DVI KVM communication protocol

1. RS232 Serial port:

Baud rate: 9600 bps
Stop bit: 1 bit
Data length: 8 bits

Commands **(Hex format)**:
Switch to PC1: 0xAA 0xBB 0x03 0x01 0x01 0xEE
Switch to PC2: 0xAA 0xBB 0x03 0x01 0x02 0xEE
Switch to PC3: 0xAA 0xBB 0x03 0x01 0x03 0xEE
Switch to PC4: 0xAA 0xBB 0x03 0x01 0x04 0xEE

2. LAN port:

IP address: 192.168.1.10
Port: 5000
Gate way: 192.168.1.1
Mask address: 255.255.255.0

Commands **(Hex format)**:
Switch to PC1: 0xAA 0xBB 0x03 0x01 0x01 0xEE
Switch to PC2: 0xAA 0xBB 0x03 0x01 0x02 0xEE
Switch to PC3: 0xAA 0xBB 0x03 0x01 0x03 0xEE
Switch to PC4: 0xAA 0xBB 0x03 0x01 0x04 0xEE

3. Cording Examples for Windows VS2012:

3.1 RS232:

```
private: System::IO::Ports::SerialPort^  serialPort1;
private: System::Windows::Forms::ComboBox^  CBSerialPort;
private: array<Byte>^ cmdBuf;

// Get current available Serial Port list
inti;
for(i=0;i<(System::IO::Ports::SerialPort::GetPortNames()->Length);i++)
{
        this->CBSerialPort->Items->Add(System::IO::Ports::SerialPort::GetPortNames()[i]);
}
if(i>=1)
{
        this->CBSerialPort->SelectedIndex = 0;
```

```
}
//Connect to selected Serial Port
try
	{
		serialPort1->BaudRate = 9600;
		serialPort1->PortName = this->CBSerialPort->Text;
		timer1->Enabled = true;
		serialPort1->Open();
	}
	catch(System::Exception^ e)
	{
		// Initializes the variables to pass to the MessageBox::Show method.
		String^ message = "Cann't open port "+ this->CBSerialPort->Text;
		String^ caption = "Open port error!";
		MessageBoxButtonsbuttons = MessageBoxButtons::OK;
		System::Windows::Forms::DialogResultresult;

		// Displays the MessageBox.
		result = MessageBox::Show( this, message, caption, buttons );
		if(result == System::Windows::Forms::DialogResult::OK)
		{
			BTConnect->Text = "Connect";
		}
	}
//Send command strings to Matrix
cmdBuf[0] = 0xAA;
cmdBuf[1] = 0xBB;
cmdBuf[2] = 0x03;
cmdBuf[3] = 0x01;
cmdBuf[4] = 0x01;
cmdBuf[5] = 0xEE;

if(serialPort1->IsOpen)
```

```
{
    serialPort1->Write(cmdBuf,0,6);
}
```

3.2 LAN(TCP/IP connection):

```
private: array<Byte>^ cmdBuf;
private: System::Net::Sockets::Socket^ netCtlSocket;

//Connect to Matrix
netCtlSocket = ConnectSocket("192.168.1.10",5000);          //Connect to matrix with default IP address and port number.

//Send command strings to Matrix
cmdBuf[0] = 0xAA;
cmdBuf[1] = 0xBB;
cmdBuf[2] = 0x03;
cmdBuf[3] = 0x01;
cmdBuf[4] = 0x01;
cmdBuf[5] = 0xEE;
if(netCtlSocket!=nullptr)
{
    if(netCtlSocket->Connected)
    {
        netCtlSocket->Send(cmdBuf, 6, static_cast<SocketFlags>(0) );
    }
}
```