **Name, ID** Each module has an ID along with its name. ID is a number 0..63. Please note, ID is shown in HEX form at the table. $10 is equal to decimal 16. Name and ID are tied together in DLL procedure calls, separated by a colon. Both versions HEX and decimal are allowed for the IDs. For example, **LDM6A:16** and **LDM6A:$10** are valid references for module LDM6A? as **devname** argument in function calls. The network may contain several modules with the same name but their IDs always differ.
- Column **Menu** is not used in remote control.
- Column **Type** register data type, see table below. REMOTECONTROL.dll operates in converted to floating point or to text representation numbers, therefore type may be ignored in most applications.

Type	Description
u8	byte 0..255 unsigned 8-bit
s8	shortint -128..127 signed 8-bit
u16	word 0..65535 unsigned 16-bit
s16	smallint -32768..32767 signed 16-bit
u32	longword 0..4294967295 unsigned 32-bit
s32	long integer -2147483648..2147483647 signed 32-bit
float	single 1.5 x 10^-45 .. 3.4 x 10^38 Significant digits 7-8
string8	array of bytes

- Column **User rights**. In remote control **ArUrSr** value marks read only parameter. All others values are for writable parameters.
- Column **Nonvolatile**. Value of **NV** indicates that parameter may be written to a non volatile memory.
- Columns **Min value**, **Max value** parameter bounds. REMOTECONTROL.dll will not send command if parameter will be off bounds.
- Column **Short name** not used.
- Column **Print format**. Data display format helps with numerical value interpretation. Functions GetRegAsString and SetRegFromString uses format for forth and back conversion.
    - Integer types **%u[x]** There x is parameter dimension.
    - Decimal types. Register value is integer. Print format e.g. **%.1f** indicated that register value should be divided by 10 for display, **%.2f[x]** - by 100...
    - Float type **%f[x]**
    - Set type [XXX, YYY, ...]. Register type is u8. Register value points to corresponding actual element from the set. Zero value points to first element of the set.
- Column **Name**. Full register name, used by REMOTECONTROL.dll for register identification. Copy value to **regname** argument in function calls.
- Column **Value** not used.

REMOTECONTROL.INI

- REMOTECONTROL.INI is in standard Windows INI files format.
- REMOTECONTROL.INI provides following information for control panel application:
    - Product name
    - Master module (CPU) name.
    - List of registers that are polled periodically and displayed on control panel, **PollList**.
    - List of registers that appears on drop list and can be set or programed, **SetProgramList**
- List are text lines with module name, register and description separated by double spaces.

```
[Product]
Name=Laser PL2241
[PollList]
Name0=SY3PL50M:32  State  Laser State
[Modules]
CPUModuleName=SY3PL50M:32
[SetProgramList]
Name1=SY3PL50M:32  PRE-T delay
```

Product remote control panels

- Product remote control panel is executable application for the basic remote functions: starting/stopping the laser, watching essential operational data, changing user settings.