

Praktikum Mikrocontrollertechnik SS 2015

Versuch 3

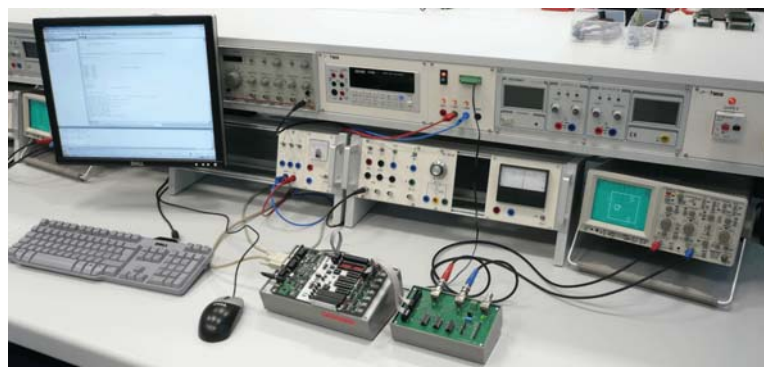
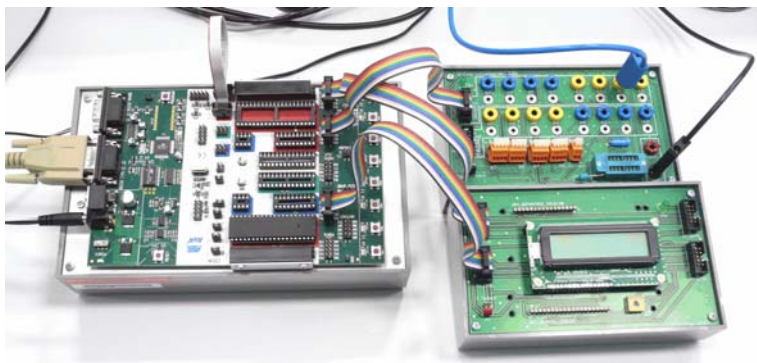
Stand: 30. 6. 2015

Elementare Anwendungs- und Schnittstellenprogrammierung in C. Nutzung von Zähler- und Zeitgebereinheiten (Timer-Counter Units).

1. Frequenzmessung.
2. Periodendauermessung.
3. Pulsweitenmodulation.
4. XY-Darstellung.

Versuchsanordnung:

Starterkit Atmel STK 500 mit Atmel ATmega 16. Peripherie: PC mit Terminalprogramm, LCD-Anzeige 09 mit Punktmatrix-LCD 2 • 16, Universaladapter 10b, XY-Adapter 09a, Lüftertrainer 14a.



Zum Versuchsablauf:

Mikrocontrollerschaltkreise haben eine jeweils bestimmte, fest vorgegebene Belegung ihrer Anschlüsse (Pins) mit Signalen von Schnittstellensteuerungen und peripheren Einrichtungen. Wird eine eingebaute Einrichtung genutzt, so sind die zugehörigen Pins nicht mehr für die universelle Ein- und Ausgabe verfügbar. Somit können externe Einrichtungen nur an jene Ports angeschlossen werden, die dafür noch frei sind. Deshalb muß die Versuchsanordnung von Zeit zu Zeit umgebaut werden. Um nicht unnötig Zeit mit dem Umschreiben der E-A-Funktionen zu verlieren, werden mehrere Musterprogramme (templates) bereitgehalten. Nutzung: den Quelltext des Projektprogramms im Eingabefenster löschen und das jeweils neue Musterprogramm hineinkopieren.

Praxistip: Alle Festwerte (Texte, Ports, Bits usw.) mit eigenen Symbolen bezeichnen. Die jeweils konkreten Werte z. B. über **def**-Anweisungen (C) oder **equ**-Anweisungen (Assembler) zuordnen. Das habe wir hier nur deshalb unterlassen, um schnell zu ersten Erfolgserlebnissen zu kommen.

Kurzausbildung Zähler und Zeitgeber (Counter/Timer)

Zähler (Counter):

- Zählimpulse kommen von außen.
- Vorwärtszählen von geladenem Wert an.
- Nulldurchgang (Wrap Around) wird registriert. Kann abgefragt werden oder Interrupt auslösen.

Zeitgeber (Timer):

- Zählt mit internem Takt.
- Takt kann über Vorteiler (Prescaler) geführt werden. Typische Teilverhältnisse sind Zweierpotenzen.
- Vorwärtszählen von geladenem Wert an.
- Nulldurchgang (Wrap Around) wird registriert. Kann abgefragt werden oder Interrupt auslösen.

Zeiterfassungsfunktion (Capture)

Externes Signal bewirkt Übernahme des aktuellen Zählerstandes in ein programmseitig abfragbares Haltereister. Das Auftreten des externen Signals wird registriert. Kann abgefragt werden oder Interrupt auslösen.

Vergleichsfunktion (Compare)

Zählerstand wird mit dem Inhalt eines programmseitig ladbaren Vergleichsregisters verglichen. Bei Gleichheit wird eine entsprechende Bedingung gesetzt. Kann abgefragt werden oder Interrupt auslösen. Ggf. weitere Wirkungen (programmseitig einstellbar):

- bei Gleichheit Zähler löschen (so daß er von Null an weiterzählt) oder durchlaufen lassen.
- Ausgangssignal setzen / löschen /umschalten.

Interrupts einer Zähler-Zeitgeber-Einheit:

- Nulldurchgang (Überlauf),
- Capture,
- Compare.

PWM (Pulsweitenmodulation):

- fortlaufendes Zählen,
- Zählperiode fest, Duty Cycle veränderlich (Vergleichsregister).

PWM-Betriebsarten:

Sind von Bedeutung, wenn mehrere synchrone PWM-Signale zu erzeugen sind.

1. Synchronisation durch Rücksetzen. Zähler zählt aufwärts und wird am Ende der Periode wieder gelöscht. Alle PWM-Impulse starten bzw. enden (je nach Polarität) zur gleichen Zeit
2. Komplementäre Zählweise. Zähler zählt aufwärts bis zur Hälfte der Zählweite und dann wieder abwärts bis Null. Der Zähler durchläuft den jeweiligen Vergleichswert zweimal. Der eine Durchlauf bewirkt das Einschalten des PWM-Signals, der andere das Ausschalten.

Beispiel (ATmel ATmega 16): drei Zähler-Zeitgeber-Einheiten Timer/Counter TC0, TC1, TC2.

- Timer/Counter 0. 8 Bits mit Vergleich und PWM.
- Timer/Counter 1. 16 Bits mit Zählwerterfassung (Capture), Vergleich und PWM.
- Timer/Counter 2. 8 Bits mit Vergleich und PWM. Zusätzlicher Taktgenerator wählbar (vorzugsweise für Uhrenquarz 32,763 kHz).

(Bildquellen: Atmel)

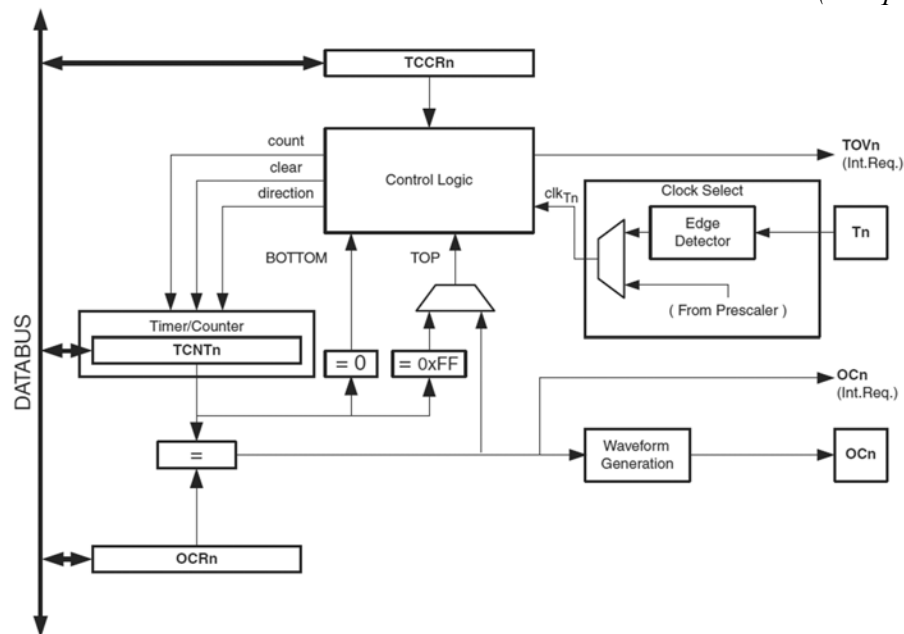


Abb. 1 Timer/Counter TC0 (8 Bits).

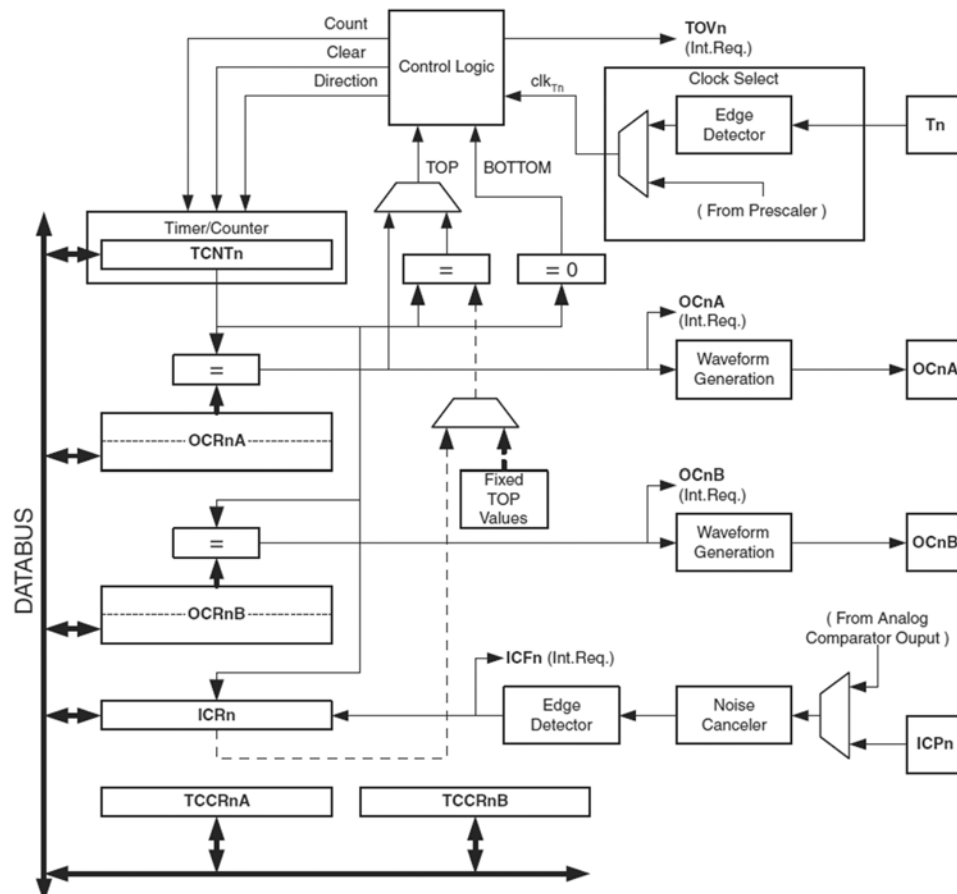


Abb. 2 Timer/Counter TC1 (16 Bits).

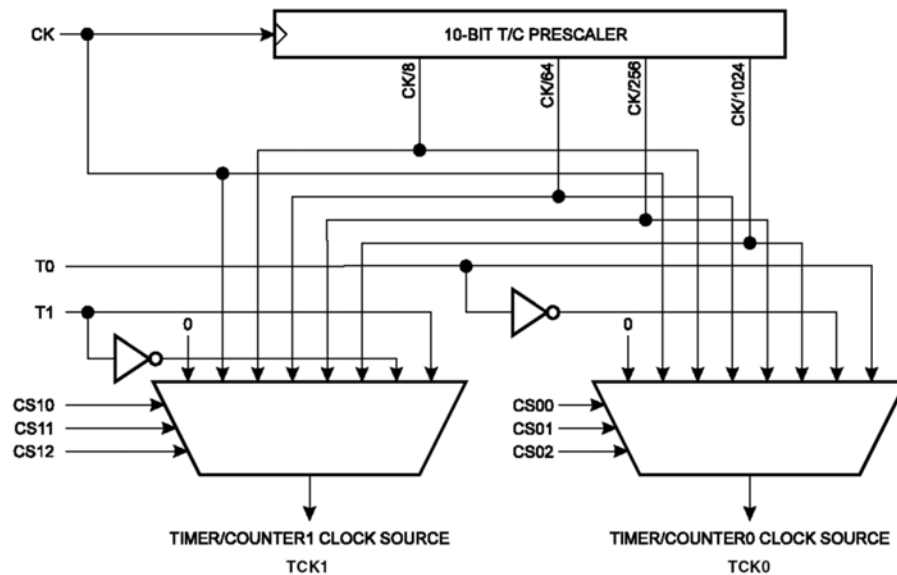


Abb. 3 Vorteiler (Prescaler) für Timer/Counter 0 und 1.

Programmseitige Steuerung:

- Zeitgeberinterrupts zulassen oder verhindern: Timer/Counter Interrupt Mask Register TIMSK.
- Abfrage von Interruptanforderungen der Zeitgeber: Timer/Counter Interrupt Flag Register TIFR.
- Betriebsarteneinstellung: Timer/Counter Control Register TCCR0, TCCR1A, TCCR1B, TCCR2.
- Zählwert einstellen und abfragen.
- Vergleichswert einstellen.
- Erfassten Zählwert abfragen.

Timer/Counter Interrupt Mask Register TIMSK

7	6	5	4	3	2	1	0
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0

- OCIE_x: Interruptauslösung bei Zählwertvergleich (Zeitgeber 1 und 2).
- TICIE1: Interruptauslösung bei Zählwerterfassung (Capture; nur Zeitgeber 1).
- TOIE_x: Interruptauslösung bei Zählerüberlauf (vom Endwert nach Null). Betrifft alle Zeitgeber.

Timer/Counter Interrupt Flag Register TIFR

7	6	5	4	3	2	1	0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

- Setzen der Flagbits: Hardware (bei Auftreten der jeweiligen Bedingung).
- Löschen der Flagbits:
 - bei Einleitung der Interruptbehandlung (Hardware),
 - durch Schreiben von Einsen in die jeweiligen Bitpositionen (Software).
- OCF_x: Es ist eine Vergleichsbedingung aufgetreten (Zählwert = Vergleichswert). Betrifft die Zeitgeber 1 und 2.
- ICF1: ein Zählwert wurde erfaßt (= ins Haltereister übernommen). Betrifft nur Zeitgeber 1 (Capture).
- TOV_x: ein Zählerüberlauf (vom Endwert nach Null) ist aufgetreten. Betrifft alle Zeitgeber.

Timer/Counter Control Register TCCR0, TCCR1A, TCCR1B, TCCR2

Register	7	6	5	4	3	2	1	0
TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20

- CSx2...CSx0: Auswahl des Zähltaktes (über Vorteiler (Prescaler)).
- COMx1, COMx0: Betriebsart des Vergleichers.
- WGM: Zählerbetriebsart (Impulserzeugung).
- ICNC1: Störunterdrückung für Erfassungsfunktion (Capture). 0 = ausgeschaltet, 1 = eingeschaltet. Nur Zeitgeber 1.
- ICES1: Flankenauswahl für Erfassungsfunktion (Capture). 0 = fallende Flanke (High-Low), 1 = steigende Flanke (Low-High). Nur Zeitgeber 1.

Aufgabe 1: Frequenzmessung.

Das Impulssignal, dessen Frequenz zu messen ist, wird als Takt auf den Zähler gegeben. Der Zähler wird anfänglich gelöscht. Dann wird eine bestimmte Zeit lang gezählt. Ist diese Meßzeit abgelaufen, wird der Zählwert ausgegeben. Bei einer Meßzeit von 1 s entspricht der Zählwert direkt der Frequenz in Hz.

Versuchsanordnung:

- Wir verwenden die Zähler-Zeitbereinheit TC1 (16 Bits). Meßsignal T1 = Port B1.
- Meßsignalquelle: Funktionsgenerator, Synchronsignalausgang. Anschluß über Meßadapter 09a und Universaladapter 10a an Port B.
- Ergebnisanzeige: LCD-Anzeige 10. Steuersignale Port D, Datenbus Port C.

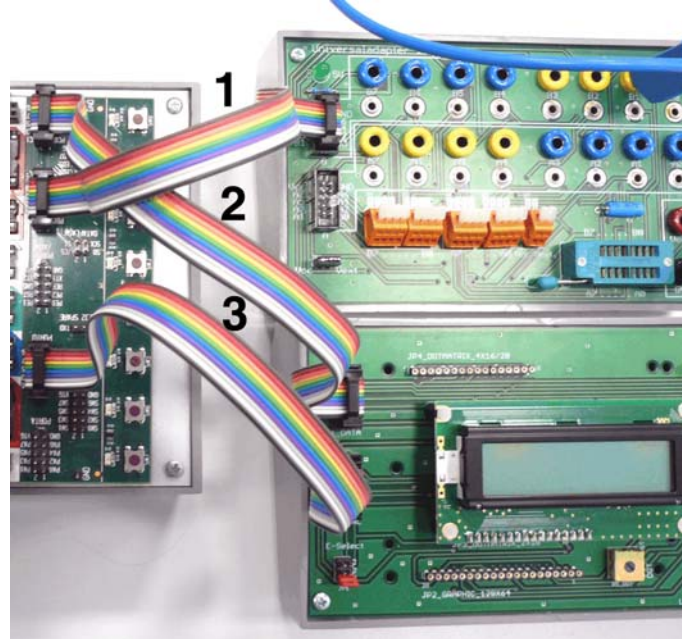


Abb. 4 Frequenzmessung. 1 - Universaladapter Port B an Port B. 2 - LCD-Datenbus an Port C. 3 - LCD-Steuerung an Port D.

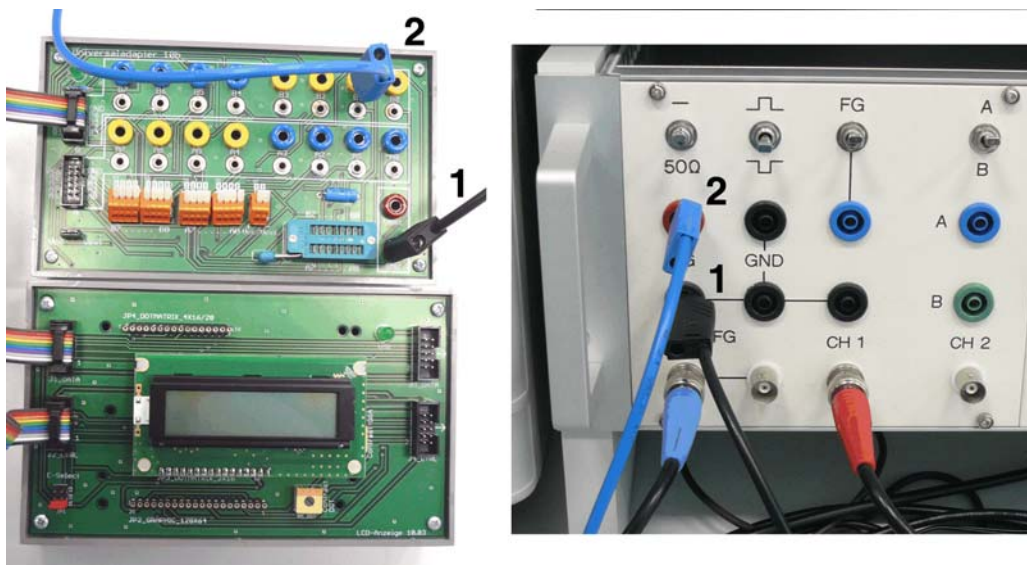


Abb. 5 Meßsignaleinspeisung vom Funktionsgenerator. 1 - Masse; 2 - Meßsignal an Universaladapter Port B1.

C-Programmierung:

- Das Projekt des ersten Versuchs aufrufen oder eine neues C-Projekt einrichten (s. die Anleitung zum ersten Versuch).
- Das Programm **V3_15_template_LCD_D.c** in das Editorfenster des Projekts kopieren.
- Die Datei **V3_15_examples.c** im Editor öffnen, um ggf. Programmbeispiele übernehmen zu können.
- Optimierung ausschalten ("O0").

Das Programm übersetzen und laufen lassen. Auf dem LCD muß ein Zeichen 'A' erscheinen (Funktionskontrolle).

Frequenzmessung mit 1 s Meßzeit.

Die Meßzeit wird mit der Millisekunden-Zählfunktion gewonnen. Das ist eine Einfachlösung mit sehr beschränkter Genauigkeit. Das Ergebnis wird mit 5 Dezimalstellen angezeigt. Ein 16-Bit-Zähler kann bis 65 535 zählen. Die höchste zu messende Frequenz liegt somit knapp über 64 kHz.

Programmbeispiel: 01.

- Zusätzlich ausprobieren: Meßzeit = 100 ms.
- Selbst (ohne Vorlage) zu programmieren: Führende Nullen unterdrücken.

Aufgabe 2: Periodendauermessung.

Der Zähler läuft mit einem internen Takt (Zeitgeberfunktion). Die Zeit zwischen zwei gleichartigen Flanken des Meßsignals wird ausgezählt. Läuft der Zähler mit 1 MHz und werden 533 Takte gezählt, so beträgt die Periodendauer 533 μ s.

Versuchsanordnung:

- Wir verwenden die Input-Capture-Funktion (ICP) der Zähler-Zeitgebereinheit TC1 (16 Bits). Meßsignal ICP1 = Port D6.
- Meßsignalquelle: Funktionsgenerator, Synchronsignalausgang. Anschluß über Meßadapter 09a und Universaladapter 10a an Port D.
- Ergebnisanzeige: LCD-Anzeige 10. Steuersignale Port B, Datenbus Port C.

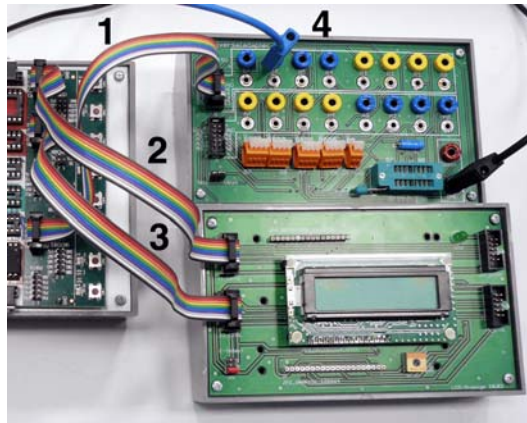


Abb. 6 Periodendauermessung: 1 - Universaladapter Port B an Port D. 2 - LCD-Datenbus an Port C. 3 - LCD-Steuerung an Port B; 4 - an Universaladapter Port B6.

C-Programmierung:

- Das Programm **V3_15_template_LCD_B.c** in das Editorfenster des Projekts kopieren (es ist ein komplett neues Programm).
- Die Datei **V3_15_examples.c** im Editor öffnen, um ggf. Programmbeispiele übernehmen zu können.

Das Programm übersetzen und laufen lassen. Auf dem LCD muß ein Zeichen 'A' erscheinen (Funktionsprobe).

Programmbeispiel: 02.

Zählfrequenz = Taktfrequenz : 8. Bei 4 MHz Takt ergibt sich so eine Zählfrequenz von 500 kHz bzw. eine Periode von 2 μ s. Periodendauer = Anzeigewert \cdot 2 μ s. Ein 16-Bit-Zähler kann bis 65 535 zählen. Die längste zu messende Periodendauer liegt somit knapp über $64\,000 \cdot 2 = 128\,000 \mu\text{s} = 128 \text{ ms}$. Das entspricht einer Frequenz von ca. 7,8 Hz.

Mit verschiedenen Frequenzen (bis hin zu $> 1 \text{ kHz}$) ausprobieren. Was müßte herauskommen, was kommt heraus? Woran könnte es liegen? Siehe auch den Anhang zu den Mysterien der Input-Capture-Funktion.

Selbst (ohne Vorlage) zu programmieren:

1. Die Anzeige direkt in Mikrosekunden.
2. Anstelle der Periodendauer die Frequenz anzeigen (umrechnen – sollte so schwierig nicht sein...).

Probieren Sie es ggf. mit den Programmbeispielen **02X** und **02Y**. Praxistip: Selbst denken schadet wirklich nicht ...

Aufgabe 3: Pulsweitenmodulation (PWM).

Kurzausbildung Pulsweitenmodulation

Die Pulsweitenmodulation dient dazu, kontinuierliche Größen, wie die Helligkeit einer Lichtquelle, die Drehzahl eines Motors oder die Amplitude einer Spannung, auf rein digitale Weise einzustellen, also durch einfaches Ein- und Ausschalten. Hierzu werden Impulsfolgen mit fester Periodendauer, aber veränderlicher Breite abgegeben. Je breiter der Impuls, desto höher die Helligkeit, Drehzahl usw. Gar kein Impuls ergibt den Aus-Zustand, eine ständige Erregung (Dauersignal) ergibt den jeweiligen Maximalwert ("Vollgas"). Die kontinuierliche Wirkung ergibt sich als Folge von Integrationswirkungen. Bei Lichtquellen integriert das Auge des Betrachters (eine entsprechend angesteuerte LED flimmert im Takt der Erregung, wir können das aber nicht wahrnehmen), beim Motor ist es die mechanische Trägheit.

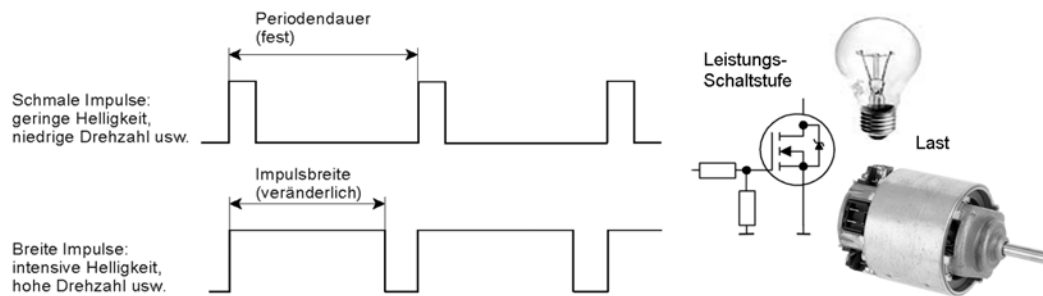


Abb. 7 Zum Prinzip der Pulsweitenmodulation (PWM).

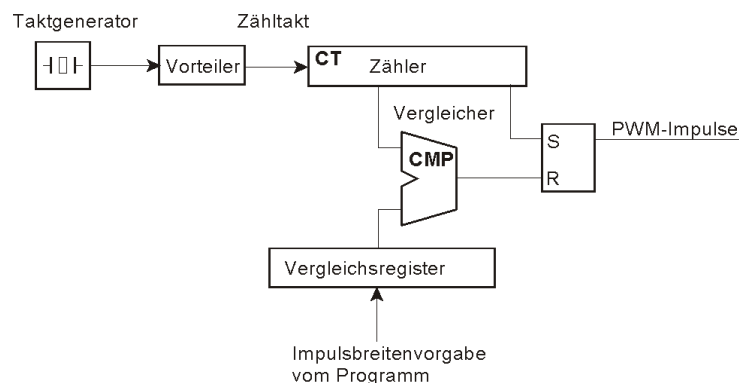


Abb. 8 Das Prinzip der Impulserzeugung. Der Periodenzähler läuft ständig um. Läuft er über das Zählende hinaus wieder in die erste Stellung ein, setzt er das Flipflop. Der PWM-Impuls wird aktiv. Entspricht der Zählerstand dem Wert der vorgegebenen Impulsbreite, so setzt der Vergleicher das Flipflop wieder zurück. Der Impuls ist damit zu Ende.

Versuchsanordnung:

- Wir verwenden die Zähler-Zeitgeber-Einheit (Timer/Counter) TC0. Deren Ausgang: OC0 = Port B3.
- Bedienung: Zunächst über die serielle Schnittstelle.
- Anzeige: Oszilloskop.
- Demonstration: Lüftertrainer 14a an Port B.
- Die LCD-Anzeige abbauen.
- Die serielle Schnittstelle in Betrieb nehmen (Kabelverbindung, Terminalprogramm).
- Zunächst nur das Oszilloskop anschließen und die Schaltung ausprobieren.
- Wenn es auf dem Oszilloskop funktioniert, den Lüftertrainer anschließen und in Betrieb nehmen.

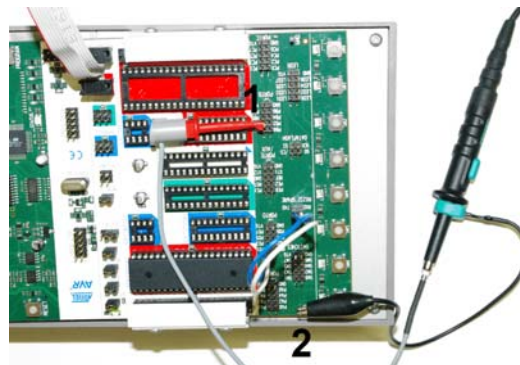


Abb. 9 So wird das Oszilloskop angeschlossen: 1 - der Haken des Tastkopfes an PB3 (Port B). 2 - die Masseklemme an einen Meßpunkt GND.

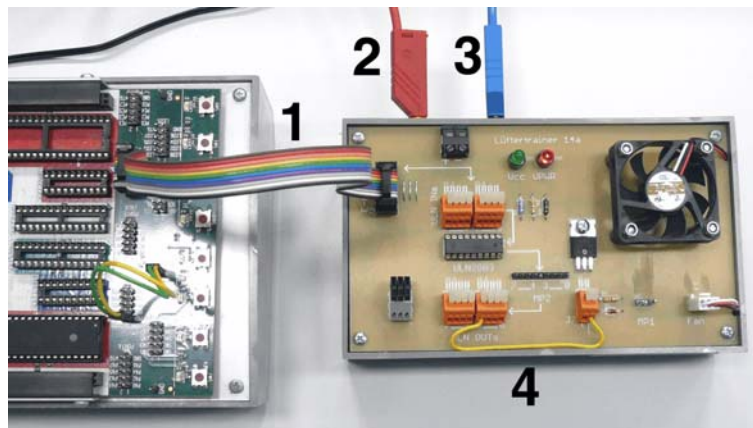


Abb. 10 Der Versuchsaufbau im Überblick. 1 - Port B. 2 - Masse Lüftermotor; 3 - Lüftermotor +12 V (2 und 3 mit Hameg-Netzgerät verbinden). 4 - Lüftermotorerregung über Bitposition 3. Ggf. das Brückenkabel entsprechend umklemmen.



Abb. 11 Ans Hameg-Netzgerät anschließen.

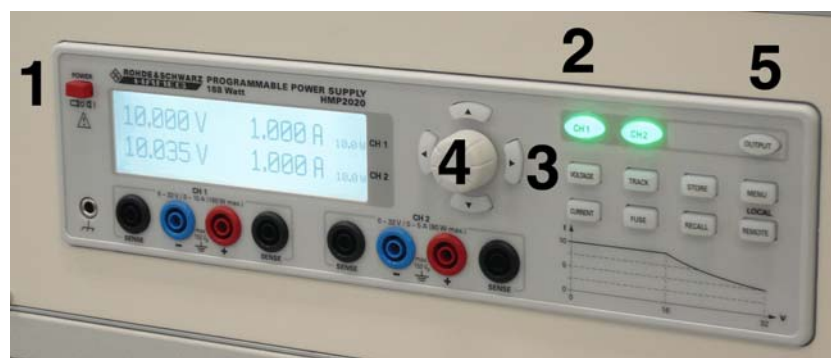


Abb. 12 Spannungseinstellung in folgenden Schritten: 1. Einschalten. 2 - Kanal 1 wählen. 3. Spannungseinstellung (VOLTAGE). 4. Spannungswert eindrehen. 5 - Spannung auf die Buchsen schalten (OUTPUT).

C-Programmierung:

- Das Programm `V3_15_template_serial.c` in das Editorfenster des Projekts kopieren (es ist ein komplett neues Programm).
- Die Datei `V3_15_examples.c` im Editor öffnen, um ggf. Programmbeispiele übernehmen zu können.

Das Programm übersetzen und laufen lassen. Das Terminalprogramm im PC starten (vgl. Versuch 2). Auf dem Terminalbildschirm muß ein Zeichen 'A' erscheinen (Funktionsprobe).

Die Wirkungsweise der Schaltung veranschaulichen wir uns mittels Oszilloskop, die anwendungspraktische Wirkung der Pulsweitenmodulation durch Anschließen des Lüftertrainers 14a.

Die aktuelle Zählweite wird in einem Universalregister gehalten. Wir beginnen mit einem mittleren Wert (128). Um die Zählweite zu verändern, nutzen wir das Terminal, die serielle Schnittstelle und die Interruptserviceroutine.

Programmseitige Steuerung:

- Einstellen der Betriebsart: Timer/Counter Control Register TCCR0.
- Laden des Vergleichswertes: Output Compare Register OCR0.

Timer/Counter Control Register TCCR0

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
0	1	1	0	1	0	1	1

Wir nutzen die Betriebsart "Fast PWM". WGM01 = 1, WGM00 = 1.

Der Lüftertrainer erwartet positive Impulse. Also den Signalpegel am Anfang setzen und bei Zählwertgleichheit (Compare Match) löschen (Non-inverting Mode). COM01 = 1, COM00 = 0. Also TCCR0 = 6BH.

Takterzeugung: Der Zählerumlauf entspricht 256 heruntergeteilten Taktimpulsen. Wir brauchen eine Impulsfolgefrequenz von wenigen hundert Hz. $4 \text{ MHz} : 256 = 15,6 \text{ kHz}$. Eine Takteilung 1:64 ergibt rund 244 Hz. Also wie folgt probieren: CS2...0 = 3H.

Steuerung der Pulsweitenmodulation über die serielle Schnittstelle:.

Wir werten die folgenden Zeichen aus:

- 0 = aus ("0" = 0x30).
- + = beschleunigen ("+" = 0x2b).
- = verringern ("- " = 0x2d).
- V = Vollgas ("V" = 0x56).

Programmbeispiel: 03.



Abb. 13 Die Impusfolge nach der Initialisierung.

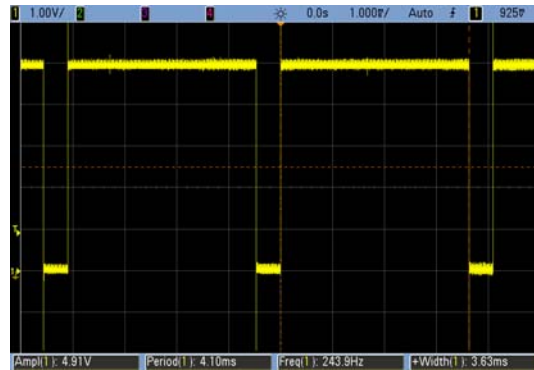
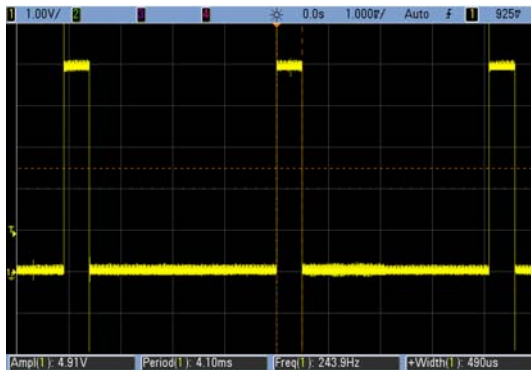


Abb. 14 Typische Impulsfolgen.

Drehzahlsteuerung mittels Joystick (1)

Der Joystick wird an Port A angeschlossen. Wir verwenden zunächst den Koordinatentaster (Hat Switch):

- Nach oben: Drehzahl erhöhen.
- Nach unten: Drehzahl verringern.
- Nach links: aus,
- Nach rechts: Vollgas.

Programmbeispiel: 04.

Drehzahlsteuerung mittels Joystick (2)

Jetzt nutzen wir den analogen Steuerweg der Höhenruderbetätigung (Ziehen – Drücken). Hierzu wird der A-D-Wandler in Betrieb genommen

Programmbeispiel: 05.



Abb. 15 Der Joystick am Universaladapter.

– Port B Universaladapter –							
7	6	5	4	3	2	1	0
COM1	COM2	S4	S3	S2	Plus G	Plus Q	Plus H
Down	Up	Left	Right	S1 (Trigger)	Schleifer Gas	Schleifer Quer	Schleifer Höhe
7	6	5	4	3	2	1	0
– Port A Universaladapter –							

COM1: Potis und S5 bis S7. COM2: Hat Switch und S1 bis S4. Mit Masse verbinden. Plus G, Q, H = das andere Ende des jeweiligen Potentiometers. An die Betriebsspannung anschließen.

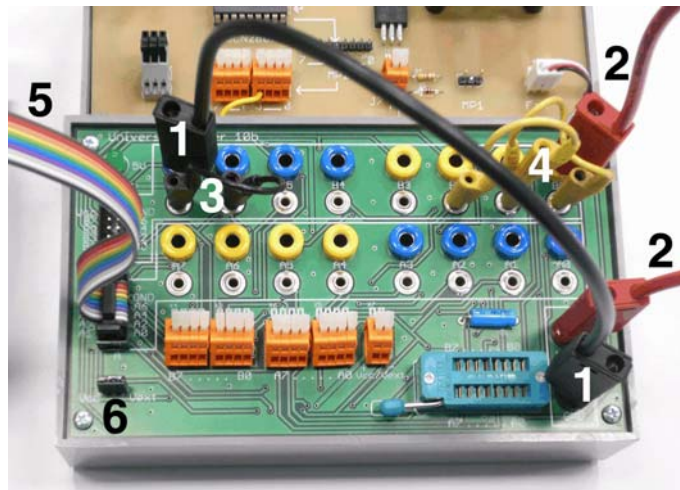


Abb. 16 So wird der Joystick adaptiert. 1 - Masse; 2 - Betriebsspannung (brauchen wir für die Potentiometer); 3 - die Bits 7 und 6 Port B an Masse; 4 - die Bits 2, 1, 0 Port B an Betriebsspannung; 5 - Port A Universaladapter mit Port A des Mikrocontrollers verbinden; 6 - dieser dreipolige Spezialjumper muß gesteckt sein – es ist echte Hochtechnologie...

Aufgabe 4: XY-Darstellung.

Den XY-Adapter 09 an die Ports B und C anschließen und mit dem Oszilloskop verbinden (BNC-Kabel). Oszilloskop auf XY-Betrieb schalten. Der Joystick bleibt angeschlossen.

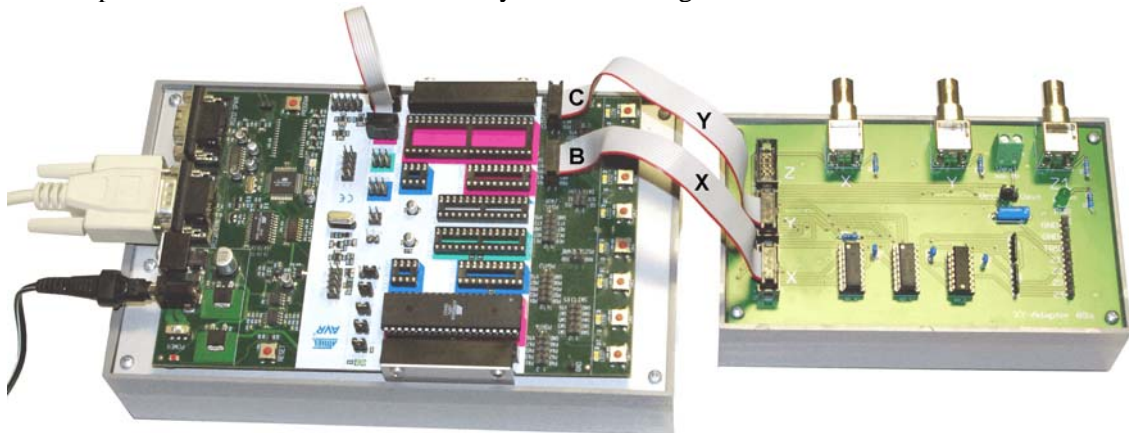


Abb. 17 Starterkit mit XY-Adapter 09. Verbinden: Port C mit Y, Port B mit X.

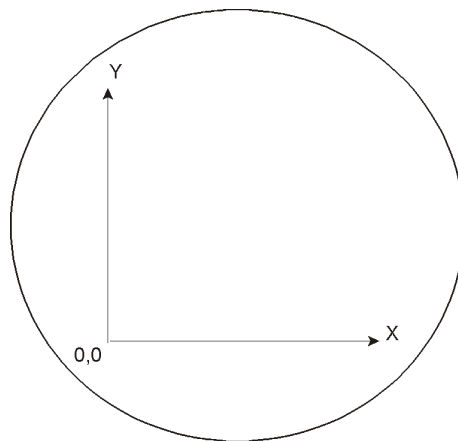


Abb. 18 Das Prinzip der Darstellung auf dem Oszilloskop.

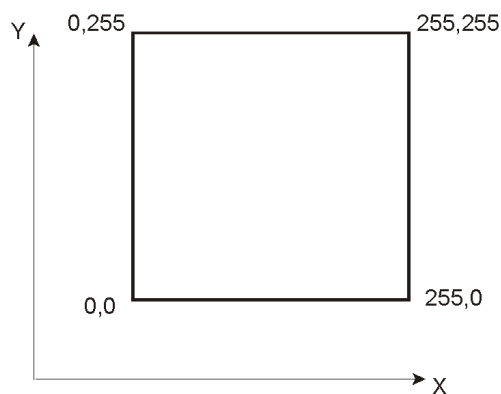


Abb. 19 Die Grenzen des Darstellungsbereichs. Die Darstellung besteht aus Bildpunkten (Pixeln). In jeder Koordinatenachse sind es maximal 256 Pixel.

1. Eine waagrechte Linie maximaler Länge.

Zunächst naiv mit einer FOR-Schleife probieren.

Programmbeispiel: 06.

2. Eine senkrechte Linie maximaler Länge anhängen.

Das klappt offensichtlich nicht. Weshalb?

Programmbeispiel: 07.

3. So umbauen, daß es klappt.

Es ist aber immer noch ein Pixel zuwenig. Wie kann man das korrigieren? (Ob es zuwenig Pixel sind oder nicht, können wir sehen, wenn wir zusätzlich Pixel in die äußersten Ecken schreiben und sie etwas länger stehenlassen..)

Programmbeispiel: 08.

4. Ein Quadrat maximaler Größe.

Die fehlenden Pixel werden zusätzlich eingetragen. Man könnte die Schleifen jedoch auch ganz anders organisieren. Es muß nicht unbedingt FOR sein...

Programmbeispiel: 09.

5. Ein kleineres Quadrat einschreiben.

Die Seitenlänge soll ein Festwert sein (z. B. 20 Pixel). Das Quadrat soll beweglich sein und an beliebiger Stelle dargestellt werden können (Basiskoordinaten).

Programmbeispiel: 10.**6. Das kleinere Quadrat über den Bildschirm bewegen (1).**

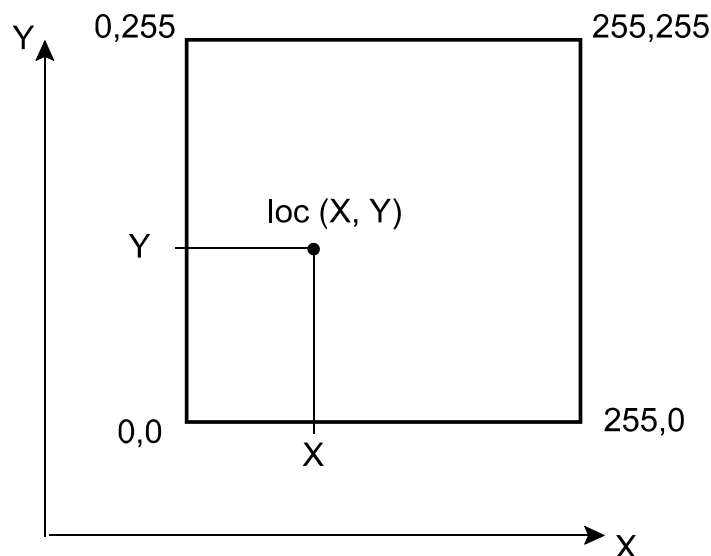
Wir verwenden den Koordinatentaster (Hat Switch). Auf die Bildränder nehmen wir keine Rücksicht.

Programmbeispiel: 11.**7 und 8. Das kleinere Quadrat über den Bildschirm bewegen (2).**

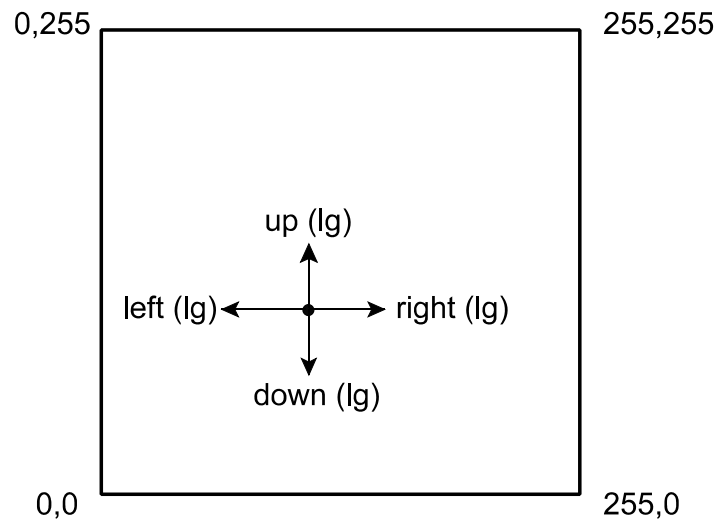
Wir verwenden die analogen Steuerwege der Höhen- und Querruderbetätigungen. Ggf. Wertewandlung so, daß sich ein intuitiv vernünftiger Zusammenhang zwischen den Bewegungen des Hebels und des Quadrates ergibt.

Programmbeispiele: 12 und 13.**9. Symbole mit schrägen Linen (45°) zeichnen.****Programmbeispiel: 11.****Kurzausbildung zum Prinzip:**

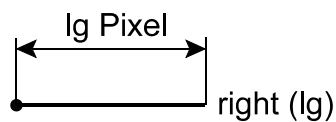
Mit der Funktion `loc (X,Y)` lenken Sie den Elektronenstrahl auf eine bestimmte Position, so daß dort ein Bildpunkt dargestellt wird.



Senkrechte und waagerechte Linien zeichnen Sie mit diesen vier Funktionen:



Der Funktionsparameter lg drückt die Länge der Linie aus. Das lässt sich einfach anhand eines Beispiels erklären:

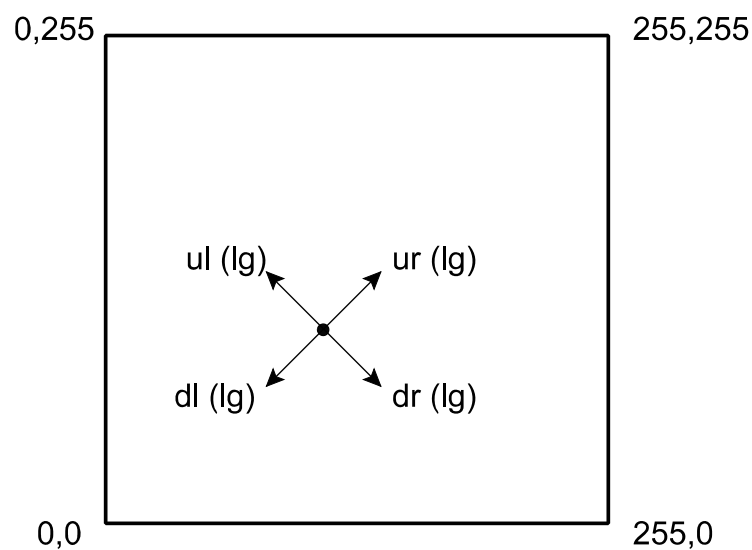


right (0): Länge = Wert der Variablen **len**

right (27): Länge = 27 Pixel

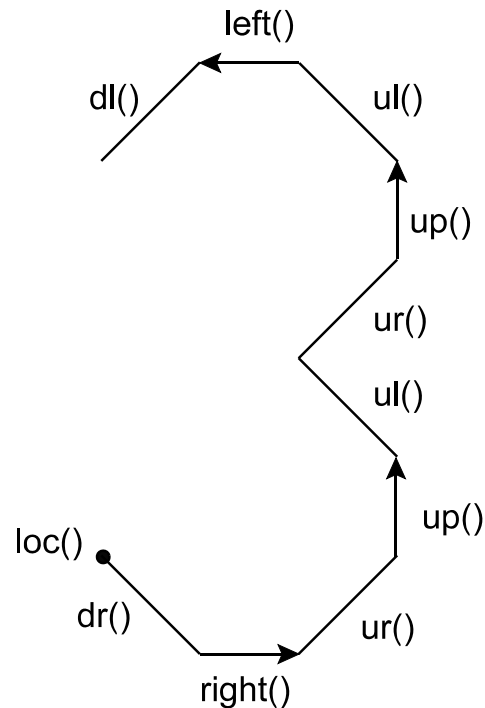
-- Das gilt sinngemäß für alle anderen Funktionen --

Mit diesen vier Funktionen können Sie schräge Linien zeichnen, die um 45° geneigt sind:



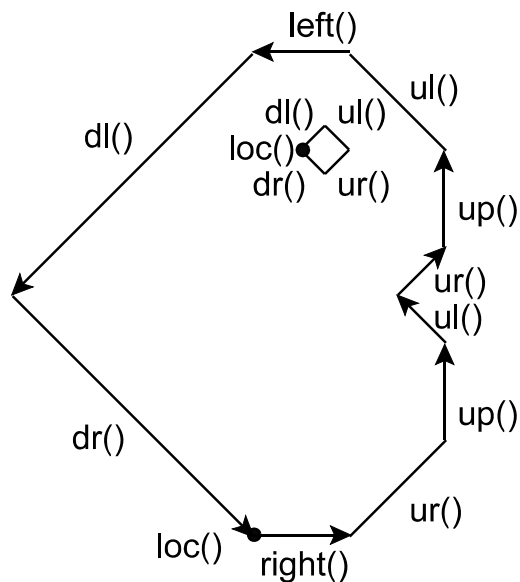
Alle Bezeichnungen sind aus dem Englischen abgeleitet. Schwierig sollte es eigentlich nicht sein...

So wird die Ziffer Drei dargestellt:



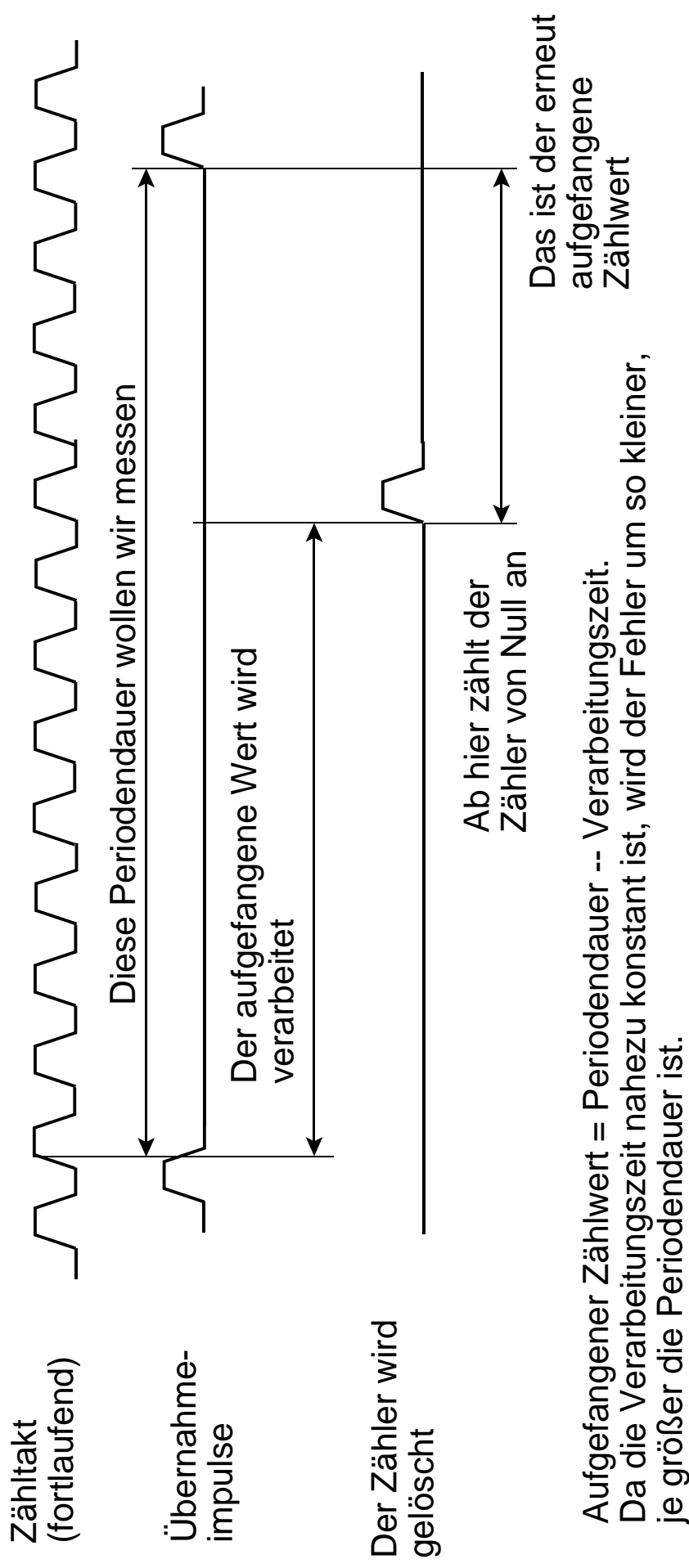
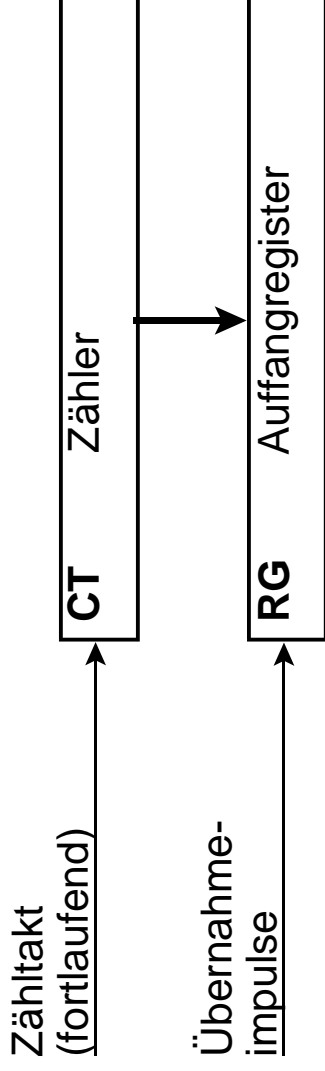
In die Klammern passende Werte einsetzen. Z. B. locate (20,20) und sonst immer eine Null. Dann die Längenzuweisung im fertigen Text z. B. auf len = 10 ändern.

Das (vereinfachte) Pacman-Symbol ist schon etwas anspruchsvoller...



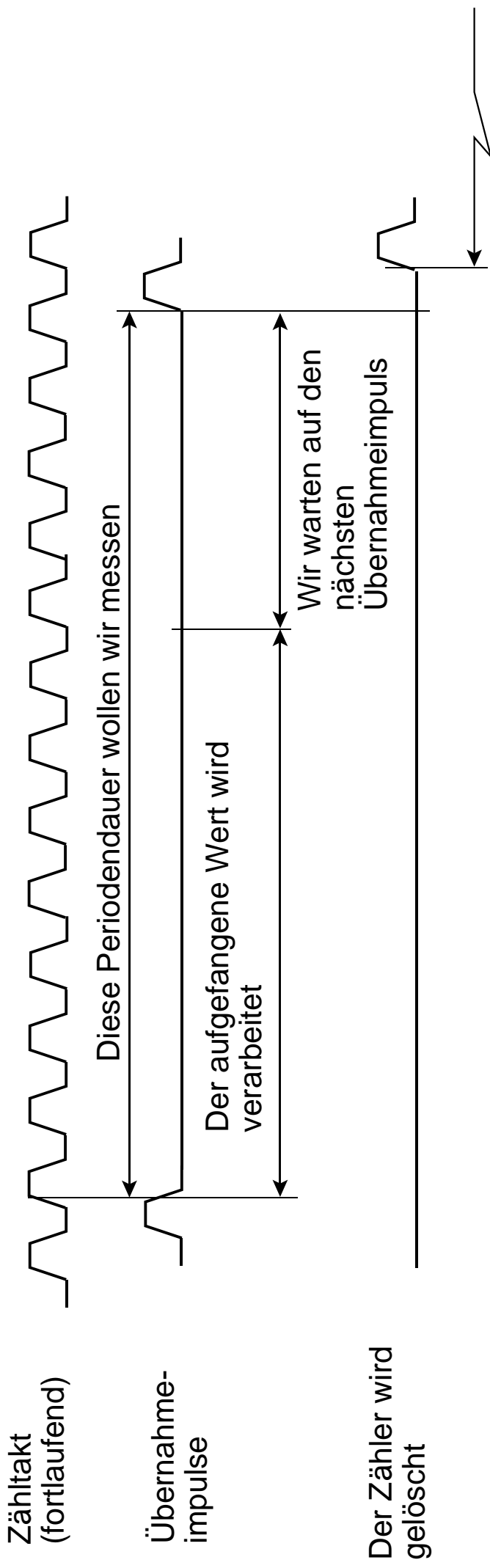
Selbst (ohne Vorlage) zu programmieren:

1. Einige Symbole (z. B. Ziffern) nach Wahl.
2. Das PacMan-Symbol mit dem Joystick bewegen.



Mysterien der Input-Capture-Funktion (1)

Ein Ausweg:



Um auf den nächsten Übernahmeimpuls zu warten, nutzen wir die Fangschaltung der Capture-Funktion erneut aus (Flagbit löschen, dann wieder abfragen).

Jetzt wird vom Übernahmeimpuls an gezählt. O.k.

Grundsätzliche Alternative: Nutzung der Interruptfunktion.

Mysterien der Input-Capture-Funktion (2)

Meßbeispiele

Prüffrequenz: 1 kHz. Periodendauer 1 ms.

Oberer Ablauf: Zyklische Umschaltung bei jedem Durchlauf der Endlosschleife.

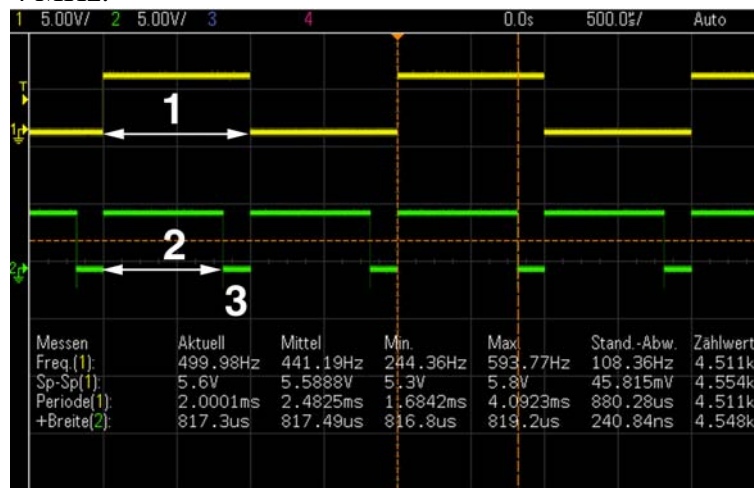
Unterer Ablauf: Kennzeichnet die Dauer der Behandlung (Abfrage, Anzeige usw.).

a) Herkömmlicher Meßablauf.

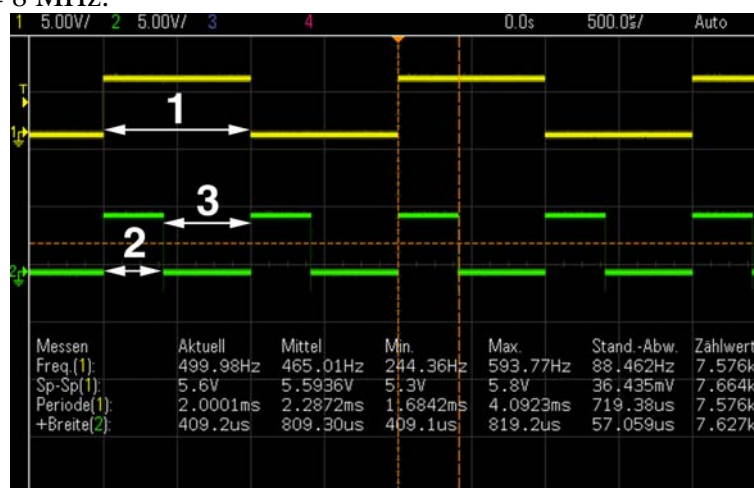
Die Capture-Bedingung abfragen und behandeln.

- 1 - ein Umlauf der Endlosschleife = 1 Periode des Prüfsignals.
- 2 - Dauer der Behandlung. Bei 4 MHz Takt rund 817 μ s, bei 8 MHz Takt rund 409 μ s.
- 3 - In dieser Zeit wird von Null an gezählt. Dieser Zählwert wird angezeigt. Meßfehler.

Quarzfrequenz = 4 MHz:



Quarzfrequenz = 8 MHz:

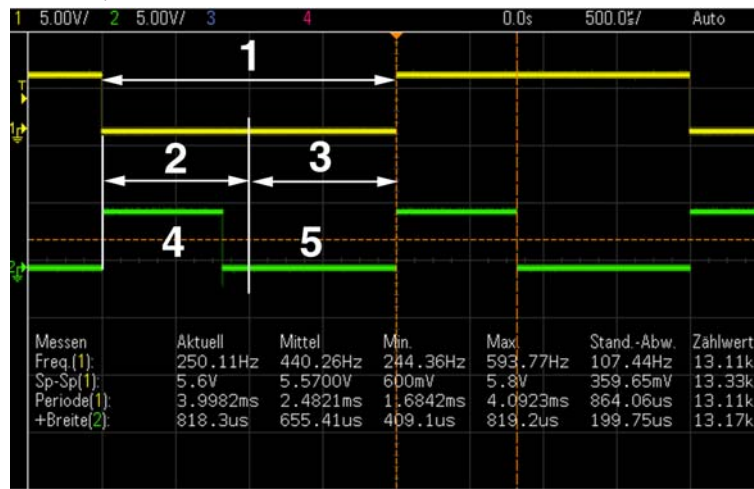


b) Verbesserter Meßablauf.

Die Capture-Bedingung abfragen und behandeln. Dann nochmals auf das Eintreten der Capture-Bedingung warten, um die Vorderflanke des Meßimpulses zu erkennen. Von da an zählen.

- 1 - ein Umlauf der Endlosschleife = 2 Perioden des Prüfsignals, die erste zum Anzeigen, die zweite zum Zählen für die nachfolgende Anzeige.
- 2 - Während dieser Prüfsignalperiode (= Halbperiode des Schleifenumlaufs) wird angezeigt.
- 3 - In dieser Zeit wird von Null an gezählt. Dieser Zählwert wird angezeigt. Kein Meßfehler.
- 4 - Dauer der Behandlung bei 4 MHz Takt rund 818 μs , bei 8 MHz Takt rund 409 μs .
- 5 - Warten auf die nächste Vorderflanke des Prüfsignals.

Quarzfrequenz = 4 MHz:



Quarzfrequenz = 8 MHz:

