



ERIK
BARTMANN

www.erik-bartmann.de

DAS EEBOARD VON DIGILENT

Das EEBoard der Firma Digilent | Erik Bartmann

Autor:	Erik Bartmann
Thema:	Das <i>EEBoard</i> der Firma <i>Digilent</i>
Dokumentversion:	1.1
Datum:	03.09.2012
Internet:	www.erik-bartmann.de

Inhalt

Einleitung.....	4
Funktionen.....	6
Wir verschaffen uns einen Überblick	7
Was ist überhaupt ein Breadboard?	8
Die Funktionsblöcke im Detail	9
Das Oszilloskope	10
Impulsformer	15
Kennlinienaufnahme	21
Der Waveform-Generator	32
Die unterschiedlichen Standard-Signalformen	32
Signalform Gleichstrom	33
Signalform Sinus	33
Signalform Rechteck	35
Signalform Dreieck	36
Eigene Signalformen kreieren	36
Wie hören sich die Signalformen an?	37
Das Powersupply	38
Das Powersupply ein- bzw. ausschalten.....	39
Einzelne Spannungsquellen ein- bzw. ausschalten	40
Einzelne Spannungsquellen regeln.....	40
Den Spannungs- bzw. Stromverlauf anzeigen.....	41
Exportieren der angezeigten Werte	41
Die Voltmeter	42
Der Logic-Analyzer	43
Die unterstützen Protokolle	43
Die Kontroll-Schaltflächen	44
Die Triggerung	44
Anzeigen von Binärcode	45
Der Digital-Pattern-Generator.....	48
Die Kontroll-Schaltflächen	48
Unterschiedliche Pattern.....	49
Static I/O	50
Der Taster (Eingabe)	50
Die Schalter (Eingabe)	51

Die LED's (Ausgabe).....	52
Die Siebensegmentanzeige (Ausgabe)	52
Der Slider (Eingabe).....	52
Der Progress Bar (Ausgabe).....	53
Abschließende Informationen.....	54

Einleitung

Wer sich für Elektronik interessiert, der hat sicherlich schon einmal einen Blick in ein Elektroniklabor riskiert und war von der Anzahl der dort stehenden Messgeräte geradezu erschlagen. Mir erging es da nicht anders und als ich das erste Mal mit meinem Vater auf der Arbeitsstelle war. Er arbeitete seinerzeit im *Deutschen Luft- und Raumfahrtzentrum* in Köln-Porz, und ich bekam ganz feuchte Hände, denn an einem Windkanal-Prüfstand war so einiges vertreten, was gut und teuer war. Angefangen von XY-Messschreibern über eine ganze Batterie von Oszilloskopen bis hin zu Funktionsgeneratoren bzw. Messgeräten zur Strom- bzw. Spannungsaufnahme. Und dann noch viele andere Gerätschaften, von denen ich nicht mal heute den Namen kenne. Für einen kleinen Elektronik begeisterten Jungen, der ich nun damals einmal war, hatte das schon etwas Überirdisches. Genug der Vorrede und des Schwelgens in der Vergangenheit. Woraus ich hinaus möchte, ist die Tatsache, dass die Entwicklung natürlich nicht stehen geblieben ist und sich auf dem Gebiet der Messgeräte so einiges getan hat.

Gerade im Zeitalter der Computertechnik, wo jeder Haushalt mit mindestens einem Rechner ausgestattet ist, haben findige Entwickler ein Gerät konstruiert, das mehrere elektronische Geräte, die man so in einem Elektroniklabor anzutreffen pflegt, vereint. Wollten wir uns z.B. ein *Oszilloskope*, einen *Funktionsgenerator*, ein *Power-Supply*, ein Voltmeter, einen *Logic-Analyzer*, um nur einige der Geräte zu nennen, anschaffen, dann würde dieses Vorhaben unsere Geldbörse ganz schon in die Knie zwingen.

Die Firma [Digilent](#) hat aus diesem Grund ein Gerät entwickelt, das sich **Electronics Explorer Board** – kurz **EEBoard** – nennt, und eine ganze Reihe von elektronischen Komponenten in einem Gerät vereint. Dazu gehören auch u.a. *die* Geräte, die ich gerade genannt habe. Werfen wir zu Beginn einen kurzen Blick auf das Board.

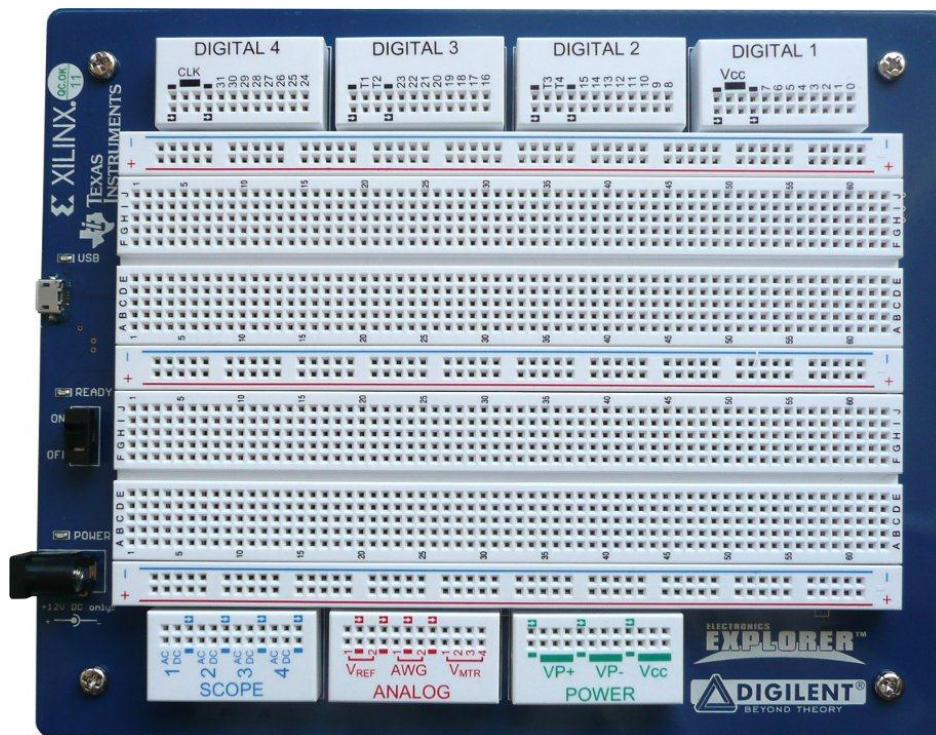


Abbildung 1 EEBoard

Manch einer mag sich jetzt vielleicht zu der Aussage hinreißen lassen, dass es sich wohl um ein ganz normales *Steckbrett* – auch *Breadboard* genannt – handelt. Nun, der erste Blick trügt in diesem Fall. Die ganze Power, die sich dahinter verbirgt, kommt dann zum Vorschein, wenn wir das Board einmal von Hinten betrachten.

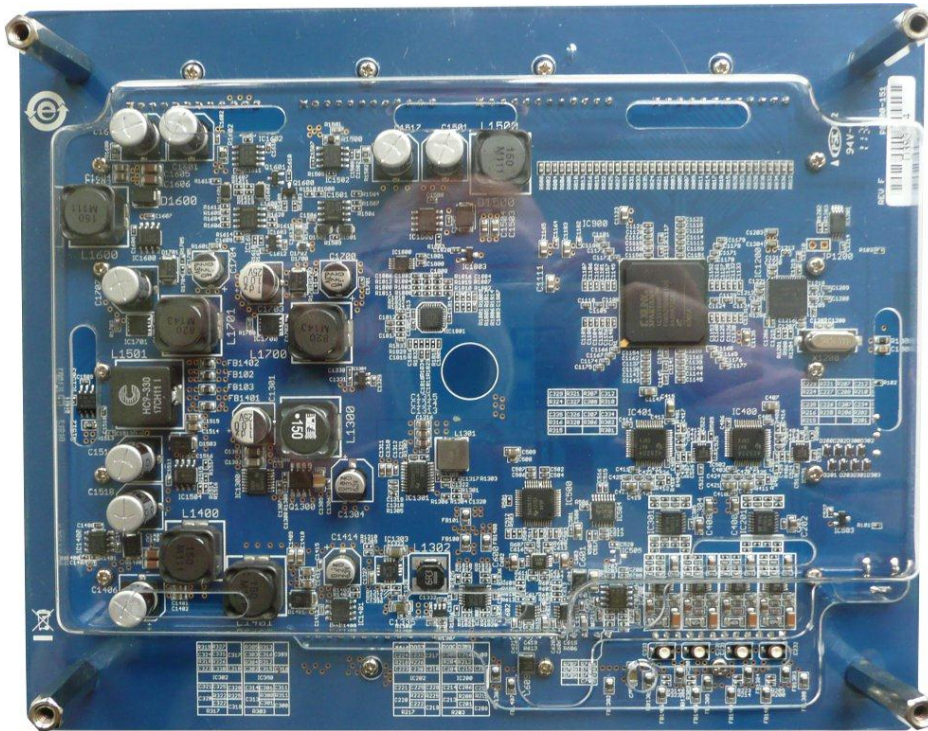


Abbildung 2 EEBoard von der Rückseite gesehen

Zu diesem Zeitpunkt können wir nur erahnen, was das Board im Stande ist zu leisten. Doch das werden wir in Kürze nachholen.

Funktionen

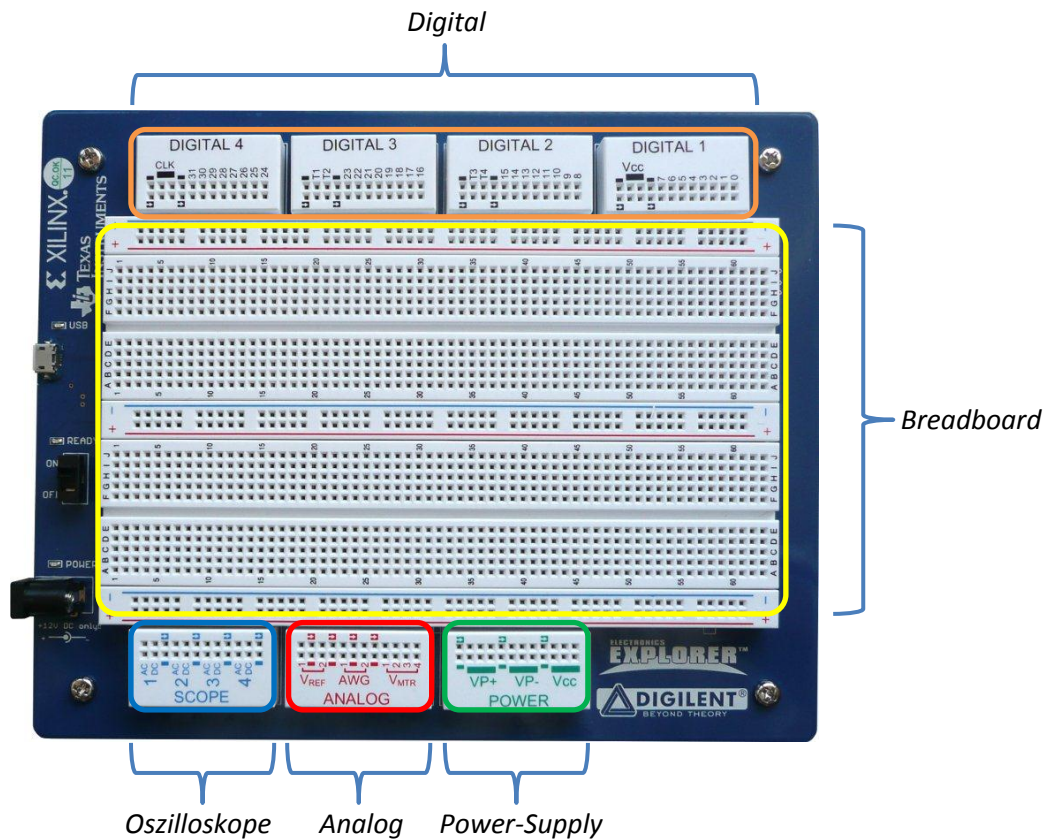
Kommen wir jetzt zu den einzelnen elektronischen Geräten, die das Board in sich vereint.

- *4 Kanal Digital Oszilloskope*
 - 40 MSps (Mega-Samples)
 - 10-Bit Converter mit bis zu 16 KByte Puffer
 - Leistungsstarke Trigger-Optionen
 - AC/DC Ankopplung mit +/- 20V Eingangsspannung
 - Schnelle Fourier-Transformation (FFT – Fast Fourier Transform)
 - XY-Darstellung
 - Zooming
 - Daten-Export in unterschiedliche Dateiformate
- *32 Kanal Logic Analyzer*
 - Eingangssignale bis zu 5V
 - 100 MSps (Mega-Samples)
 - 16 KSa/pin Puffer
 - Internal/External Clock und Trigger
 - Daten-Export in unterschiedliche Dateiformate
- *2 Kanal Waveform Generator*
 - Beliebige Waveform-Erzeugung
 - 40 MSps (Mega-Samples)
 - 14-Bit Converter mit bis zu 32 KByte Puffer
 - 4 MHz Bandbreite
 - 10 V Peak-To-Peak
 - AM/FM-Modulation
- *32 Kanal Digital-Pattern-Generator*
 - 2 KSa/pin Puffer
 - Unterschiedliche Pattern-Generierung (Zeit, Zähler, Konstant, Zufall, User-Definiert)
- *Power-Supply + Voltmeter*
 - 2 x Feste 5V + 3.3V Spannungsversorgungen (bis zu 1.5A)
 - 2 x Variable Spannungen von -10V bis +10V (bis zu 1.5A)
 - 4 Kanal Voltmeter (Eingangsschutz bis 200V mit 1.2 MOhm Impedanz)
 - 2 Referenz-Spannungen +/- 10V
- *Static I/O*
 - Taster
 - Schalter
 - LED's
 - Siebensegmentanzeige
 - Schieberegler
 - Fortschrittsanzeige

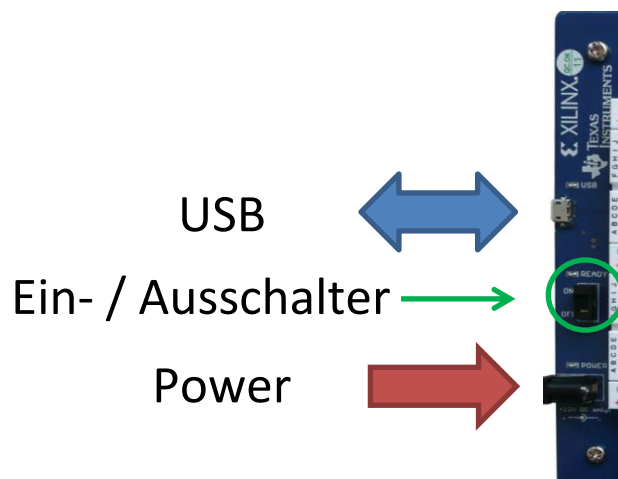
Die hier aufgezählten Funktionen werden wir in diesem Dokument näher beleuchten, so dass die Arbeitsweise klar wird und spätere elektronische bzw. elektrische Versuche mit dem Board keine Schwierigkeiten mehr bedeuten. Ich werde die einzelnen Bereiche jedoch nicht streng voneinander trennen, sondern je nach Thema ineinander fließen lassen.

Wir verschaffen uns einen Überblick

Bevor wir mit den Erläuterungen der diversen Funktionen beginnen, ist es sicherlich ratsam, sich vorab ein wenig auf dem Board zu orientieren. Damit es zu keiner Verwirrung hinsichtlich der Beschaltung kommen kann, sind die einzelnen Funktionsblöcke zusätzlich noch farblich hervorgehoben. Ich zeige euch auf dem folgenden Bild die operativen Ein- bzw. Ausgänge des Boards.



Natürlich muss das Board über die USB-Schnittstelle mit dem PC verbunden werden, damit der Datenaustausch zustande kommen kann. Zusätzlich wird das Board noch über ein separates Netzteil mit Spannung versorgt, da die USB-Schnittstelle die notwendigen Spannungen und Ströme alleine nicht liefern kann. Zum aktivieren bzw. deaktivieren des Boards ist zusätzlich noch ein Schalter vorhanden.



Was ist überhaupt ein Breadboard?

Ich habe den mittleren Bereich des *EEBoard* mit der Bezeichnung *Breadboard* versehen. Vielleicht stellt sich der eine oder andere jetzt die Frage: „Was um Himmels Willen ist denn ein *Breadboard*?“ Nun ja, es handelt sich – wenn ich einmal die Übersetzung an den Mann bringen darf – um ein *Steckbrett*. Hilft auch nicht so richtig weiter, oder!? Na jedenfalls sagt den Name *Steckbrett* etwas darüber aus, was die grobe Funktion dieses Boards ist. Wir können also etwas darauf stecken. Da wir es hier mit *Elektronik* zu tun haben, werden wir elektronische Bauteile wie z.B. *Widerstände*, *Kondensatoren*, *integrierte Schaltkreise* oder auch *Steckbrücken* – um nur einige zu nennen – darauf stecken. Ein Breadboard bietet also eine fantastische Möglichkeit, schnell und flexibel elektronische Schaltungen zusammen zu bauen, ohne einen Lötcolben zu benutzen. Diese Art des Schaltungsaufbaus wird auch *Prototyping* genannt. Es handelt sich um eine gängige Praxis, der sich u.a. auch professionelle Elektroniker bedienen, wenn es z.B. darum geht, eine neue Schaltung zu entwickeln. Du möchtest aber sicherlich wissen, wie die unzähligen Löcher, die *Buchsen* genannt werden, auf dem Board zu nutzen sind. Schau einmal her:

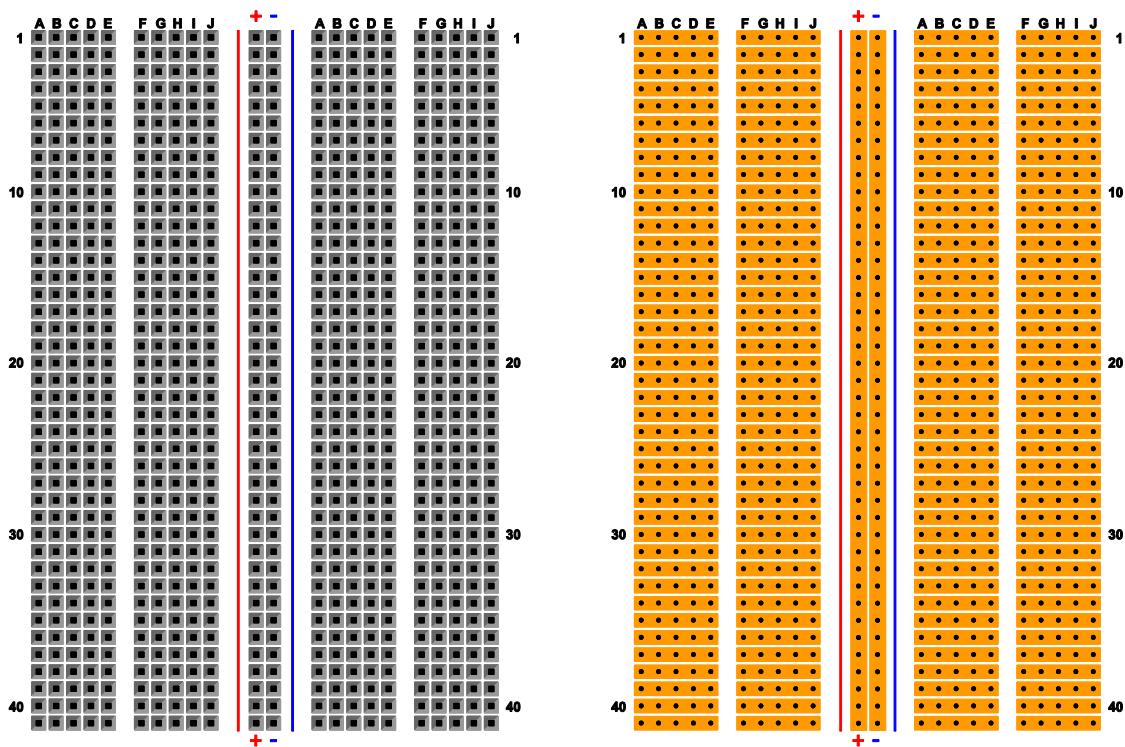


Abbildung 3 Ein Breadboard von außen (links) und von innen (rechts)

Hier siehst Du das *Breadboard* einmal von *außen* und einmal von *innen*. Die einzelnen Buchsen sind in den Spalten A bis E bzw. F bis J jeweils elektrisch waagerecht miteinander leitend verbunden. Es gibt auch senkrechte Verbindungen, die sich über das gesamte Steckbrett erstrecken. Sie sind hier mit + bzw. – gekennzeichnet und bilden die sogenannten *Power-Rails*. Das sind quasi Stromschienen, die die Spannungsversorgung komfortabel fast an jeden Punkt des Steckbrettes verteilen.

Möchtest Du z.B. einen integrierten Schaltkreis auf das Breadboard platzieren, dann sieht das wie folgt aus:

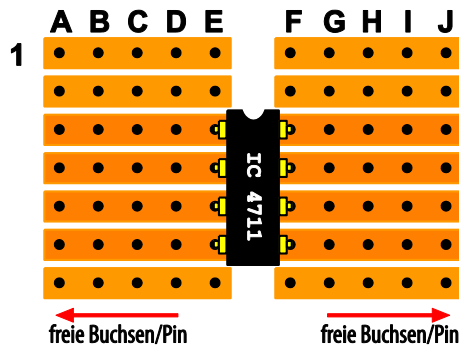
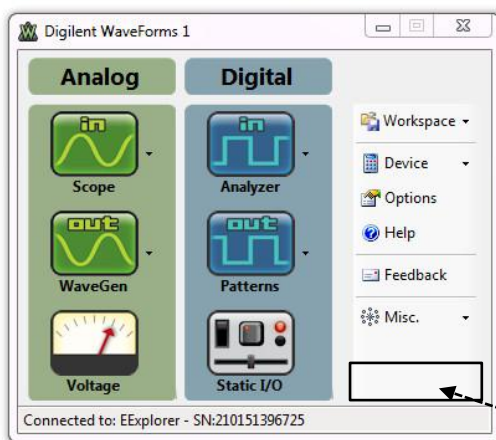


Abbildung 4 Ein integrierter Schaltkreis auf einem Breadboard

Wie Du siehst, ist das Bauteil in die etwas größere Lücke des *Breadboards* gesteckt worden und die einzelnen Anschlüsse werden jetzt nach links bzw. rechts weiter geführt, so dass Du dort die Steckbrücken einstecken kannst, um weitere elektrische Verbindungen zu schaffen, die z.B. zu weiteren elektronischen Bauteilen führen. Es wird also kein Lötkolben benötigt und der gesamte Aufbau ist sehr flexibel und kann jederzeit ohne größere Probleme angepasst werden.

Die Funktionsblöcke im Detail

Ich denke, dass es jetzt an der Zeit ist, sich mit den einzelnen angesprochenen Funktionsblöcken auseinander zu setzen. Bevor wir dies jedoch machen, sollte ich ein wenig auf die Software eingehen, die zur Kommunikation zwischen *EEBoard* und *PC* erforderlich ist. Das Board für sich alleine genommen ist zwar schön und gut, kann jedoch nicht ohne eine Anwendung auskommen, die die angesprochenen Funktionen verwaltet bzw. steuert. Diese Software nennt sich *Waveforms* und kann von der [Anbieterseite](#) kostenlos heruntergeladen werden.



Achtung:

Wenn Du den Schalter am *EEBoard* noch auf Position *OFF* stehen hast, dann wird in der Steuerzentrale *Board OFF* in der rechten oberen Ecke angezeigt. Ist das Fall, dann befindet sich das Board nicht im Arbeitsmodus.

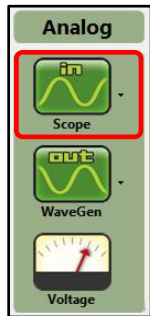


Abbildung 5 Waveforms Steuerzentrale

Der Startbildschirm der Waveforms-Anwendung ist in zwei Hauptbereiche - *Analog* bzw. *Digital* – unterteilt, der die einzelnen Funktionsblöcke unterstehen. Zudem gibt es noch einen Konfigurationsbereich zur Verwaltung diverser Parameter.

Das Oszilloskope

Das *Oszilloskope* ist im analogen Bereich beheimatet und wird über den ersten Button gestartet.



Bei einem *Oszilloskope* handelt es sich um ein elektronisches Messgerät, das elektrische Spannungen und deren zeitliche Verläufe optisch auf einem Bildschirm darstellt. Wie schaut so Bild eines Oszilloskopes – auch *Oszillogramm* genannt – eigentlich aus? Nun, das ist nicht so einfach zu beantworten, denn es verändert sich in Abhängigkeit des anliegenden Eingangssignales. Eines können wir aber festhalten. Es handelt sich um ein *2-Dimensionales* Bild, wobei es eine horizontale *X-Auslenkung* bzw. eine vertikale *Y-Ausrichtung* gibt. Sehen wir uns den einfachsten Fall einmal genauer an. Die *X-Auslenkung* wird durch eine interne Zeitbasis gesteuert und die *Y-Auslenkung* durch das an *Channel 1* anliegende Signal. Ich habe für unser erstes Beispiel ein *Sinussignal* mit dem Eingang verbunden, wobei ich noch nicht verraten möchte, wie groß die *Frequenz* bzw. die *Spannung* ist.

Wir sehen hier eine *Sinusschwingung*, die am *Channel 1* des Oszilloskopes anliegt.

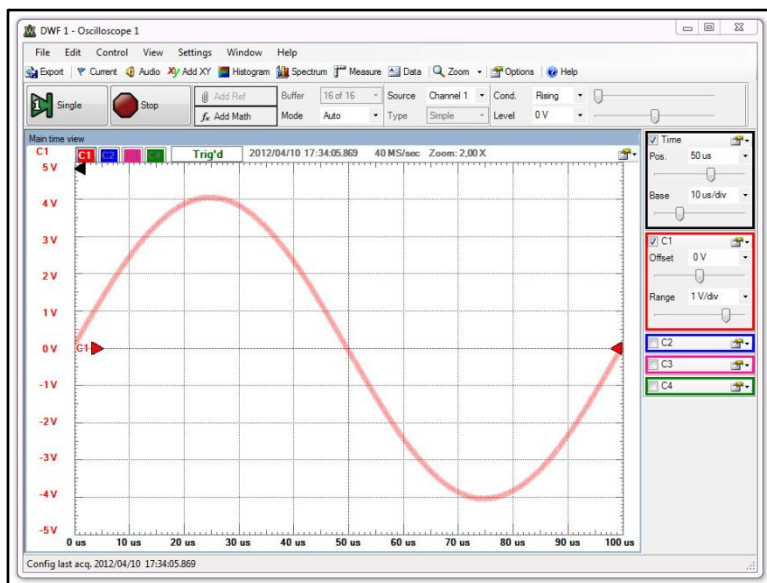


Abbildung 6 Eine Sinusschwingung

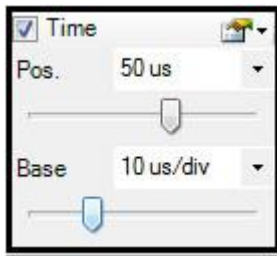
Ich sollte kurz die beiden großen Schaltflächen und deren Funktion erläutern:



Starten einer *einzigsten Aufnahme*, die zu diesem Zeitpunkt vorherrscht. Es handelt sich quasi um einen Snapshot.

Starten der *kontinuierlichen Aufnahme* des anliegenden Signales. Nach dem Start wandelt sich die Schaltfläche in eine *Stopp-Taste*, wie in der Abbildung zu sehen ist.

Bei einem reinen Sinussignal handelt es sich um eine periodische Schwingung und wir wollen zuerst einmal die *Frequenz* ermitteln. Dazu schauen wir uns einmal die horizontale *X-Achse* genauer an.



Die Einstellung, die uns erst einmal interessiert, ist die untere mit der Bezeichnung *Base* und bezieht sich auf die *X-Achse*, die in unserem Fall auf 10 us/div steht. Das sind 10^{-6} Sekunden pro Kästchen. Für eine komplette Schwingung werden 100^{-6} Sekunden benötigt. Wie können wir jedoch daraus die Frequenz f ermitteln? Ganz einfach:

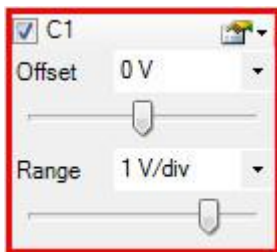
Die Formel dazu lautet:

$$f = \frac{1}{T}$$

Das Formelzeichen für die *Frequenz* wird mit einem kleinen f gekennzeichnet und die *Periodendauer* mit einem großen T . Was ist aber die *Periodendauer*? Das ist eben genau *die* Zeit, die für eine komplette Schwingung - einen einzigen Kurvenzug - benötigt wird. Also in unserem Fall sind das 100^{-6} Sekunden. Wir können dazu auch *100 Mikrosekunden* sagen, was wie folgt geschrieben werden kann:

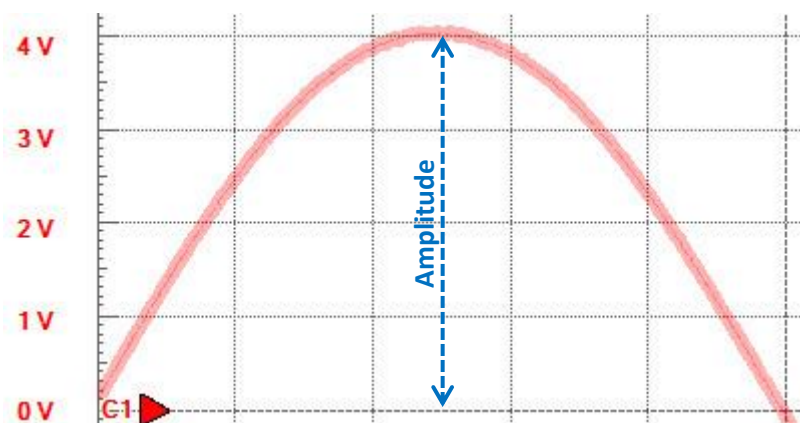
$$f = \frac{1}{100\mu\text{s}} = 10.000\text{ Hz} = 10\text{ KHz}$$

Die Einheit für die Frequenz lautet *Hertz* (Einheitenzeichen: *Hz*) und gibt die Anzahl der *Schwingungen pro Sekunde* an. Unsere Sinusschwingung hat demnach eine Frequenz von 10.000 Hz bzw. 10 KHz . ($K = \text{Kilo}$ und bedeutet 1000).



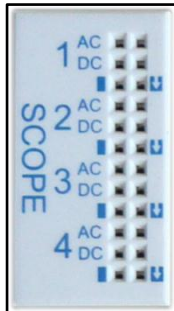
Neben der *Frequenz* des Eingangssignals ist natürlich noch die *Spannung* ein wichtiges Kriterium. Die *Spannung* können wir ohne langes bzw. aufwendiges Rechnen einfach ablesen. Warum? Nun, die Einstellung der *Spannung*, die ein einzelnes Kästchen in *Y-Richtung* einnimmt, wird über *Range* definiert bzw. festgelegt. Das sind in unserem Fall 1 V pro Kästchen. Beachte, dass das kleine Kästchen in der linken oberen Ecke bei *C1* ausgewählt ist. Das Häkchen muss also sichtbar sein. Ansonsten wird das am Kanal anliegende Signal nicht angezeigt.

Wie sieht das für unser Eingangssignal aus?

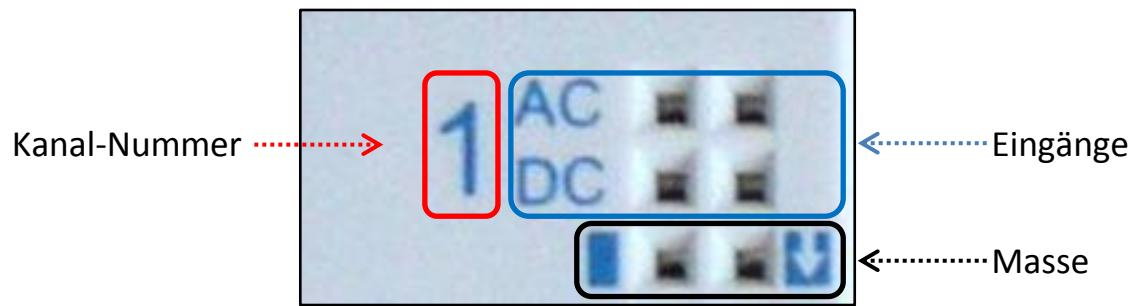


Das ist der Abstand der Sinusschwingung von der Nulllinie bis zur *Amplitude*, also der maximalen Auslenkung und beträgt bei uns ganze 4 Volt. Auf der linken Seite befinden sich die Wertangaben der Spannung für die jeweilige Zeile und das ist sehr einfach abzulesen.

Die Frage, die sich jetzt der eine oder andere stellt ist die, wo befinden sich die Eingänge des Oszilloskopes auf dem *EEBoard*?



Ich habe den entsprechenden Eingangsblock auf dem *EEBoard* einmal senkrecht dargestellt, so dass die Beschriftung der 4 Kanäle besser zu erkennen sind. Dafür hat sich die *Scope*-Bezeichnung gedreht. Man kann eben nicht alles haben 😊 Wenn wir uns einen einzelnen Kanal genauer anschauen, dann können wir folgendes erkennen:



Auf der linken Seite befindet sich die *Kanal-Nummer*, die in unserem Fall Kanal 1 ist. Rechts daneben befinden sich die Eingänge, die mit AC bzw. DC versehen sind. AC steht für *Alternating Current* und bedeutet *Wechselstrom*, wohingegen DC für *Direct Current* steht und *Gleichstrom* bedeutet. Es kann immer nur einer von beiden genutzt werden! Darunter befindet sich der *Masseanschluss* der beiden Eingänge, der beim Abgriff eines zu messenden Signales nicht vergessen werden darf. Masseanschlüsse werden sowohl mit einem Rechteck, als auch mit dem nach unten weisenden Pfeil gekennzeichnet.



Wie verändert sich denn die Anzeige eines Signals, wenn ich es einmal an AC und ein anderes Mal an DC anschließe?

Hey, wer bist Du denn?

Mein Name ist *Ardus* und ich lese den Kram, den Du hier zum Besten gibst. Wie sollen die Leser denn Fragen stellen, wenn ich es nicht für sie tue?



Ich denke, dass das ein guter Ansatz ist und die erste Frage, die Du gestellt hast, ist auch an dieser Stelle wirklich berechtigt! Verwendest Du den *AC-Eingang*, wird nur der *Wechselspannungsanteil* des Eingangssignals zur Anzeige gebracht. Ich denke, dass ich an dieser Stelle etwas weiter ausholen muss.

Bleiben wir beim *Sinussignal*, was ja quasi ein *Wechselspannungssignal* darstellt.

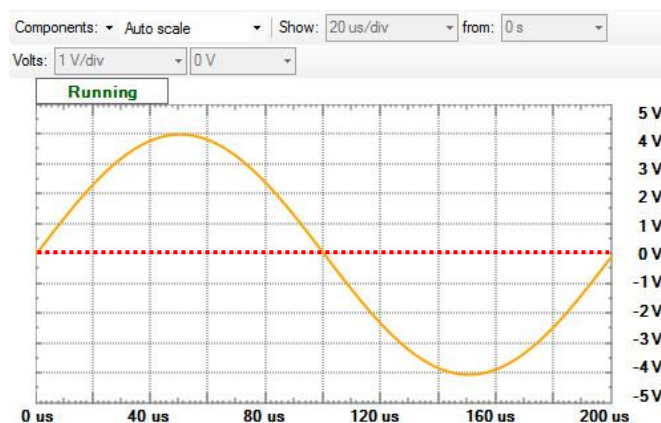


Abbildung 7 Sinussignal

Wenn Du Dir die Nulllinie (hier rot gestrichelt) der Sinuskurve anschaust, dann stimmt sie mit dem Spannungspegel von 0V überein. Dieses Signal besitzt keinen *Gleichspannungsanteil*. Was ist aber ein *Gleichspannungsanteil*, wirst Du Dich sicherlich fragen? Schau Dir die folgende Sinuskurve an und lenke Dein Augenmerk auf die Nulllinie der Kurve.

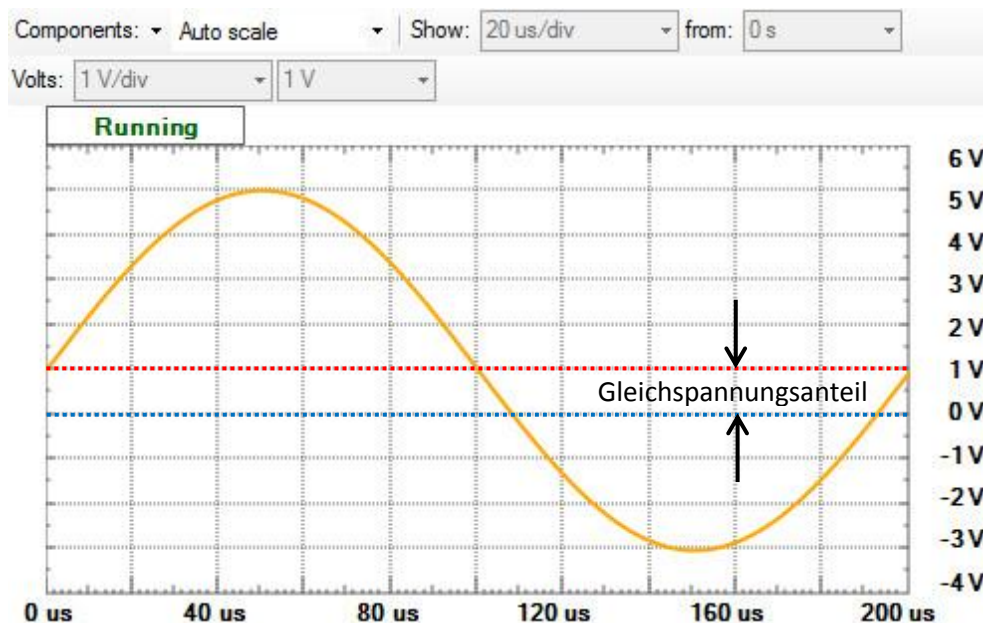


Abbildung 8 Sinussignal mit Gleichspannungsanteil von 1V

Die Sinuskurve ist um den Gleichspannungsanteil von 1V nach oben verschoben. Mit Hilfe des *Funktionsgenerators* kann ein derartiger Kurvenverlauf ohne Mühe erstellt werden.

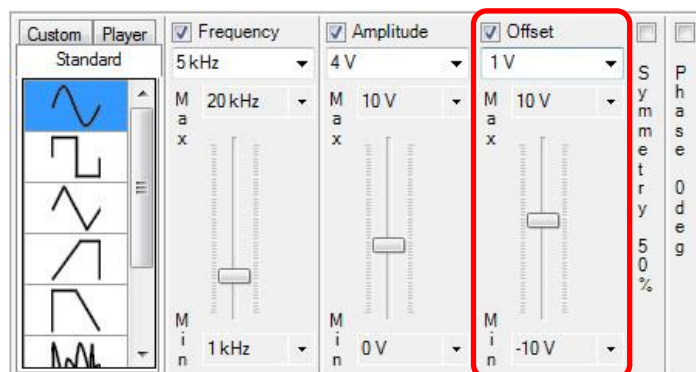
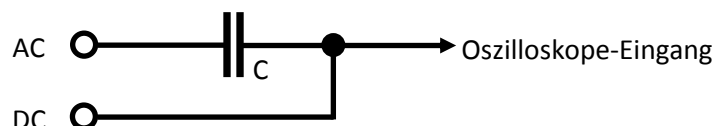


Abbildung 9 Einstellungen am Funktionsgenerator mit Offsetspannung

Im Bereich *Offset* können wir den besagten *Gleichstromanteil* einstellen, der hier auf 1V justiert ist. Doch dazu später mehr. Du wolltest aber eigentlich etwas anderes wissen. Was nämlich der Unterschied zwischen der AC bzw. dem DC Eingangsbuchse ist. Nutzt Du die AC-Eingangsbuchse, dann möchtest Du sicherlich nur den *Wechselspannungsanteil* Deines Eingangssignals zur Anzeige bringen. Wie können wir aber den ggf. vorhandenen Gleichspannungsanteil eliminieren? Dazu wird der DC-Eingangsbuchse quasi ein *Kondensator* vorgeschaltet, der für eine *Gleichspannung* ein unüberwindliches Hindernis darstellt bzw. einen unendlich großen Widerstand bedeutet, wohingegen er für eine Wechselspannung mehr oder weniger durchlässig ist. Das ist jedoch von der vorherrschenden Frequenz abhängig, die wir an dieser Stelle aber erst einmal vernachlässigen wollen. Du kannst Dir das folgendermaßen vorstellen:



Impulsformer

Ich würde einmal vorschlagen, dass wir uns zwei interessante Schaltungen anschauen, die wir jedoch nicht mit dem – sagen wir mal – sonst üblichen *Sinussignal* versorgen, sondern mit einem *Rechtecksignal*. Bei den beiden Schaltungen handelt es sich um einen *Tief-* bzw. *Hochpass*. Ich kann vorausnehmen, dass das hineingeschickte Rechtecksignal in einer bestimmten Weise *verformt* wird und auf eben diese *Verformung* kommt es uns an. Sie wollen wir genauer betrachten.

RC-Integrierglied

Fangen wir mit dem *Tiefpass* an, der auch *RC-Integrierglied* genannt wird. Hört sich ja ziemlich wichtig und kompliziert an, aber es handelt sich um eine einfache Form der Zusammenschaltung von *Widerstand* und *Kondensator*, die wir uns genauer anschauen wollen. Zuerst sollten wir einen Blick auf die Schaltung werfen, damit Du eine Vorstellung davon bekommst, wovon ich hier rede.

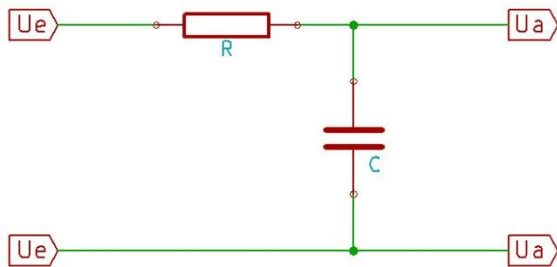
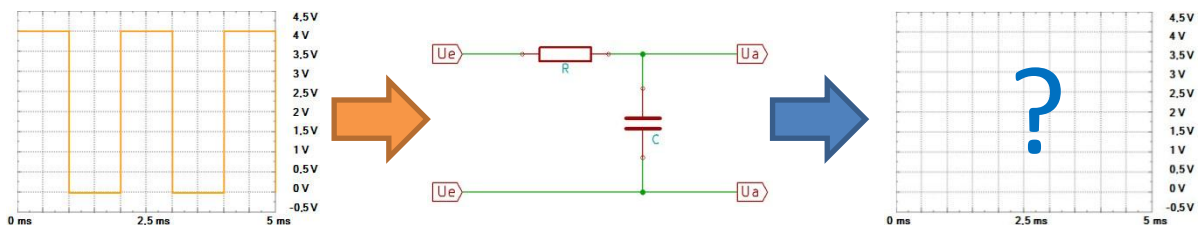


Abbildung 10 RC-Integrierglied

Auf der linken Seite haben wir den Eingang und auf der rechten den Ausgang. Was passiert jetzt mit dem angelegten Rechteck-Eingangssignal auf dem Weg durch diese Schaltung? Klar, ich hatte es schon verraten! Es erfolgt eine Art von *Verformung*. Wie stellt sie sich aber dar und von welchen Faktoren hängt sie ab?



Bei einem *RC-Integrierglied* handelt es sich – wie schon erwähnt - um einen *Tiefpass*. Wie der Name schon klar zu erkennen gibt, lässt diese Schaltung tiefe Frequenzen besser durch, als hohe.



Jetzt erkläre er mir doch mal bitte, warum das so ist? Warum lässt dieser *Tiefpass* niedrige Frequenzen besser passieren, als höhere?

Nun *Ardus*, ich habe Dir ja schon gesagt, dass ein Kondensator für ein Wechsignal wie z.B. eine Sinusspannung kein Hindernis darstellt.

Der Kondensator verhält sich durch das ständige Auf- bzw. Umladen durch diese Spannung wie ein Widerstand, der frequenzabhängig reagiert. Je höher die Frequenz, desto geringer ist der Widerstand. Wir haben es aber an dieser Stelle mit keinem *Sinussignal* zu tun, sondern mit einem *Rechtecksignal*.

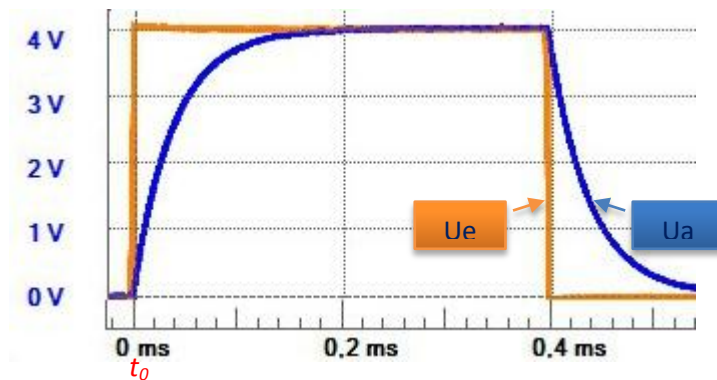


Abbildung 11 Ein- bzw. Ausgangssignal an einem RC-Integrierglied

Das Eingangssignal wird durch das orangefarbene Rechtecksignal dargestellt und das Ausgangssignal durch die blaue, leicht verformte Kurve. Wie kommt diese Verformung jetzt zustande?

Zum Zeitpunkt t_0 ($0ms$) steigt die Eingangsspannung sprunghaft an und benötigt für dieses Aktion theoretisch keine Zeit, was real natürlich nicht zu erreichen ist. Für diesen kurzen Augenblick bei t_0 ist der Kondensator C noch nicht geladen und stellt einen Kurzschluss dar, der am Ausgang sofort Wirkung in der Art zeigt, dass die Ausgangsspannung $0V$ beträgt. Erst mit der Zeit lädt sich der Kondensator über den Widerstand R auf, was Du daran erkennen kannst, dass die blaue Kurve sich langsam dem Maximum der orangenen Kurve nähert. Die Aufladezeit hängt maßgeblich von den Werten R und C ab und kann folgendermaßen formuliert werden:

$$\tau = R \cdot C$$

Der kleine griechische Buchstabe τ (Tau) steht für die Zeitkonstante, die die Zeit definiert, zu welchem Zeitpunkt der Kondensator auf 63% geladen ist. In meiner Schaltung habe ich die folgenden Werte verwendet:

$$\tau = 2K\Omega \cdot 20nF = 0,04ms$$

Nach $0,04ms$ beträgt die Spannung an Kondensator also 63% der Ladespannung U_e von $4V$.

$$U_c = 4V \cdot 0,63 = 2,52V$$

Das wollen wir doch einmal in unserem Oszillogramm überprüfen, obwohl ich an dieser Stelle erwähnen sollte, dass Bauteiltoleranzen und Ablesfehler zu ungenauen Ergebnissen beitragen können.

Aber schauen wir einmal genauer hin...

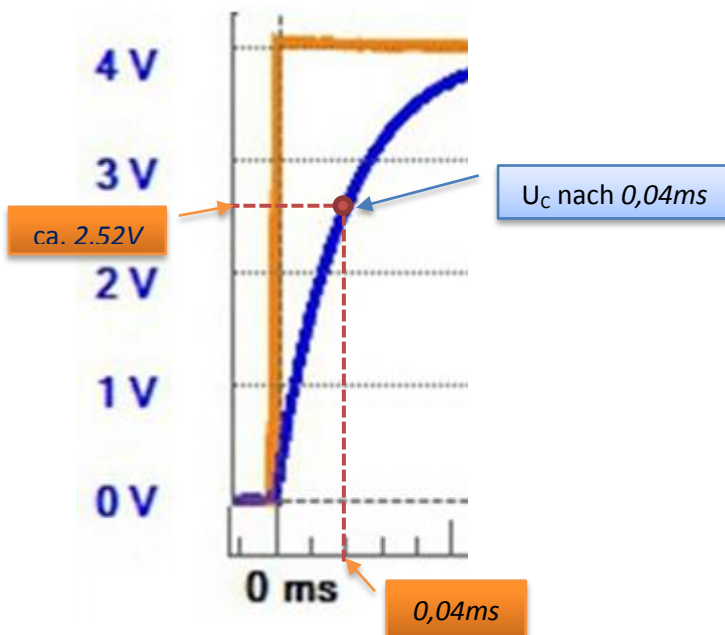


Abbildung 12 Spannung am Kondensator nach 0,04ms



Ok, das habe ich soweit verstanden. Wenn ich jetzt aber wissen möchte, nach welcher Zeit der Kondensator komplett geladen ist, wie komme ich dann an *das* Ergebnis?

Gute Frage an dieser Stelle, *Ardus*! Also ich muss hier vielleicht einmal betonen, dass ein Kondensator niemals völlig geladen, bzw. entladen wird. Wir können aber sagen, dass er nach einer Zeit von 5τ als geladen angesehen werden kann. Die Spannung beträgt dann ca. $0,99 \cdot U_e$.

Das habe ich zwar verstanden und dann doch wieder nicht. Wenn ich die Ladespannung nun 10 Tage angelegt lasse, ist der Kondensator dann nicht zu 100% geladen?



Das sollte man vielleicht vermuten, doch dem ist nicht so. Das hat folgende Bewandtnis. Der Kurvenverlauf der Ladespannung folgt einer sogenannten *e-Funktion* (natürliche Exponentialfunktion). Diese Funktion ist die Basis zur Berechnung aller Wachstums- bzw. Zerfallsprozesse, wobei e für die natürliche Zahl 2,718 281... steht.

Auf diese Weise kannst Du die Spannung bzw. den Strom nach einer bestimmten Zeit berechnen.

Nutze dafür die beiden Formeln:

$$U_C = U \cdot (1 - e^{-\frac{t}{\tau}})$$

$$I_C = \frac{U}{R} \cdot e^{-\frac{t}{\tau}}$$

CR-Differenzierglied

Das *CR-Differenzierglied* gleicht dem des *Integriergliedes*, jedoch sind hier Widerstand und Kondensator vertauscht. Schaltungstechnisch ist es ein *Hochpaß*, der hohe Frequenzen besser passieren lässt, als niedrige.

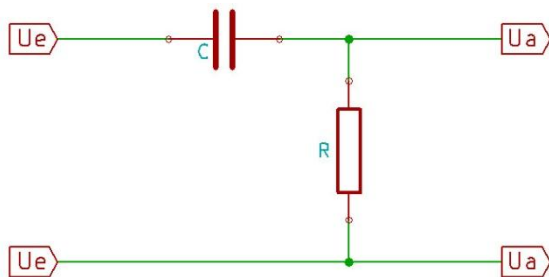
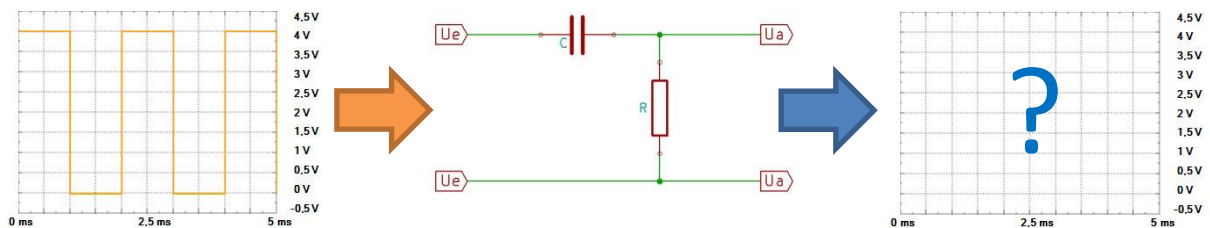


Abbildung 13 CR-Differenzierglied

Wie mag wohl der Kurvenverlauf am Ausgang aussehen?



Du erinnerst Dich sicherlich noch an die Feststellung, dass der Kondensator im ersten Moment einen Kurzschluss darstellt und den Strom ungehindert durchlässt. Erst einen kurzen Augenblick später, je mehr er sich auflädt, wird sein Widerstand größer und der Spannungsabfall steigt.

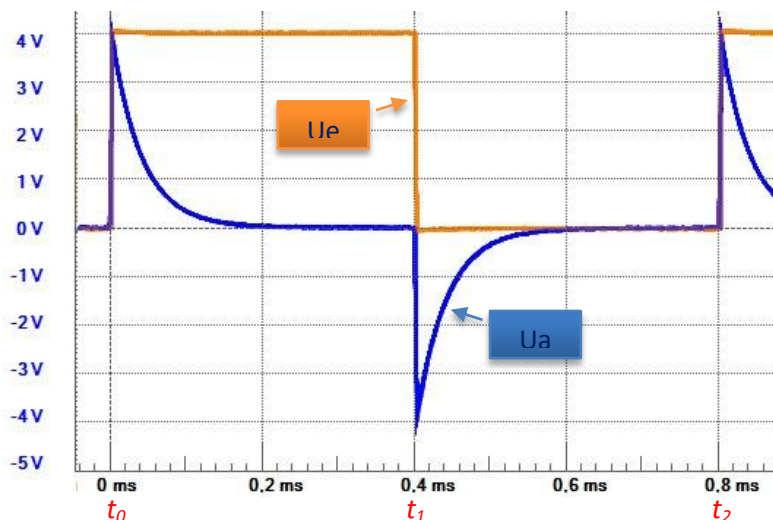


Abbildung 14 Ein- bzw. Ausgangssignal an einem CR-Differenzierglied

Das sieht doch sehr interessant aus, nicht wahr!? Zum Zeitpunkt t_0 (0ms) steigt die Eingangsspannung U_e sprunghaft an. Für diesen kurzen Augenblick bei t_0 ist der Kondensator C noch nicht geladen und stellt einen Kurzschluss dar, so dass die komplette Spannung am Widerstand R abfällt. Je mehr sich jedoch der Kondensator nach der schon bekannten *e-Funktion* auflädt, desto weiter sinkt die Spannung am Widerstand. Auch hier kommt die Zeitkonstante τ zum Tragen.

$$\tau = R \cdot C$$

Nach der Zeit von 1τ ist die Ausgangsspannung auf 37% gesunken und erreicht nach 5τ annähernd 0V.



Irgendetwas stimmt doch hier nicht so ganz. Wenn die Ausgangsspannung einen Peak nach oben macht, sehe ich das ja noch ein, doch warum geht der Peak dann später nach unten in den *negativen Bereich* rein? Das verstehe ich nicht!

Das ist natürlich eine berechtigte Frage und gut beobachtet! Wir schauen uns die einzelnen Phasen einmal genauer an, denn dann erkennst Du das Verhalten der Ausgangsspannung sicherlich sofort. Der ganze Vorgang läuft natürlich kontinuierlich ab, kann aber in 2 *Phasen* unterteilt werden:

- *Ladephase* des Kondensators
- *Entladephase* des Kondensators

Ladephase:

In der *Ladephase* zwischen den Punkten t_0 und t_1 liegt die Gleichspannung U_e von 4V am Differenzierglied an und der Kondensator C wird nach der *e-Funktion* geladen.

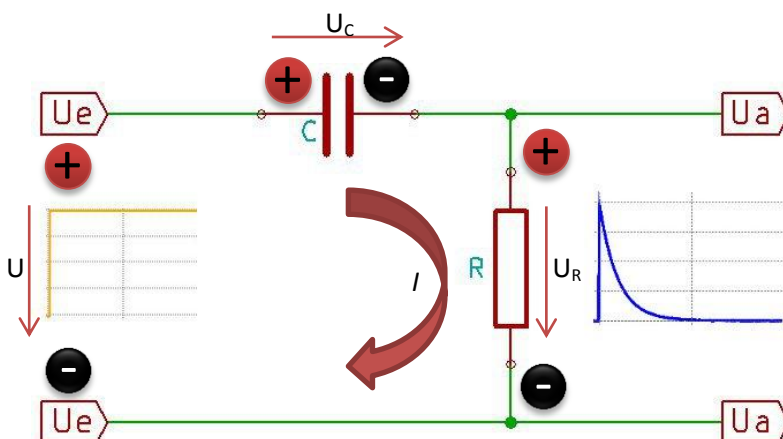


Abbildung 15 Stromfluss in der gezeigten Richtung (*Lade-Phase*)

Entladephase:

In der *Entladephase*, die zwischen den Zeitpunkten t_1 und t_2 liegt, ist die Spannungsquelle U_e abgeschaltet. Wenn wir davon ausgehen, dass der Innenwiderstand des Funktionsgenerators sehr viel niederohmiger ist, als der verwendete Widerstand R , dann wird sich der geladene Kondensator C seinerseits wie eine Spannungsquelle für den Widerstand R verhalten. Die Entladung erfolgt über den Widerstand R nach der *e-Funktion*.

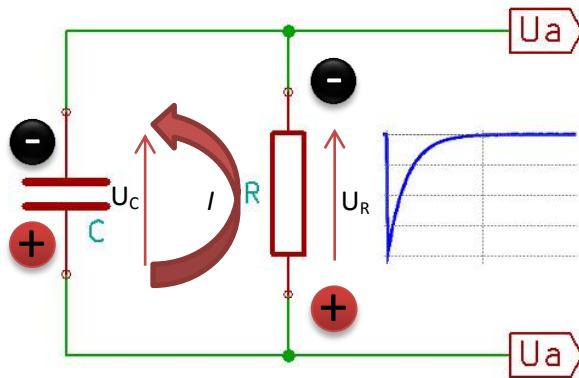


Abbildung 16 Stromfluss in der gezeigten Richtung (*Entlade-Phase*)

Der Stromfluss I hat jetzt aber seine Richtung geändert, da der Kondensator anstelle der Spannungsquelle als Energielieferant arbeitet und diese quasi ersetzt. Jetzt jedoch mit umgekehrten Vorzeichen.

Experiment

Wie werden sich wohl die die Ausgangssignale von *Integrierglied* bzw. *Differenzierglied* ändern, wenn Du z.B. zu dem vorhandenen Kondensator einen weiteren parallel schaltest? Was bedeutet das für die Gesamtkapazität und wie ändert sich die Lade- bzw. Entladezeit?



Kannst Du mir einen Eindruck darüber geben, wie so eine Schaltung auf dem *EEBoard* denn aussieht? Ich habe das noch nie gesehen.

Klar, *Ardus*! Schau her.

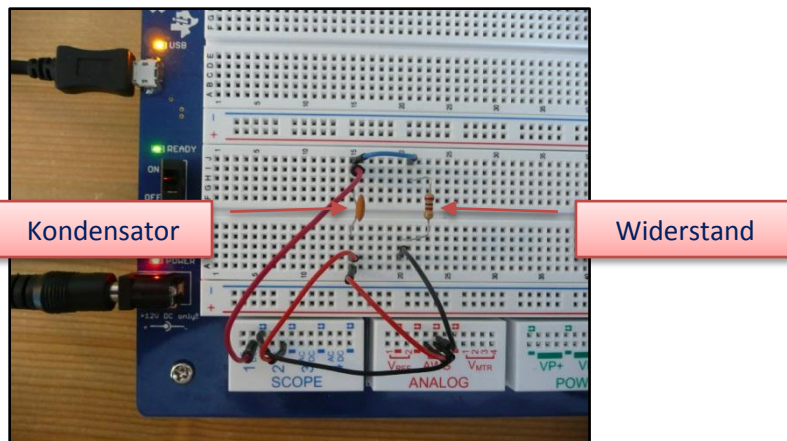


Abbildung 17 Aufbau der Differenzglied-Schaltung auf dem EEBoard

Sieht doch garnicht so wild aus – oder!?

Kennlinienaufnahme

Kommen wir zu einem sehr interessanten Teil. Ok, ich weiß, dass alles sehr interessant ist, aber das Folgende ist sehr, sehr spannend. Es geht um die *Kennlinienaufnahme*. Ich hatte ganz zu Beginn einmal sehr kurz erwähnt, dass das Oszillogramm eine *2-dimensionale Darstellung* ist, die sich über eine horizontale *X-Achse* bzw. eine vertikale *Y-Achse* erstreckt. Die horizontale *X-Achse* wird in der Regel über eine *interne Zeitablenkung* gesteuert, wohingegen die vertikale *Y-Achse* durch das anliegende Signal beeinflusst wird. Was wäre aber die Regel ohne Ausnahme? Kommen wir also zur Ausnahme. Du kannst das Oszilloskope auch im sogenannten *XY-Betrieb* benutzen.



Das verstehe ich nicht! Wir haben doch bisher auch die *X-* bzw. die *Y-Achse* des Oszilloskopes genutzt. Was soll denn jetzt da anders sein?

Nun ja, Du hast nicht *ganz* Unrecht, doch die horizontale *X-Achse* wurde bisher nicht von einem externen anliegenden Signal beeinflusst, sondern eben durch die *interne Zeitbasis*. Klar!?! Nun wollen wir aber eben diese Ausnahmemöglichkeit nutzen und ein externes Signal auf den *X-Eingang* schicken.

Wo aber befindet sich denn der *X-Eingang*? Wir haben doch nur 4 *Y-Eingänge*, so wie ich das gesehen habe.



Ok *Ardu*s, ganz einfach! Wir machen einfach durch eine Anpassung in der Konfiguration einen der vorhandenen *Y-Eingänge* zu einem *X-Eingang*.

Das ist über die Software ganz einfach zu realisieren. Ich zeige Dir das gleich. Ein wenig Geduld also noch. Für unser erstes vorbereitendes Experiment benötigen wir zusätzlich zu unserem *EEBoard* noch 2 Potentiometer. Das Ganze schaut dann wie folgt aus:

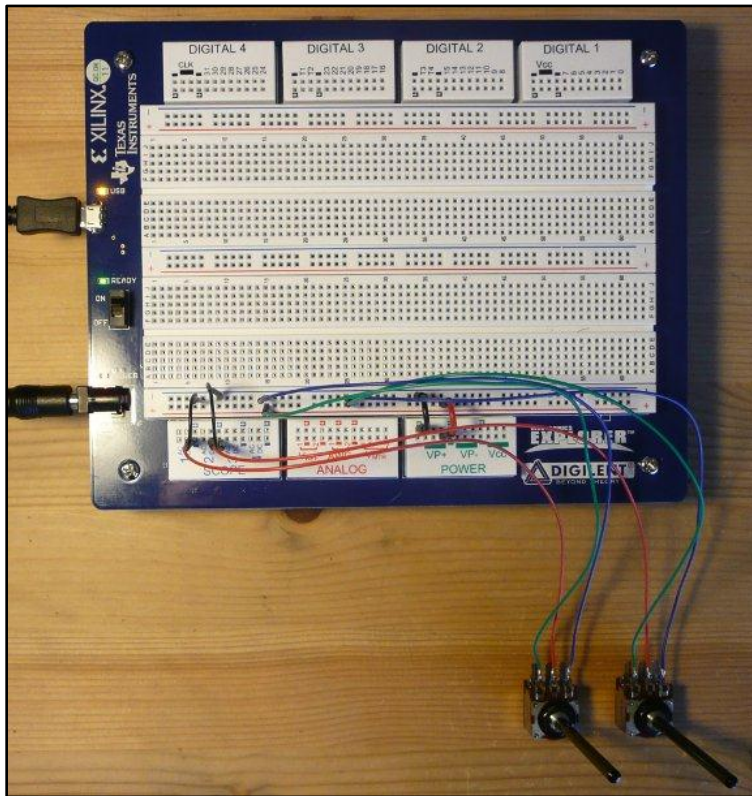


Abbildung 18 2 Potentiometer für den Oszilloskope XY-Betrieb

Auf diesem Foto ist natürlich kam zu erkennen, wie alles angeschlossen bzw. verkabelt ist. Deswegen hier einmal der Schaltplan.

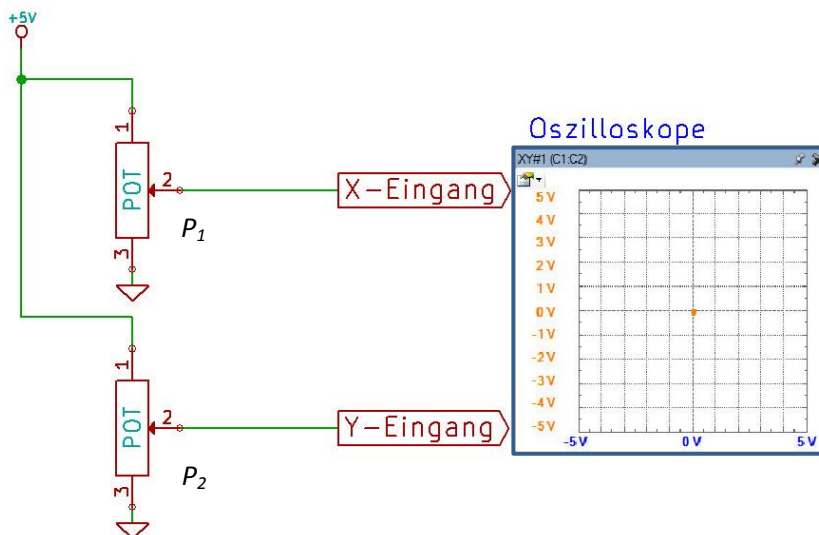


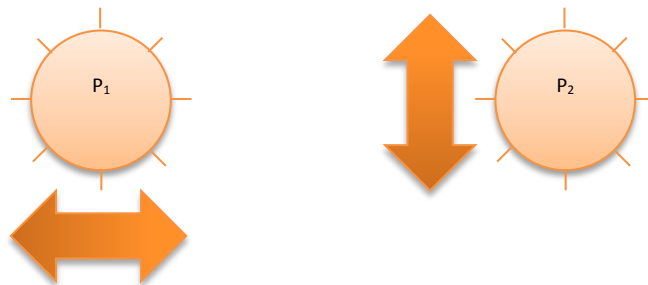
Abbildung 19 Schaltplan der 2 Potentiometer

Du kannst auf dem Schaltplan erkennen, dass die beiden Potentiometer jeweils als variable Spannungsteiler eingesetzt werden und ihr Ausgangssignal, also der Schleiferkontakt, zu den Eingängen des Oszilloskopes führen.



Kannst Du mir bitte verraten, was Du mit dieser Schaltung erreichen willst. Ich erahne den Sinn, doch ganz durchblicke ich das noch nicht.

Das ist ganz einfach, *Ardus*! Es handelt sich um eine Testschaltung, um einen einzelnen Punkt auf dem Oszilloscope mit den beiden Potentiometern steuern zu können.



Mit dem linken Potentiometer P_1 kannst Du den Punkt *horizontal*, also von links nach rechts und umgekehrt bewegen, wohingegen Du mit dem Potentiometer P_2 den Punkt vertikal, also von unten nach oben und umgekehrt bewegen kannst. Die Anzeige schaut wie folgt aus:

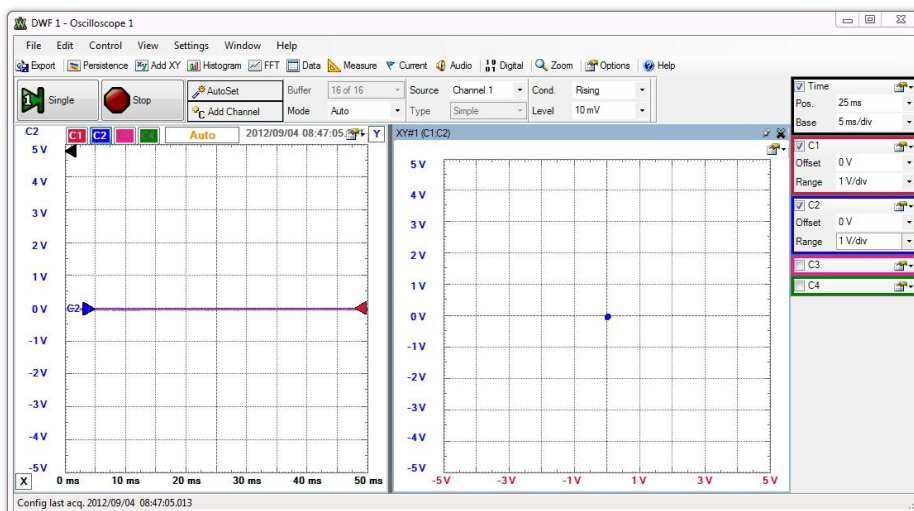


Abbildung 20 XY-Darstellung

Ich habe die Anzeige etwas verkleinert, denn auf der rechten Seite ist ein weiteres Fenster hinzugekommen. Wie habe ich das aber gemacht? Schau her:

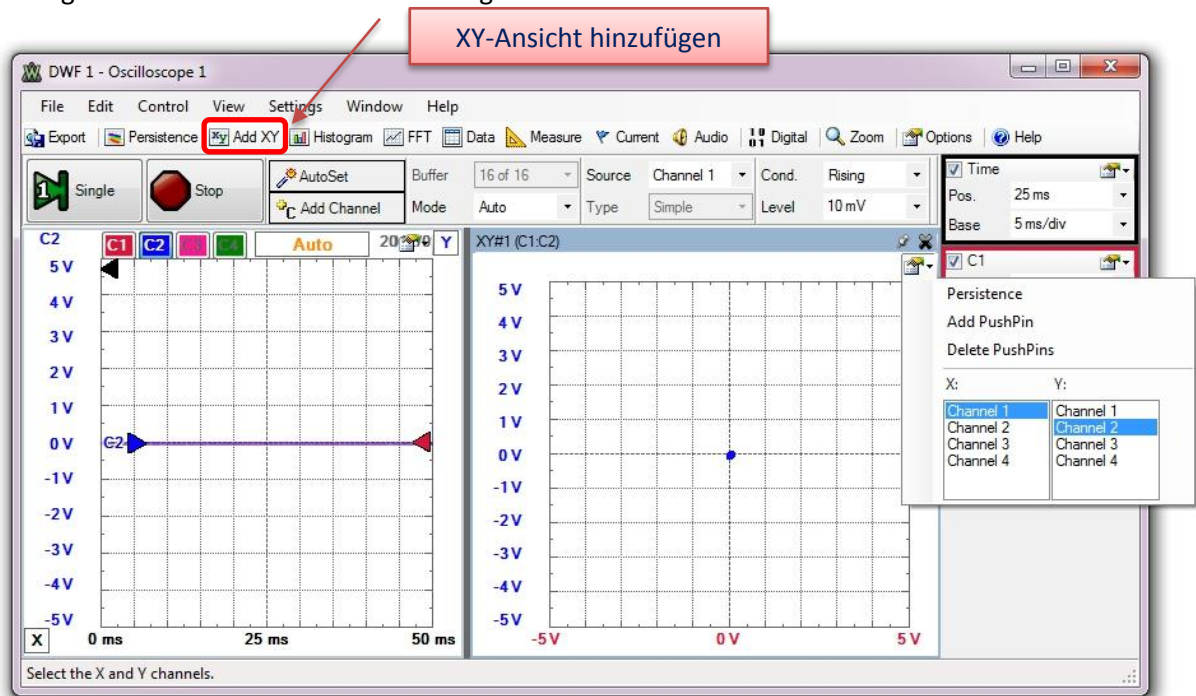


Abbildung 21 XY-Ansicht hinzufügen

In der *Toolleiste* unterhalb der Menüleiste befindet sich eine Schaltfläche mit der Bezeichnung *Add XY*. Wenn Du sie anklickst, dann wird dem Oszilloskope ein weiteres Fenster auf der rechten Seite hinzugefügt. Zusätzlich öffnet sich ein kleines Dialogfenster, in dem Du die einzelnen Eingangskanäle 1 bis 4 dem X- bzw. Y-Kanal zuweisen musst. In meinem Fall habe ich *Kanal 1* auf die X-Achse, *Kanal 2* auf die Y-Achse gelegt. Das folgende Bild zeigt Dir *beide* Ansichten und das ist wirklich fantastisch.

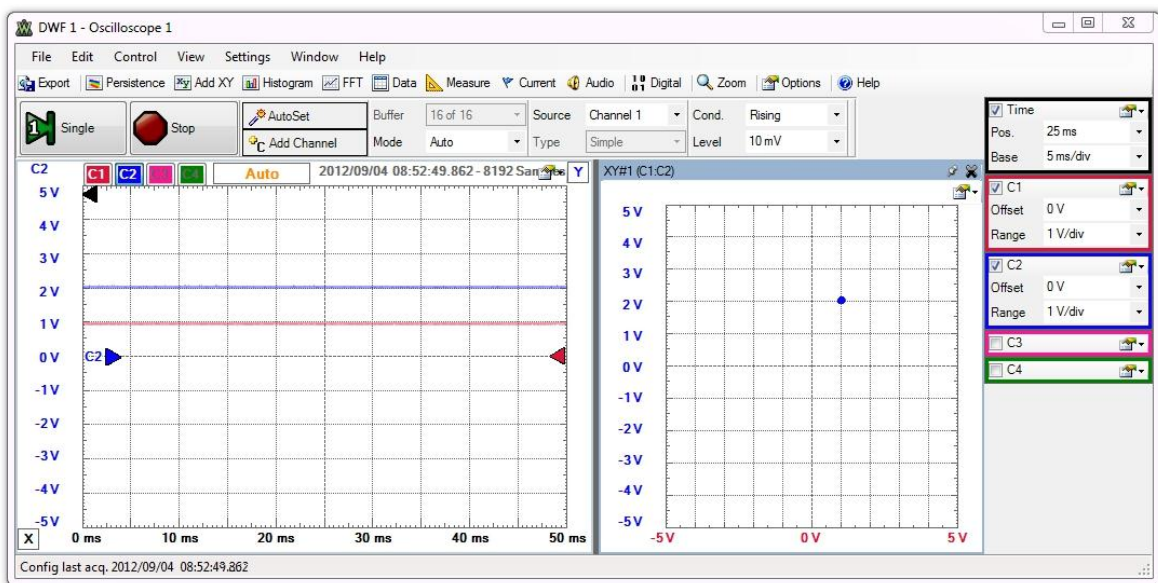


Abbildung 22 Main time- und XY-Ansicht

Wenn ich sage, dass *beide* Ansichten zur Verfügung stehen, dann meine ich damit die *normale*, bisher bekannte *Eingangssignal/Zeitbasis*-Anzeige auf der linken Seite und zusätzlich die *XY-Darstellung* der beiden Eingangskanäle 1 und 2 auf der rechten Seite. In diesem Fall habe ich das linke Potentiometer für die *X-Position* auf 1V justiert. Das kannst Du an der blauen Kurve erkennen. Das rechte Potentiometer für die *Y-Position* habe ich auf 2V gestellt. Die Reaktion siehst Du anhand der Lage der orangenen Kurve.

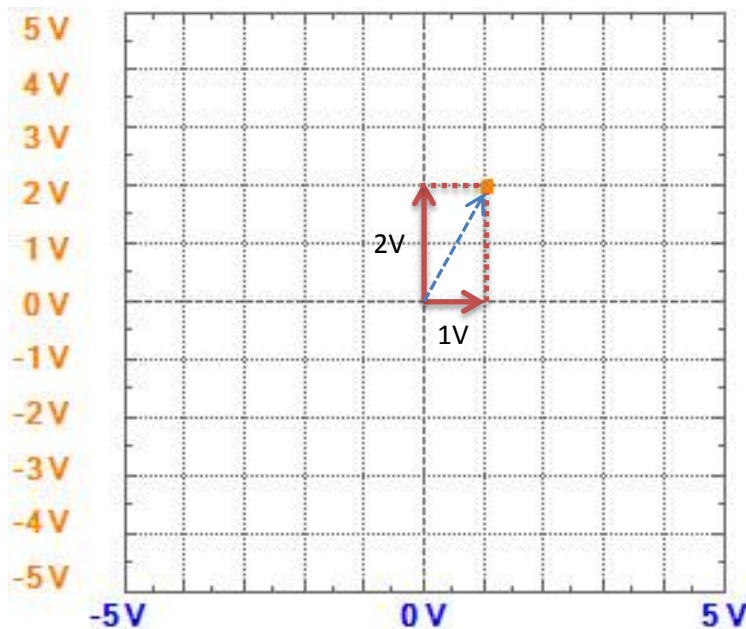
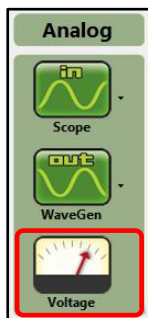
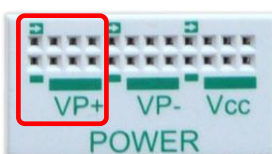


Abbildung 23 XY-Darstellung

Wenn Du mit den Stellungen der Potentiometer spielst, kannst Du den Punkt an jede beliebige Stelle innerhalb des zur Verfügung stehenden Spannungsbereiches positionieren. Wenn Du diese Schaltung aufbaust, dann kannst Du *EEBoard* eigene *Power-Supply* verwenden. Du siehst also, dass Du keine zusätzlichen Gerätschaften benötigst. Ich werde Dir gleich zeigen, wie Du die entsprechende Spannung einstellst. Zuvor musst Du das Power-Supply jedoch erst einmal öffnen und zur Anzeige bringen. Das geschieht mit einem Klick auf das *Voltage-Symbol*, das ich hier rot umrandet habe.



Du kannst nach der Einstellung der Power-Supply Parameter die Spannung auf dem *EEBoard* am gezeigten Block abgreifen:



Ich habe auf dem *EEBoard* die Anschlüsse für *VP+* und *Masse* direkt auf die *Power-Rails* gelegt, so dass sie überall verfügbar waren.

Danach öffnet sich das folgende Fenster:

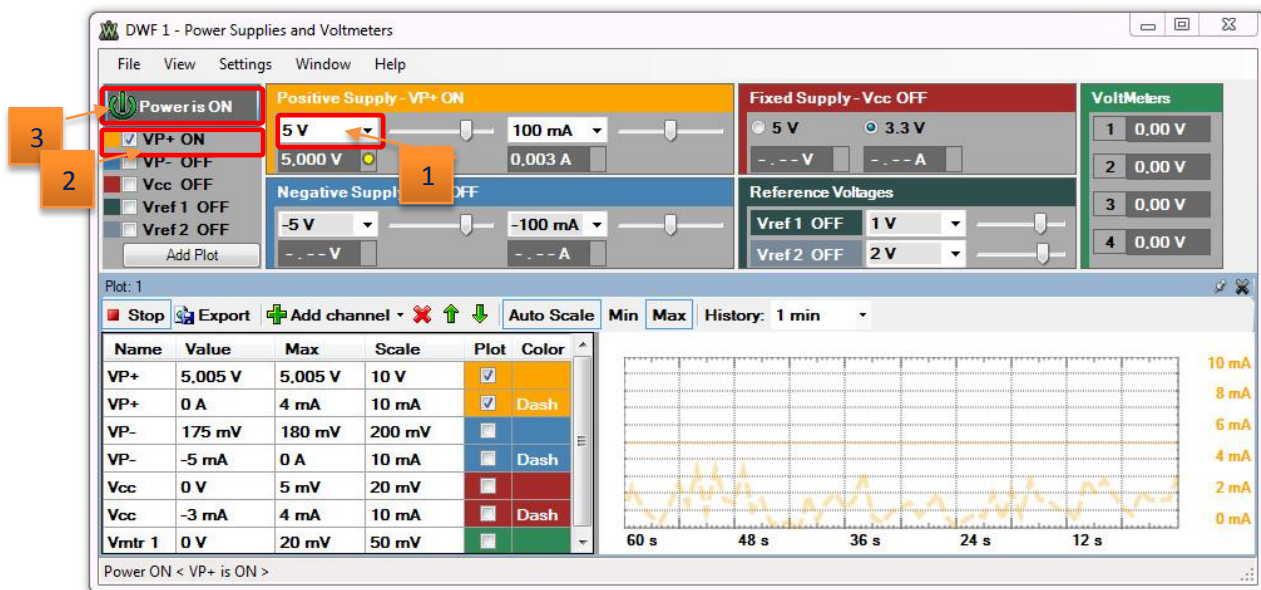


Abbildung 24 Power-Supply und Voltmeter

An den markierten Stellen habe ich die notwendigen Einstellungen vorgenommen, damit das *Power-Supply* die gewünschte Spannung liefert.

1. Maximalen Spannungswert auswählen (hier 5V)
2. Spannungsausgang wählen (hier V+)
3. Power-Supply einschalten (über die *Power-On* Schaltfläche)

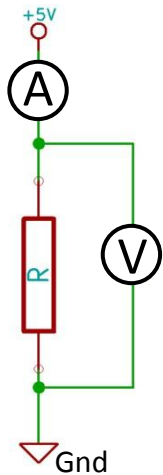
Zu den restlichen Einstellungen für das *Power-Supply* komme ich zu gegebener Zeit noch zurück.

Ich denke, dass Du jetzt soweit bis, dass ich Dir das zeigen möchte, was ich beabsichtigt hatte. Es geht um die Aufnahme einer *Kennlinie* für unterschiedliche elektronische Bauteile.



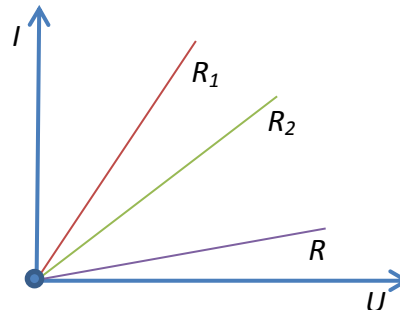
Kennlinie... Das habe ich ja noch nie gehört! Was soll das denn sein?

Immer mit der Ruhe, *Ardus*! Du musst mir schon die Möglichkeit geben, die Sache zu erklären. Bei einer *Kennlinie* handelt es sich um die grafische Darstellung zweier physikalischer Größen in einem Koordinatensystem. Elektronische Bauteile, wie z.B. Widerstände, Kondensatoren oder Spulen haben einen charakteristischen Kennlinienverlauf, anhand dessen ein Bauteil mehr oder weniger identifiziert werden kann. Quasi ein elektronischer Fingerabdruck.



Diese einfache Schaltung mit einem ohmschen Widerstand dient zur Aufnahme einer sogenannten *Strom/Spannungs-Kennlinie*. In diesem Fall kann diese Kennlinie auch *Widerstandskennlinie* genannt werden.

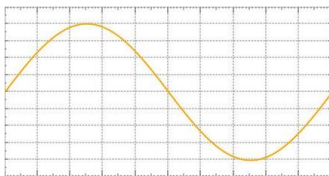
Bei einem ohmschen Widerstand handelt es sich um ein Bauteil mit einem linearen *Strom/Spannungsverhalten*.



Auf dem rechten Diagramm kannst Du das Verhalten des Stroms in Abhängigkeit des verwendeten Widerstandes erkennen, wenn die Spannung sich ändert. Ein derartiges Diagramm mit mehreren Kennlinien wird auch *Kennlinienfeld* genannt. Strom, Spannung und Widerstand stehen in einem bestimmten Verhältnis zueinander, das im *Ohm'schen Gesetz* festgehalten ist. Es lautet:

$$U = R \cdot I$$

Ich möchte Dich jetzt endlich mit einer Schaltung vertraut machen, die Du zur Kennlinienaufnahme nutzen kannst. Natürlich ist es keinem Elektroniker zumutbar, dass er zur Aufnahme einer Kennlinie neben der Schaltung sitzt und ständig am Potentiometer dreht, damit die Spannung auf- bzw. abgeregelt wird. Was können wir stattdessen machen? Nun, das ist ganz einfach. Denken wir einmal scharf nach und rekapitulieren, welche Kurvenform uns bekannt ist, bei der die Spannung ein automatisches und wiederkehrendes *Auf* und *Ab* darstellt. Richtig, die *Sinuskurve*.



Wenn wir jetzt diese Kurvenform an den Eingang unserer Schaltung legen, dann können wir das *Strom/Spannungsverhalten* wunderbar in der *XY-Ansicht* darstellen.

Der Widerstand

Mit der folgenden Schaltung können wir das *Strom/Spannungsverhalten* eines Widerstandes über eine Kennlinie sichtbar machen.

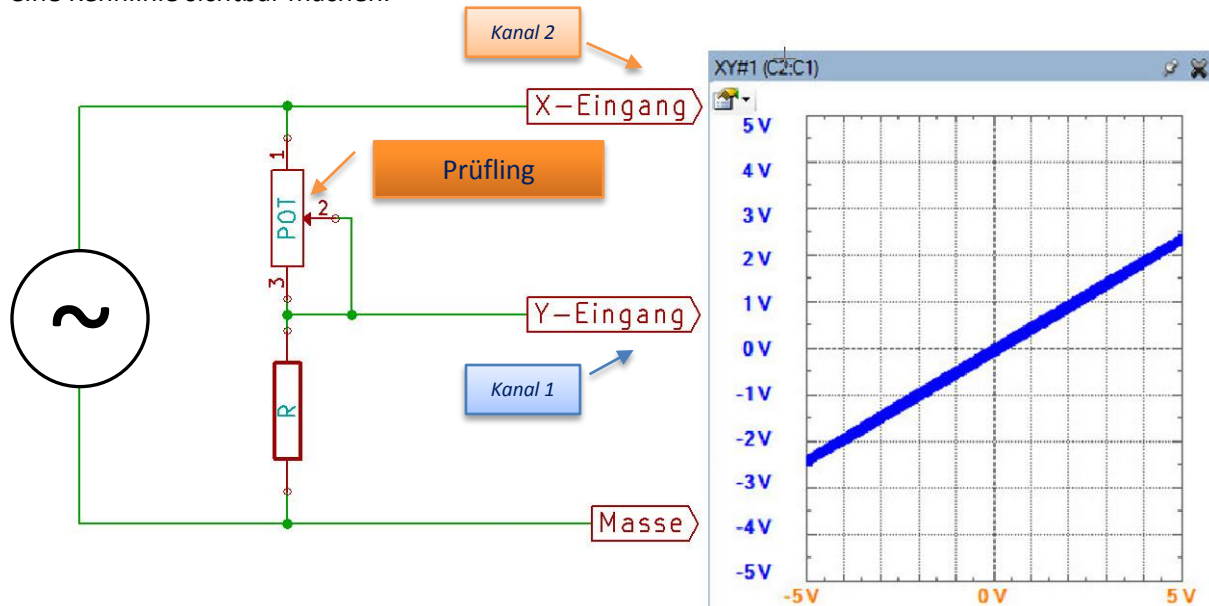
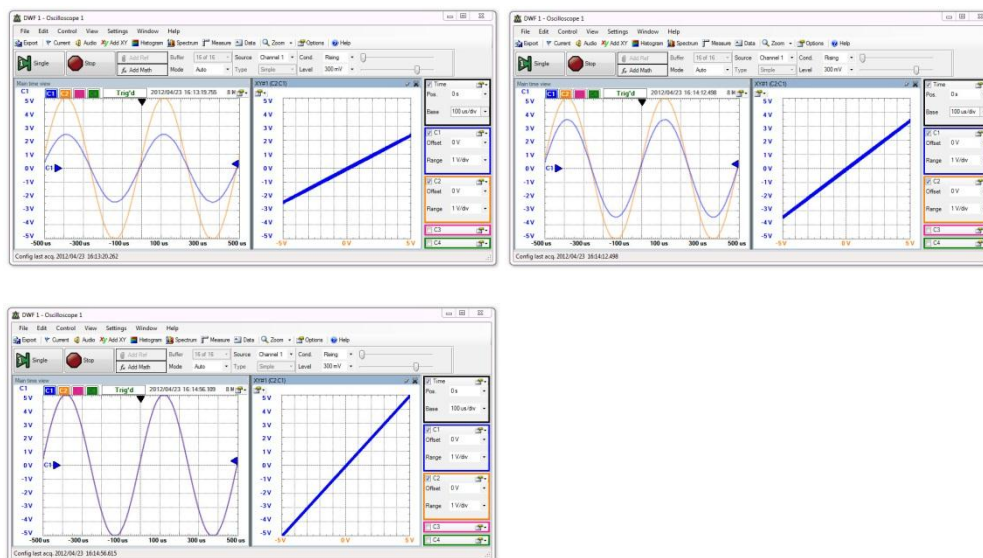


Abbildung 25 Strom/Spannungs-Kennlinienaufnahme eines Widerstandes

Das mit *Prüfling* bezeichnete Bauteil ist ein Potentiometer. Änderst Du seinen Widerstand, bekommst Du unterschiedliche Kennlinien angezeigt. Sie weisen aber allesamt ein lineares Verhalten auf. Wir haben es lediglich mit einer variierenden Steigung zu tun. Die nachfolgenden Bilder zeigen Dir ein paar unterschiedliche Widerstandswerte mit den entsprechenden Kennlinien.



Die orangene Kurve repräsentiert das Signal an *Kanal 1*, was dem unveränderten Ausgangssignal des Funktionsgenerators entspricht. Das Signal des *Prüflings*, also des *Potentiometer*, ist das blaue Signal, das sich nach und nach dem Ausgangssignal angleicht. Der Widerstand wird also geringer.



Das ist ja schon beeindruckend! Doch erkläre mir bitte einmal ganz kurz, warum da eine Gerade angezeigt wird. Das habe ich nicht ganz verstanden.

Hey *Ardus*, ist doch kein Problem. Beim ersten Mal habe ich auch ganz schön meine Haare gerauft und wenn Du wüsstest, wie viel (wenig) Haare ich jetzt noch auf meinem Schädel habe. Erinnere Dich noch einmal kurz an das Beispiel mit den beiden Potentiometern. Du konntest da die Position des Punktes frei einstellen. Hast Du die beiden Potentiometer gleichförmig von *links* nach *rechts* und zurück bewegt, dass beschrieb der Punkt eine Gerade von *links unten* nach *rechts oben* und zurück. Also immer dann, wenn beide die gleichen Signale an die Eingänge des Oszilloskopes schicken, kannst Du dieses Verhalten beobachten. Bei der *Sinusfunktion* ist das nichts anderes.

Die Diode

Kommen wir zu einem elektronischen Bauteil, das in die Kategorie *Halbleiter* fällt. Wir wollen uns eine *Diode* anschauen. Der Begriff *Halbleiter* ist auf den ersten Blick etwas verwirrend und drückt das Verhalten dieses Bauteils nicht korrekt aus. Ein Bauteil kann nicht nur *halb leiten*. Das ist völliger Unsinn. Die korrekte Bewertung von Halbleitern ist *die*, dass ihre *Leitfähigkeit* von gewissen Faktoren abhängt. Eine Diode arbeitet quasi wie ein Ventil. Wenn wir uns mit einer Hochgeschwindigkeitskamera den Fluss der Elektronen durch einen Leiter anschauen, dann eröffnet sich uns folgendes Bild:

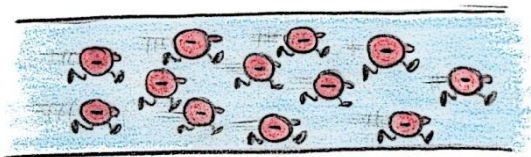


Abbildung 26 Hochgeschwindigkeitsaufnahme von Elektronen durch einen Leiter

Wenn wir jetzt eine Diode in diesen Fluss einbauen und die Elektronen noch einmal beobachten, dann siehst Du, dass sie keine Probleme haben, den Halbleiterbaustein zu überwinden.

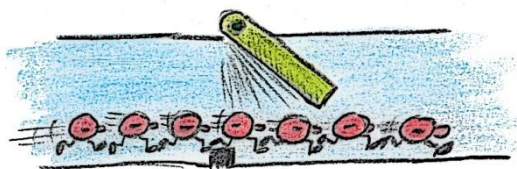


Abbildung 27 Elektronen, die auf eine Diode stoßen

Ein paar Millisekunden später wurde der Stromfluss jedoch umgekehrt und die Elektronen flitzen in die entgegengesetzte Richtung. Wir wollen mal sehen, was passiert.

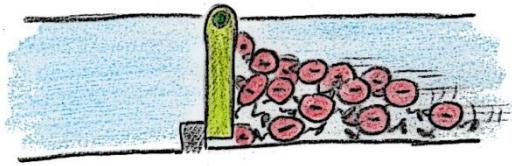
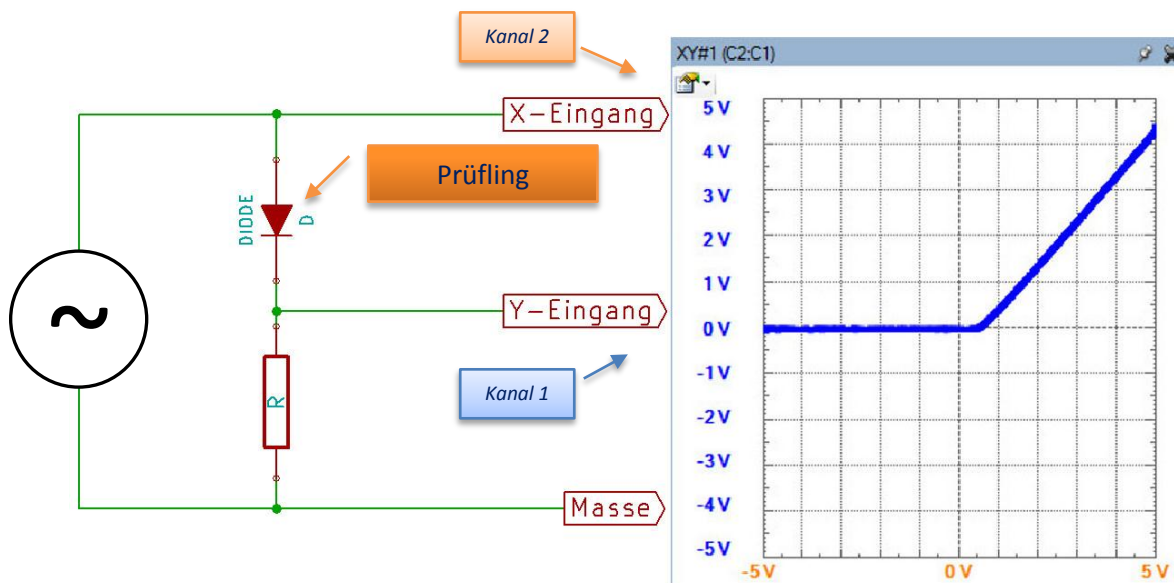
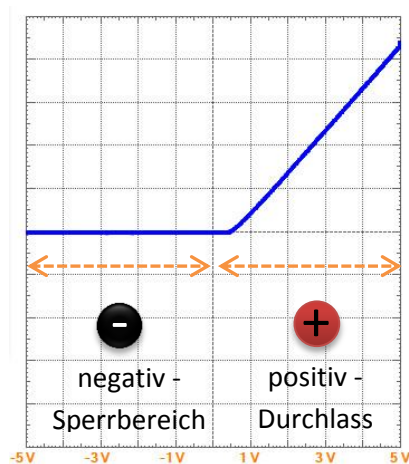


Abbildung 28 Elektronen rasen in Sperrrichtung gegen eine Diode

Das sieht ja nun Mal nicht ganz so gut aus. Irgendwie kommen sie auf diese Weise nicht weiter. Das hat den Grund, dass sie versuchen, sich in *Sperrrichtung* durch die Diode zu bewegen. Ok, dann wollen wir doch in unserer Kennlinienschaltung den Prüfling austauschen. Statt des Widerstandes setzen wir nun eine Diode ein.



Die Kennlinie einer *Diode* schaut etwas anders aus, als Du das von dem *Widerstand* gewohnt bist. Aber warum sieht sie so aus? Wenn Du das verstehen willst, muss ich ein wenig weiter ausholen. Du hast gesehen, dass sich eine Diode wie ein Ventil verhält und den Strom nur in einer Richtung durchlässt. Das erklärt schon mal den groben Verlauf der Kennlinie, die Du hier siehst.



Im *negativen Bereich* der anliegenden Sinusspannung, den Du auf der linken Seite des Diagramms siehst, sperrt die Diode und die blaue Kurve bleibt auf dem Level von 0V. Erst, wenn der *positive Bereich* der Sinusspannung an der Diode anliegt, den Du auf der rechten Seite des Diagramms siehst, fängt die Diode an zu leiten und die Spannung geht nach oben.

Aber lassen wir uns einen Blick auf die einzelnen Signale werfen, dann wird das Ganze noch etwas deutlicher.

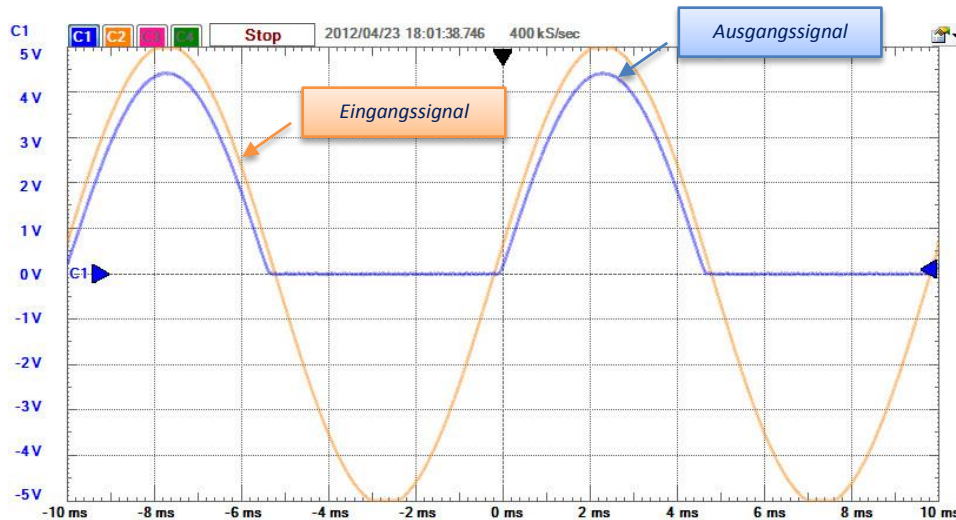


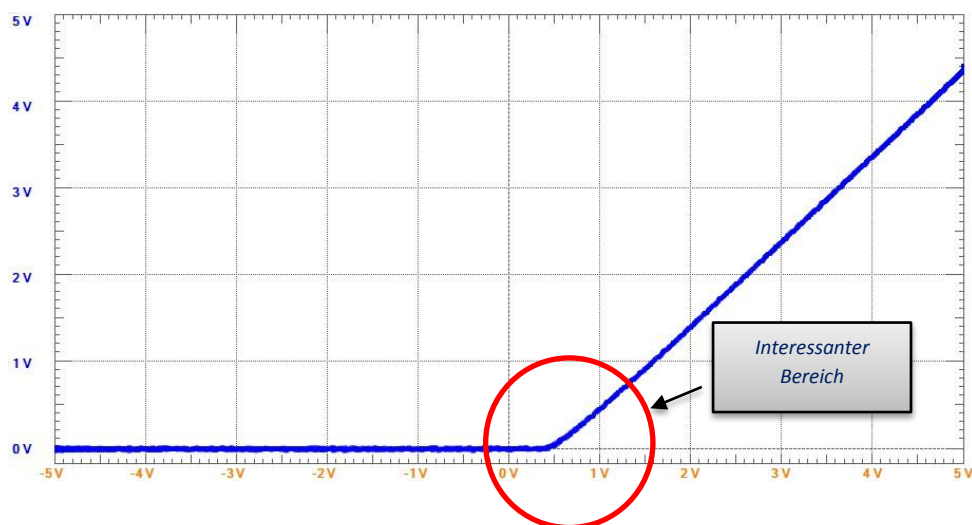
Abbildung 29 Eingangs- bzw. Ausgangsspannung der Diode

Nur, wenn die Sinuskurve den *positiven Bereich* durchläuft, folgt die Spannung an der Diode. Wird der *negative Bereich* erreicht, dann sperrt die Diode und die Spannung bleibt auf dem 0V-Level.



Stopp mal kurz! Da ist mir bei genauerem Hinsehen etwas aufgefallen. An der *Kennlinie* kann ich erkennen, dass die Spannung an der Diode nicht sofort nach Verlassen des Null-Durchgangs der Sinusspannung nach oben geht. Handelt es sich dabei um einen Anzeigefehler oder was ist der Grund?

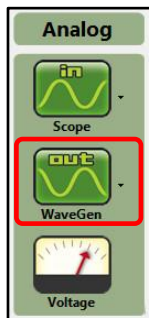
Hey, Du hast gute Augen, *Ardus*! Das hat natürlich einen Grund und es ist kein Anzeigefehler. Ich zeige Dir das Verhalten der Kennlinie an dem besagten Punkt noch einmal aus der Nähe.



Eine *Diode* weist ein spezielles Verhalten auf, wenn es um den Übergang von ihrem Sperrverhalten zum Durchlassverhalten geht. Dieser Bereich bzw. diese Spannung wird *Schleusenspannung* oder auch *Schwellenspannung* genannt und sie beträgt bei *Siliziumdioden* ca. 0,6 V. Genau diesen Wert kannst Du hier im Diagramm gut ablesen.

Der Waveform-Generator

Der Waveform-Generator ist ebenfalls wie das Oszilloskope im analogen Bereich beheimatet und wird über den zweiten Button gestartet.



Bei einem *Waveform-Generator* handelt es sich um ein elektronisches Gerät, das unterschiedliche Signalformen generieren kann. Eine Signalform hast Du schon kennen gelernt. Es handelt sich um die *Sinuskurve*. Es gibt jedoch unendlich verschiedene Kurvenformen, von denen sich aber einige als Standard-Signalformen zu erkennen geben. Auf diese Formen wollen wir an dieser Stelle eingehen.

Die unterschiedlichen Standard-Signalformen

Hast Du den *Waveform-Generator* gestartet, dann bietet sich Dir folgende Oberfläche mit diversen Einstellmöglichkeiten.

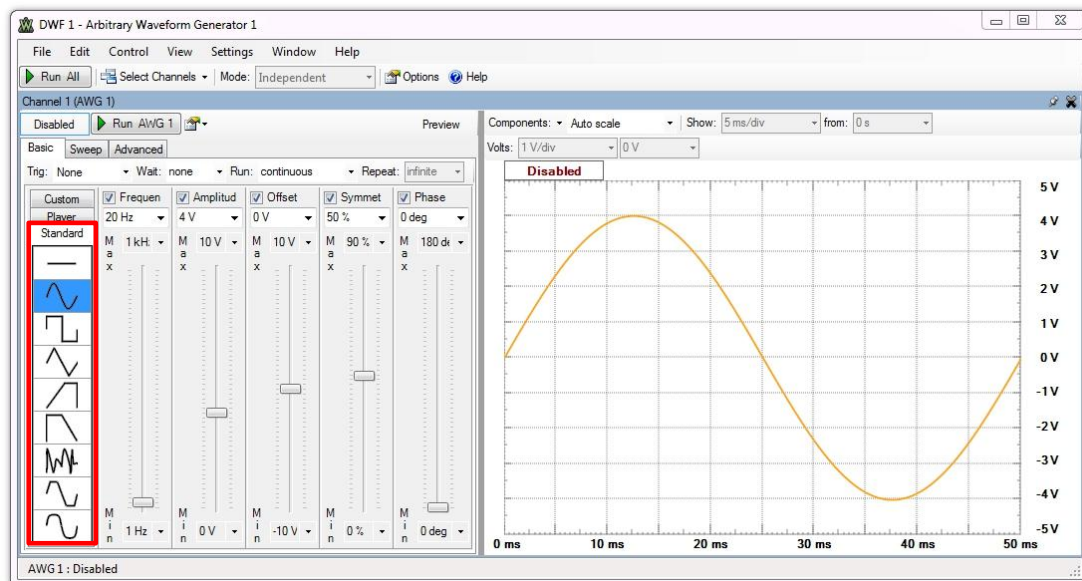


Abbildung 30 Der Waveform-Generator

Auf der linken Seite kannst Du die unterschiedlichen Signalformen auswählen, wobei Du auf der rechten Seite das Ergebnis erkennen kannst. Eine *Signalform* wird aber nicht nur anhand ihrer Form charakterisiert. Es gibt noch weitere Parameter, die einen Einfluss auf das Erscheinungsbild haben. Da sind z.B.

- die Frequenz
- die Amplitude
- der Offset

- die Symmetrie
- die Phase

Doch dazu später mehr. Im Folgenden gehe ich auf den *Standard-Mode* ein, der grundlegende Signalformen zur Verfügung stellt. *Custom-* und *Player-Mode* werden in den Hilfen detailliert erklärt.

Signalform Gleichstrom

Die Signalform *DC (Direct Current)* bedeutet, dass wir es mit einem *Gleichstrom* zu tun haben. Der einzige Parameter hierbei ist der *Offset*, der den *Abstand* zur Nulllinie definiert.

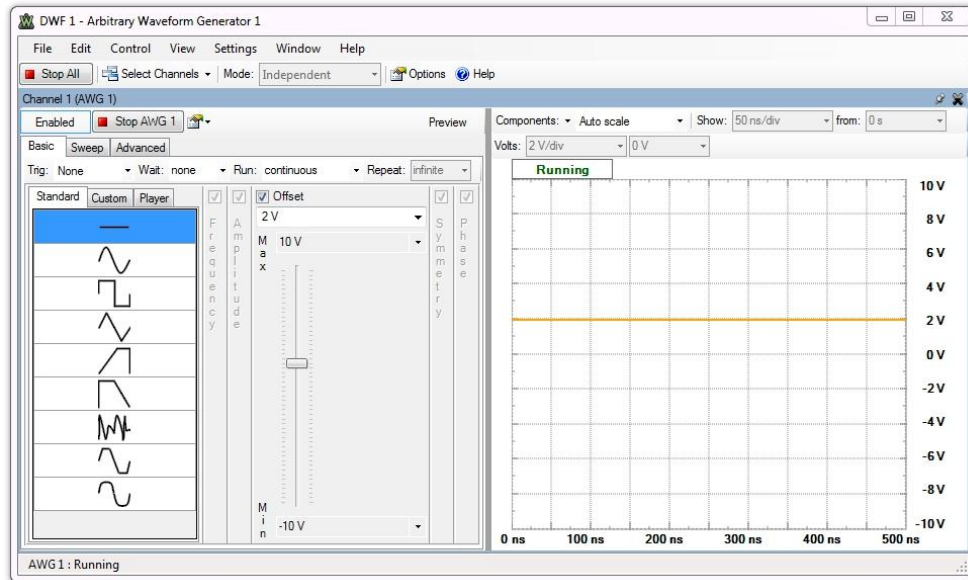


Abbildung 31 Signalform DC - Gleichstrom

Für dieses Beispiel habe ich den *Offset* auf 2V eingestellt und Du kannst auf der rechten Seite erkennen, dass sich die Kurve um diesen Betrag von der Nulllinie nach oben gewandert ist.

Signalform Sinus

Das *Sinussignal* ist Dir schon bekannt und hier gibt es die unterschiedlichsten Einstellmöglichkeiten.

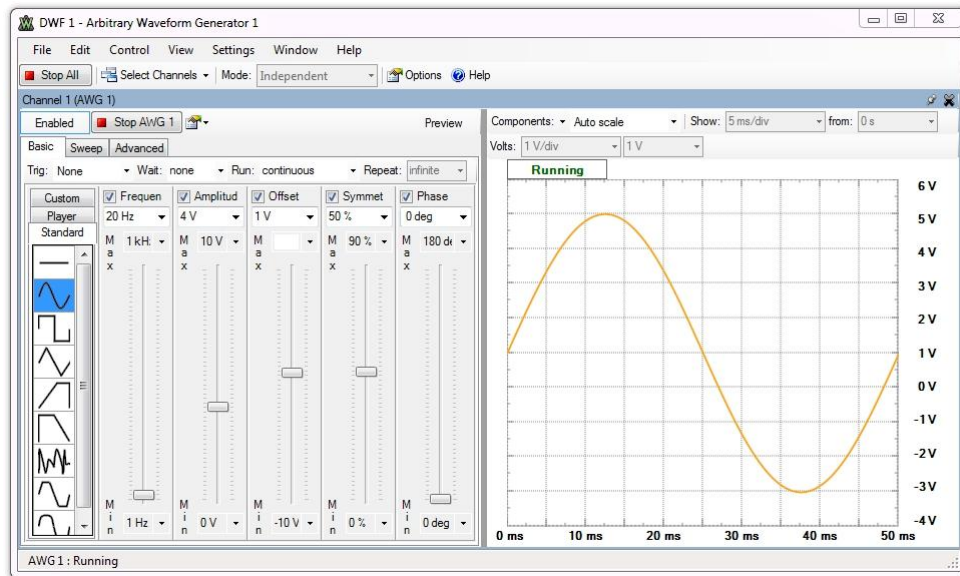


Abbildung 32 Signalform Sinus

Die Parameter *Frequenz*, *Amplitude* und *Offset* sind an dieser Stelle wohl nicht mehr zu erwähnen. Was jedoch noch wichtig ist, sind die folgenden:

- Symmetrie
- Phase

Schauen wir uns zuerst die *Symmetrie* genauer an. Sie steht standardmäßig auf 50%. das bedeutet, dass sowohl der *positive*, wie auch der *negative* Bereich gleichermaßen mehr oder weniger in die entsprechende Richtung gestaucht sind.

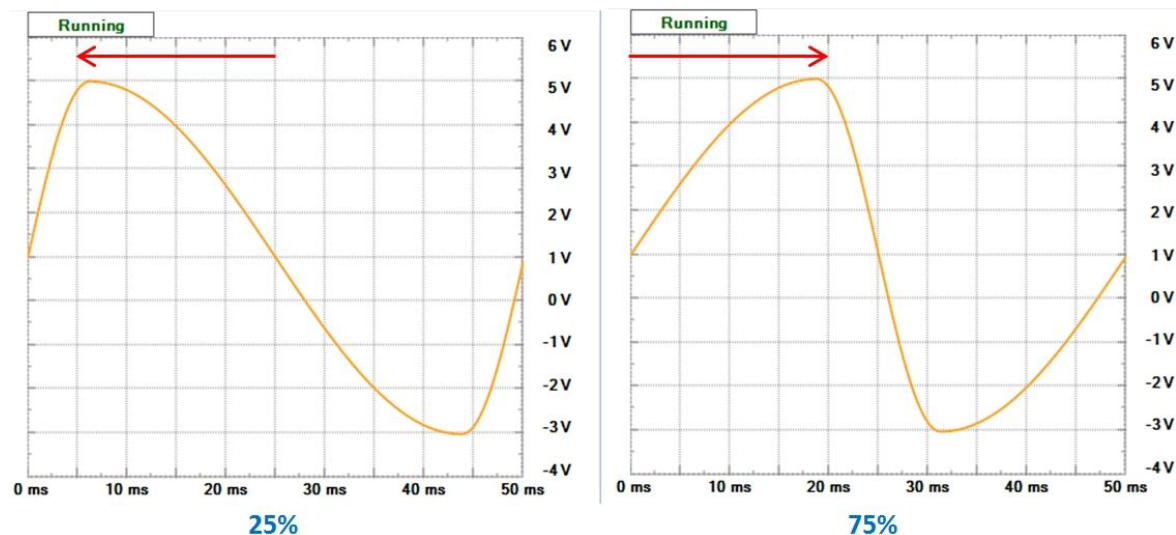


Abbildung 33 Veränderte Symmetrie der Sinuskurve

Bei der *Phase* wird der komplette Kurvenzug in Richtung der *Zeitachse* (X-Achse) verschoben. Damit Du es besser verstehst, schauen wir uns in diesem Zusammenhang einmal den *Einheitskreis* an.

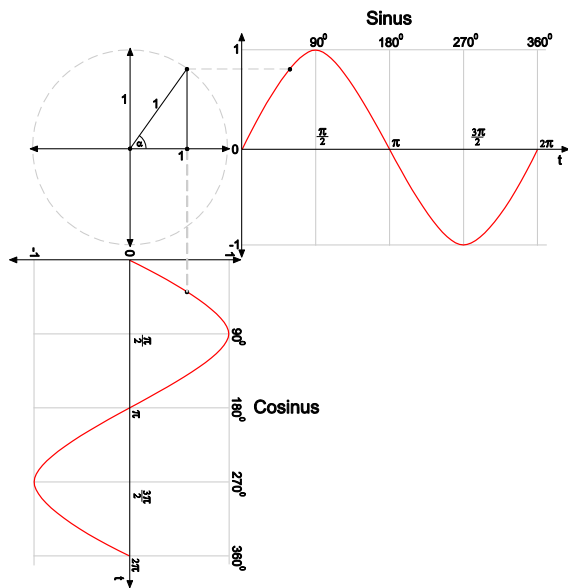


Abbildung 34 Der Einheitskreis

Bei einer vollen Umdrehung am Einheitskreis von 360° wird eine komplette *Sinuskurve* gezeichnet. Erfolgt eine Phasenverschiebung um 90° , dann haben wir es mit einer *Kosinuskurve* zu tun. Um eben diese Verschiebung zu erreichen, wird der *Phasenwert* angepasst. Der folgende Kurvenverlauf zeigt eine Phasenverschiebung von 90° an.

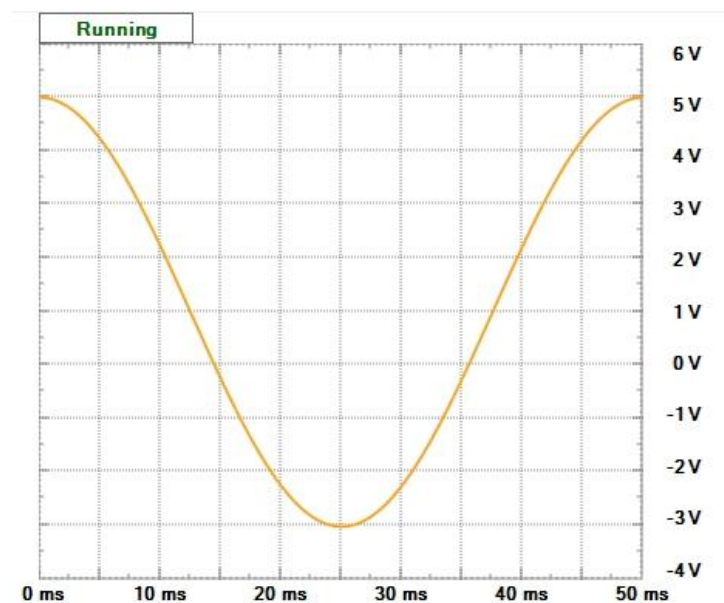


Abbildung 35 Phasenverschiebung um 90 Grad

Signalform Rechteck

Die Signalform *Rechteck* findet in der Digitaltechnik große Verwendung, wenn es darum geht, bestimmte Impulse zu verarbeiten.

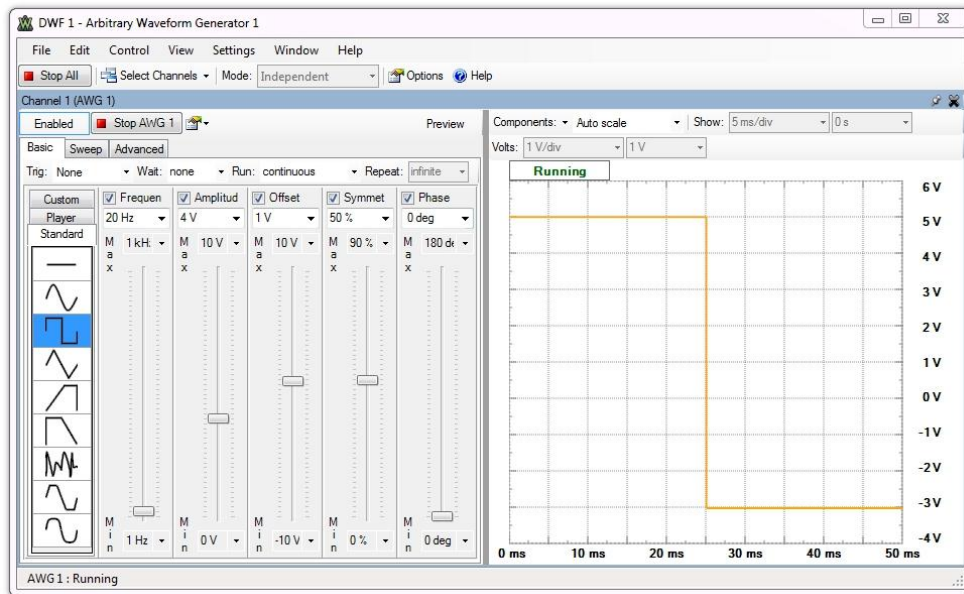


Abbildung 36 Signalform Rechteck

Signalform Dreieck

Die Signalform *Dreieck* zeichnet sich durch einen mehr oder weniger steilen An- bzw. Abstieg aus, wobei beide gleichsam erfolgen.

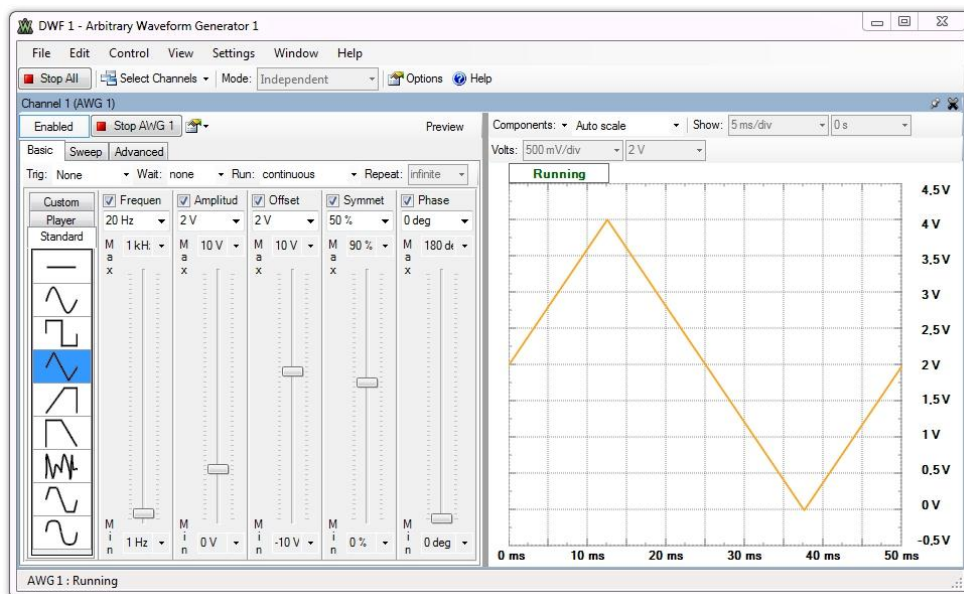
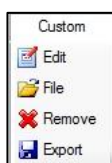


Abbildung 37 Signalform Dreieck

Eigene Signalformen kreieren

Bisher haben wir die grundlegenden Signalformen aus der Rubrik *Standard* kennengelernt. Ich sagte aber zu Beginn, dass es unendlich unterschiedliche Signalformen gibt. In der Rubrik *Custom* ist es Dir möglich, eigene Signalverläufe zu entwerfen. Wenn Du die *Edit-Schaltfläche* anklickst, kannst Du über ein Editor-Fenster mit der Maus eigene Kurvenverläufe erstellen.



Ganz zu Anfang hast Du eine *Nulllinie*, die Du mit gedrückter linker Maustaste nach Deinen Vorstellungen anpassen kannst.

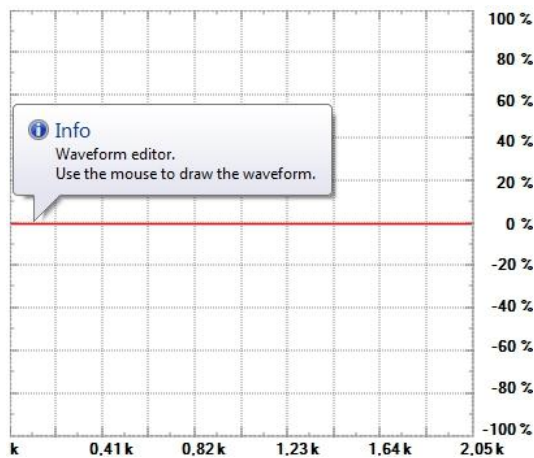


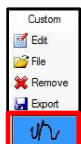
Abbildung 38 Anpassen der Nulllinie

Ich habe meiner Kreativität freien Lauf gelassen, wobei das bestimmt nicht sehr sinnvoll erscheint. Aber sei es drum.



Abbildung 39 Willkürlich generierter Kurvenverlauf

Nach einem Klick auf die *Save-Schaltfläche* wird der von Dir ersonnene Kurvenverlauf gespeichert, so dass Du auch in Zukunft darauf zugreifen kannst. Ein entsprechendes Icon erscheint in der *Custom-Rubrik*.



Wie hören sich die Signalformen an?

Jede einzelne Signalform hat ein charakteristisches Aussehen und natürlich haben auch Klänge eine ähnliche Form, würden wir sie auf dem Oszilloskope darstellen. Du kannst einen sehr einfachen Versuch machen, um die einzelnen Signalformen *hörbar* zu machen. Dem *EEBoard* liegen einige elektrische und elektronische Komponenten bei. So auch ein kleiner *Lautsprecher*. Verbinde ihn, so wie Du es auf dem folgenden Bild siehst, mit Deinem *EEBoard* und starte den *Waveform-Generator* mit den unterschiedlichsten Signalformen.

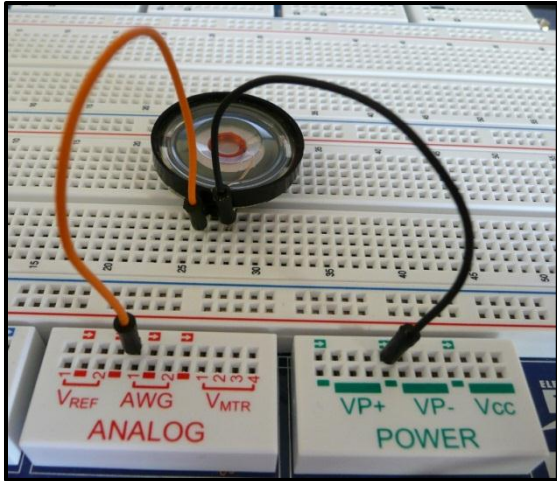


Abbildung 40 Der Lausprecher am EEBoard

Verwende dabei den *AWG (Arbitrary-Waveform-Generator) Ausgang*, um den Lautsprecher anzusteuern. Behältst Du die Frequenz bei, so wirst Du feststellen, dass sich die Klangfarbe bei den unterschiedlichen Formen ändert. Ein *Sinussignal* hört sich irgendwie weicher an, als ein *Rechtecksignal*. Das kommt daher, dass das *Sinussignal* weiche Übergänge zwischen den Amplituden hat. Das *Rechtecksignal* springt sehr hart und ohne erkennbaren Übergang zwischen An bzw. Aus hin und her.

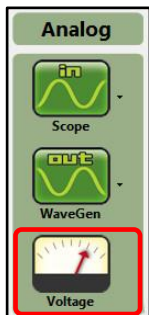


Das ist ja toll. Dann kann ich doch auch sicherlich mal einen richtig großen Lautsprecher an das *EEBoard* anschließen. Dann wird das richtig krachen!

Tue das nicht! Das einzige, was dann krachen wird, ist Dein *EEBoard*. Es ist nicht für die Dimensionen eines großen Lautsprechers ausgelegt und wird möglicherweise Schaden nehmen. Verwende nur *die* Komponenten, die dafür vorgesehen sind! Andernfalls hast Du nicht mehr lange Spaß mit Deinem *EEBoard*.

Das Powersupply

Was wäre ein Experimentierboard ohne die Möglichkeit einer Spannungsversorgung. Das *EEBoard* gibt Dir die Möglichkeit, die unterschiedlichsten Spannungsquellen zu generieren. Über die *Voltage-Schaltfläche* kannst Du die Einstellungen für das *Powersupply* öffnen. Alle einzelnen Powersupplies liefern ihre Spannungen relativ zur Masse des *EEBoards*.



Die Oberfläche gestaltet sich dabei wie folgt.

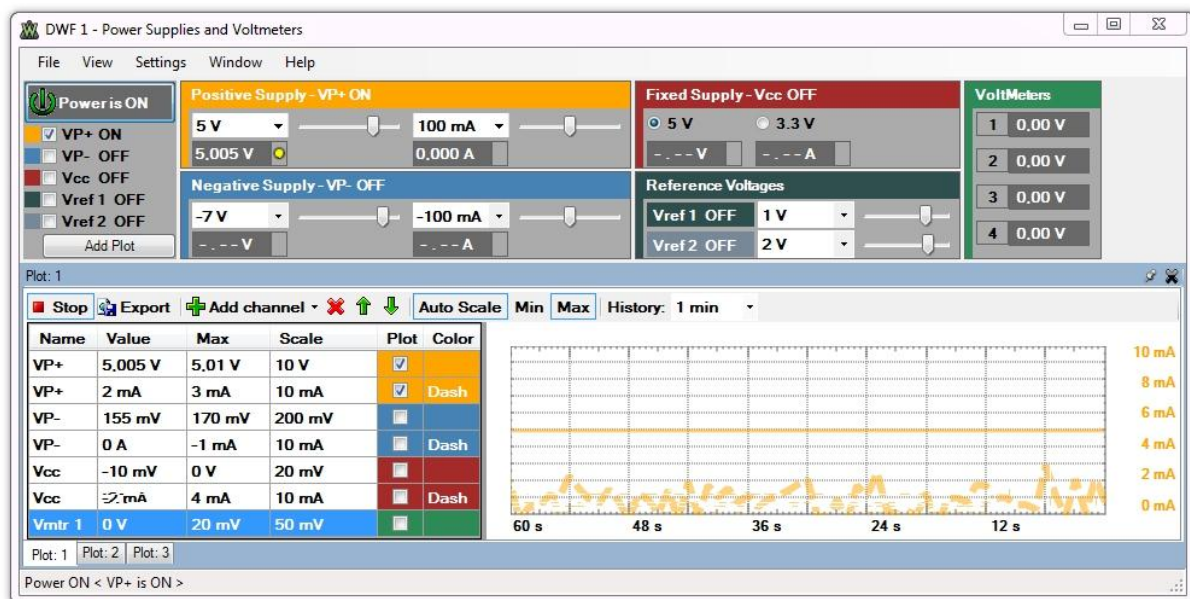
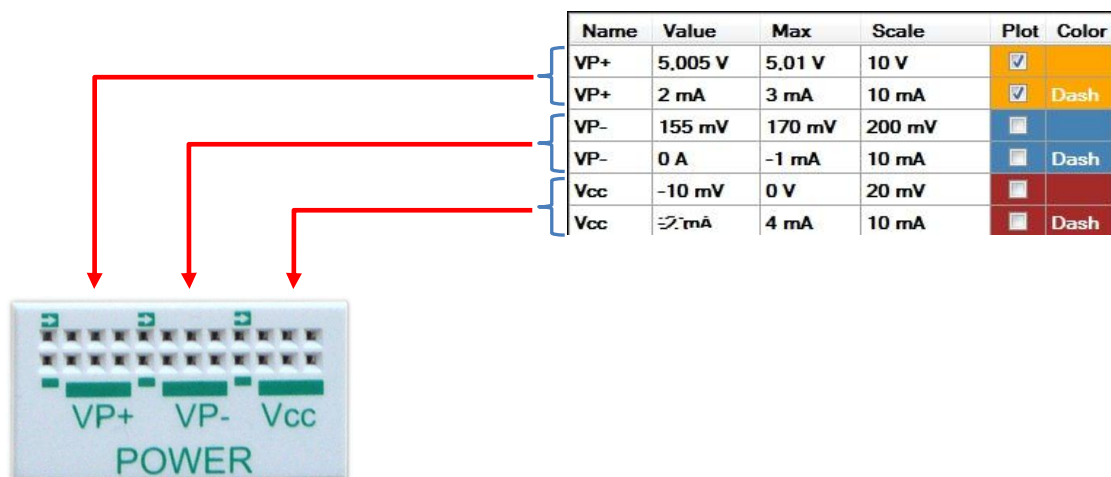


Abbildung 41 Das Powersupply

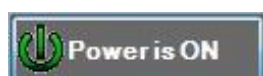
Ein Powersupply liefert eine Spannung nach außen, ganz ähnlich wie der Waveform-Generator. Doch hier wird lediglich eine Gleichspannung (DC) geliefert. Mehr benötigen wir auch für unsere Experimente nicht.

Schauen wir uns doch einmal die entsprechenden Anschlüsse auf dem EEBoard an.



Das Powersupply ein- bzw. ausschalten

Wie jedes dedizierte Powersupply, das über einen Ein- / Ausschalter zu bedienen ist, gibt es auch hier diese Möglichkeit.



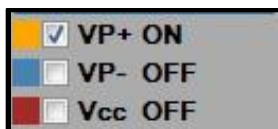
Über diesen Button werden alles zur Verfügung stehenden Spannungen + Referenzspannung ein- bzw. ausgeschaltet.

Einzelne Spannungsquellen ein- bzw. ausschalten

Zudem hast Du die Möglichkeit, einzelne Spannungen wie

- $VP+$
- $VP-$
- V_{CC}

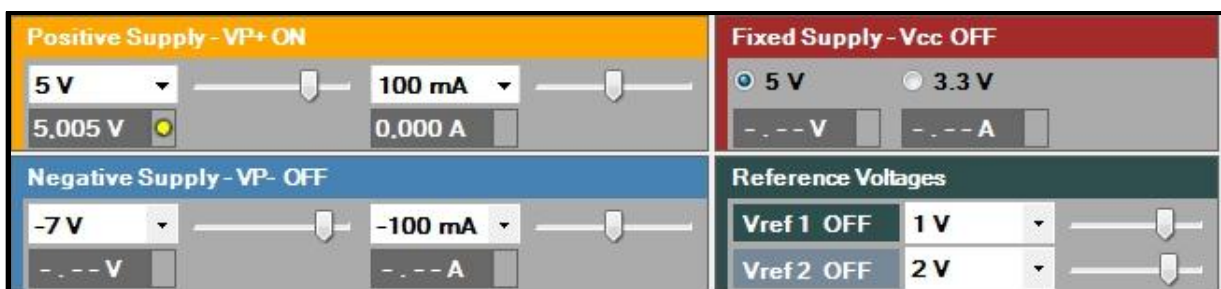
einzelnen ein- bzw. auszuschalten. Das kannst Du über die entsprechenden Checkboxes kontrollieren.



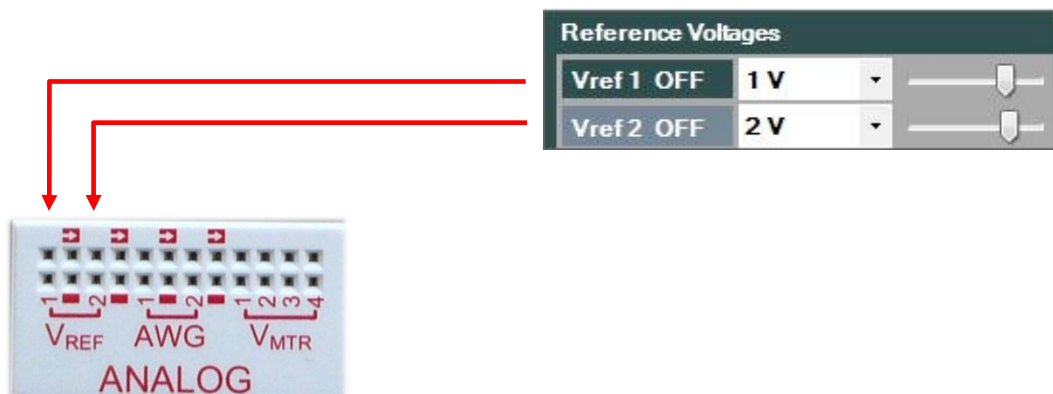
Bei diesem Beispiel ist $VP+$ aktiviert und der Rest deaktiviert. So kannst Du die unterschiedlichen Spannungsquellen ganz individuell regeln.

Einzelne Spannungsquellen regeln

Natürlich kannst Du die einzelnen Spannungsquellen mehr oder weniger separat innerhalb bestimmter Grenzwerte manipulieren. Das erfolgt über die weiter rechts gelegenen Bereiche.



Auf der linken Seite siehst Du die Einstellungen für $VP+$ und $VP-$. Es können Werte von $0V$ bis $9V$ ($VP+$) bzw. $0V$ bis $-9V$ ($VP-$) eingestellt werden. Es handelt sich dabei um die *variablen* Spannungsquellen. Des Weiteren verfügst Du über eine *Fest-Spannungsquelle* für so markante Werte wie $5V$ bzw. $3,3V$, die in der Digital- bzw. Mikrocontrollertechnik üblich sind. Dafür nutzt Du den V_{CC} Ausgang. Uns zu guter Letzt verfügst Du über zwei *Referenz-Spannungsquellen*. Diese werden über die analoge Buchsenleiste abgegriffen.



Jeder dieser Werte kann sich in den Grenzen von $-10V$ bis $+10V$ bewegen. Entsprechende Schieberegler ermöglichen ein schnelles Anpassen der Werte, obwohl ich von dieser Form der Anpassung abrate. Es wird schnell einmal ein Wert ausgewählt, der für eine Schaltung ggf. zu hoch ist. Die Auswahl über die Auswahlliste ist da allemal sicherer.

Den Spannungs- bzw. Stromverlauf anzeigen

Über die Zeit gesehen, kann natürlich ein Verbraucher, also z.B. eine Schaltung, mehr oder weniger belastend für ein Powersupply sein. Du kannst Dir alle Werte der einzelnen Powersupplies in einem Graphen anzeigen lassen.

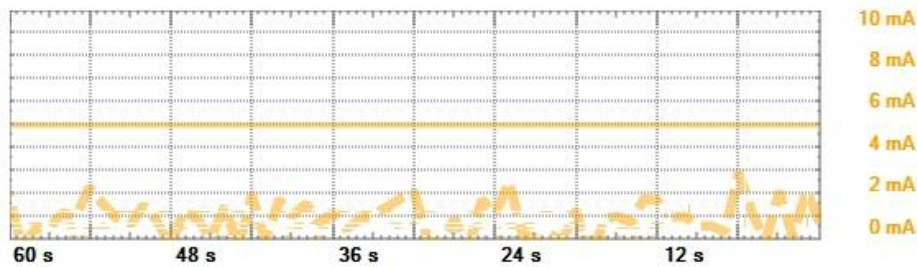


Abbildung 42 Spannung und Strom über die Zeit gesehen

Diese Ansicht kann für jedes einzelne Powersupply getrennt geregelt werden.

Exportieren der angezeigten Werte

Über zwei Schaltflächen (Stop bzw. Export) können die angezeigten Werte eingefroren und *exportiert* werden.



Der nachfolgende Dialog bietet Dir die Möglichkeit, eine Werteliste in unterschiedliche Dateiformate zu exportieren. (*Tab-Separiert* bzw. *Komma-Separiert*)

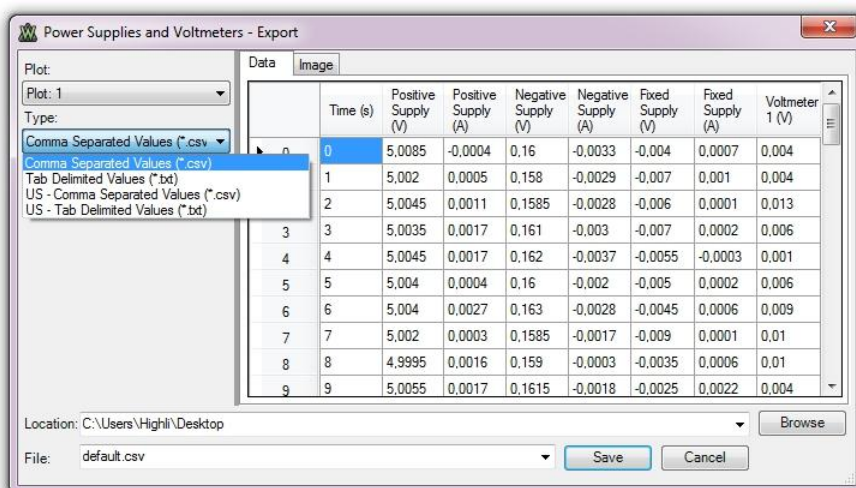


Abbildung 43 Exportieren von angezeigten Werten

Die Voltmeter

Dieser Punkt passt zwar nicht ganz in dieses Kapitel, aber es geht um Spannungen und deswegen können wir es an dieser Stelle erwähnen. Das *EEBoard* verfügt über vier getrennte Voltmeter.



Wir können im Moment folgende Werte ablesen:

- Voltmeter 1: 1,00V
- Voltmeter 2: 2,03V

Diese Anzeige findest Du in der rechten oberen Ecke des Fensters.

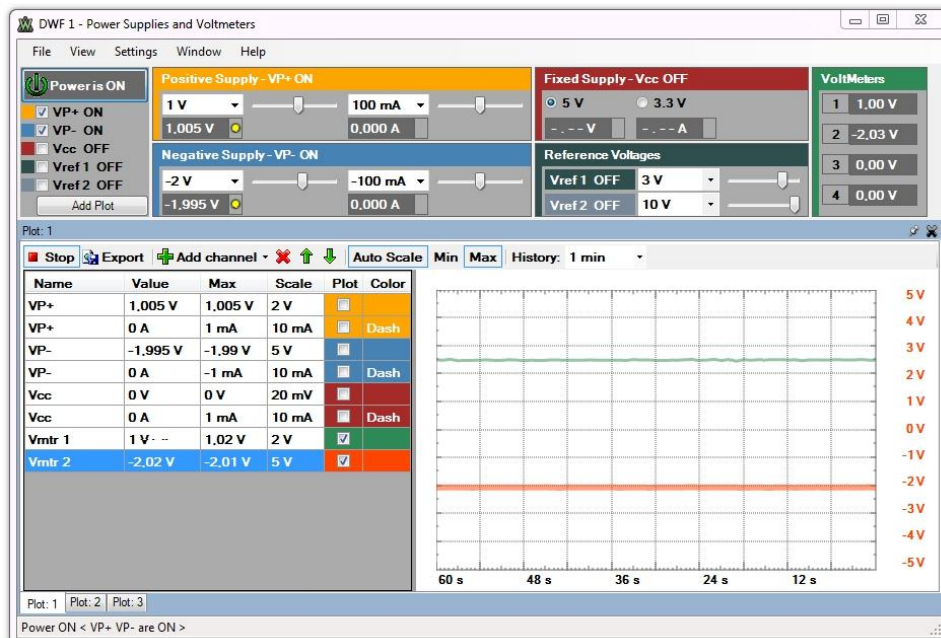
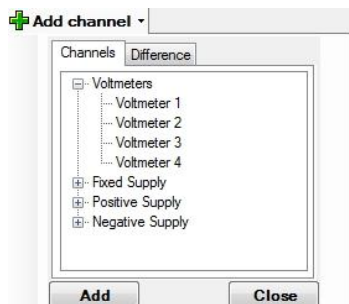


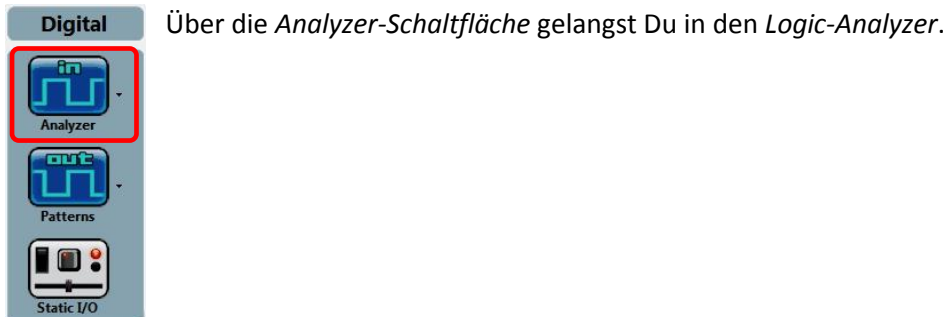
Abbildung 44 Die Voltmeter

Zusätzlich kannst Du natürlich den zeitlichen Verlauf der einzelnen Spannungen in dem rechts unten befindlichen Graphen anzeigen lassen. Standardmäßig wird nur ein einziges Voltmeter angezeigt. Um weitere Voltmeter hinzuzufügen, klickst Du auf die *Add channel-Schaltfläche*.

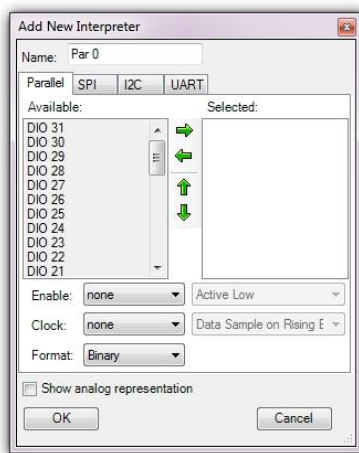


Der Logic-Analyzer

Ein *Logic-Analyzer* ist ein Messgerät, das den zeitlichen Verlauf von digitalen Impulsen darstellt. Der *Logic-Analyzer* des *EEBoards* kann nicht nur einzelne Impulse quasi parallel darstellen, sondern ist in der Lage, Impulsgruppen aus unterschiedlichen Protokollen zu erkennen und anzuzeigen.



In der folgenden Grafik siehst Du die unterstützten Protokolle des Interpreters.



Die unterstützten Protokolle

Als da wären:

- Parallel
- SPI (Serial Peripheral Interface)
- I2C (I-Quadrat-C)
- UART (Serielle Schnittstelle)

Es können unterschiedliche Protokolle gleichzeitig angezeigt werden, wie Du das in der folgenden Grafik sehen kannst.

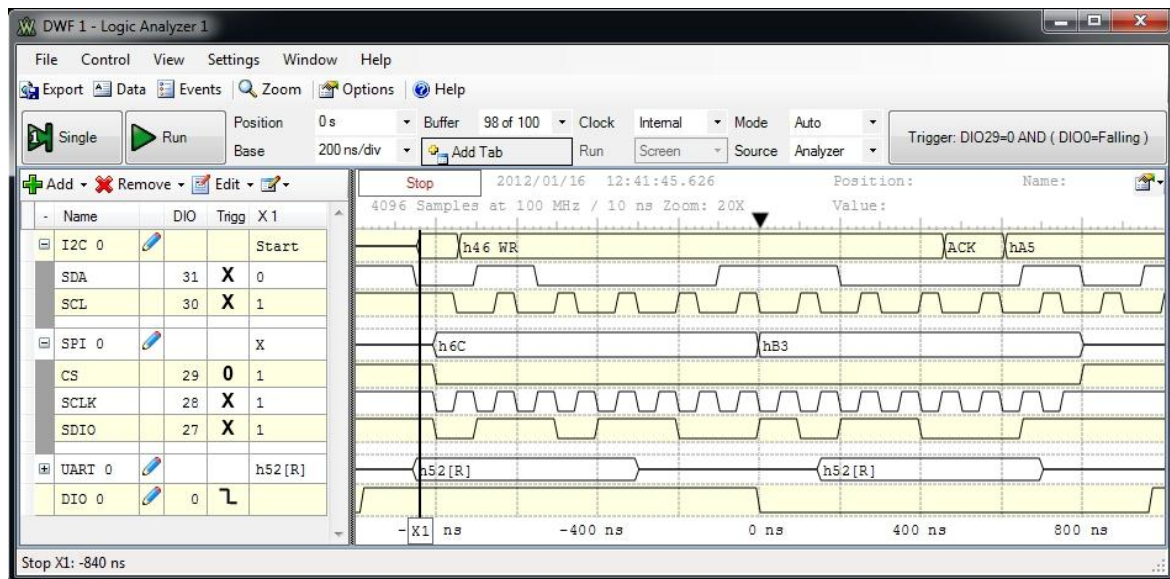





Abbildung 45 Der Logic-Analyzer (Quelle: Hilfe-Seite)








Die Kontroll-Schaltflächen

Die Schaltflächen zur Steuerung der Aufnahme haben folgende Bedeutungen:

Schaltfläche	Bedeutung
 Single	Startet eine einzige Momentaufnahme
 Run	Startet eine Endlosaufnahme
 Stop	Stoppt eine Endlosaufnahme

Die Triggerung

Wir können uns die Frage stellen, wann bei einer gewünschten Aufnahme die unterschiedlichen Logik-Pegel angezeigt werden sollen. Das kann unmittelbar erfolgen oder aber zu einem bestimmten *Ereignis*. Dieses *Ereignis* wird in diesem Kontext *Trigger* genannt.

+	Name	DIO	Trigger
	GP23	0	
	GP22	1	 Don't Care (X)
			 Low (0)
			 High (1)
			 Rising Edge (R)
			 Falling Edge (F)
			 Any Edge (E)

Jeder Kanal des *Logic-Analyzers* verfügt über eine *Triggerspalte*. Wenn Du darauf klickst, dann bekommst Du die Möglichkeit auszuwählen, wann ein *Trigger* ausgelöst werden soll.

Die Symbole haben unterschiedliche Bedeutungen.

Symbol	Bedeutung
X	Unabhängig von irgendwelchen Impulsänderungen
0	Startet bei LOW-Pegel
1	Startet bei HIGH-Pegel
┐	Startet bei ansteigender Flanke
┌	Startet bei abfallender Flanke
↑	Startet bei ansteigender bzw. abfallender Flanke

Weitere umfangreiche Hinweise finden sich in der Hilfe.

Anzeigen von Binärcode

Aber was nützt alle Theorie, wenn wir uns nicht einmal ein konkretes Beispiel anschauen können. Wir wollen einmal den Inhalt einer Variablen hinsichtlich des *Binärcodes* sichtbar machen. Ich beschränke mich dabei auf die untersten 4 Bits, denn das sollte als Anschauung vollkommen ausreichen. Als Controller-Board verwende ich dafür das *chipKIT Max32*,

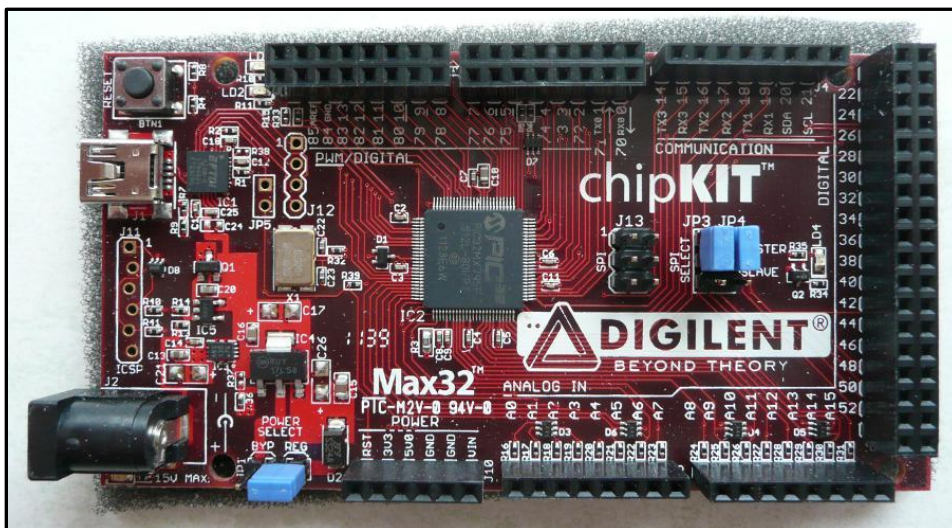


Abbildung 46 Das chipKIT Max32 -Board

das ein hervorragendes Mikrocontroller-Board ist. Nachfolgend siehst Du den Schaltungsaufbau in Kombination mit dem *EEBoard*.

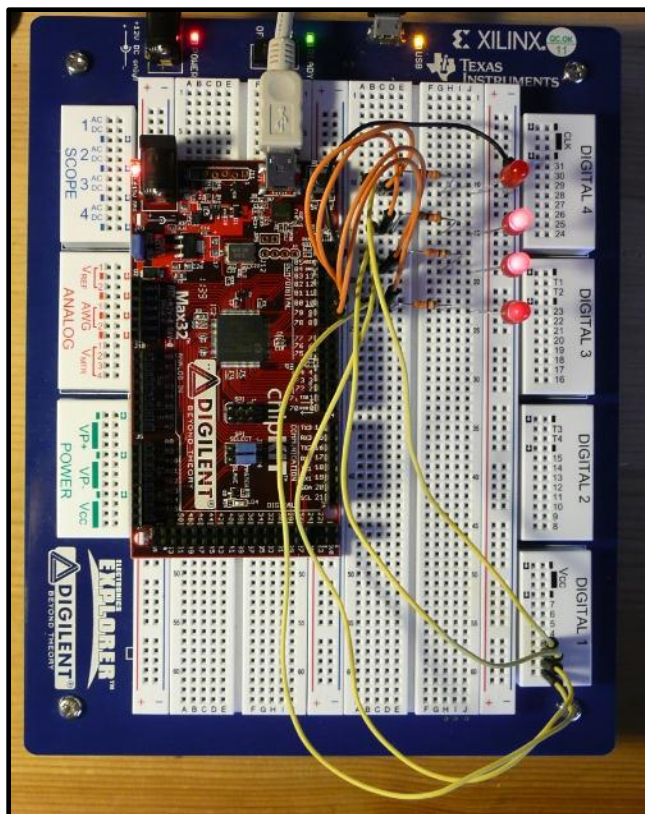


Abbildung 47 Das chipKIT Max32-Board im Zusammenspiel mit dem EEBoard

Auf dem *chipKIT Max32 Board* habe ich zur Ansteuerung der vier Leuchtdioden über 330 *Ohm* Vorwiderstände, die digitalen Ausgänge 8, 9, 10 und 11 verwendet. Schauen wir uns dazu erst einmal den Programmcode an.

```

LogicAnalyzer001 | Mptide 0023-windows-20111221
File Edit Sketch Tools Help
LogicAnalyzer001
byte zaehler; // Zählervariable

void setup(){
  pinMode(8, OUTPUT); // digitaler Ausgang Pin 8
  pinMode(9, OUTPUT); // digitaler Ausgang Pin 9
  pinMode(10, OUTPUT); // digitaler Ausgang Pin 10
  pinMode(11, OUTPUT); // digitaler Ausgang Pin 11
}

void loop(){
  digitalWrite(8, (zaehler&1)==1?HIGH:LOW); // Bit 0 abfragen
  digitalWrite(9, (zaehler&2)==2?HIGH:LOW); // Bit 1 abfragen
  digitalWrite(10, (zaehler&4)==4?HIGH:LOW); // Bit 2 abfragen
  digitalWrite(11, (zaehler&8)==8?HIGH:LOW); // Bit 3 abfragen
  delay(100); // 10ms Pause
  zaehler++; // Zähler um den Wert 1 erhöhen
}

Done compiling.

Binary sketch size: 6640 bytes (of a 520192 byte maximum)

14 chipKIT MAX32 on COM9

```

Abbildung 48 Programmcode zur Ansteuerung der 4 LED's

Innerhalb der *setup*-Funktion, die *einmalig* zu Beginn des Programmlaufes ausgeführt wird, legen wir fest, dass die benötigten digitalen Pins 8, 9, 10 und 11 als *Ausgänge (OUTPUT)* arbeiten sollen. Das erfolgt über die *pinMode*-Funktion. Innerhalb der *loop*-Funktion, die eine Endlosschleife darstellt, werden die einzelnen LED's über die *digitalWrite*-Funktion entsprechend angesteuert. Als Zähler arbeitet eine Variable des Datentyp *byte*, die innerhalb der *loop*-Funktion bei jedem Durchlauf um den Wert 1 erhöht wird. Ist der maximale Wert von 255 erreicht, beginnt die Zählung wieder bei 0 und stellt kein Problem dar. Schauen wir uns nun das entsprechende Diagramm des *Logic-Analyzers* an.

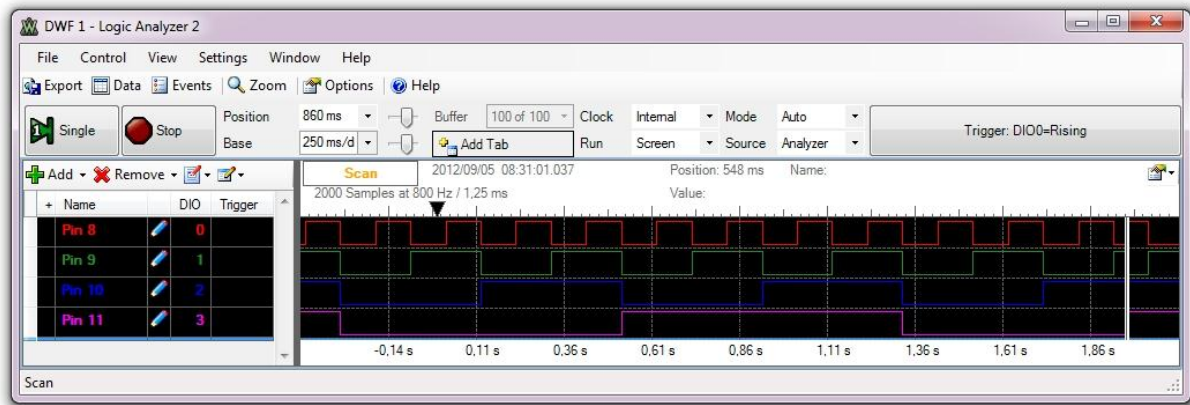


Abbildung 49 Anzeige der 4 Bytes im Logic-Analyzer

Vielleicht sagt dem einen oder anderen diese Anzeige nichts, doch wer sich mit Binärzahlen auskennt, der wird sofort den Zusammenhang erkennen. Vergrößern wir doch einfach einmal den Ausschnitt der Logik-Pegel und schreiben die entsprechenden Werte darunter.

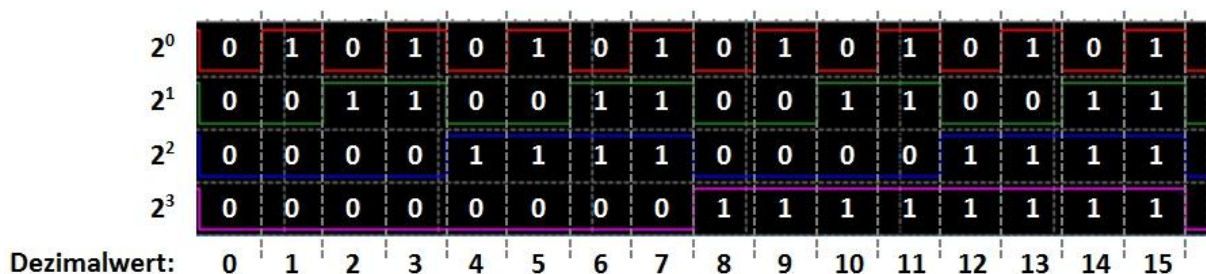


Abbildung 50 Die Logik-Pegel mit den entsprechenden Werten

Jetzt erkennen wir recht schnell, dass die einzelnen Pegel untereinander gesehen genau den Binärzahlen von 0000 bis 1111 entsprechen. Das sind die dezimalen Werte von 0 bis 15.

Der Digital-Pattern-Generator

Der *Digital-Pattern-Generator* wird über die *Patterns-Schaltfläche* gestartet und ermöglicht Dir, unterschiedliche digitale Muster zu generieren.

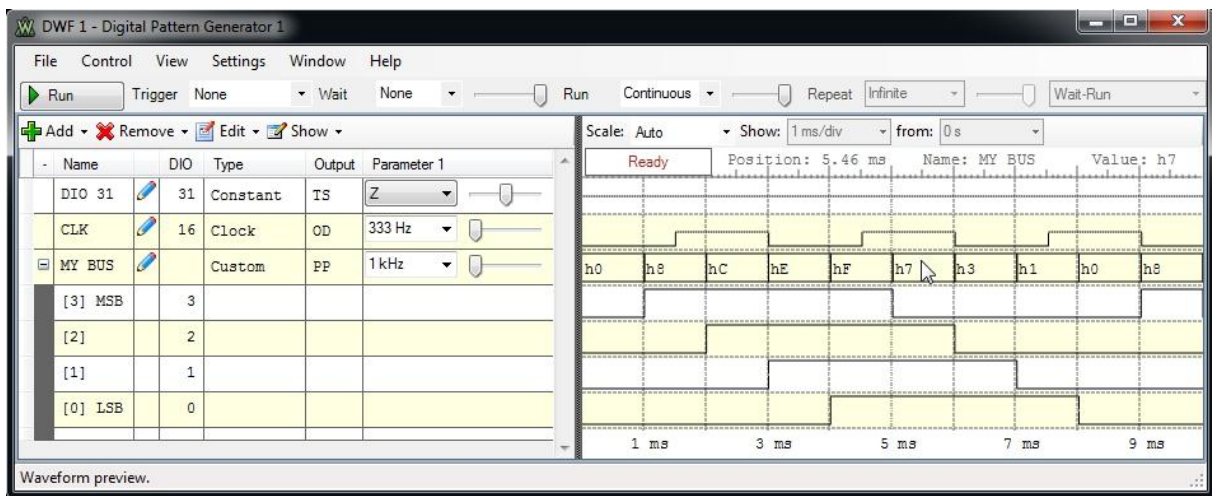
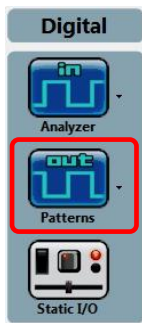
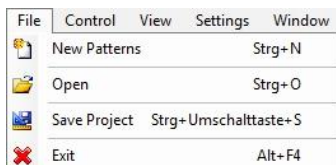


Abbildung 51 Der Digital-Pattern-Generator (Quelle: Hilfe-Seite)

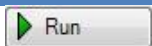
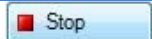
Über das *File-Menu* kannst Du die unterschiedlichen Aktionen zur Steuerung auswählen:



- *New Patterns*: Öffnet ein neues Pattern-Instrument
- *Open*: Öffnet ein bestehendes Pattern-Projekt von einer Datei
- *Save Project*: Speichert ein Pattern-Projekt in eine Datei
- *Exit*: Schließt den Digital-Pattern-Generator

Die Kontroll-Schaltflächen

Die Schaltflächen zur Steuerung haben folgende Bedeutungen:

Schaltfläche	Bedeutung
 Run	Startet den Digital-Pattern-Generator
 Stop	Stoppt den Digital-Pattern-Generator

Unterschiedliche Pattern

Du hast die Möglichkeit, die unterschiedlichsten Pattern generieren zu lassen. Auf dem folgenden Bild erkennst Du in der Spalte *Type* die verschiedenen Pattern *Constant*, *Clock*, *Random* und *Custom*.

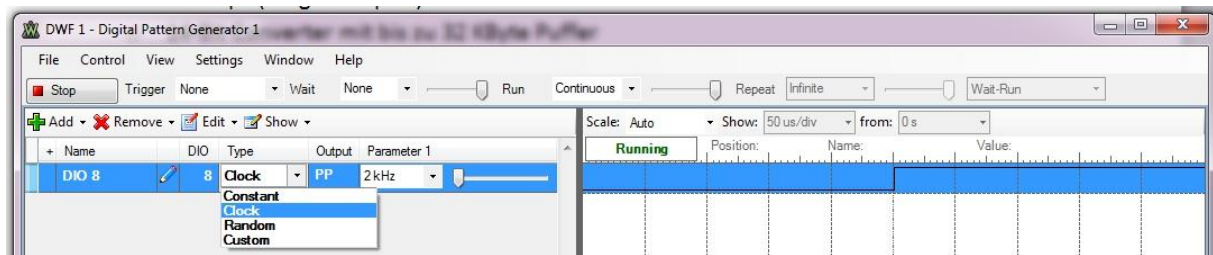
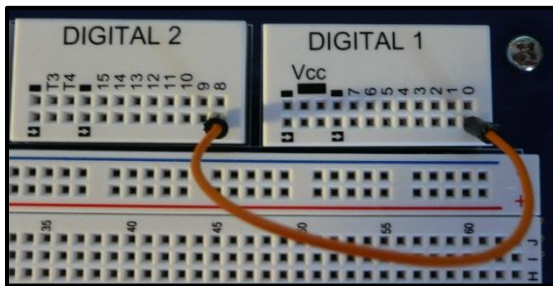
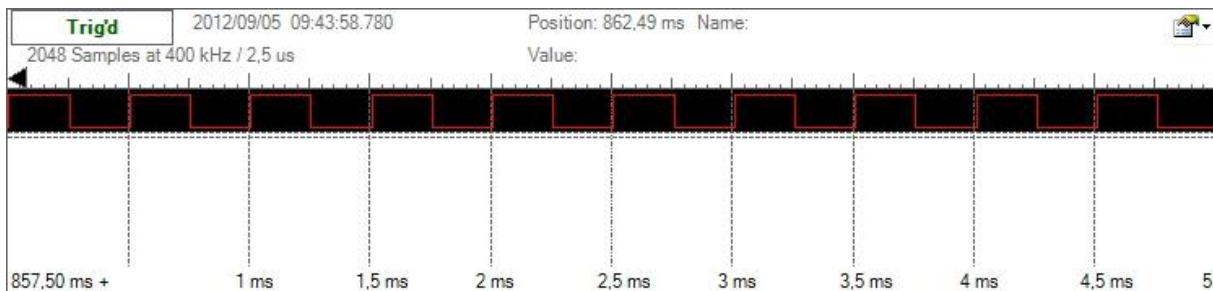


Abbildung 52 Unterschiedliche Pattern-Types

Ich habe mich für *Clock* entschieden und den *Pin 8* auf dem *Digital 2-Block* mit dem *Pin 0* auf dem *Digital 1-Block* verbunden.



Somit kann ich z.B. im *Logic-Analyzer* sehen, was ich für ein Signal generiere. Mein *Clock-Parameter 1*, mit dem ich die *Frequenz* eingestellt habe, zeigt den Wert *2KHz* an. Wir wollen einmal sehen, was unser *Logic-Analyzer* dazu sagt.



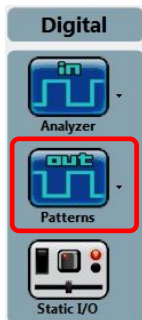
Für eine Periodendauer werden *0,5ms* benötigt. Die Frequenz wird wie folgt berechnet:

$$f = \frac{1}{T} = \frac{1}{0,5ms} = 2000 \text{ Hz} = 2\text{KHz}$$

Hey, das passt! Für weitere Einstellmöglichkeiten, wirf bitte einen Blick in die Hilfe.

Static I/O

Das *Static I/O-Panel* öffnest Du über die *Static I/O-Schaltfläche* - wer hätte das gedacht. Mit diesem Tool hast Du die Möglichkeit, die verschiedensten statischen Signale zu generieren, bzw. abzufragen. Das wären also z.B.



- Taster
- Schalter
- Slider
- Progress-Bar

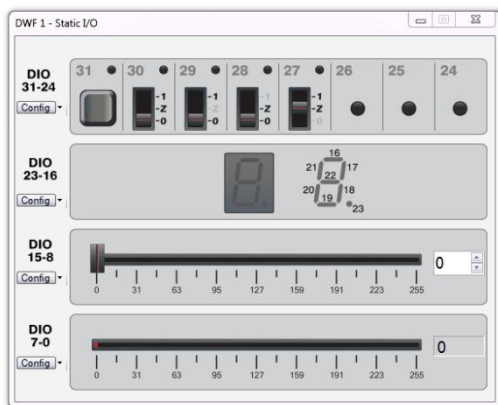
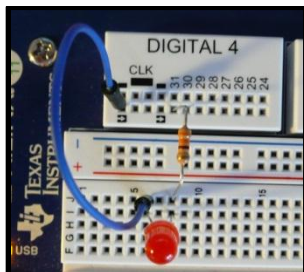


Abbildung 53 Das Static I/O-Panel

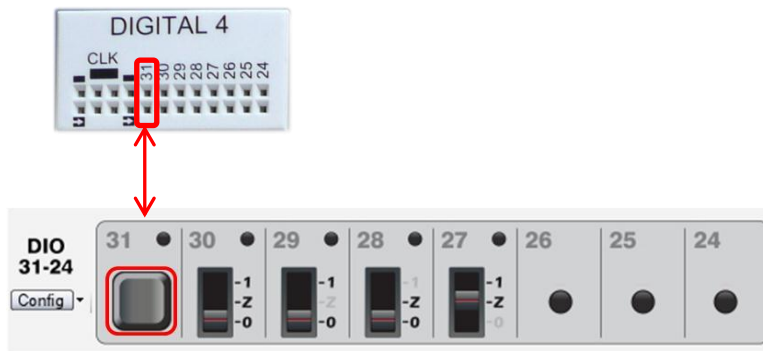
Alle Anschlüsse sind über die digitalen Blöcke *Digital 1* bis *Digital 4* zu erreichen. Die nachfolgend gezeigten Beispiele beziehen sich auf eine von mir erstellte Konfiguration, die jedoch hinsichtlich der Pinbelegung abweichen kann. Über die *Config-Schaltfläche*, der in jeder *I/O-Zeile* vorhanden ist, können unterschiedliche Pinbelegungen erzielt werden.

Der Taster (Eingabe)

Sehen wir uns dazu doch einfach einmal ein konkretes Beispiel an. Ich möchte über einen Taster eine LED ansteuern, die ich über einen Vorwiderstand von $330\ \Omega$ an den *Pin 31* im *Digital-Block 4* angeschlossen habe. Doch schau her:



Jetzt kann ich über den *Taster 31* eben diese LED ansteuern.

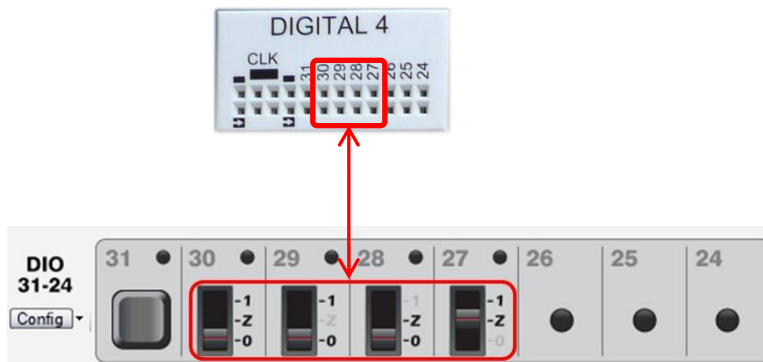


Was soll denn das für einen Sinn machen, eine LED anzusteuern? Darunter kann ich mir ja nun Garnichts vorstellen.

Ok, ok! Das war doch nur ein Beispiel. Du kannst das Ausgangssignal nutzen, um irgendeine Schaltung mit einem Signal zu versorgen, das Du über den *Taster 31* oder auch die *Schalter 27 bis 30* manipulierst.

Die Schalter (Eingabe)

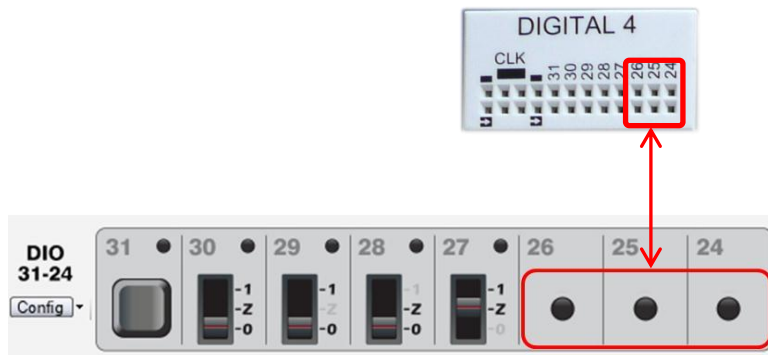
Ein *Taster* schließt den Kontakt nur solange, wie Du ihn auch betätigst. Ein Schalter hat mehrere stabile Schaltzustände.



Die digitalen Anschlüsse 27, 28, 29 und 30 werden für diese Schalter verwendet.

Die LED's (Ausgabe)

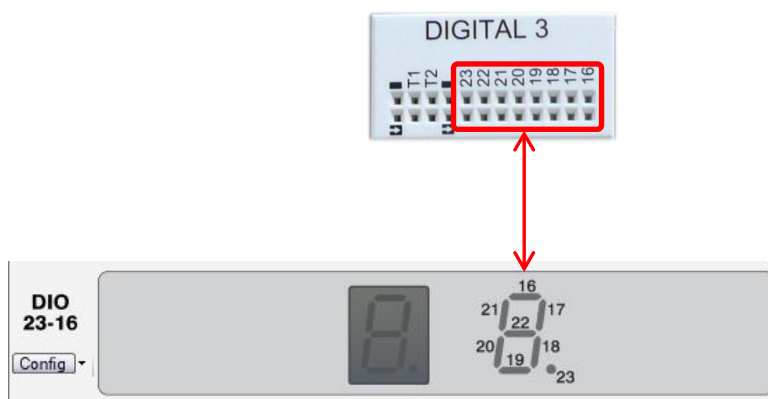
Um nicht unbedingt LED's an Deine Schaltung anzuschließen, weil Du vielleicht gerade keine bei der Hand hast, dann kannst Du auch die drei LED's an den Anschlüssen 24, 25 und 26 verwenden.



Die Siebensegmentanzeige (Ausgabe)

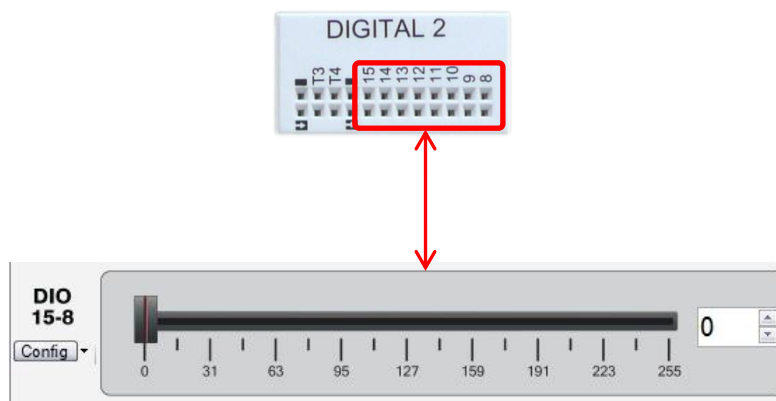
Eine *Siebensegmentanzeige* eignet sich hervorragend, um Werte von 0 bis 9 oder auch ein paar Buchstaben wie z.B. A, b, C, d, E, F, G, H, I, J, L, n, o, P oder U darzustellen. Die Anschlüsse der einzelnen Segmente stehen direkt daneben und müssen mit den digitalen Pins verbunden werden.

Schau her:



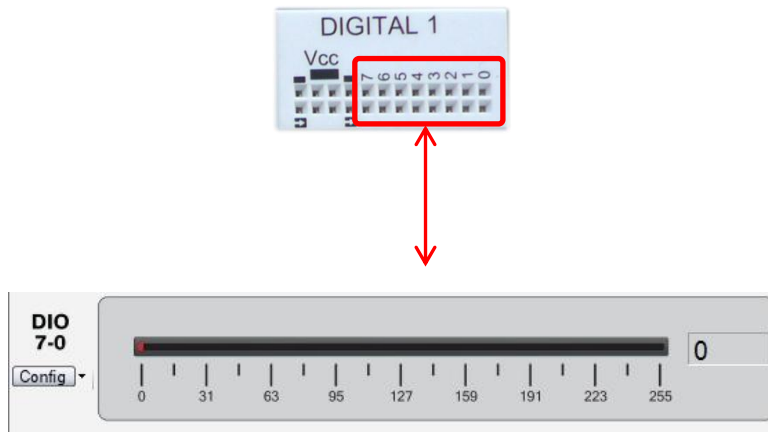
Der Slider (Eingabe)

Der *Slider* ist ein Schieberegler, der 8-Bit Werte von 0 bis 255 an die entsprechenden digitalen Ausgänge 0 bis 7 liefert. Es werden dabei die Werte im *Binärcode* übermittelt.



Der Progress Bar (Ausgabe)

Der *Progress Bar* dient zur Anzeige von Werten (0 bis 255), die im 8-Bit Binärformat vorliegen.



Abschließende Informationen

Der Hersteller *Digilent* ist unter der folgenden Internetseite zu finden:

<http://www.digilentinc.com/>

Details zum *EEBoard* finden sich unter:

<http://www.digilentinc.com/Products/Detail.cfm?Prod=EEBOARD>

www.trenz-electronic.de

Informationen zur Software *Waveform* finden sich unter:

<http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,849&Prod=WAVEFORMS>

Informationen zu meiner Person sind unter

www.erik-bartmann.de

zu finden.

Viel Spaß

Erik Bartmann