

InterRadio – Vortrag

Stefan Lehmann (DD4GS)
Peter Hansmann (DH0GHP)

06.11.2004

Schrägglage
erkennen

**Neigungssensor mit „Flash Of Inspiration“
bequem auswerten**

Sehr geehrte Damen und Herren,

ich möchte Sie alle recht herzlich zu unserem Vortrag „Schräglage erkennen“ hier auf der InterRadio 2004 begrüßen.

Mein Name ist Stefan Lehmann. Ich werde heute von Herrn Hansmann unterstützt. Während meines Vortrages wird er den Laptop und den Beamer bedienen. Am Ende des Vortrages werden wir ihnen beiden für weitere Fragen gerne zur Verfügung stehen.

In diesem Vortrag werden wir ihnen zeigen, wie man mit Hilfe eines PIC-Microcontrollers die Signale eines Sensors erfasst, umrechnet und anzeigen kann. Der PIC-Microcontroller sitzt auf der Universal-Experimentierplatine „Flash Of Inspiration“ die wir ihnen kurz vorstellen wollen. „Flash Of Inspiration“ bedeutet „Geistesblitz“. Ziel bei der Entwicklung von „Flash Of Inspiration“ durch den OM DF2OAG war ein möglichst variables System zu bekommen. Es sollte die wichtigsten Komponenten des Elektronikertages beinhalten. Auf dem **Bild 1** sehen Sie eine Abbildung einer fertig aufgebauten Platine. Hier ist die LC-Anzeige, darunter 2 Siebensegment-Anzeigen sowie ein Miniaturlautsprecher und unter diesen eine Matrixtastatur mit 12 Tasten. Daneben sieht man zwei einzelne Taster, 4 Leuchtdioden sowie zwei Widerstandstrimmer.

Neben der LCD befinden sich der Microcontroller PIC 16F628 und verschiedene I2C-Bausteine. Es sind dies ein 8-kanaliger Portbaustein, ein EEPROM, eine Echtzeituhr mit RAM und ein Temperatursensor.

Rechts neben dem PIC befinden sich zwei Miniaturrelais. Sie erlauben die Programmierung des PICs in der Schaltung. Man erspart sich somit das lästige „raus aus der Fassung, rein ins Programmiergerät“. Neben der wesentlichen Erleichterung der Arbeit, werden auch die Beinchen des PICs geschont.

Links davon befindet sich das Schaltnetzteil. Gegenüber einem Linearregler hat es den Vorteil, dass die Verlustwärme deutlich geringer ist. Dafür ist der Bauteileaufwand etwas größer. Dennoch macht der Einsatz eines Schaltreglers an dieser Stelle durchaus Sinn.

Unterhalb des Spannungsreglers befindet sich die RS232-Schnittstelle. Damit kann ein Datenaustausch mit einem PC realisiert werden.

Das wichtigste aber ist das darunterliegende Anschlussfeld. Hier werden alle wichtigen Signale auf einzelne Buchsen geführt. Auf diese Weise kann man mittels eines Drahtes diese Komponenten x-beliebig mit den einzelnen I/O-Pins des PICs verbinden. Dies bringt die erforderliche Flexibilität die im Alltag des Entwicklers so wichtig ist. Zusätzlich sind die I/O-Pins noch auf Stiftleisten herausgeführt, so dass einer Erweiterung mittels Flachbandkabel zu externen Platinen nichts im Wege steht. Von dieser Möglichkeit machen wir an dieser Stelle auch tatsächlich Gebrauch.

Eine Beschreibung der gesamten Platine sowie eine detaillierte Aufbauanleitung finden Sie im Internet unter

www.T555.de/flash

oder

www.iL-online.de

Da 45 Minuten sehr schnell vorbei sind, möchte ich gleich mit dem Vortrag beginnen.

(Bild 2)

„Schräglage erkennen“. Wie lässt sich so etwas realisieren? Im Allgemeinen verwendet man dazu eine Wasserwaage oder ein Senkblei. Um einen Neigungswinkel mittels Elektronik messen zu können, bedarf es eines speziellen Sensors. Solch ein Sensor wird u.a. auch von Analog Device hergestellt. Wir haben uns für den ADXL213 entschieden. **(Bild 3)** Auf der Leinwand sehen Sie einen abgebildet.

Wie arbeitet nun solch ein Sensor? Im Prinzip ganz einfach. **(Bild 4)** Sie sehen, dass eine Masse an zwei Bändern aufgehängt ist. Diese Masse wird je nach ihrer Lage durch die Erdbeschleunigung mehr oder weniger verschoben. Die Bänder sorgen dafür, dass sie bei absolut horizontaler Lage wieder in ihre Ursprungsstellung zurückkehrt. Die durch ein Kippen verursachte Lageänderung wird durch einen Sensor erfasst. Dieser Sensor ist nichts anderes als ein Kondensator, dessen eine Platte fest an dieser Masse befestigt ist. Aus Symmetriegründen verwendet man einen Differenzkondensator. Dieser besteht aus 2 feststehenden Platten. In Ruhelage sind beide Kapazitätswerte gleich groß. Bei einer Auslenkung der Masse wird dann die eine Kapazität größer und im gleichen Maße die andere kleiner.

Der Sensor wertet nun diese Kapazitätsänderungen aus und gibt an seinem Ausgang ein pulsweitenmoduliertes Signal ab **(Bild 5)**. In Ruhelage liefert der Sensor ein Impulsverhältnis von 1:1. Je nach dem in welche Richtung man den Sensor neigt, wird die Impulsbreite T1 größer oder kleiner. Die Gesamtperiodendauer T2 bleibt dabei konstant.

Kommen wir zum Schaltbild. **(Bild 6)** Sie sehen, dass neben dem Sensor nur sehr wenige passive Bauteile notwendig sind. Die Spannungsversorgung des Sensors erfolgt über den Tiefpass R1 und C1. Dieser sorgt dafür, dass die Spannungsversorgung des Sensors sehr sauber und rauschfrei ist. In Geräten mit einem Mikrocontroller haben sie immer ein Rauschen auf den Versorgungsleitungen. Diese entstehen durch das schnelle Schalten der I/O-Pins. Dabei entstehen starke Lastwechsel, die Stromschwankungen zur Folge haben. Diese Stromänderungen erzeugen nun an den Leitungswiderständen entsprechende Spannungsänderungen, welche sehr steilflankig sein können. Und je steilflankiger ein Signal ist, desto höhere Frequenzanteile sind vorhanden, die wiederum einfacher in anderen Signalkreise einkoppeln können.

Die Kondensatoren C2 und C3 legen die Bandbreite des internen Sensorsignals fest. Diese Bandbreite bestimmt damit auch die maximale Abtastrate mit der das Signal über einen AD-Wandler eingelesen werden kann. In diesem Fall ist aber vor allem die Minimierung des Rauschens von Interesse. Je niedriger die Grenzfrequenz liegt, desto besser können die tieferfrequenten Rauschanteile unterdrückt werden. Die Rauschanteile bestimmen die Auflösung des Signals. Wenn Sie beispielsweise einen Wert mit 10-Bit auflösen wollen, müssen die Rauschanteile deutlich niedriger sein als bei einer Auflösung mit 8-Bit, sonst kann es sein, dass die niederwertigsten Stellen entsprechend schwanken. Wir haben eine Bandbreite von ca. 23Hz eingestellt.

Die Periodendauer T2 lässt sich in weiten Grenzen verändern. In diesem Aufbau beträgt sie ca. 8,2 ms, was einer Frequenz von etwa 120 Hz entspricht. Da wir diese Zeiten in unserem Programm mit Hilfe von Zählschleifen erfassen, hat die Periodendauer direkten Einfluss auf unsere Messgenauigkeit.

Die pulsweitenmodulierten Signale werden nun einfach auf die Eingänge des PIC 16F628 gegeben. Das nächste Bild **(Bild 7)** zeigt die Sensorschaltung zusammen mit den Komponenten, die wir in diesem Aufbau verwenden.

Nachdem nun unsere Sensorschaltung über ein Flachbandkabel mit der Platine „Flash Of Inspiration“ verbunden ist, können wir mit der Erstellung des PIC-Programmes beginnen.

Programmiert wird das Ganze mit Hilfe von iL_BAS16. (**Beamer auf iL_EDy umschalten**) Das ist ein PIC-BASIC Compiler, der für diese Aufgabe ideal geeignet ist. (Umschalten zu iL_EDy). Sie sehen hier die integrierte Entwicklungsoberfläche iL_EDy. Damit lässt sich sehr schnell und effizient programmieren.

Wir wollen nun das Programm für die Auswertung der Signale des ADXL 213 Schritt für Schritt entwickeln.

Den Programmkopf haben wir bereits eingegeben. Nun müssen wir den PIC und die Variablen die wir verwenden wollen, definieren. Der PIC wird einfach mit der Zeile:

```
define device 16F628, wdt_off, cmcfig7, irc_osc, MCLR_INT, OSC2_IO  
festgelegt.
```

Die Parameter, die hier aufgelistet sind, legen bestimmte Funktionen des PICs fest. So haben wir hier den Watchdog und die Komparatoren abgeschaltet, verwenden den internen RC-Oszillator, der Pin 4 soll als I/O-Pin arbeiten und der Reset soll dafür intern generiert werden. Auch der Pin 15 (OSC2) soll als gewöhnlicher I/O-Pin verwendbar sein.

Es folgt die Definition der Variablen. Hier ist es sinnvoll, die Variablennamen so zu wählen, dass sie einen direkten Bezug zum Signal haben. Da Symbolnamen bis zu 16 Zeichen lang sein dürfen, entsteht somit ein leicht lesbares Programm.

Tx1 heißt unsere erste Variable. In ihr soll der Impuls T1 erfasst werden und soll einen 16-Bit Wert aufnehmen können. Deswegen müssen wir sie als Wortvariable definieren. Dabei bekommt sie die RAM-Speicherstelle 20H und 21H zugewiesen. Im BASIC wird einer Hexzahl ein \$-Zeichen vorangestellt. Ein %-Zeichen als Präfix bedeutet, dass es sich um eine Binärzahl handelt. Fehlt der Präfix, ist es eine gewöhnliche Dezimalzahl. Damit wir später noch wissen, was wir an dieser Stelle gedacht haben, folgt ein möglichst sinnvoller Kommentar.

Als weiteres definieren wir die Variable für die Y-Achse. Danach folgen die Variablen für die Gesamtperiodendauer, Ergebnisse und Vorzeichen. Dabei werden die Vorzeichenbits in einer Byte-Variablen hinterlegt.

Für das Vorzeichen benötigt man nur ein Bit. Deshalb definieren wir das Vorzeichen für den X- bzw. Y-Wert als Bitkonstanten. Dieses wird die Lesbarkeit des Programms weiter verbessern, außerdem ersparen wir uns dadurch etwas Schreibarbeit.

Die nächsten Bitdefinitionen dienen ebenfalls zur Vereinfachung. Die I/O-Pins, an denen die Sensorsignale anliegen, bekommen die Bezeichnungen XIN und YIN. Der Summer, den wir dann ansteuern wollen, wenn der Winkel größer als 65° ist, wird über ein Stück Draht vom Verdrahtungsfeld (links unten auf der Platine), Anschluss LS, mit dem Kontakt RB7 verbunden.

Nun definieren wir noch die Frequenz, mit der der PIC betrieben wird. Beim 16F628 schwingt der interne RC-Oszillator mit ca. 4 Mhz. Diesen Wert geben wir hinter dem Schlüsselwort XTAL ein. Mit dieser Angabe kann der Compiler alle notwendigen Zeitschleifen richtig berechnen und initialisieren.

Alle bisher gemachten Angaben dienen lediglich dazu, dem Compiler und PIC mitzuteilen, wie die grundsätzliche Funktionsweise ist.

Nun wird es spannend. Denn jetzt verwenden wir zum ersten Mal einen BASIC-Befehl. Mit dem Befehl TRIS kann man die Datenrichtung der einzelnen I/O-Pins umschalten. Nach dem Einschalten sind alle I/O-Pins auf Eingang gestellt. Will man den einen oder anderen als Ausgang arbeiten lassen, muss dieser umgeschaltet werden. Dabei ist die Bitnummer des Ports auch die Bitnummer im Argument des XTAL-Befehls. Soll ein Pin als Ausgang arbeiten, muss an der entsprechenden Position eine 0, für einen Eingang eine 1 stehen. Somit lautet die erste BASIC-Anweisung:

```
TRIS RA,%01111111
```

Führt der PIC diesen Befehl aus, arbeitet der I/O-Pin RA7 als Ausgang, alle übrigen Pins des Port RA sind weiterhin Eingänge. In der nächsten Zeile steht die Anweisung

```
TRIS RB,0
```

was bedeutet, dass der gesamte Port RB als Ausgang arbeitet. Bitte beachten Sie, dass im ersten Fall das Argument als Binärzahl angegeben wurde, im zweiten Fall jedoch als normale Dezimalzahl.

Als nächstes folgt nun eine BASIC-Anweisung die zeigt, um wieviel einfacher die Programmierung eines solchen PICs mittels BASIC-Compiler ist. LCDINIT schickt eine ganze Reihe von Befehlen an die LCD, damit diese arbeitet. Gäbe es diesen Befehl nicht, so müsste man im Datenblatt der LCD nachschauen, welche Befehle mit welchen Argumenten die LCD benötigt, um sie richtig zu initialisieren. Dank des BASIC Compilers reduziert sich der Aufwand zu einer Zeile.

```
LCDINIT RB,2,16
```

Somit wird die LCD über den Port RB angesteuert. Sie besitzt 2 Zeilen mit je 16 Zeichen. Es gibt noch weitere Argumente für diesen Befehl. Diese sind hier jedoch nicht erforderlich.

Die nächsten drei Zeilen lassen beim Einschalten des Gerätes für 2 Sekunden den Eröffnungstext erscheinen. Auch hier reduziert sich der Aufwand für die Textausgabe auf der LCD zu einem Einzeiler. Nach dem Schlüsselwort LCDWRITE erfolgt die Positionsangabe, ab der der Text geschrieben werden soll.

Das folgende GOTO START überspringt die nachfolgenden Unterprogramme. Dazu eine kleine Bemerkung: Es gibt PIC-Bausteine mit einem Programmspeicher von 4 oder 8k Worten. In diesen ist aber der Programmspeicher nicht linear adressierbar sondern in Segmente eingeteilt. Diese Segmente werden als Pages bezeichnet. Wenn nun das Programm von einer Page in eine andere Page verzweigen soll, müssen ganz bestimmte Vorkehrungen getroffen werden, damit die Seite auch aktiviert wird. Dieser Mechanismus beruht auf dem sogenannten PCLATH-Register. iL_BAS16 unterstützt diesen Mechanismus. Beim Assemblerprogrammieren muss der Programmschreiber selbst darauf achten, wenn der Programmcode so groß wie eine Page wird, dass er dieses PCLATH-Register zum richtigen Zeitpunkt entsprechend der neuen Programm-Page setzt. Aber wehe dem, wenn man zusätzlich Code einfügen muss. Dann verschiebt sich nämlich alles, auch die Programmstellen an denen die Seite wechselt. All diese Probleme nimmt uns iL_BAS16 ab. Der Algorithmus, der dahinter steckt, ist nicht ganz trivial und dass dieser auch sicher arbeiten kann, sollte man sich angewöhnen, die Unterprogramme soweit wie möglich an den Programmanfang zu stellen. Deshalb an dieser Stelle die GOTO START Anweisung.

Welche Unterprogramme wir benötigen, entscheidet sich später. Wir schreiben zuerst das Hauptprogramm weiter.

Das Hauptprogramm bekommt als erstes die Sprungmarke START: Damit kann nun der Compiler dem GOTO START auch tatsächlich eine bestimmte absolute Stelle im Programm zuordnen. Sprungmarken dürfen wie die anderen Symbole bis zu 16 Zeichen haben und werden mit einem Doppelpunkt abgeschlossen.

Mit LCDWRITE beschreiben wir die LCD mit einem neuen Text. Anschließend kommt erneut eine Sprungmarke. Wir nennen sie LOOP: um anzudeuten, dass es sich hier um eine Programmschleife handelt. Danach wäre es nun an der Zeit, den Sensor auszulesen.

Das soll ein Unterprogramm mit dem Namen MESSEN übernehmen. Eigentlich wäre an dieser Stelle kein Unterprogramm notwendig, da es im ganzen Programm nur an einer Stelle aufgerufen wird. Der Aufruf erfolgt mit der BASIC-Anweisung GOSUB. Die Sprungmarke soll MESSEN: heißen.

Nachdem das Unterprogramm abgearbeitet ist und die Werte für T1x, T2x, T1y und T2y ermittelt worden sind, werden diese dann weiterbearbeitet.

Doch schreiben wir zuerst unser Unterprogramm. Dazu geben wir an der entsprechenden Stelle die Sprungmarke MESSEN: ein. Die Messung der Periodendauer der einzelnen Signale erfolgt mit Hilfe von Zählschleifen. Das Ablaufdiagramm verdeutlicht, wie das Programm arbeiten soll.

(Bild 8)

Für die Y-Achse gilt prinzipiell der gleiche Ablauf, nur dass eben die Variablen T1y und T2y verwendet werden.

(iL_EDy wieder zeigen)

Die Umsetzung dieses Ablaufdiagramms ist relativ einfach. Zuerst werden die Zählervariablen gelöscht.

```
LET T1x = 0
LET T2x = 0
LET T1y = 0
LET T2y = 0
```

Um eine steigende Flanke erkennen zu können, muss man auf einen Low-Pegel warten. Ist nämlich der aktuelle Pegel an diesem Pin auf High, so liegt die steigende Flanke ja bereits hinter uns und wir müssen auf die nächste steigende Flanke warten. Falls dieser Fall vorliegt, verzeigen wir auf die Sprungmarke MESSEN1: zurück. Diese Abfrage der Signalfanken erfolgt mit der BASIC-Anweisung IF...THEN...ELSE.

```
IF Xin = 1 THEN GOTO MESSEN1
```

Wenn also der Pegel an Xin, also an RA,2, auf 1 ist, heißt das: verzweige auf den Anfang dieser Anweisung. Sollte Xin aber Low sein oder aber nach einer kurzen Zeit Low werden, wird die nachfolgende BASIC-Anweisung ausgeführt.

Nun führt der Eingang Xin Low-Pegel und wir können auf die steigende Flanke warten. Auch dies erfolgt mittels IF-Anweisung.

MESSEN2:

```
IF Xin = 0 THEN GOTO MESSEN2
```

Sobald die steigende Flanke kommt, also der Pegel an Xin von 0 auf 1 wechselt, wird diese Warteschleife beendet. Diese zwei IF-Abfragen entsprechen somit unserer ersten Raute im Ablaufdiagramm. **(Bild 8 nochmals kurz zeigen)**

Bei MESSEN3 beginnt die Zählschleife für T1x. Die Zählvariable T1x wird mit der BASIC-Anweisung um eins erhöht. Dazu verwenden wir den Befehl INC. Prinzipiell wäre auch die Anweisung LET T1x = T1x + 1 möglich, jedoch benötigt sie mehr Speicherplatz und auch mehr Zeit für die Abarbeitung.

```
LET T1x = T1x + 1 (schreiben und dann wieder löschen)
```

Die Erhöhung des Zählers erfolgt bis der Pegel an Xin von 1 auf 0 wechselt. Deshalb folgt der Erhöhung der Variablen erneut die Abfrage des Pins.

```
IF Xin = 1 THEN GOTO MESSEN3
```

Wenn der Pegel an Xin wechselt, ist die Dauer von T1x erfasst. Es folgt die Messung von T2x. Da T2x die gesamte Periodendauer ist, wir ab dieser Stelle aber nur den LOW-Anteil der Periode erfassen können, berechnen wir am Ende dieses Unterprogramms die Gesamtperiodendauer einfach dadurch, dass wir T1x und T2x addieren.

Bei der Sprungmarke MESSEN4: erfassen wir nun die Zeit des LOW-Anteils der Periode. Nach dem Wechsel von 0 auf 1 an Xin ist die Erfassung der 'Zeiten für die X-Achse beendet.

Ab MESSEN5 bis einschließlich MESSEN8 wird die Periodendauer der Y-Achse gemessen.

(Eingabe der entsprechenden Zeilen)

Anschließend berechnen wir Gesamt-T2 durch Addition des High- und Low-Anteils der Periode. Es folgt eine Rundung und Division durch 10. Das ist notwendig, damit sich bei der später durchgeführten Verhältnisbildung auch ein Wert größer 0 bildet.

Zur Erinnerung: Wir haben ja nur Zahlen zwischen 0 und 65535 zur Verfügung.

Die BASIC-Anweisung RETURN sorgt dafür, dass das Programm bei der Anweisung, die auf GOSUB folgt, fortgeführt wird.

Das Verhältnisses von T1x zu T2x ist ja unser Beschleunigungswert. Damit bei der Division das Ergebnis eine möglichst große Zahl wird, muss der Zähler so groß wie möglich sein. Deshalb wird er mit 50 multipliziert.

Warum gerade mit 50 und nicht mit 100?. Die gemessenen Werte von T1x und T1y können bis ca. 900 groß werden. 900×100 ergibt 90000, die Grenze für Wortvariablen liegt aber bei 65535. Eine Multiplikation mit 50 ergibt dagegen einen Maximalwert von 45000.

Mit

```
LET Ex = T1x / T2x
```

ermitteln wir die Beschleunigung. Allerdings muss noch der Nullwert abgezogen werden. Dieser Nullwert wird vom Hersteller als $T1x = 0,5 T2x$ definiert. Bei einem Tastverhältnis von 1:1 ist somit die Beschleunigung 0, aber das Verhältnis von T1x zu T2x beträgt dann 0,5. Somit muss dieser Nullwert von 0,5 abgezogen werden.

In unserer Berechnung muss aber als Nullwert die Zahl 25 abgezogen werden. Das entspricht dem Multiplikationswert 50 mal 10 mal 0,5. Der Faktor 10 stammt von der Division von T2 durch 10 und da T2 im Nenner steht, muss er an dieser Stelle multipliziert werden.

Bevor subtrahiert werden kann, muss allerdings geprüft werden, ob der ermittelte Wert größer bzw. gleich oder kleiner 250 ist. Denn ist dieser Wert kleiner 250, ist die Beschleunigung negativ.

Ist E_x größer oder gleich 250 wird das Vorzeichen positiv und der richtige Wert von E_x berechnet.

```
IF     $E_x \geq 250$  THEN GOTO Xpos ELSE Xneg
```

Xpos:

```
LET   $E_x = E_x - 250$ 
```

Ein positives Vorzeichen wird durch zurücksetzen des entsprechenden Vorzeichenbits gekennzeichnet.

```
RES   $V_x$   
GOTO LOOP1
```

Xneg:

```
LET   $E_x = 250 - E_x$   
SET   $V_x$ 
```

Dasselbe gilt auch für die Y-Achse.

(Eingabe der entsprechenden Zeilen).

Somit hätten wir jetzt das Programm soweit entwickelt, dass die angezeigten Ergebnisse bereits mit dem Neigungswinkel korrespondieren. Wir erhalten bei 0° einen Anzeigewert von ebenfalls 0, bei 90° erscheint der Wert 152 auf der LCD. Dies ist auch der Maximalwert, denn wenn wir weiterdrehen, wird der Wert wieder kleiner.

(Effekt zeigen, möglichst mittels Kamera)

Prüfen wir jetzt einmal nach, in wie weit unser Anzeigewert mit dem Winkel, den wir ja eigentlich anzeigen wollen übereinstimmt. Dazu drehen wir den Sensor um 45°. Das geschieht am Einfachsten mit Hilfe eines Geodreiecks. Sicherheitshalber achten wir auch darauf, dass der jeweils andere Messwert auf 0 steht.

(Zeigen)

Zu unserem Erstaunen lesen wir den Wert von 107 ab. Der Wert kann auch 1 darüber oder darunter liegen. Erwartet haben wir eigentlich den halben Wert vom Maximalwert, also etwa 76. Woher kommt diese große Diskrepanz?

Man könnte jetzt natürlich eine Messreihe erfassen, um so den Zusammenhang zwischen Soll- und Istwert zu ermitteln. Um das ganze abzukürzen will ich ihnen die Lösung sofort verraten. Der eine oder andere von ihnen hat es vielleicht auch schon erkannt.

Die Lösung liegt beim Sinus. Wir müssen nämlich wissen, dass bei der Gravitationskraft für alle Winkeln $> 0^\circ$ nur die jeweilige Komponente wirksam ist, die auch in Richtung Erdmittelpunkt zeigt. Bei einem Winkel von 45° beträgt die wirksame Gravitationskomponente $\sin(45^\circ)$ *Maximalwert. Der Sinus von 45° beträgt 0,707. Somit erhalten wir $0,707 * 152 = 107,5$. Dieses Ergebniss stimmt mit unserem Anzeigewert auch überein.

Wir müssen, um den Winkel im Display anzeigen zu können, unseren errechneten Wert entsprechend umrechnen. Die dazu notwendige Formel lautet:

$$\text{Wert}^\circ = \text{arc sin} (\text{Messwert} / 152)$$

Bemerkung:

arcsin ist die inverse Funktion des Sinus und wird auf dem Taschenrechner durch die Taste SIN^{-1} dargestellt.

Beispiel:

$$\begin{aligned} &\text{arc sin} (107 / 152) \\ &\text{arc sin} (0,704) \\ &= 44,7^\circ \end{aligned}$$

Leider kennt der BASIC-Compiler keine trigonometrischen Funktionen, zu solchen gehört u.a. der Sinus und ARC Sinus aber auch der Cosinus und Tangens. In diesen Fällen muss der Griff zur „Mathematischen Formelsammlung“ erfolgen. Dort finden wir für den arcsin folgende Reihenentwicklung.

$$\text{Arcsin } x = x + \frac{1}{2} * x^3/3 + \frac{1}{2} * \frac{3}{4} * x^5/5 + \frac{1}{2} * \frac{3}{4} * \frac{5}{6} * x^7/7 + \dots$$

Diese Reihenentwicklung ist also eine Näherungsformel, die bei Berücksichtigung von unendlich vielen Gliedern ein genaues Ergebnis liefert.

Bei Verwendung von Wortvariablen müssen leider sehr viele Rechenschritte und Rundungen durchgeführt werden um den Zahlenbereich von 65535 nicht zu überschreiten. Dabei werden nach dem Runden die Zahlen entweder durch 10 oder 100 dividiert damit das Ergebnis der darauffolgenden Multiplikation wieder in die 16-Bit-Variable passt. Diese vielen Rundungsvorgänge führen zu nicht mehr akzeptierbar großen Fehlern. Im Anhang ist eine solche Beispielrechnung für den Winkel von 45° beschrieben.

Die Alternative für solche Potenzreihenentwicklungen oder iterativen Berechnungen ist eine entsprechend aufgebaute Tabelle. Solche Tabellen haben allerdings den Nachteil, dass sie i.d.R. sehr viel Speicherplatz brauchen. In unserem Fall sind es 152 Einträge. Reicht eine Anzeigeaufösung von 1°-Schritten aus, dann bleibt es auch bei 152 Speicherplätzen. Will man noch eine Nachkommastelle mit anzeigen, so benötigt man pro Eintrag zwei Speicherplätze. Die Tabellenwerte lassen sich mittels Kalkulationsprogramm sehr einfach berechnen.

(Bild 9 zeigt die Tabellenwerte für Anzeigenwerte mit Nachkommastellen)

(zurück zu iL_EDy)

Eine Tabelle wird im iL_BASIC mit den Anweisungen DATA und READDATA gebildet. Die eigentliche Tabelle besteht aus dem Schlüsselwort DATA und den Argumenten, die immer Byte-Konstanten sein müssen. Das Auslesen erfolgt durch READDATA. Es können mehrere Variablen als Argumente diesem Befehl folgen, dann werden eben nicht nur ein sondern mehrere Tabelleneinträge ausgelesen. Nach einem Lesezugriff wird der interne Zeiger auf das nächste Datenelement gestellt. Das Schlüsselwort RESTORE setzt diesen Zeiger auf das erste vorkommende Datenelement.

Es ist sehr wichtig, dass ein Lesen über das Feldende hinaus verhindert wird. Deshalb prüfen wir ab, ob der Messwert größer als 152 ist, denn dieser wird in der nachfolgenden Berechnung als Index auf die Feldelemente verwendet.

```
IF    Ex > 152 THEN GOTO LOOP
IF    Ey > 152 THEN GOTO LOOP
```

Im Fehlerfall wird die aktuelle Messung einfach ignoriert und eine neue Messung initiiert.

Mit RESTORE stellen wir den Datenzeiger auf das erste Datenelement und addieren darauf den Messwert. Anschließend zeigt dieser Datenzeiger auf das korrespondierende Element, welches mit READDATA in die Variable W_x eingelesen wird. Das Gleiche geschieht für die Y-Achse.

```
RESTORE
LET  DATPTR_ = DATPTR_ + Ex
READDATA Wx
RESTORE
LET  DATPTR_ = DATPTR_ + Ey
READDATA Wy
```

Diese Winkelwerte werden auf der LCD in Zeile 2 ausgegeben. Falls der eine oder andere Wert negativ sein sollte, wird in zwei weiteren Zeilen das Minuszeichen dem Ergebnis vorangestellt.

```
LCDWRITE 2,1," ",Wx," ",Wy
IF    Vx = 1 THEN LCDWRITE 2,2,"-"
IF    Vy = 1 THEN LCDWRITE 2,10,"-"
```

Als nächstes wird geprüft, ob mindestens einer der Winkelwerte größer als 65 ist. Falls dem so ist, wird ein Signalton mit 800 Hz für die Dauer von 1 Sekunde aktiviert. Ansonsten wird einfach nur eine Sekunde gewartet. Danach kehrt das Programm zur Hauptschleife LOOP: zurück und ein neuer Messzyklus kann beginnen.

```
IF    Wx > 65 THEN GOTO LOOP3
IF    Wy > 65 THEN GOTO LOOP3
GOTO  LOOP4
LOOP3:
SOUND SUMMER,800,1000
GOTO LOOP
LOOP4:
WAIT  1000
GOTO LOOP
```

Nun können wir das Programm compilieren und auf die Platine „Flash OF Inspiration“ runter laden.

Erwähnen möchte ich noch, dass dieser Vortrag inclusive Folien und Programmbeispiele von den eingangserwähnten Internetseiten heruntergeladen werden kann.

(Compilieren & runter laden, eventl. Tippfehler korregieren)

Wir sind an Ende unseres Vortrages angekommen und ich möchte mich recht herzlich bei Ihnen für Ihre Aufmerksamkeit bedanken.

Während Herr Hansmann das Programm compiliert und in den PIC programmiert, können wie ja schon einmal mit der Diskussion beginnen.

(Demonstration)

Anhang

Programmlisting 1 (Anzeige der relativen Winkelwerte)

```

*****
'          NEIGUNG.BAS
'          Flash of Inspiration I
'
'(c) by Ing.Büro St.Lehmann, D-77756 Hausach, SL@iL-online.de
*****
rem Demoprogramm für Neigungs-/Beschleunigungssensor ADXL213 +
rem      LC-Anzeige sowie 16-Bit Arithmetik
rem
rem      Einrichten der Platine
rem
rem      LCD an PortB -> glatt durchschalten
rem      Zusatzplatine über Flachbandkabel an K2 anschließent
rem      PortA,7 mit Stift LS verbinden (Summer an PortA,7)
rem
rem      Programm in PIC 16F628 laden und starten
rem
rem      -----
rem      demonstrates how to work with accelerometer sensor ADXL213
rem      LCD and buzzer
rem
rem      the following settings and connections must be done
rem
rem      put switches "LCD an PORTB" 1 .. 6 to on
rem      additional pcb with sensor with flat cable to K2
rem      connect LS to PortA,7 (buzzer onto port RA,7)
rem
rem      load compiled program into "Flash Of Inspiration I"
rem

      SNCALDEF

'Watchdog off, all comparators off, MCLR- and OSC2-Pin are used for io
define device 16f628,wdt_off,irc_osc,cmcfg7,MCLR_INT,OSC2_IO

rem Variablen
define Tx1=$20 as word      'Zählvariable für X-Achse
define Ty1=$22 as word      'Zählvariable für Y-Achse
define Tx2=$24 as word      'Periodendauer X-Achse
define Ty2=$26 as word      'Periodendauer Y-Achse
define Ex=$28 as word       'Ergebnis X-Achse
define Ey=$2A as word       'Ergebnis Y-Achse
define Vorzeichen=$2c as byte 'Vorzeichen

rem Bit-Konstanten
define Vx = Vorzeichen,0
define Vy = Vorzeichen,1

rem Pindefinitionen
define xin = ra,2
define yin = ra,3
define summer = ra,7

      xtal 4.00          'xtal frequency

      tris ra,%01111111      'RA7 = Ausgang, Rest Eingang
      tris rb,0              'alles Ausgang
      lcdinit rb,2,16        'init. lcd with each 16 chars on 2 lines

      lcdwrite 1,1,"Lagemessung mit" 'Eröffnungstext anzeigen
      lcdwrite 2,1,"ADXL213 c)jL V1"
      wait 2000              '2 Sekunden anzeigen
      goto start            'Unterprogramme überspringen

***** es folgen die Unterprogramme *****
'MESSEN liest das Tastverhältnis für X- und Y-Achse ein
messen:

```

```

      let Tx1=0              'Startwerte
      let Ty1=0
      let Tx2=0
      let Ty2=0

messen1:
      if xin = 1 then goto messen1 'steigende Flanke für X suchen
messen2:
      if xin = 0 then goto messen2
messen3:
      inc Tx1                'Länge des H-Impulses
      if xin = 1 then goto messen3
messen4:
      inc Tx2                'Länge des L-Impulses
      if xin = 0 then goto messen4
messen5:
      if yin = 1 then goto messen5 'steigende Flanke für Y suchen
messen6:
      if yin = 0 then goto messen6
messen7:
      inc Ty1                'Länge des H-Impulses
      if yin = 1 then goto messen7
messen8:
      inc Ty2                'Länge des L-Impulses
      if yin = 0 then goto messen8
      let Tx2 = Tx2 + Tx1      'X2 = gesamte Periodendauer
      let Tx2 = Tx2 / 5        'gerundet durch 10 Teilen
      let Ty2 = Ty2 + Ty1      'y2 = gesamte Periodendauer
      let Ty2 = Ty2 / 5
      let Ty2 = Ty2 / 10
      return

***** hier beginnt das Hauptprogramm *****

start:
      lcdwrite 1,1,"Lage X Lage Y "
      lcdwrite 2,1," messen      "

loop:

      gosub messen
      let Tx1 = Tx1 * 50        '100 wäre zu groß
      let Ty1 = Ty1 * 50
      let Ex = Tx1 / Tx2
      if Ex >= 250 then goto Xpos else goto Xneg

Xpos:
      let Ex = Ex - 250
      res Vx
      goto Loop1

Xneg:
      let Ex = 250 - Ex
      set Vx

Loop1:
      let Ey = Ty1 / Ty2
      if Ey >= 250 then goto Ypos else goto Yneg

Ypos:
      let Ey = Ey - 250
      res Vy
      goto loop2

Yneg:
      let Ey = 250 - Ey
      set Vy

Loop2:
      lcdwrite 2,1, Ex, " ",Ey
      if Vx=1 then lcdwrite 2,2,"-"
      if Vy=1 then lcdwrite 2,10,"-"
      wait 1000
      goto loop

```

Programmlisting 2 (Winkelangabe ohne Nachkommastelle)

```

*****
'          NEIGUNG.D.BAS
'          Flash of Inspiration I
'
'(c) by Ing.Büro St.Lehmann, D-77756 Hausach, SL@iL-online.de
*****
rem Demoprogramm für Neigungs-/Bescheinigungssensor ADXL213 +
rem LC-Anzeige sowie 16-Bit Arithmetik
rem incl. Umrechnung in absolute Winkelangaben (Potenzreihe f. arcsin)
rem
rem Einrichten der Platine
rem LCD an PortB -> glatt durchschalten
rem Zusatzplatine über Flachbandkabel an K2 anschließen
rem PortA,7 mit Stift LS verbinden (Summer an PortA,7)
rem
rem Programm in PIC 16F628 laden und starten
rem -----
rem demonstrates how to work with accelerometer sensor ADLX213
rem LCD and buzzer
rem incl. calculation of degree values (power series for arcsin)
rem
rem the following settings and connections must be done
rem put switches "LCD an PORTB" 1 .. 6 to on
rem additional pcb with sensor with flat cable to K2
rem connect LS to PortA,7 (buzzer onto port RA,7)
rem
rem load compiled program into "Flash Of Inspiration I"

        SNCALDEF

'Watchdog off, all comparators off, MCLR- and OSC2-Pin are used for io
define device 16f628,wdt_off,irc_osc,cmcfg7,MCLR_INT,OSC2_IO

rem Variablen
define Tx1=$20 as word    'Zählvariable für X-Achse
define Ty1=$22 as word    'Zählvariable für Y-Achse
define Tx2=$24 as word    'Periodendauer X-Achse
define Ty2=$26 as word    'Periodendauer Y-Achse
define Ex=$28 as word     'Ergebnis X-Achse
define Ey=$2A as word     'Ergebnis Y-Achse
define Vorzeichen=$2c as byte 'Vorzeichen
define Wx = $2d as byte   'Ergebnis X-Winkel
define Wy = $2e as byte   'Ergebnis Y-Winkel

rem Bit-Konstanten
define Vx = Vorzeichen,0
define Vy = Vorzeichen,1

rem Pindefinitionen
define xin = ra,2
define yin = ra,3
define summer = rb,7

        xtal 4.00          'xtal frequency

        tris ra,%11111111    'RA7 = Ausgang, Rest Eingang
        tris rb,0           'alles Ausgang
        lcdinit rb,2,16      'init. lcd with each 16 chars on 2 lines

        lcdwrite 1,1,"Lagemessung mit" 'Eröffnungstext anzeigen
        lcdwrite 2,1,"ADXL213 c)jL V1"
        wait 2000           '2 Sekunden anzeigen
        goto start         'Unterprogramme überspringen

***** es folgen die Unterprogramme *****
'MESSEN liest das Tastverhältnis für X- und Y-Achse ein
messen:
        let Tx1=0          'Startwerte
        let Ty1=0
        let Tx2=0
        let Ty2=0

messen1:
        if xin = 1 then goto messen1 'steigende Flanke für X suchen
messen2:
        if xin = 0 then goto messen2

messen3:
        inc Tx1            'Länge des H-Impulses
        if xin = 1 then goto messen3
messen4:
        inc Tx2            'Länge des L-Impulses
        if xin = 0 then goto messen4
messen5:
        if yin = 1 then goto messen5 'steigende Flanke für Y suchen
messen6:
        if yin = 0 then goto messen6
messen7:
        inc Ty1            'Länge des H-Impulses
        if yin = 1 then goto messen7
messen8:
        inc Ty2            'Länge des L-Impulses
        if yin = 0 then goto messen8
        let Tx2 = Tx2 + Tx1    'X2 = gesamte Periodendauer
        let Tx2 = Tx2 / 5      'gerundet durch 10 Teilen
        let Ty2 = Ty2 + Ty1    'y2 = gesamte Periodendauer
        let Ty2 = Ty2 / 5
        let Ty2 = Ty2 / 10
        return

***** hier beginnt das Hauptprogramm *****

start:
        ledwrite 1,1,"Lage X  Lage Y "
        lcdwrite 2,1," messen  "

loop:
        gosub messen
        let Tx1 = Tx1 * 50      '100 wäre zu groß
        let Ty1 = Ty1 * 50
        let Ex = Tx1 / Tx2
        if Ex >= 250 then goto Xpos else goto Xneg
Xpos:
        let Ex = Ex - 250
        res Vx
        goto Loop1
Xneg:
        let Ex = 250 - Ex
        set Vx

Loop1:
        let Ey = Ty1 / Ty2
        if Ey >= 250 then goto Ypos else goto Yneg
Ypos:
        let Ey = Ey - 250
        res Vy
        goto loop2
Yneg:
        let Ey = 250 - Ey
        set Vy

Loop2:
        if Ex > 152 then goto loop    'fehlerhafte Messung
        if Ey > 152 then goto loop    'fehlerhafte Messung
        restore
        let Datptr_ = Datptr_ + Ex
        readdata Wx
        restore
        let Datptr_ = Datptr_ + Ey
        readdata Wy

        lcdwrite 1,1,"EX=",Ex," ",Ey
        lcdwrite 1,10,"Ey="
        lcdwrite 2,1," ",Wx," ",Wy
        if Vx=1 then lcdwrite 2,2,"-"
        if Vy=1 then lcdwrite 2,10,"-"

'Falls einer oder beide Winkel große als 65° soll Summer ertönen
        if Wx > 65 then goto loop3
        if Wy > 65 then goto loop3
        goto loop4
loop3:
        sound summer,800,1000

```

```
goto loop
loop4:
wait 1000
goto loop
```

Tabelle:

```
data 00,00,01,01,02,02,02,03,03,03,04,04,05,05,05,06,06,06,
data 07,07,08,08,08,09,09,09,10,10,11,11,11,12,12,13,13,
```

```
data 14,14,14,15,15,16,16,16,17,17,18,18,19,19,20,20,20,
data 21,21,22,22,22,23,23,24,24,24,25,25,26,26,27,27,27,28,
data 28,29,29,30,30,30,31,31,32,32,33,33,34,34,34,35,35,36,
data 36,37,37,38,38,39,39,40,40,41,41,42,42,43,43,44,44,45,
data 45,46,46,47,47,48,49,49,50,50,51,52,52,53,53,54,55,55,
data 56,57,57,58,59,60,60,61,62,63,63,64,65,66,67,68,69,70,
data 71,73,74,75,77,79,81,83,90,90
```

Programmlisting 3 (Winkelangabe mit einer Nachkommastelle)

```

*****
'
      NEIGUNGW.BAS
      Flash of Inspiration I
'
'(c) by Ing.Büro St.Lehmann, D-77756 Hausach, SL@iL-online.de
*****
rem Demoprogramm für Neigungs-/Bescheinigungssensor ADXL213
und
rem      LC-Anzeige sowie 16-Bit Arithmetik
rem incl. Umrechnung in absolute Winkelangaben (Potenzreihe f. arcsin)
rem
rem      Einrichten der Platine
rem
rem      LCD an PortB -> glatt durchschalten
rem      Zusatzplatine über Flachbandkabel an K2 anschließt
rem      PortA,7 mit Stift LS verbinden (Summer an PortA,7)
rem
rem      Programm in PIC 16F628 laden und starten
rem
rem      -----
rem      demonstrates how to work with accelerometer sensor ADLX213
rem      LCD and buzzer
rem incl. calculation of degree values (power series for arcsin)
rem
rem      the following settings and connections must be done
rem
rem      put switches "LCD an PORTB" 1 .. 6 to on
rem      additional pcb with sensor with flat cable to K2
rem      connect LS to PortA,7 (buzzer onto port RA,7)
rem
rem      load compiled program into "Flash Of Inspiration I"
rem

      $NCALDEF

'Watchdog off, all comparators off, MCLR- and OSC2-Pin are used for io
define device 16f628,wdt_off,irc_osc,cmcfg7,MCLR_INT,OSC2_IO

rem Variablen
define Tx1=$20 as word      'Zählvariable für X-Achse
define Ty1=$22 as word      'Zählvariable für Y-Achse
define Tx2=$24 as word      'Periodendauer X-Achse
define Ty2=$26 as word      'Periodendauer Y-Achse
define Ex=$28 as word       'Ergebnis X-Achse
define Ey=$2A as word       'Ergebnis Y-Achse
define Vorzeichen=$2c as byte 'Vorzeichen
define temp = $2d as byte   'Hilfvariable
define Grad = $2e as byte   'Zeichencode für °
define Wx = $30 as word     'Ergebnis X-Winkel
define Wy = $32 as word     'Ergebnis Y-Winkel

define Datptr = Datptr_ as word

rem Bit-Konstanten
define Vx = Vorzeichen,0
define Vy = Vorzeichen,1

rem Pindefinitionen
define xin = ra,2
define yin = ra,3
define summer = rb,7

      xtal 4.00          'xtal frequency

      tris ra,%11111111      'RA7 = Ausgang, Rest Eingang
      tris rb,0              'alles Ausgang
      lcdinit rb,2,16        'init. lcd with each 16 chars on 2 lines

      lcdwrite 1,1,"Lagemessung mit" 'Eröffnungstext anzeigen
      lcdwrite 2,1,"ADXL213 e)iL V1"
      wait 2000              '2 Sekunden anzeigen
      goto start            'Unterprogramme überspringen

***** es folgen die Unterprogramme *****
'MESSEN liest das Tastverhältnis für X- und Y-Achse ein
messen:
      let Tx1=0              'Startwerte
      let Ty1=0
      let Tx2=0
      let Ty2=0

messen1:
      if xin = 1 then goto messen1 'steigende Flanke für X suchen
messen2:
      if xin = 0 then goto messen2
messen3:
      inc Tx1                'Länge des H-Impulses
      if xin = 1 then goto messen3
messen4:
      inc Tx2                'Länge des L-Impulses
      if xin = 0 then goto messen4
messen5:
      if yin = 1 then goto messen5 'steigende Flanke für Y suchen
messen6:
      if yin = 0 then goto messen6
messen7:
      inc Ty1                'Länge des H-Impulses
      if yin = 1 then goto messen7
messen8:
      inc Ty2                'Länge des L-Impulses
      if yin = 0 then goto messen8
      let Tx2 = Tx2 + Tx1      'X2 = gesamte Periodendauer
      let Tx2 = Tx2 + 5        'gerundet durch 10 Teilen
      let Tx2 = Tx2 / 10
      let Ty2 = Ty2 + Ty1      'y2 = gesamte Periodendauer
      let Ty2 = Ty2 + 5
      let Ty2 = Ty2 / 10
      return

***** hier beginnt das Hauptprogramm *****

start:
      lcdwrite 1,1," Lage X  Lage Y "
      lcdwrite 2,1," messen      "

loop:
      gosub messen
      let Tx1 = Tx1 * 50        '100 wäre zu groß
      let Ty1 = Ty1 * 50
      let Ex = Tx1 / Tx2
      if Ex >= 250 then goto Xpos else goto Xneg

Xpos:
      let Ex = Ex - 250
      res Vx
      goto Loop1

Xneg:
      let Ex = 250 - Ex
      set Vx

Loop1:
      let Ey = Ty1 / Ty2
      if Ey >= 250 then goto Ypos else goto Yneg

Ypos:
      let Ey = Ey - 250
      res Vy
      goto loop2

Yneg:
      let Ey = 250 - Ey
      set Vy

Loop2:
      if Ex > 152 then goto loop      'fehlerhafte Messung
      if Ey > 152 then goto loop      'fehlerhafte Messung
      restore
      let Datptr = Datptr + Ex + Ex  'Index 2 x, wegen 16-Bit Tabelle
      readdata temp
      let Wx = temp * 100
      readdata temp
      let Wx = Wx + temp
      restore
      let Datptr = Datptr +Ey + Ey
      readdata temp

```

```

let Wy = temp * 100
readdata temp
let Wy = Wy + temp

let grad=223
lcdwrite 2,1,Wx,:1,grad,#," ",Wy,:1,grad,#
if Vx=1 then lcdwrite 2,2,"-"
if Vy=1 then lcdwrite 2,11,"-"
'Falls einer oder beide Winkel grÖÙe als 65° soll Summer ertÖnen
if Wx > 659 then goto loop3
if wy > 659 then goto loop3
goto loop4
loop3:
sound summer,800,1000
goto loop
loop4:
wait 1000
goto loop

Tabelle:

```

```

data 00,00,00,04,00,08,00,11,00,15,00,19,00,23,00,26,00,30,
data 00,34,00,38,00,42,00,45,00,49,00,53,00,57,00,60,00,64,
data 00,68,00,72,00,76,00,79,00,83,00,87,00,91,00,95,00,98,
data 01,02,01,06,01,10,01,14,01,18,01,22,01,25,01,29,01,33,
data 01,37,01,41,01,45,01,49,01,53,01,56,01,60,01,64,01,68,
data 01,72,01,76,01,80,01,84,01,88,01,92,01,96,02,00,02,04,
data 02,08,02,12,02,16,02,20,02,24,02,28,02,32,02,37,02,41,
data 02,45,02,49,02,53,02,57,02,62,02,66,02,70,02,74,02,78,
data 02,83,02,87,02,91,02,96,03,00,03,04,03,09,03,13,03,18,
data 03,22,03,26,03,31,03,35,03,40,03,45,03,49,03,54,03,58,
data 03,63,03,68,03,72,03,77,03,82,03,87,03,92,03,97,04,01,
data 04,06,04,11,04,16,04,21,04,27,04,32,04,37,04,42,04,47,
data 04,53,04,58,04,64,04,69,04,75,04,80,04,86,04,92,04,97,
data 05,03,05,09,05,15,05,21,05,28,05,34,05,40,05,47,05,53,
data 05,60,05,67,05,74,05,81,05,88,05,95,06,03,06,10,06,18,
data 06,26,06,35,06,43,06,52,06,61,06,71,06,81,06,91,07,02,
data 07,13,07,25,07,38,07,53,07,68,07,86,08,07,08,34,09,00,
data 09,00

```

Winkelberechnung (ohne Nachkommastellen)

Beispiel:

Messwert: 107
 Max. Wert 152
 => sinus (ger. 0,704
 => Winkel 44,745

Messwert:	Sinus (gerundet.)	Winkel	Messwert	Sinus (gerundet)	Winkel	Messwert	Sinus (gerundet)	Winkel	Messwert	Sinus (gerundet)	Winkel
1	0,0066	0	51	0,3355	20	101	0,6645	42	151	0,9934	83
2	0,0132	1	52	0,3421	20	102	0,6711	42	152	1,0000	90
3	0,0197	1	53	0,3487	20	103	0,6776	43			
4	0,0263	2	54	0,3553	21	104	0,6842	43			
5	0,0329	2	55	0,3618	21	105	0,6908	44			
6	0,0395	2	56	0,3684	22	106	0,6974	44			
7	0,0461	3	57	0,3750	22	107	0,7039	45			
8	0,0526	3	58	0,3816	22	108	0,7105	45			
9	0,0592	3	59	0,3882	23	109	0,7171	46			
10	0,0658	4	60	0,3947	23	110	0,7237	46			
11	0,0724	4	61	0,4013	24	111	0,7303	47			
12	0,0789	5	62	0,4079	24	112	0,7368	47			
13	0,0855	5	63	0,4145	24	113	0,7434	48			
14	0,0921	5	64	0,4211	25	114	0,7500	49			
15	0,0987	6	65	0,4276	25	115	0,7566	49			
16	0,1053	6	66	0,4342	26	116	0,7632	50			
17	0,1118	6	67	0,4408	26	117	0,7697	50			
18	0,1184	7	68	0,4474	27	118	0,7763	51			
19	0,1250	7	69	0,4539	27	119	0,7829	52			
20	0,1316	8	70	0,4605	27	120	0,7895	52			
21	0,1382	8	71	0,4671	28	121	0,7961	53			
22	0,1447	8	72	0,4737	28	122	0,8026	53			
23	0,1513	9	73	0,4803	29	123	0,8092	54			
24	0,1579	9	74	0,4868	29	124	0,8158	55			
25	0,1645	9	75	0,4934	30	125	0,8224	55			
26	0,1711	10	76	0,5000	30	126	0,8289	56			
27	0,1776	10	77	0,5066	30	127	0,8355	57			
28	0,1842	11	78	0,5132	31	128	0,8421	57			
29	0,1908	11	79	0,5197	31	129	0,8487	58			
30	0,1974	11	80	0,5263	32	130	0,8553	59			
31	0,2039	12	81	0,5329	32	131	0,8618	60			
32	0,2105	12	82	0,5395	33	132	0,8684	60			
33	0,2171	13	83	0,5461	33	133	0,8750	61			
34	0,2237	13	84	0,5526	34	134	0,8816	62			
35	0,2303	13	85	0,5592	34	135	0,8882	63			
36	0,2368	14	86	0,5658	34	136	0,8947	63			
37	0,2434	14	87	0,5724	35	137	0,9013	64			
38	0,2500	14	88	0,5789	35	138	0,9079	65			
39	0,2566	15	89	0,5855	36	139	0,9145	66			
40	0,2632	15	90	0,5921	36	140	0,9211	67			
41	0,2697	16	91	0,5987	37	141	0,9276	68			
42	0,2763	16	92	0,6053	37	142	0,9342	69			
43	0,2829	16	93	0,6118	38	143	0,9408	70			
44	0,2895	17	94	0,6184	38	144	0,9474	71			
45	0,2961	17	95	0,6250	39	145	0,9539	73			
46	0,3026	18	96	0,6316	39	146	0,9605	74			
47	0,3092	18	97	0,6382	40	147	0,9671	75			
48	0,3158	18	98	0,6447	40	148	0,9737	77			
49	0,3224	19	99	0,6513	41	149	0,9803	79			
50	0,3289	19	100	0,6579	41	150	0,9868	81			

Winkelberechnung (mit einer Nachkommastelle)

Beispiel:

Messwert: 107
Max.Wert 152

=> sinus (ge 0,704
=> Winkel 44,745

Messwert:	Sinus (gerundet.)	Winkel	Messwert	Sinus (gerundet)	Winkel	Messwert	Sinus (gerundet)	Winkel	Messwert	Sinus (gerundet)	Winkel
1	0,0066	0,4	51	0,3355	19,6	101	0,6645	41,6	151	0,9934	83,4
2	0,0132	0,8	52	0,3421	20,0	102	0,6711	42,1	152	1,0000	90,0
3	0,0197	1,1	53	0,3487	20,4	103	0,6776	42,7			
4	0,0263	1,5	54	0,3553	20,8	104	0,6842	43,2			
5	0,0329	1,9	55	0,3618	21,2	105	0,6908	43,7			
6	0,0395	2,3	56	0,3684	21,6	106	0,6974	44,2			
7	0,0461	2,6	57	0,3750	22,0	107	0,7039	44,7			
8	0,0526	3,0	58	0,3816	22,4	108	0,7105	45,3			
9	0,0592	3,4	59	0,3882	22,8	109	0,7171	45,8			
10	0,0658	3,8	60	0,3947	23,2	110	0,7237	46,4			
11	0,0724	4,2	61	0,4013	23,7	111	0,7303	46,9			
12	0,0789	4,5	62	0,4079	24,1	112	0,7368	47,5			
13	0,0855	4,9	63	0,4145	24,5	113	0,7434	48,0			
14	0,0921	5,3	64	0,4211	24,9	114	0,7500	48,6			
15	0,0987	5,7	65	0,4276	25,3	115	0,7566	49,2			
16	0,1053	6,0	66	0,4342	25,7	116	0,7632	49,7			
17	0,1118	6,4	67	0,4408	26,2	117	0,7697	50,3			
18	0,1184	6,8	68	0,4474	26,6	118	0,7763	50,9			
19	0,1250	7,2	69	0,4539	27,0	119	0,7829	51,5			
20	0,1316	7,6	70	0,4605	27,4	120	0,7895	52,1			
21	0,1382	7,9	71	0,4671	27,8	121	0,7961	52,8			
22	0,1447	8,3	72	0,4737	28,3	122	0,8026	53,4			
23	0,1513	8,7	73	0,4803	28,7	123	0,8092	54,0			
24	0,1579	9,1	74	0,4868	29,1	124	0,8158	54,7			
25	0,1645	9,5	75	0,4934	29,6	125	0,8224	55,3			
26	0,1711	9,8	76	0,5000	30,0	126	0,8289	56,0			
27	0,1776	10,2	77	0,5066	30,4	127	0,8355	56,7			
28	0,1842	10,6	78	0,5132	30,9	128	0,8421	57,4			
29	0,1908	11,0	79	0,5197	31,3	129	0,8487	58,1			
30	0,1974	11,4	80	0,5263	31,8	130	0,8553	58,8			
31	0,2039	11,8	81	0,5329	32,2	131	0,8618	59,5			
32	0,2105	12,2	82	0,5395	32,6	132	0,8684	60,3			
33	0,2171	12,5	83	0,5461	33,1	133	0,8750	61,0			
34	0,2237	12,9	84	0,5526	33,5	134	0,8816	61,8			
35	0,2303	13,3	85	0,5592	34,0	135	0,8882	62,6			
36	0,2368	13,7	86	0,5658	34,5	136	0,8947	63,5			
37	0,2434	14,1	87	0,5724	34,9	137	0,9013	64,3			
38	0,2500	14,5	88	0,5789	35,4	138	0,9079	65,2			
39	0,2566	14,9	89	0,5855	35,8	139	0,9145	66,1			
40	0,2632	15,3	90	0,5921	36,3	140	0,9211	67,1			
41	0,2697	15,6	91	0,5987	36,8	141	0,9276	68,1			
42	0,2763	16,0	92	0,6053	37,2	142	0,9342	69,1			
43	0,2829	16,4	93	0,6118	37,7	143	0,9408	70,2			
44	0,2895	16,8	94	0,6184	38,2	144	0,9474	71,3			
45	0,2961	17,2	95	0,6250	38,7	145	0,9539	72,5			
46	0,3026	17,6	96	0,6316	39,2	146	0,9605	73,8			
47	0,3092	18,0	97	0,6382	39,7	147	0,9671	75,3			
48	0,3158	18,4	98	0,6447	40,1	148	0,9737	76,8			
49	0,3224	18,8	99	0,6513	40,6	149	0,9803	78,6			