NanoWave MPM300 Programming Library

NanoWave Inc

March 8, 2010

Revision History		
Rev. 3	3/7/10	Spelling and corrections.
Rev. 2	12/11/09	Changes and corrections.
Rev. 1	10/29/09	Initial release,

Contents

1 MPM.dll Overview

In addition to the complete support within CP300, MPM hardware is also supported through a provided *C#* class library. The library is available to Windows programmers to interface their own software to NanoWave's MPM hardware.

For MPM.dll to work correctly, both CyUSB.dll and firmware.hex must be in the same folder as the MPM.dll file.

The library can also be wrapped and called from programs such as National Instrument's Labview and Mathworks's Matlab.

1.1 Function Summary

Below is a summary of the functions provided by the Class Library.

USB_Transaction()
Constructor, creates the object.

void Initialize()
Must be called to initialize parameters after creating the constructor and assigning event handlers.

int GetDeviceCount()
Get the number of compatible MPM hardware connected to the host machine.

int[] GetData()
Gets an array of long (Int32) containing 50 points of data representing position information. The 50 points are sampled at 20 KHz. Data is truncated at 32 bit and may not be suitable for some encoders such as Nikon ABS rotary encoder.

long[] GetData64()
Gets an array of long (Int64) containing 50 points of data representing position information. The 50 points are sampled at 20 KHz.

float GetDropRatio()
Gets dropped packet rate.

int GetEncoderType()
Returns the type of encoder setting being selected for current device. The returned result corresponds to the list below:

0. Digimicro Encoder
1. ABS Linear Encoder
2. A-Quad-B Input

3. ABS Rotary (17 bit)
4. ABS Rotary (20 bit)

short GetFirmwareVersion()
Gets the MPM firmware version. GetData() function must be called at least once before getting the firmware version.

int32 GetMostRecentData()
Returns the single most recent data after GetData()/GetData64() is called.

long GetMostRecentData64()
Returns the single most recent data after GetData()/GetData64() is called.

void Reset()
Resets MPM hardware.

void SelectDevice(int index)
If multiple MPM hardware are connected to one host machine, switch to the specified 'index' MPM device.

void SetEncoderType(int type)
Set the encoder type connected to MPM300 counter. This function is only valid when non-Digimicro encoder, particularly Nikon's absolute rotary encoder, Nikon's absolute linear encoder and A-Quad-B type digital encoder is connected to MPM300. The parameter *type* corresponds to the list below:

0. Digimicro Encoder
1. ABS Linear Encoder
2. A-Quad-B Input
3. ABS Rotary (17 bit)
4. ABS Rotary (20 bit)

void TransferInDummyData(int n)
Transfers dummy data to clear the USB bus if data corruption is detected.

void UpdateOutData()
Resend all parameters (such as PID parameters, cutoff frequency) to MPM hardware, particularly after resetting MPM hardware.

void Zero()
Sets the current encoder position as zero.

public event DeviceFound
Event triggered when a MPM family device is connected to or removed from a host PC. Event must be assigned a handler/listener before calling **Initialize()**.

2 Information

NanoWave's SPPE technology is patent protected. The following U.S. patents have been issued to NanoWave: 5,589,686; 5,744,799; 6,639,686 and corresponding foreign patents. In addition NanoWave has several patents pending.

For more information on products, services, prices and deliveries, please visit our website at *www.nanowave.com*.
Please send questions to $info$@*nanowave.com*.

NanoWave, Inc.
PO Box 490
Sutton MA 01590 USA
(888) 415-6555 Toll-free
(857) 413-4901 Office
(857) 413-4906 Fax