

EDU05D.DLL

Technical Guide

Introduction

General

The EDU05 USB tutor module has 8 digital input/output channels, five analog input channels and two PWM outputs. There on the board is also 16 character LCD display.

All communication routines are contained in a Dynamic Link Library EDU05D.DLL.

In this manual we will describe each of these functions provided by the DLL in detail. Calling the functions exported by the DLL, you can write custom Windows applications in Visual Basic or any other 32-bit Windows application development tool that supports calls to a DLL.

A complete overview of the procedures and functions that are exported by the EDU05D.DLL follows.

Note that all the examples in the function description section are written in C++.

EDU05 examples folder includes examples written in Visual Basic 2008 Express, Visual C# 2008 Express and Visual C++ 2008 Express.

Readers should have an understanding of the basic data types as well as basic knowledge of the Microsoft Windows operating system.

Microsoft Visual Studio users please note: The EDU05D.DLL is a standard Windows DLL, you cannot reference it.

Calling convention

A calling convention is a scheme for how functions receive parameters from their caller and how they return a result. Different programming languages use different calling conventions, so it is important to know which calling convention is used by your programming language and which calling convention is used by the EDU05D.DLL.

The most common calling convention is the *stdcall* calling convention, and this is also the one we have used for our DLL.

If you are using .NET (VB.NET or C#) you do not need to worry about this since the calling convention in .NET is also *stdcall*. However if you are using C to import the functions provided by the DLL, you will need to pay special attention to this.

Overview of the Functions

General functions

<code>int</code> <code>OpenDevice()</code>	<i>Opens the communication link to the EDU05 device</i>
<code>void</code> <code>CloseDevice()</code>	<i>Closes the link to the EDU05 device</i>
<code>int</code> <code>Connected()</code>	<i>Checks that the USB connection to the card is valid</i>
<code>void</code> <code>InOutMode(int IOMode)</code>	<i>Set the digital terminals either inputs or outputs</i>

Analog to Digital converter function

<code>int</code> <code>ReadAnalogChannel(int Channel)</code>	<i>Reads the status of one analog input-channel</i>
--	---

PWM Output function

<code>void</code> <code>SetPWM(int Channel, int Data)</code>	<i>Sets the status the PWM output</i>
--	---------------------------------------

Digital Output functions

<code>void</code> <code>OutputAllDigital(int Data)</code>	<i>Sets the digital outputs according to the data</i>
<code>void</code> <code>ClearDigitalChannel(int Channel)</code>	<i>Clears the output channel</i>
<code>void</code> <code>ClearAllDigital()</code>	<i>Clears all output channels</i>
<code>void</code> <code>SetDigitalChannel(int Channel)</code>	<i>Sets the output channel</i>
<code>void</code> <code>SetAllDigital()</code>	<i>Sets all output channels</i>

Digital Input functions

<code>int</code> <code>ReadAllDigital()</code>	<i>Reads the status of all the input channels</i>
--	---

LCD functions

<code>void</code> <code>LCDInit()</code>	<i>Initializes the LCD display module</i>
<code>void</code> <code>LCDClear()</code>	<i>Clears the text on the LCD display</i>
<code>void void</code> <code>LCDWriteString(String^ Data, int Position);</code>	<i>Writes text on the LCD display</i>

Function List

OpenDevice

Syntax

```
int OpenDevice();
```

Result

int: If succeeded the return value 1 indicates that EDU05 card was found.

Return value 0 indicates that no EDU05 card found.

Description

Opens the communication link to the EDU05 card. Loads the drivers needed to communicate via the USB port. This procedure must be performed before any attempts to communicate with the EDU05 card.

Example

```
int h = OpenDevice();
switch (h)
{
    case 0:
        Label1->Text = "EDU05 not found";
        break;
    case 1:
        Label1->Text = "EDU05 connected";
        IOMode = 1;
        InOutMode(IOMode);
        RadioButton3->Checked = true;
        Timer1->Enabled = true;
        break;
}
```

CloseDevice

Syntax

```
void CloseDevice();
```

Description

Unloads the communication routines for EDU05 card and unloads the driver needed to communicate via the USB port. This is the last action of the application program before termination.

Example

```
private: System::Void Form1_FormClosed(System::Object^ sender,
System::Windows::Forms::FormClosedEventArgs^ e)
{
    CloseDevice();
}
```

ReadAnalogChannel

Syntax

```
int ReadAnalogChannel(int Channel);
```

Parameter

Channel: Value between 1 and 5 which corresponds to the AD channel whose status is to be read.

Result

int: The corresponding Analog to Digital Converter data is read.

Description

The input voltage of the selected 10-bit Analog to Digital converter channel is converted to a value which lies between 0 and 1023.

Example

```
int Buffer[5];
int i;

for (i=0; i<5; i++)
    Buffer[i] = ReadAnalogChannel(i+1);
```

InOutMode

Syntax

```
void InOutMode(int IOMode);
```

Parameter

IOMode: 0: the digital I/O terminals are outputs

IOMode: 1: the digital I/O terminals are inputs.

Description

Set the digital terminals either inputs or outputs.

Example

```
InOutMode(0); // All the digital I/O pins of card are set to outputs.
```

OutputAllDigital

Syntax

```
void OutputAllDigital(int Data);
```

Parameter

Data: Value between 0 and 255 that is sent to the digital output port (8 channels).

Description

The channels of the digital output port are updated with the status of the corresponding bits in the data parameter.

Example

```
InOutMode(0);
OutputAllDigital(0x55);
```

ClearDigitalChannel

Syntax

```
void ClearDigitalChannel(int Channel);
```

Parameters

Channel: Value between 1 and 8 which corresponds to the output channel that is to be cleared.

Description

The selected channel is cleared.

Example

```
InOutMode(0);
ClearDigitalChannel(3);
```

ClearAllDigital

Syntax

```
void ClearAllDigital();
```

Result

All digital outputs are cleared.

Example

```
InOutMode(0);  
ClearAllDigital(CardAddress);
```

SetDigitalChannel

Syntax

```
void SetDigitalChannel(int Channel);
```

Parameter

Channel: Value between 1 and 8 which corresponds to the output channel that is to be set.

Description

The selected digital output channel is set.

Example

```
if (CheckBox3->Checked)  
    SetDigitalChannel(3);  
else  
    ClearDigitalChannel(3);
```

SetAllDigital

Syntax

```
void SetAllDigital();
```

Description

All the digital output channels are set.

Example

```
InOutMode(0);  
SetAllDigital(CardAddress);
```

SetPWM

Syntax

```
void SetPWM(int Channel, int Data);
```

Parameters

Channel: The PWM output channel 1 or 2.

Data: Value between 0 and 255 which is to be sent to the PWM output of the card. The duty cycle of the PWM output corresponds to the data value: 0 = 0%, 255 = 100% duty cycle.

Example

```
SetPWM(1, 128); // The duty cycle of the PWM output #1 is set to 50%
```

Connected

Syntax

```
bool Connected();
```

Result

bool: *true* if the card is connected, *false* if not connected.

Description

Checks that USB connection to the cards is valid.

Example

```
if (Connected())  
    Label1->Text = "EDU05 connected";  
else
```

```
Label1->Text = "EDU05 not connected";
```

LCDInit

Syntax

```
void LCDInit();
```

Description

Initializes the LCD display module.

Example

```
InOutMode(0);  
LCDInit();
```

LCDClear

Syntax

```
void LCDClear();
```

Description

Clears the LCD display.

Example

```
InOutMode(0);  
LCDClear();
```

LCDWriteString

Syntax

```
void LCDWriteString(String^ Data, int Position);
```

Parameters

Data: Pointer to the text string.

Position: Text position (0...15) on the LCD display.

Example

```
InOutMode(0);  
LCDWriteString(textBox1->Text, Convert.ToInt32(textBox2->Text));
```

Function declarations in other programming languages

Visual Basic 2008 Express

```
Private Declare Function OpenDevice Lib "edu05d.dll" () As Integer
Private Declare Sub CloseDevice Lib "edu05d.dll" ()
Private Declare Function Connected Lib "edu05d.dll" () As Boolean
Private Declare Function ReadAnalogChannel Lib "edu05d.dll" (ByVal Channel As Integer) As Integer
Private Declare Function ReadAllDigital Lib "edu05d.dll" () As Integer
Private Declare Sub SetPWM Lib "edu05d.dll" (ByVal Channel As Integer, ByVal Data As Integer)
Private Declare Sub InOutMode Lib "edu05d.dll" (ByVal IOMode As Integer)
Private Declare Sub OutputAllDigital Lib "edu05d.dll" (ByVal Data As Integer)
Private Declare Sub ClearAllDigital Lib "edu05d.dll" ()
Private Declare Sub SetAllDigital Lib "edu05d.dll" ()
Private Declare Sub ClearDigitalChannel Lib "edu05d.dll" (ByVal Channel As Integer)
Private Declare Sub SetDigitalChannel Lib "edu05d.dll" (ByVal Channel As Integer)
Private Declare Sub LCDInit Lib "edu05d.dll" ()
Private Declare Sub LCDClear Lib "edu05d.dll" ()
Private Declare Sub LCDWriteString Lib "edu05d.dll" (ByVal Data As String, ByVal Position As Integer)
```

Visual C# 2008 Express

```
[DllImport("edu05d.dll")]
public static extern int OpenDevice();

[DllImport("edu05d.dll")]
public static extern void CloseDevice();

[DllImport("edu05d.dll")]
public static extern bool Connected();

[DllImport("edu05d.dll")]
public static extern int ReadAnalogChannel(int Channel);

[DllImport("edu05d.dll")]
public static extern int ReadAllDigital();

[DllImport("edu05d.dll")]
public static extern void SetPWM(int Channel, int Data);

[DllImport("edu05d.dll")]
public static extern void InOutMode(int IOMode);

[DllImport("edu05d.dll")]
public static extern void OutputAllDigital(int Data);

[DllImport("edu05d.dll")]
public static extern void ClearAllDigital();

[DllImport("edu05d.dll")]
public static extern void SetAllDigital();

[DllImport("edu05d.dll")]
public static extern void ClearDigitalChannel(int Channel);

[DllImport("edu05d.dll")]
public static extern void SetDigitalChannel(int Channel);

[DllImport("edu05d.dll")]
public static extern void LCDInit();

[DllImport("edu05d.dll")]
public static extern void LCDClear();

[DllImport("edu05d.dll")]
public static extern void LCDWriteString(string Data, int Position);
```
