

HÖHERE TECHNISCHE BUNDESLEHRANSTALT HOLLABRUNN

Höhere Abteilung für Elektronik – Technische Informatik

Klasse / Jahrgang: 4BHEL	Gruppe: 4b	Übungsleiter: Dipl.-Ing. Franz Geischläger
Übungsnummer: IV / 3	Übungstitel: LabView – GPIB und DAQ	
Datum der Übung: 28.04.2015	Teilnehmer: Graussam, Holzinger, Mischling, Schwarzmann	
Datum der Abgabe: 12.05.2015	Schrittführer: Graussam	Unterschrift:

	Beurteilung
Aufgabenstellung	
Dokumentation	
Messschaltungen	
Messtabellen	
Berechnungen	
Programmlistings	
Auswertung	
Diagramme	
Berechnungen	
Simulationen	
Schlussfolgerungen	
Kommentare	
Inventarliste	
Messprotokoll	
Form	
Summe	

1 Allgemeiner Teil

Titel der Übung: LabView – GPIB und DAQ
Übungsnummer: IV / 3
Übungsplatz: T1
Datum der Übung: 28.04.2015
Klasse: 4BHEL
Schriftführer: Lukas Graussam
Übungsteilnehmer: Lukas Graussam
Thomas Holzinger
David Mischling
Philipp Schwarzmann

Inhaltsverzeichnis

1	Allgemeiner Teil.....	2
2	Generelle Informationen	3
3	ADC / DAC.....	3
3.1	Schaltung	3
3.2	Hardwaretest	3
3.3	Task – Erstellung.....	5
3.4	LabView.....	7
4	GPIB-Kommunikation	9
4.1	Schaltung	9
4.2	NI MAX - GPIB	9
4.3	LabView – GPIB	10
4.4	Library	11
5	Erkenntnisse.....	12

2 Generelle Informationen

Bei der Übung wurde ein Basiswissen über LabView und GPIB (General Purpose Interface Bus) vorausgesetzt.

Folgende Programmversionen und Betriebssystem wurden bei der Übung verwendet:

LabView: 14.0

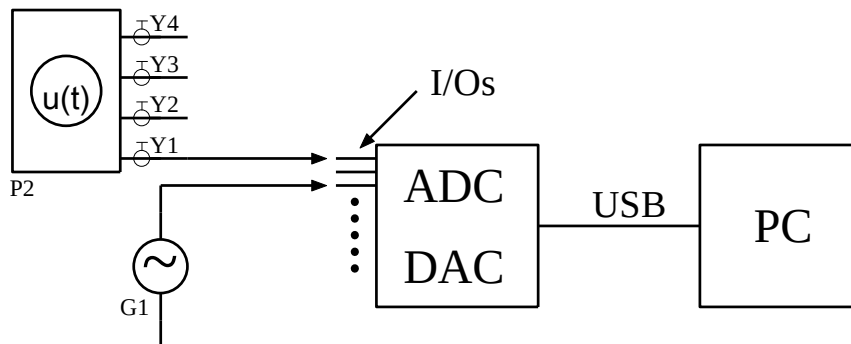
NI MAX: 14.0.0f0

Betriebssystem: Windows 7 Enterprise

3 ADC / DAC

Im ersten großen Teil der Übung ging es um DAQ (Data Acquisition, dt.: Datenerfassung). Dafür wurde ein Analog/Digital – Input/Output Gerät von National Instruments verwendet. Im Wesentlichen handelt es sich dabei um ADCs und DACs.

3.1 Schaltung

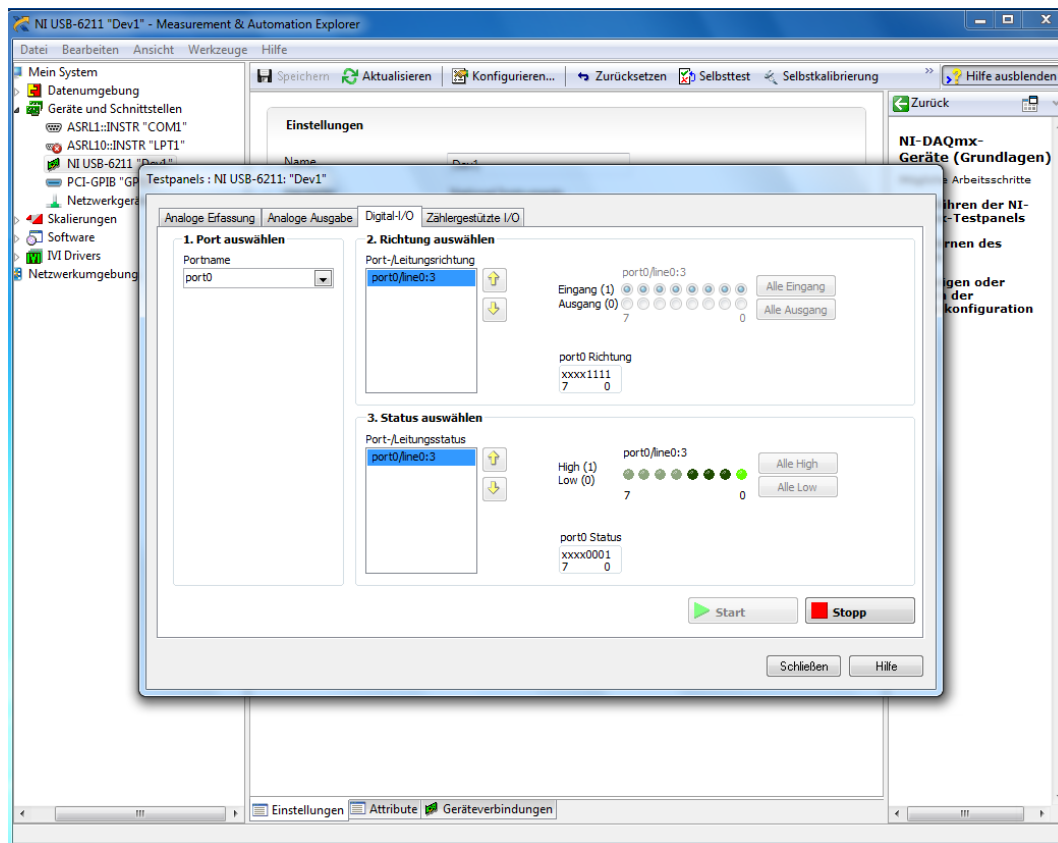


3.2 Hardwaretest

Der erste Schritt war es, die Hardware (den ADC/DAC) zu testen. Dies wurde mit der Software NI MAX durchgeführt.

Testpanel: Rechtsklick auf das Device -> Testpanel

Zuerst wurden die **digitalen** Ports getestet:

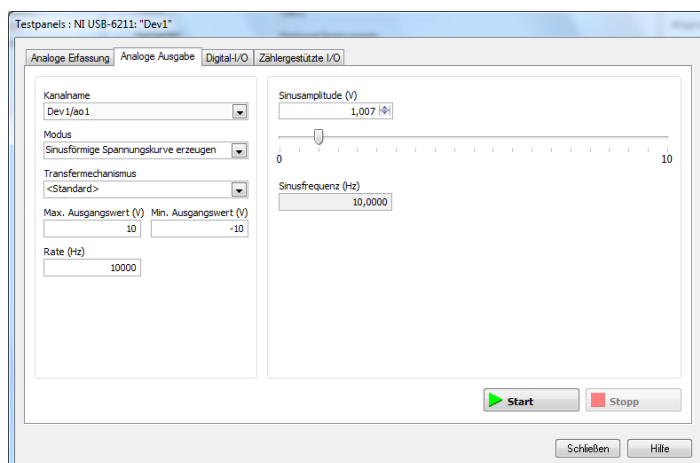


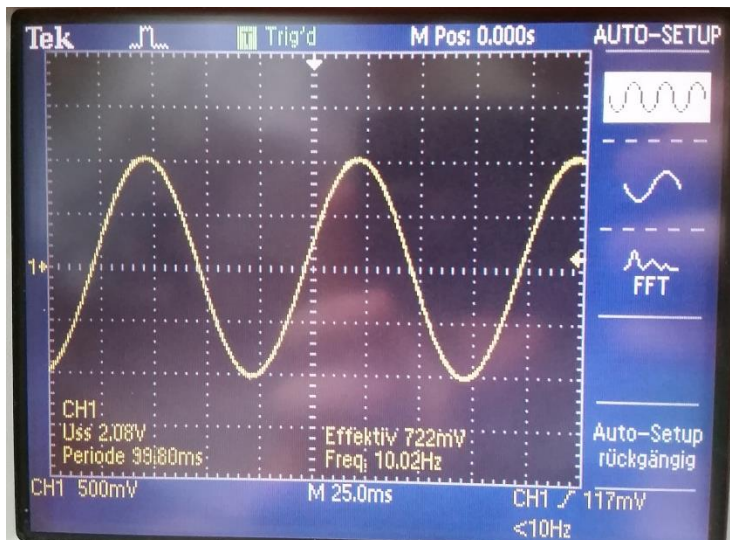
Zuerst wurden die **Input** Ports getestet. In diesem Fall wurde am Port 0 mit einem Drahtbügel 5V angelegt.

Auch der digitale **Output** wurde getestet. Geprüft wurde dies mit dem Oszilloskop P2.

Anschließend wurde der **analoge Output** getestet. Dafür wurde ein Sinus mit 10Hz ausgegeben und mit dem Oszilloskop an den entsprechenden Ports gemessen.

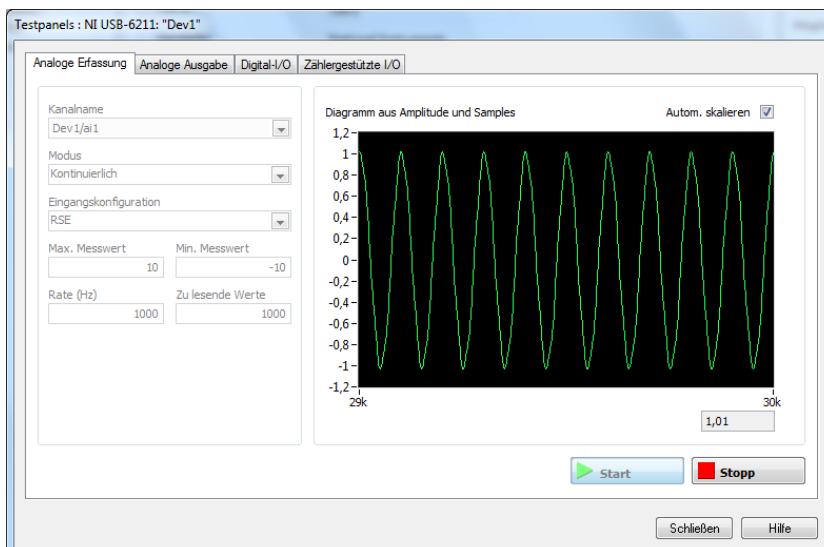
Folgend ein Screenshot von der analogen Ausgabe und vom Oszilloskop.





Anmerkung: Es ist zu erkennen, dass die Signale übereinstimmen.

Zum Abschluss des Tests wurde noch der **analoge Input** getestet. Dafür wurde mit dem Generator G1 ein Sinus mit $2V_{pp}$ und einer Frequenz von 10Hz an den zu testenden Ports erzeugt. Hier der Screenshot vom Testpanel:



In unserem Fall verliefen alle Hardware-Tests ohne Probleme.

3.3 Task – Erstellung

Die nächste Aufgabe war es, sogenannte Tasks zu erstellen. Diese können von LabView verwendet, um auf erfasste Daten zuzugreifen, oder um Daten auszugeben.

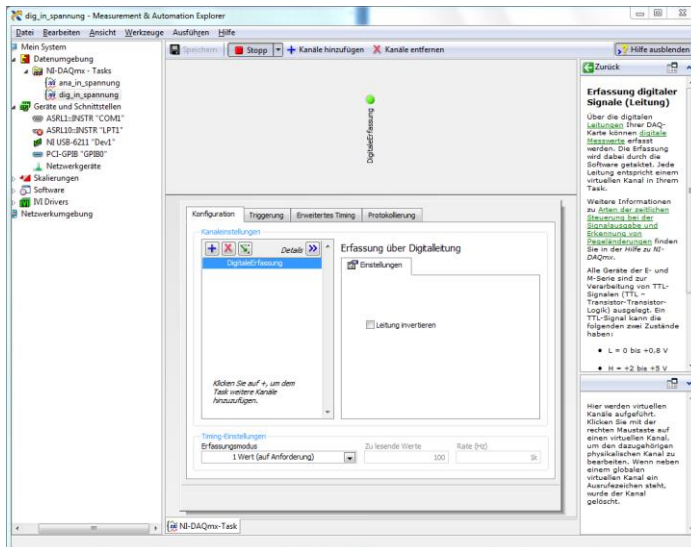
Task erstellen: Rechtsklick auf Device -> Task erstellen -> Signal erfassen / erzeugen -> Digital/Analog – I/O -> Art (Spannung, Strom, ...) -> Kanäle (Ports)

Auf diese Weise mussten 4 Tasks erstellt werden. Analog sowie digital jeweils ein Task für Input und Output.

Zur Sicherheit wurden die Input Tasks direkt mit NI MAX nochmals getestet.

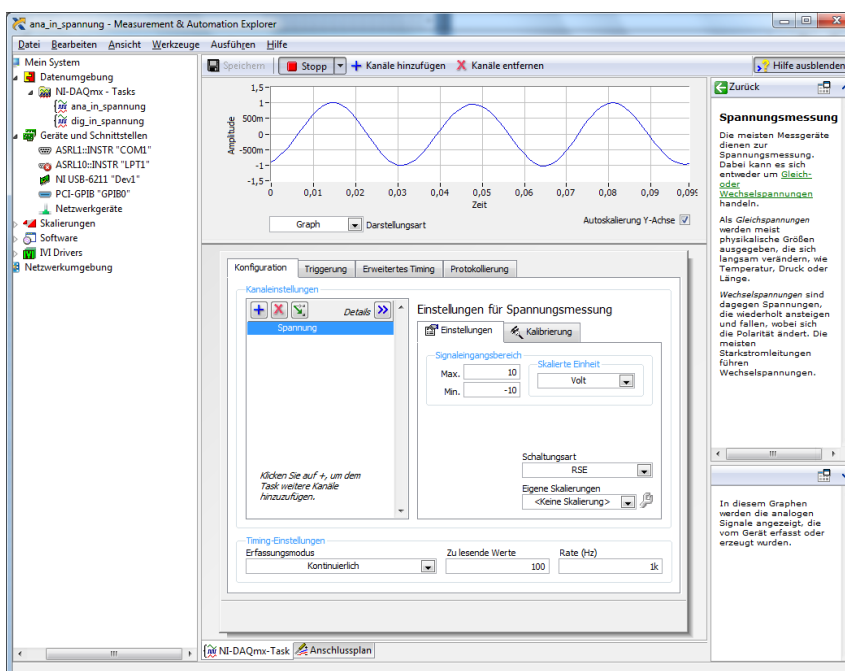
Task - Digital Input:

Dieser wurde wieder mit einer einfachen Drahtbrücke getestet. Hier ein Screenshot des Tests:



Task – Analog Input:

Dieser wurde mit dem Generator getestet. Ein Signal mit 2V_{pp} und 30Hz wurde dafür verwendet. Hier wieder ein Screenshot des Tests:



Auch diese Tests funktionierten ohne Probleme.

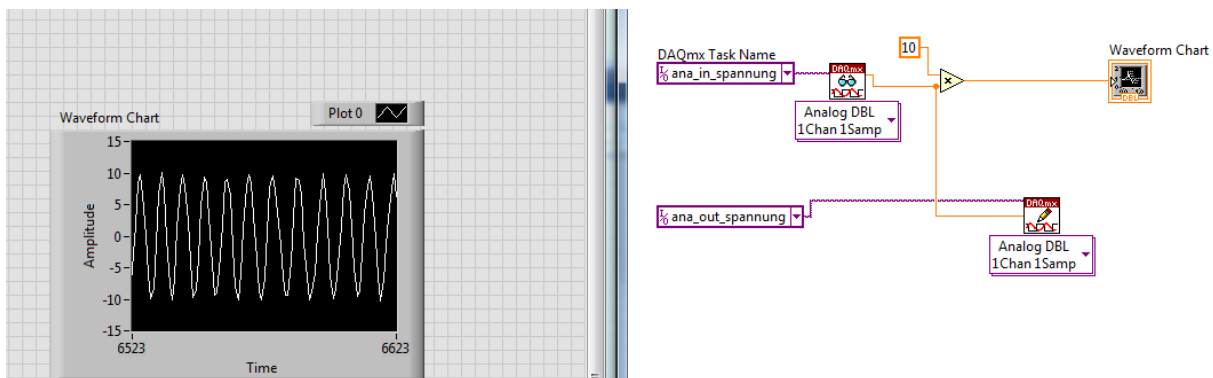
Anmerkung: Der Erfassungsmodus wurde für die weitere Verwendung auch hier auf „Wert auf Anforderung“ gestellt.

3.4 LabView

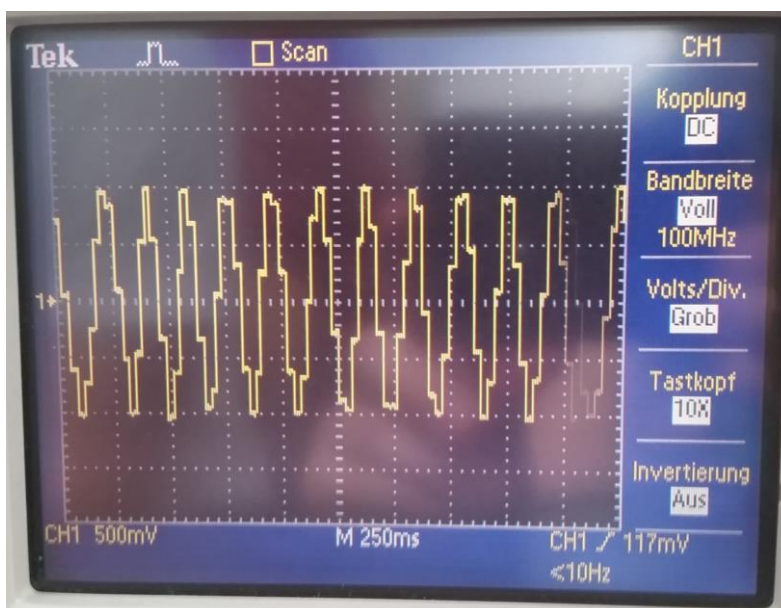
Die nächste Aufgabe war es, ein sinnvolles LabView Programm zu realisieren, welches auf die erstellten Tasks zugreifen sollte. Tasks (I/Os) in LabView:

Rechtsklick -> Measurement I/O -> NI DAQ -> Read / Write

Zuerst erstellten wir eine einfache Verbindung mit dem **analogen** Input und dem analogen Output in LabView. Um das Signal auch am PC-Bildschirm verfolgen zu können, ergänzten wir den Aufbau mit einem Multiplikator und einer Anzeige (Waveform Chart):

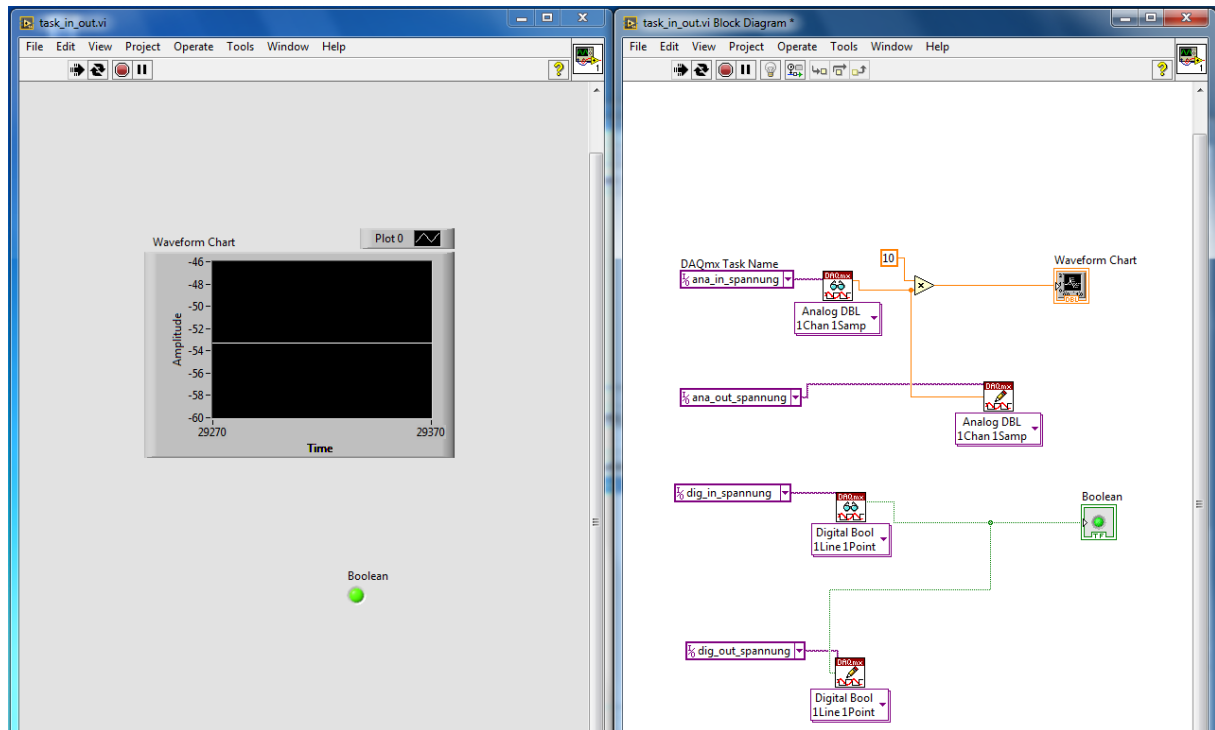


Am Screenshot kann man bereits unser Testsignal erkennen. Mit dem Generator G1 wurde ein 5Hz Sinus für den analogen Eingang erzeugt. Der Ausgang wurde außerdem mit dem Oszilloskop P2 gemessen:

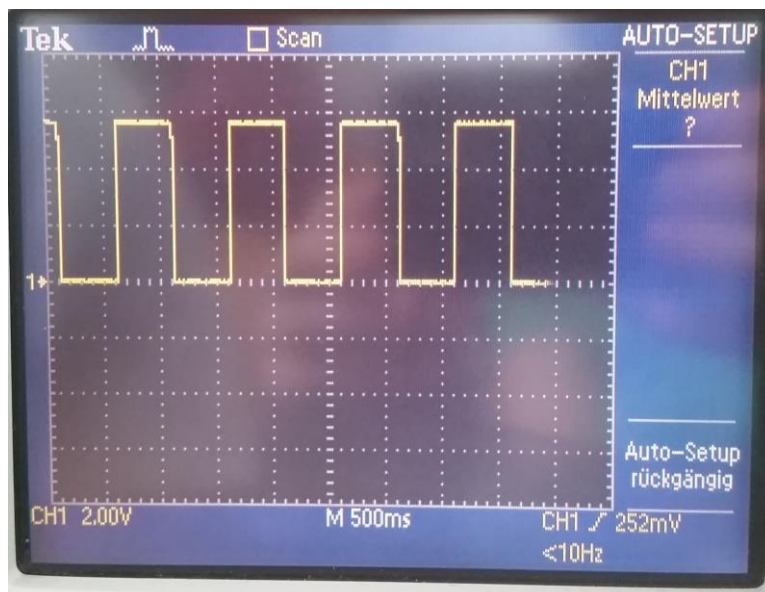


Der Aufbau funktionierte einwandfrei. An dem Signalverlauf kann man jedoch erkennen, dass der Aufbau für höhere Frequenzen nicht geeignet ist.

Dasselbe wurde danach mit dem **digitalen** Input und dem digitalen Output in LabView realisiert. Es wurde jedoch nur eine LED-Anzeige ohne Multiplikator verwendet:



Als am digitalen Input Port wurde mit dem Generator G1 ein Rechteck mit 5V_{pp} und einer Frequenz von 1Hz angelegt. Am digitalen Ausgang wurde wieder mit dem Oszilloskop P2 gemessen:

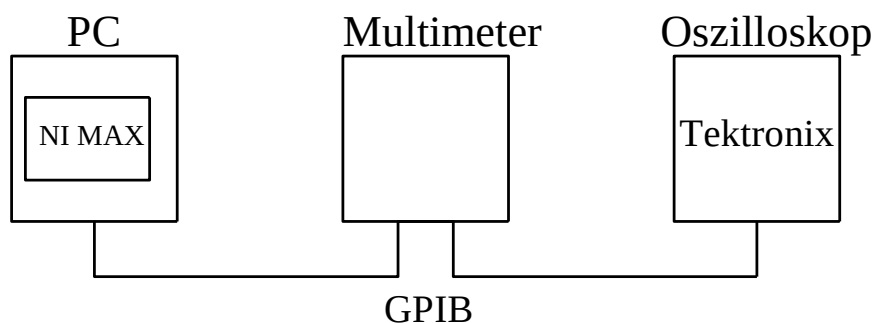


Die LED in LabView blinkte und es funktionierte alles einwandfrei.

4 GPIB-Kommunikation

Die nächste Aufgabe war es, eine Verbindung von der NI MAX Software mit dem Tektronix Oszilloskop P2, welches auch für die Messungen zuvor verwendet wurde, und mit dem etwas älteren digitalen Multimeter P1 herzustellen. Dies sollte mit dem GPIB-Bus geschehen.

4.1 Schaltung



4.2 NI MAX - GPIB

Zuerst musste ein GPIB Task in NI MAX auf diese Weise erstellt werden:

Geräte und Schnittstellen -> Rechtsklick auf GPIB -> Scan for Instruments

Beide Messgeräte wurden erkannt und folgende Adressen festgestellt:

Tektronix (P2): GPIB0::22::INSTR

Multimeter (P1): GPIB0::16::INSTR

Die Adressen sind auch manuell veränderbar, beim Oszilloskop mit diesen Schritten:

Menü -> Optionen -> GPIB Einstellungen -> Adresse

Um die Funktion zu testen änderten wir die Adresse von 22 auf 18. Der Sinn dieser Möglichkeit ist, vermeiden zu können, dass Messgeräte dieselbe Adresse besitzen.

Die Adresse des Multimeters wurde nicht geändert. Nach der Adressänderung musste in NI MAX nochmals „Scan for Instruments“ ausgelöst werden.

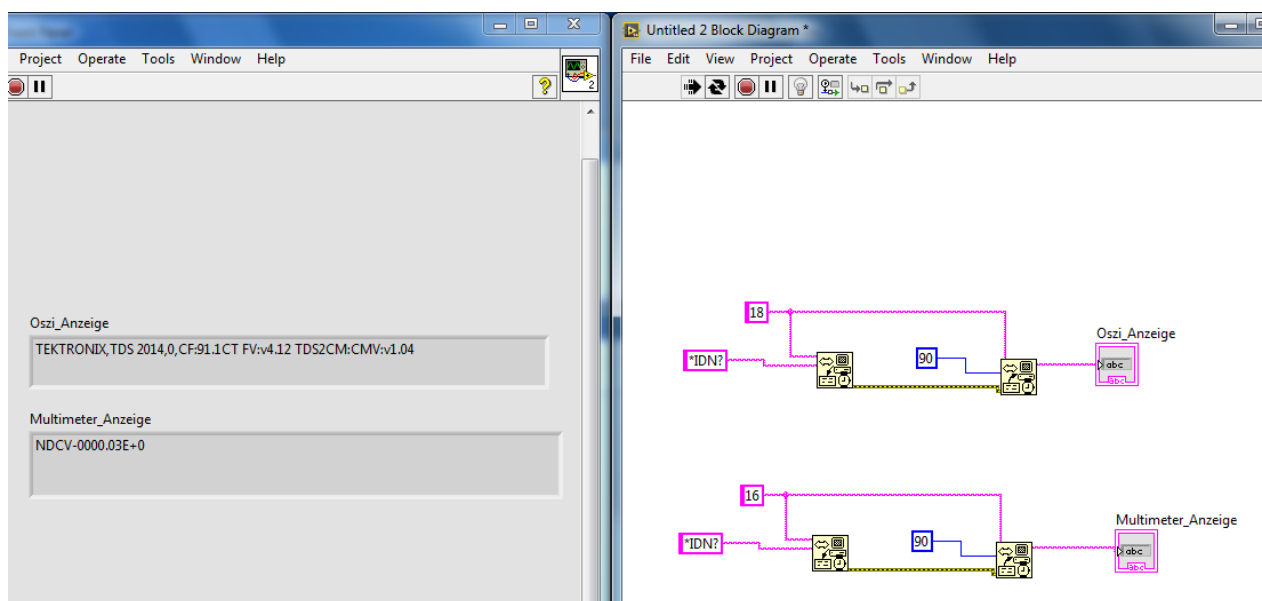
4.3 LabView – GPIB

Als nächstes sollte ein LabView Programm erstellt werden, welches den beiden Messgeräten den Befehl „*IDN?“ schicken sollte. Diese Query fordert Geräte auf, ihre ID zu senden.

In LabView auf den GPIB Task zugreifen:

Rechtsklick -> Instruments -> GPIB -> Read / Write

Natürlich musste überall die jeweilige Adresse „angeschlossen“ werden. Des Weiteren muss bei dem Write Task der String „*IDN?“ verknüpft sein. Beim Read Task gibt es noch eine Byte Count Grenze, bei der „90“ eingestellt wurde. Es wurde jeweils ein Textfeld als Anzeige verwendet. Außerdem wurden jeweils beim Read und Write Task die Fehlerleitungen verbunden, damit die Reihenfolge der Befehle garantiert ist:

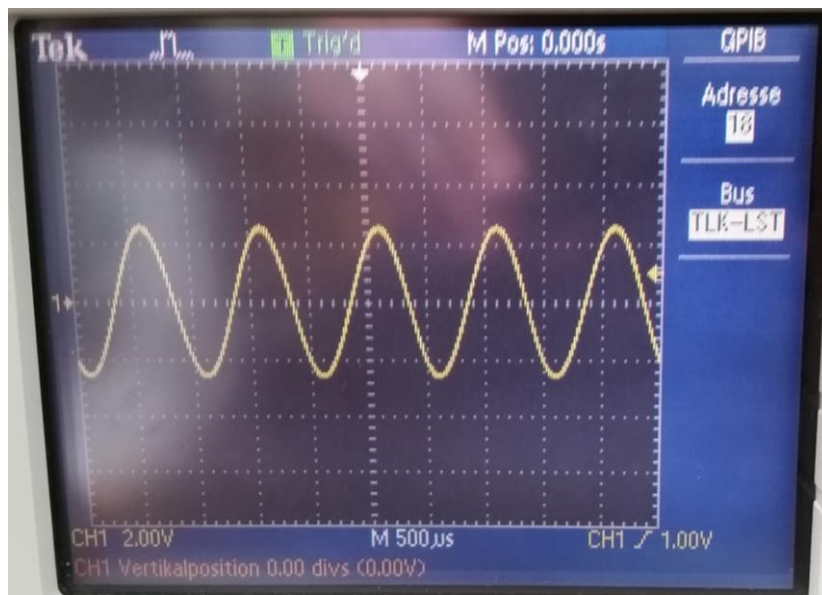
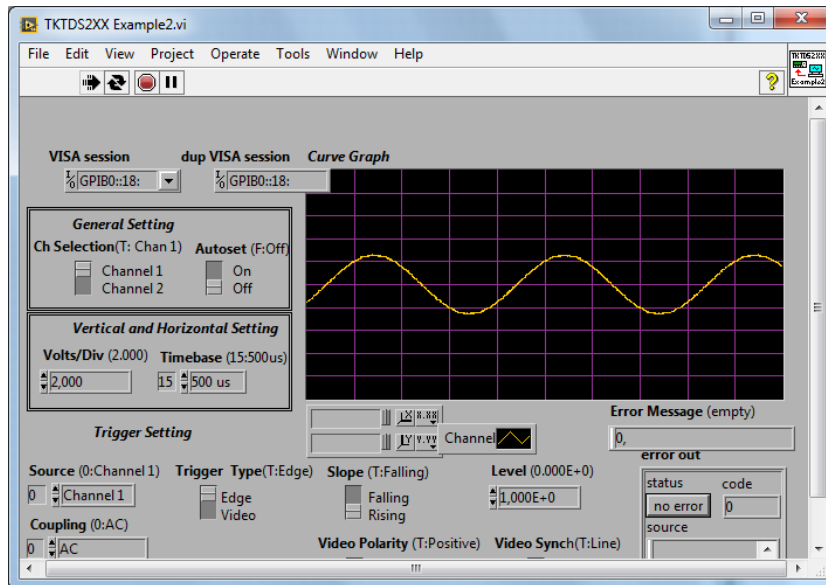


Wie hier zu sehen ist, funktionierte der Befehl bei dem **Tektronix** Oszilloskop (P2) **einwandfrei**. Beim **Multimeter** wird jedoch ein Messwert zurückgegeben. Das liegt daran, dass das Multimeter zwar einen GPIB Stecker hat, aber eigentlich nur für den IEC-Bus (Vorläufer von GPIB) ausgelegt ist und das **GPIB-Protokoll nicht unterstützt**.

4.4 Library

Zum Abschluss wurde uns vom Übungsleiter eine fertige Library für LabView zur Verfügung gestellt, mit der das Tektronix Oszilloskop über den GPIB Bus gesteuert werden kann und auch der Signalverlauf angezeigt wird.

Hier ein Screenshot der Steuerung und darunter des Oszilloskops:



Gemessen wurde ein 1kHz Signal. Über den PC konnten nun Einstellungen getätigt werden wie zum Beispiel das Einstellen der Zeitbasis.

5 Erkenntnisse

- NI MAX kennengelernt
- Kenntnisse über LabView erweitert
- GPIB-Bus
- DAQ mit National Instruments Hardware und Software