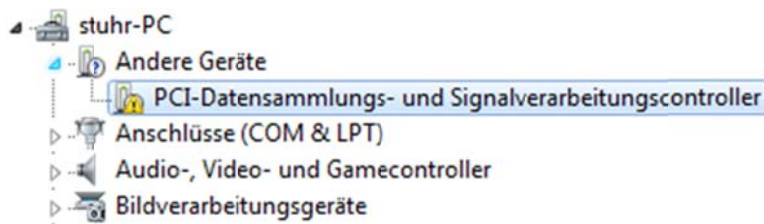


Kurs 24# Meilhaus Me4660

Installation ME 4660 – AD- Karte

Karte in eine freien PCI- Slot stecken, um zu prüfen, ob sie nicht schon installiert worden ist. Wenn nicht, dann gibt es nach PC starten folgendes Bild: Im Geräte Manager erscheint dann ein unbekanntes Gerät:



Dann muss die Karte wieder ausgebaut werden, da die Treiber ohne Karte installiert werden müssen!

Dann auf der Meilhaus Seite den neusten Treiber herunterladen:



ME-iDS und PowerLab

Bitte wählen Sie hier einen Bereich zum downloaden aus...

Download ...

Bitte beachten Sie die entsprechenden Hinweise zur Lizenzregelung der Software in unseren » [AGBs](#).

	Unbedingt zuerst lesen!	Version	Update	Größe	
	Zweisprachige Quick-Start-Anweisung zum ME-iDS	1.5	25.05.2011	3 MB	
	Software	Version	Update	Größe	
	Treiber-System ME-iDS für Windows 2000, XP, Vista, Windows 7 (32 bit) inkl. ME-PowerLab ³ (und ME-PowerLab ²)	1.3.7.0	25.05.2011	34 MB	
	Treiber-System ME-iDS für Windows XP, Vista, Windows 7 (64 bit) inkl. ME-PowerLab ³ (und ME-PowerLab ²)	1.3.7.0	25.05.2011	36 MB	
	Treiber-System ME-iDS für Linux	1.3.54	25.05.2011	3,5 MB	
	Gebrauchsanleitungen	Version	Update	Größe	
	Handbuch in deutscher Sprache	1.4D	25.05.2011	2,5 MB	

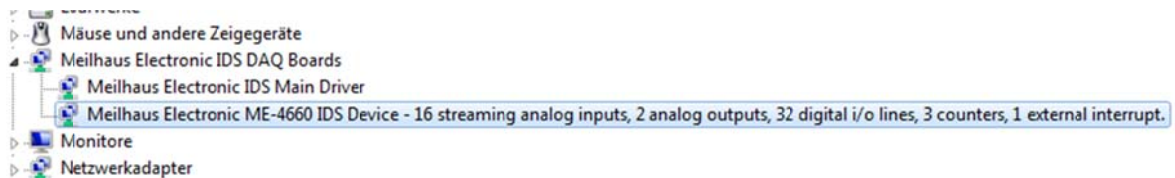
Hinweis: MEphisto Digi, MEphisto Opto und MEphisto Switch nicht unterstützt unter Linux und Windows 64 bit

» [Zur Produkt-Beschreibung](#).

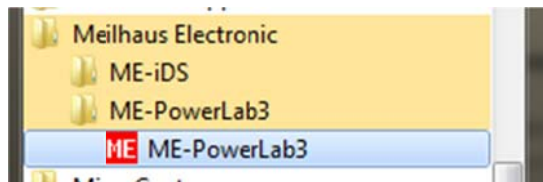
Da ich ein Win 7- 64 Bit habe, eingerahmte Version genommen. Nach Auspacken gibt es dabei eine FirstSteps.pdf, die die Installation gut beschreibt. Das hat dann ohne Probleme bei mir unter Win 7 geklappt.

Dann Karte wieder einbauen.

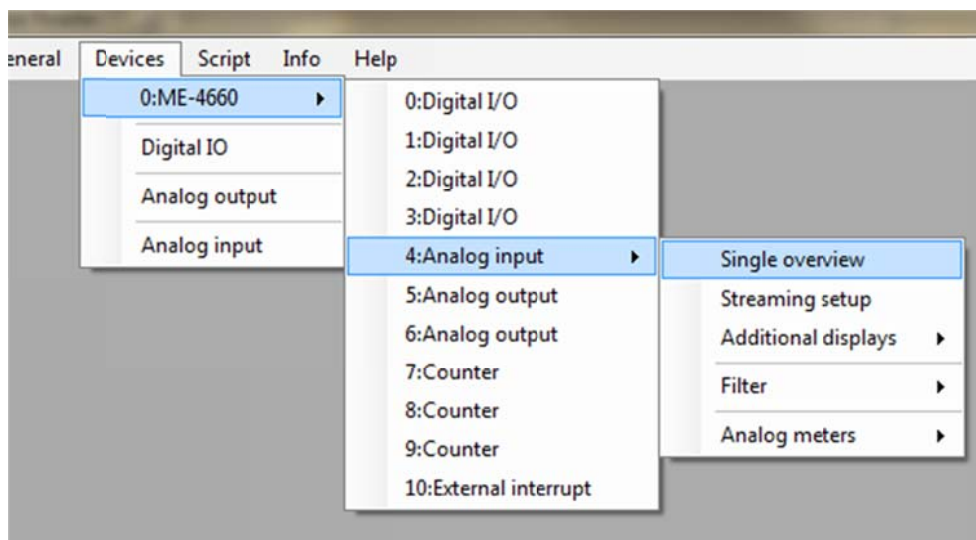
Danach sollte es im Gerätemanager so aussehen:



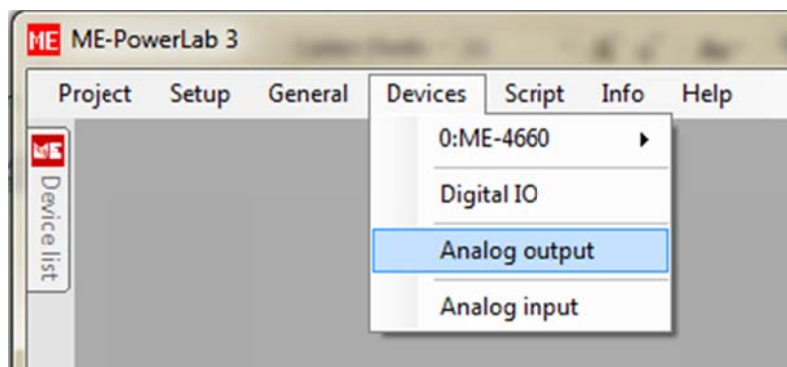
Dann sollte die Hardware getestet werden. Dazu gibt es das Programm Me Power Lab3, zu finden in Programme:



Nach Start kann man so die Analogen Eingänge sich anschauen und



So die analogen Ausgänge:



Danach am besten mit zwei Kabeln mal DA0 mit AD0 und DA1 mit AD1 verbinden und auf dem Bildschirm sieht es dann so aus: (DA0 auf +1V und DA1 auf -1V):

Channel	Output Name	Range	Value	Unit
0	0:ME-4660 5:Analog output Channel 0	-10.0..10.0 V	1.000	V
1	0:ME-4660 6:Analog output Channel 0	-10.0..10.0 V	-1.000	V

Und

Channel	Input Name	Mode	Range	Value	Unit
0	Channel 0	single	-10.0..10.0 V	0.994	V
1	Channel 1	single	-10.0..10.0 V	-1.003	V
2	Channel 2	single	-10.0..10.0 V	-6.710	V
3	Channel 3	single	-10.0..10.0 V	-6.313	V
4	Channel 4	single	-10.0..10.0 V	-6.546	V
5	Channel 5	single	-10.0..10.0 V	-6.314	V
6	Channel 6	single	-10.0..10.0 V	-8.155	V
7	Channel 7	single	-10.0..10.0 V	-7.213	V
8	Channel 8	single	-10.0..10.0 V	-4.353	V
9	Channel 9	single	-10.0..10.0 V	-3.275	V
10	Channel 10	single	-10.0..10.0 V	-5.155	V
11	Channel 11	single	-10.0..10.0 V	-3.351	V
12	Channel 12	single	-10.0..10.0 V	-3.227	V
13	Channel 13	single	-10.0..10.0 V	-2.435	V
14	Channel 14	single	-10.0..10.0 V	-3.024	V
15	Channel 15	single	-10.0..10.0 V	-2.687	V

Wie Sie sehen, zeigen AD0 und AD1 die richtigen Werte an.

Software in C#

Jetzt zur Software in C#. ME liefert Beispiel mit, die zu finden sind unter

Lokaler Datenträger (C:) > Programme (x86) > Meilhaus Electronic > ME-iDS > 01.03.07.000 > SDK > MS Visual C-Sharp >				
Bearbeiten Ansicht Extras ? Erstellen In Bibliothek aufnehmen Freigegeben für Brennen Neuer Ordner				
	Name	Änderungsdatum	Typ	Größe
Ordner	Con_meIDSSystemInfo	19.10.2011 07:25	Dateiordner	
Desktop	Con_meIOExtIrqCallback	19.10.2011 07:25	Dateiordner	
Downloads	Con_meIOResetDevice	19.10.2011 07:25	Dateiordner	
Zuletzt besucht	Con_meIOSingleReadAnalog	19.10.2011 07:25	Dateiordner	
Bibliotheken	Con_meIOSingleWriteAnalog	19.10.2011 07:25	Dateiordner	
Bilder	Con_meIOStreamRead	19.10.2011 07:25	Dateiordner	
Dokumente	Con_meIOStreamWrite	19.10.2011 07:25	Dateiordner	
Musik	Con_meUtilityPWM	19.10.2011 07:25	Dateiordner	
Videos	Executables	19.10.2011 07:25	Dateiordner	
	meIDS.cs	24.05.2011 12:48	Visual C# Source f...	
Netzgruppe	TestConsoleGeneric.cs	12.02.2010 15:13	Visual C# Source f...	

Es sind dort zwar Konsolenprogramme, aber die kann man dann nach eigenen Wünschen umgestalten.

Erst einmal folgt die Analyse, welche Kartennummer bzw. welche „Subdevices“ angesprochen werden müssen. Dazu hab ich in dem Ordner „Executables“ das Programm *Con_meIOSingleWriteAnalog.exe* gestartet.

Folgendes Ergebnis:

```

Subdevice: 5
-----
Subdevice type:      0x180001
Subdevice subtype:   0x190001
Number of channels:  1

Subdevice: 6
-----
Subdevice type:      0x180001
Subdevice subtype:   0x190001
Number of channels:  1

```

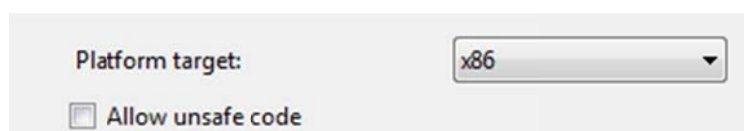
```

2 AO sub-devices found in System
1. Device: 0 Sub-device: 5 Channels: 1 Ranges: 1
2. Device: 0 Sub-device: 6 Channels: 1 Ranges: 1
Press any key to continue

```

Also sind zwei DA- Ausgänge gefunden, die den Subdevices Nr. 5 und 6 zugeordnet sind.

Die C# - Datei meIDS.cs muss dann dem Projekt hinzugefügt werden. Diese enthält alle Definitionen und insbesondere auch die Verweise auf eine DLL, die die Schnittstelle zur Hardware darstellt. Diese DLL heißt *meIDSMmain.dll* und wird bei der Installation in das Standard – Windows- Verzeichnis kopiert. Die Beispiele sind für die 32 Bit- Version geschrieben, man muss das Projekt also auch auf *platform x86* stellen, sonst geht es nicht. Also in Project → Properties dann in Tab „Build“



Bevor überhaupt irgendetwas funktioniert, muss ein „open- Befehl“ ausgelöst werden:

```
int error=meIDS.meOpen(0);
```

Wenn alles OK ist, dann ist der Rückgabewert=0;

Dann muss der Ausgang konfiguriert werden:

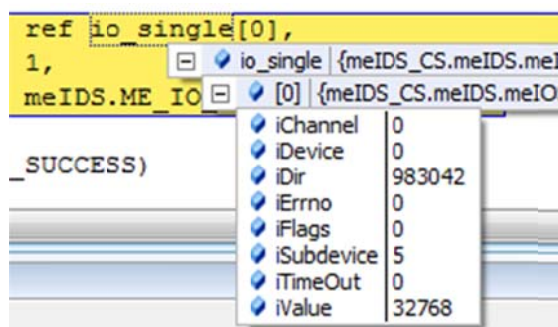
```
i_me_error = meIDS.meIOSingleConfig(
arrSubdevices[index_subdevice].m_iDeviceIndex, // Device index =0
arrSubdevices[index_subdevice].m_iSubdeviceIndex, // Subdevice index= 5 /6,
index_channel, // Channel index=0
index_range, // Range index=0 +-10V
meIDS.ME_REF_AO_GROUND, // Reference
meIDS.ME_TRIG_CHAN_DEFAULT, // Trigger channel - standard
meIDS.ME_TRIG_TYPE_SW, // Trigger type - software
meIDS.ME_VALUE_NOT_USED, // Trigger edge - not applicable
meIDS.ME_IO_SINGLE_CONFIG_NO_FLAGS ); // Flags
```

Dies muss für beide Kanäle einmal nach Start gemacht werden.

Jetzt sieht eine Ausgabe auf DA- Kanal 0 wie folgt aus:

```
i_me_error = meIDS.meIOSingle( ref io_single[0], // Output list
1, // Number of elements in the above list
meIDS.ME_IO_SINGLE_NO_FLAGS ); // Flags
```

Bei Ausgabe 0V ist ein Wert von 32768 zur Karte zu schreiben, bei 10 V 65535 und bei -10 V dann 0.
Der Speicherinhalt der Struktur io_single[0] lautet:



Programmiert:

```
meIDS.meIOSingle_t[] io_single = new meIDS.meIOSingle_t[1];

io_single[0].iDevice = 0;
io_single[0].iSubdevice = 5; // 5 für DA0 und 6 für DA1
io_single[0].iChannel = 0;
io_single[0].iDir = meIDS.ME_DIR_OUTPUT;
io_single[0].iValue = i_digital_value; // Wert 0 bis 65535 (Hex FFFF)
io_single[0].iTimeout = 0; // No timeout-not required for software output
io_single[0].iFlags = meIDS.ME_IO_SINGLE_TYPE_NO_FLAGS;
```

In iDir wird die obige Konstante geschrieben.

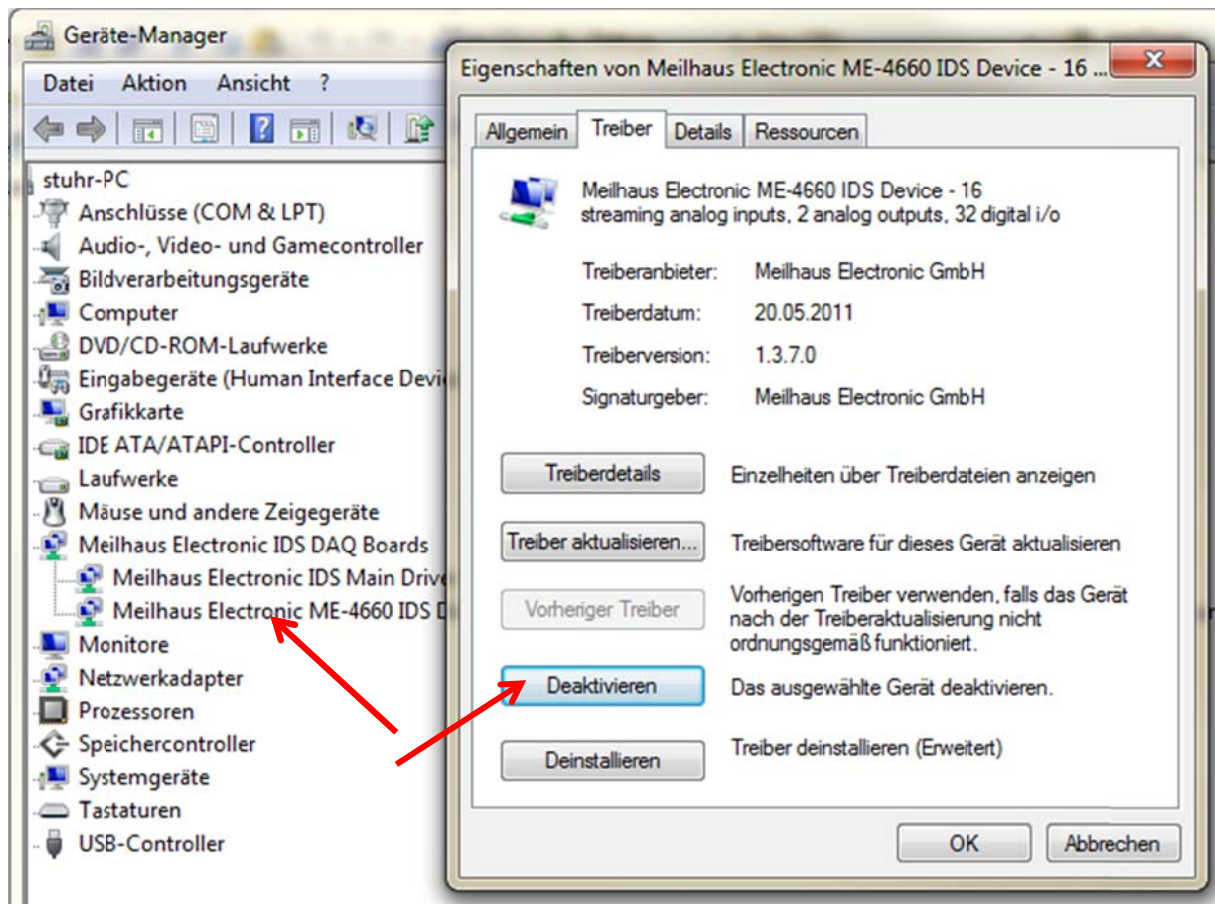
Beim Schließen des Programms sollte der Befehl

```
meIDS.meClose(0);
```

ausgelöst werden.

Wenn der Rechner in den Energiesparmodus fährt, liegen anschließend -10 V an beiden DA – Ausgängen. Weder ein Reset noch irgendein Befehl über ME- Power Lab3 können aus diesem Zustand heraushelfen.

Man muss mit dem Gerätemanager dann die Karte einmal deaktivieren und sofort wieder aktivieren:



dann läuft sie wieder.

Nun zur analogen Ausgabe. Diese geschieht mit dem Subdevice 4, alle Kanäle 0 bis 15 und iDir muss umgestellt werden auf Eingabe. Also:

```
io_single[0].iSubdevice = 4; // 5 für DA0 und 6 für DA1, 4 für AD
io_single[0].iDir = meIDS.ME_DIR_INPUT;
io_single[0].iChannel = 0; // Kanal 0 bis 15
error = meIDS.meIOSingle(ref io_single[0], 1, meIDS.ME_IO_SINGLE_NO_FLAGS);
```

Dann wird in `io_single[0].iValue` der Abtastwert zurückgeliefert, wieder 0 für -10V und 65535 für +10 V. Ausgabe in Volt:

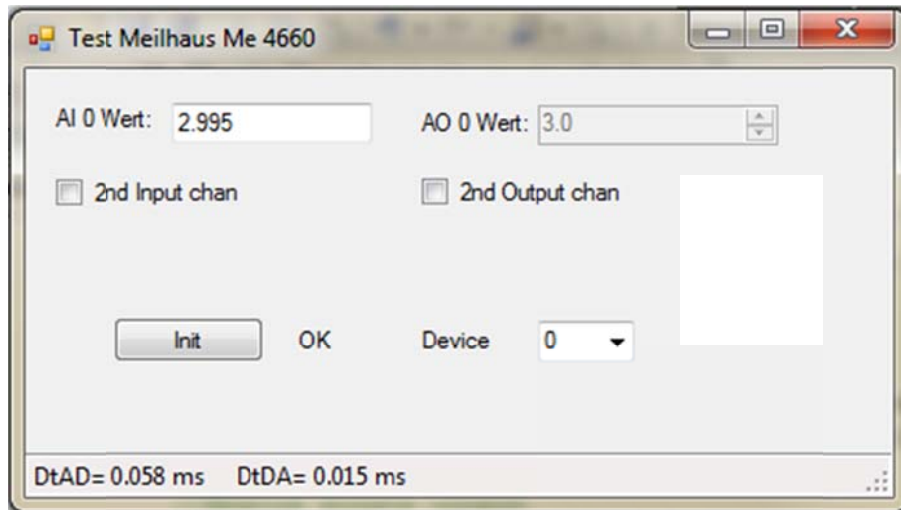
```
double outVal=io_single[0].iValue/65535.0*20.0-10.0;
```

Nun kann bei jedem neuen Aufruf ein neuer Spannungswert gemessen werden.

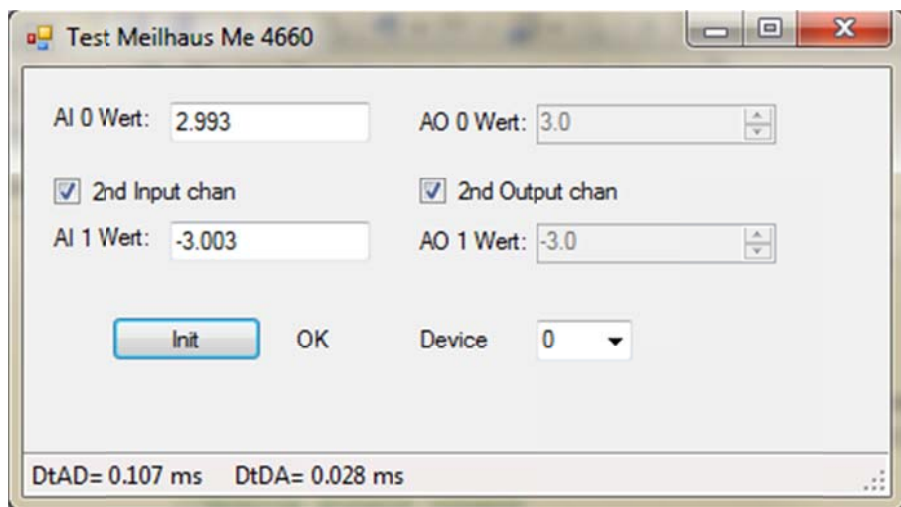
Nur dummerweise geht danach die Ausgabe nicht mehr. Man darf jetzt nämlich nicht vergessen, bei der Ausgabe den `iChannel` wieder auf 0 zu stellen.

Zeitmessung

Mit einer Stopwatch kann man nun die Laufzeiten der Ein- und Ausgabe messen. Im fertigen Projekt hab ich die Messung eingebaut. Folgende Zeiten ergeben sich mit dieser Karte auf einem Dell Optiplex 780 mit Pentium Dual Core 2,6 GHz:



Mit zwei AD – DA – Aufrufen:



So muss das sein!!!!

AD- Eingabe: ca. 60µsec!

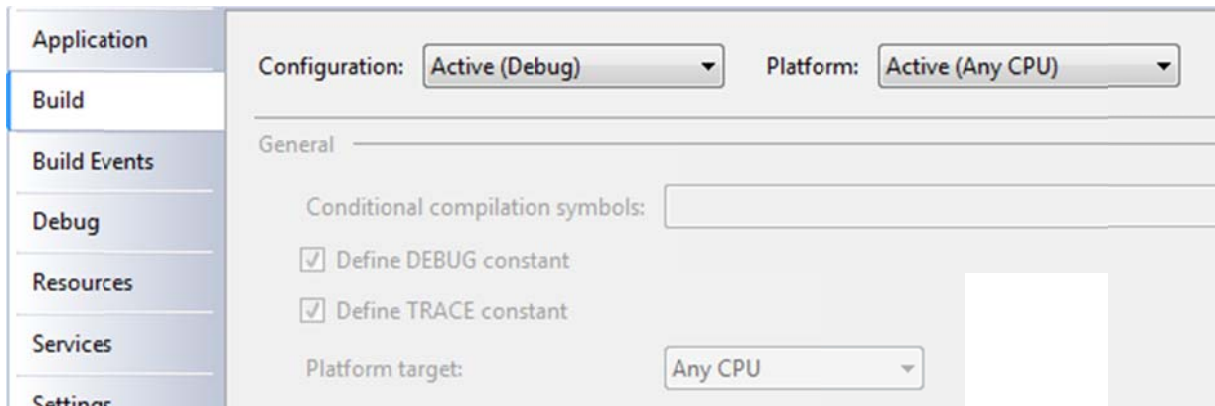
DA Ausgabe: ca. 20 µsec!

64- Bit Version

Wenn man auf einem 64 –Bit – PC eine 64 Bit – Version kompilieren will, so muss die dll *meIDSmain.dll* durch die *meIDSmain64.dll* ersetzt werden. Diese findet man in dem Verzeichnis

Computer > Lokaler Datenträger (C:) > Programme (x86) > Meilhaus Electronic > ME-iDS > 01.03.07.000 > System > ME-iDS (Driver)

Diese kopiere man in das aktuelle Verzeichnis bin/Debug des Projektes. Man muss zusätzlich dann natürlich in der Datei *meIDS.cs*, die ja die Links auf die Dll Datei erhält, alle 51 Links auf diese Dll ändern. Dann kann man mit 64 Bit kompilieren, also in Project → ...Properties, dann in Tab Build:



Dann funktioniert es, aber auch nicht schneller, aber das gesamte Projekt kann ja Teile enthalten, die mit 64 Bit schneller abgearbeitet werden.

Gezeichnet Prof. Dr. Bayerlein