

Bauanleitung und Tipps zum Bau einer Propeller-Uhr

Aktualisiert 05.04.09

# RN-Propellerclock

Sehr beeindruckendes und interessantes Projekt für SMD erfahrene Anwender.



Schaltungen mit LED-Effekten wirken immer wieder sehr faszinierend. Eine der beeindrucktesten Schaltungen der letzten Jahre ist die sogenannte „Propellerclock“ bzw. das „Propellerdisplay“. Im wesentlichen handelt es sich hier um eine Platine mit einer Reihe von Leds die in hoher Geschwindigkeit gedreht wird. Durch genaues Timing werden dann die Leds so zum Leuchten gebracht, das es für das Auge so erscheint als würde ein leuchtender Schriftzug oder auch eine Grafik mitten in der Luft geschrieben. Zustande kommt dies durch die Trägheit des Auges.

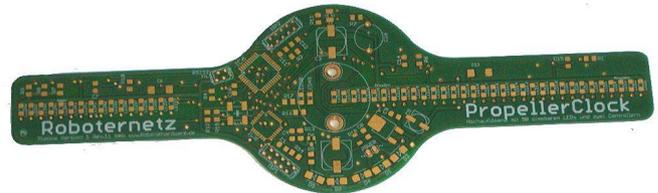
Oft werden die Leds auch so zum Leuchten gebracht das eine analoge Uhr frei schwebend in der Luft erscheint, daher der Name Propellerclock.

Zahlreiche Bastler haben schon Propelleruhren in den unterschiedlichsten Varianten gebastelt, die Idee ist also nicht neu. Da auch mich die Anzeige immer schon fasziniert hat und das ganze ein tolles Design-Objekt darstellt, bin ich dieses Projekt auch mal angegangen. Und damit auch andere Bastler was von haben, gebe ich meine Erfahrungen gerne in dieser Doku weiter.

## Fertige Platine zum Projekt erhältlich

Leider ist der Aufbau und die Programmierung einer solchen Propellerclock nicht ganz einfach für weniger erfahrene Bastler. Damit die Propelleruhr wirkt darf sie nicht zu klein sein.

Alleine das ist schon ein Problem denn viele Bastler haben bestenfalls ein Freeware CAD-Programm wie Eagle. Mit der Freeware Version sind jedoch nur sehr kleine Platinen möglich, für eine Propellerclock reicht das nicht aus.



Ein weiteres Problem ist die Bauteileauswahl. Die Bauteile müssen klein sein und so verteilt werden das möglichst wenig Unwucht entsteht. Bei hohen Drehzahlen bedeutet eine größere Unwucht erhebliche Vibrationen und einfach richtig Krach. Das stört nicht nur den Effekt sondern zerstört nach kurzer Zeit auch die Motorlager. Wenn ihr Pech habt dann fliegt euch auch noch die Platine um die Ohren.

Und noch ein weiteres größeres Problem ist die Stromversorgung der drehbar gelagerten Platine. Wie soll man die Platine mit Strom versorgen wenn sie sich schnell drehen soll, Kabel sind somit nicht möglich. Man muss also zu Schleifern oder induktiver Übertragung übergehen.

Und zu guter letzt dann noch die Programmierung! Wie bekommt man das Timing hin?

## Sehr lehrreiches Projekt

All diese und noch weitere kleine Probleme müssen gelöst werden damit am Ende ein beeindruckendes optisches Ergebnis steht. Gerade das macht diese Schaltung aber auch sehr reizvoll und lehrreich. Es gibt nur wenige Schaltungen die so viel Spaß und dennoch so viele verschiedene Grundlagen vermitteln.

Diese Dokumentation soll eine Möglichkeit aufzeigen wie man eine solche Propelleruhr realisieren kann. Robotikhardware.de bietet zu der hier beschriebenen Schaltung die passende Platine an, somit ist schon einmal ein großes Problem beseitigt. Es handelt sich dennoch nicht um ein leichtes Projekt, denn der mechanische Aufbau bleibt einem weitestgehend selbst überlassen, diese Doku soll allerdings viele hilfreiche Tipps dazu geben.

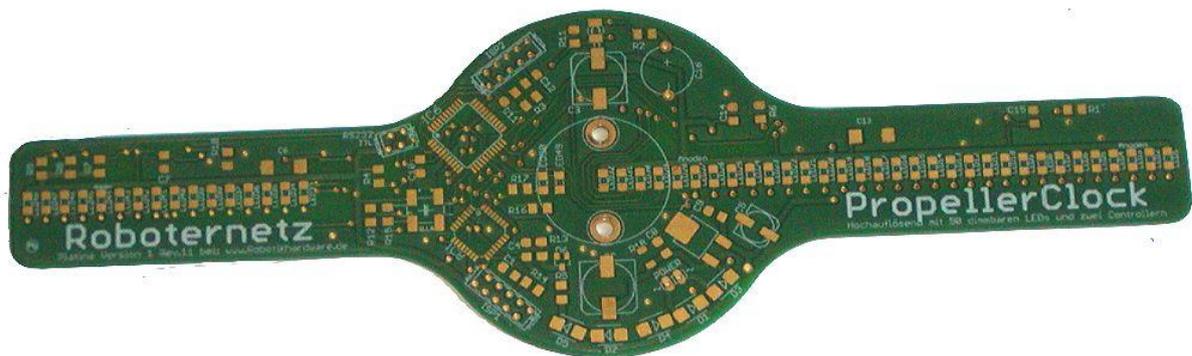
Auch die vorgestellten Programme und Lösungen sind keineswegs das Endstadium meines Propellerclock Projektes sonder zeigen nur den ersten Schritt und helfen zum ersten Erfolg zu kommen. Die Schaltung erlaubt noch viele weitere Möglichkeiten wie induktive Datenübertragung. Was letztlich draus entsteht und welche Möglichkeiten der Platine ihr nutzt bleibt jedem selbst überlassen, dieses Doku soll nur etwas helfen und praktische Erfahrungen vermitteln.

## Tipps für den Aufbau

### Die eigentliche Platine

Grundlage unserer Schaltung ist eine propellerförmige Platine mit 225mm Länge. Diese Platine wird quasi ausschließlich mit SMD-Bauteilen bestückt. Diese Bauteilformen haben das geringste Gewicht und die kleinste Größe. Fast alle Bauteile werden auf der beschrifteten Oberseite platziert, nur die LED-Treiber, Hallsensor, IR-LED und ein Fototransistor befinden sich auf der Rückseite. Die genaue Bauteileliste und deren Bezugsquelle sind weiter hinten in der Doku zu finden.

In der Mitte der Platine befinden sich zwei 3,2mm Löcher zum Befestigen der Platine an einer Motorwelle. Eignen tun sich hier zum Beispiel Radfelgen oder auch Zahnräder die man einfach mit zwei Bohrungen versieht. Verwendet man einen herkömmlichen PC-Lüfter als Motor kann man sich auch mit etwas Kunststoff (oder Acrylglas) zwischen Lüfter und Platine helfen. Geeigneter Kleber (welcher die Kunststoffe fest verschweißt) und zwei Gewinde im Kunststoff tun dann den Rest. Letzteres haben wir bei unserem Test gemacht.



Die [robotikhardware.de](http://robotikhardware.de) - Platine erleichtert den Aufbau erheblich

### Wichtig - Sicherheit beim Basteln

Wichtig ist das eine absolut feste Verbindung zwischen Platine und Motorwelle entsteht. Diese muss auch den Belastungen einer höheren Drehzahl (min. 1500 UMin) aushalten. Geht hier also auf Nummer sicher denn eine umherfliegende Platine kann erhebliche Verletzungen verursachen.

Es wird auch dringend empfohlen eine Art transparente Haube zu basteln, die über die drehende Platine gestülpt wird. Nur so ist sichergestellt das ihr euch nicht bei der drehenden Platine verletzen könnt.

## Welcher Motor / Welche Bildwiederholfrequenz

Im Grunde eignet sich fast jeder Gleichstrommotor zum drehen der Platine. Es werden keine sehr großen Anforderungen an den Motor gestellt, jedoch sollte diese bei der verwendeten Spannung eine Umdrehungszahl zwischen 800 und 1500 Umdrehungen pro Minute nicht überschreiten. Das bedeutet also das unsere Propelleruhr später eine Bildwiederholungsrate von ca. 15 bis 25 Bilder pro Sekunde hin bekommen würde. Das Reicht für hochauflösende Darstellungen und Effekte aus. Viel höhere Bildwiederholungsraten sind mit der Schaltung nicht drin, da die LED-Treiber maximal mit 1Mhz Bus per I2C angesteuert werden müssen.

Sollte der Motor eine höhere Nenndrehzahl haben, so kann man diese gewöhnlich durch reduzieren der Spannung reduzieren. Natürlich könnte man die Geschwindigkeit des Motors auch per PWM mit einem weiteren Controllerboard (z.B. RN-Control) steuern, aber gewöhnlich ist das nicht notwendig.

Ideal eignen sich auch viele PC-Lüfter. Einige Restposten-Anbieter bieten sehr günstig Lüfter mit niedrigen Drehzahlen an. Diese eignen sich sehr gut da sie meist sehr laufruhig sind und eine sehr konstante Drehzahl bieten. Bei meinem Test habe ich einen solchen Lüfter mit 1500 U/Min genutzt. Die Flügelblätter wurden einfach mit einen kleinen Minibohrmaschine mittels Trennscheibe herausgeschnitten. Der Rest wurde mit einem Minifräser sauber weggeschliffen. Dadurch entfällt dann auch das restliche Luftgeräusch und der Lüfter ist kaum noch zu hören. Mit Platine hat unser Lüfter dann bei 12V ca. 1000 und bei 16V ca. 1236 UMin geleistet, also max. 20 Bilder pro Sekunde. Die Fotos in dieser Doc wurden mit 17 Bilder pro Sekunde geschossen, genau wie das Video. Wenn man nur 16 statt 32 Leds nutzt, kann man die Bildwiederholfrequenz auch noch erheblich erhöhen.

Um die genaue Drehzahl eures Propellers zu ermitteln habe ich noch ein Programm „umdrehungenzaehlen.bas“ auf die Robotikhardware CD gepackt. Einfach in den Hauptcontroller der Propelleruhr laden und dann dieses starten. Das Programm wartet bis der Motor voll läuft und zählt dann die Umdrehungen. Nach einigen Sekunden wird das Ergebnis endlos über die RS232 ausgegeben. Aber steht alles in den Kommentarzeilen zum Programm.



## Die Stromversorgung

Wie schon erwähnt stellt die Stromversorgung der Platine eine besondere Herausforderung dar. Unsere Platine kann bis zu 50 Leds aufnehmen und ansteuern. Die Stromaufnahme kann also Spitzenwerte von ca. 1 A erreichen. Ein Betrieb per Akku macht daher wenig Sinn.

Es bleiben also nur zwei Möglichkeiten:

1. Man befestigt an der Motorachse zwei Ringe an die man einen Draht anlötet und mit der Platine verbindet. Zwei Schleifer müssten dann während sich die Achse mit den Ringen dreht die Spannung durch Berührung übertragen.  
**Nachteil:** Nicht ganz einfach umzusetzen und ein gewisser mechanischer Verschleiß der Schleifer wenn man es nicht richtig auslegt.  
**Vorteil:** Auch höhere Ströme lassen sich relativ gut übertragen
2. Eine zweite Möglichkeit ist die induktive Übertragung von Spannung. Man kennt das Prinzip vielleicht von der elektrischen Zahnbürste. Diese wird aufgeladen ohne dass sie eine elektrische Verbindung zum Ladegerät besitzt.  
**Nachteil:** Wirkungsgrad ist oft nicht toll und Aufbau ebenfalls nicht sonderlich leicht. Durch die Spulen wird die PropellerClock gewöhnlich etwas tiefer und Unwucht Probleme kommen schnell auf.  
**Vorteil:** kein Verschleiß

Ich habe mich bei diesem Projekt für Methode 2 (induktive Stromübertragung) entschieden, was nicht heißt das Methode 1 nicht auch sinnvoll ist.

Die Platine ist so ausgelegt das beide Methoden möglich sind. Sie muss einfach nur mit einer Spannung zwischen 7 und 18 Volt versorgt werden. Dabei ist es ihr sogar egal ob es Gleichstrom oder Wechselstrom ist. Auch wie der Gleichstrom an dem vorgesehenen Anschluss (POWER) gepolt wird, ist der Schaltung egal (dank Brückengleichrichterschaltung).

## Induktive Stromübertragung

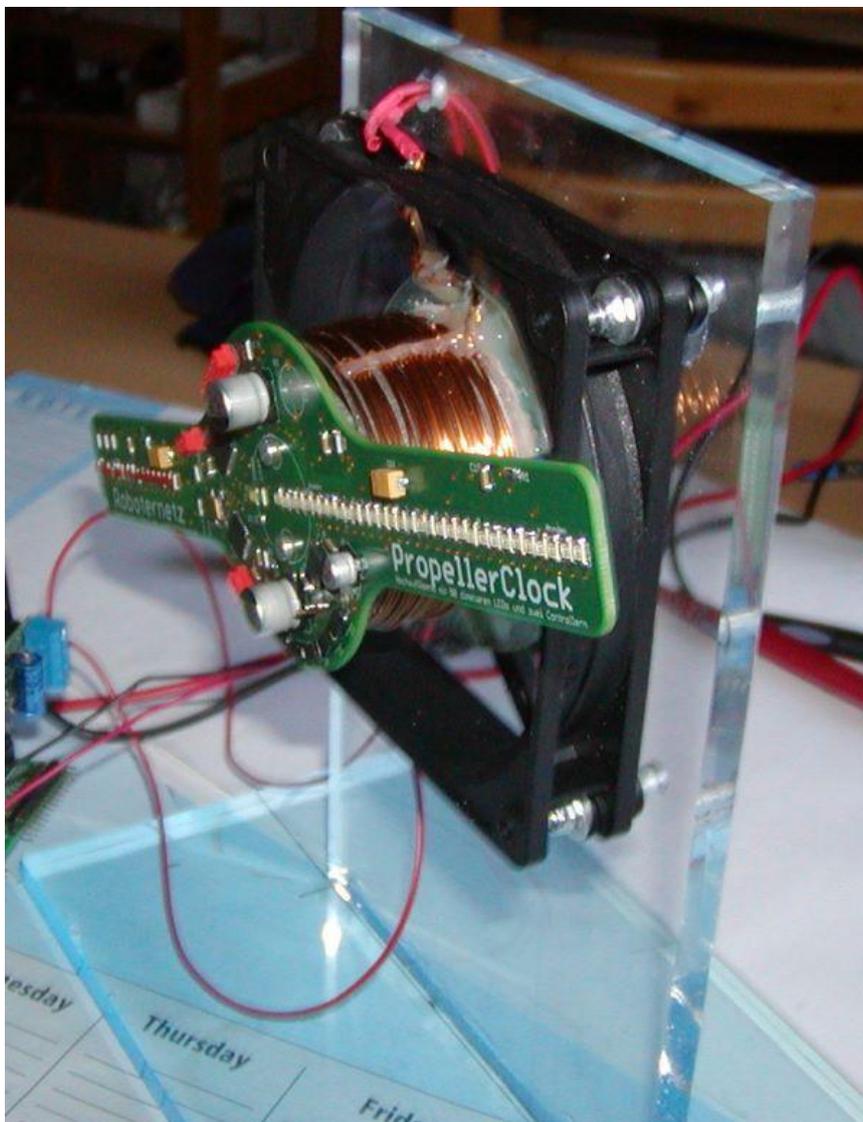
Um die Schaltung möglichst einfach mit der notwendigen Spannung zu versorgen brauchen wir zwei Spulen die ineinander stecken. Eine der Spulen muss sich mit der Platine drehen und eine andere muss fest stehen und mit einer Wechselspannung versorgt werden. Die fest stehende Spule erzeugt dadurch ein Magnetfeld und induziert somit eine Spannung in der anderen Spule. Die Spannung und der Strom müssen dann nur noch ausreichen um die Platine zu versorgen. Ob die innere oder äußere Spule als Primär- oder Sekundär –Spule genutzt wird ist dabei völlig egal.

Das ganze hört sich einfach an, aber in der Praxis ist es gar nicht so einfach die richtigen Drahtdurchmesser, Windungszahl und Frequenz der Wechselspannung zu finden. Ich habe bestimmt über 10 Spulen gewickelt um an brauchbare Ergebnisse heranzukommen. Hier müsst ihr einfach ein wenig experimentieren, wobei nachfolgende Tipps euch schon sehr helfen müssten:

1. Die Spulen sollten einen nicht zu kleinen Durchmesser besitzen. Gute Werte habe ich erst mit Spulen von mindestens 4cm Durchmesser erzielt. Ideal sind wohl für unser Projekt Spulendurchmesser zwischen 40 und 70 mm. In der Praxis eignen sich Kunststoff HT-Rohre (Abwasserrohre) aus dem Baumarkt sehr gut für solche Spulen. Auch die Stopfen solcher Rohre können gut verwendet werden, an sie kann man dann auch gut die Platine anschrauben.
2. Innere und äußere Spule sollten möglichst dicht aneinander sein. Je dichter sich die Windungen gegenüberstehen desto höher der Wirkungsgrad, also je besser die Stromübertragung. Allerdings muss man auch darauf achten das eine geringe Unwucht nicht dazu führt das sich innere und äußere Spule berühren, sie würden sich sehr schnell zerstören und zudem würde der Motor nicht mehr gleichmäßig drehen. Also keine Sorge wenn der Abstand etwas größer aus fällt, es klappt notfalls auch noch bei 3-5mm Abstand, allerdings halt mit schlechteren Wirkungsgrad.
3. Im Inneren der Spulen sollte möglichst wenig Metall vorhanden sein. Ist das Metall zu nah an den Wicklungen beeinflusst dies das Magnetfeld und den Wirkungsgrad sehr ungünstig. Also nicht auf den Gedanken kommen den Motor direkt zu umwickeln. Zumindest ein kleiner Abstand muss sein!
4. Die Wicklungszahl darf nicht zu groß gewählt werden. Oft denkt man je mehr desto besser, aber das ist leider falsch. Werden zu hohe Windungszahlen gewählt, so erhöht sich nicht nur die Induktivität der Primärspule sondern auch der induktive Widerstand. Ein hoher induktiver Widerstand gestattet dann keine größeren Ströme mehr. Gute Ergebnisse haben Wicklungszahlen von ca. 30 bis 40 Windungen bei der Primärspule ergeben. Bei der Sekundärspule sollten es einige Windungen mehr sein, hier haben 40-60 Windungen ein gutes Ergebnis gezeigt. Es kommt natürlich immer auch auf die Frequenz der Wechselspannung an, dazu gleich mehr. Bei der Primärspannung gehe ich bei den genannten Angaben mal von 10 bis 15 Volt aus.
5. Der Drahtdurchmesser der Spulen sollte nicht zu dünn werden. Unsere Schaltung kann bis zu 1A ziehen. Bedenkt man dann noch dass der Wirkungsgrad oft nicht toll ist, so kann primär durchaus 1,5 bis 2A gezogen werden, bei niedrigen Frequenzen sogar noch mehr. Die Spulen sollten sich bestenfalls nur gering bis gar nicht erwärmen. Brauchbare Ergebnisse zeigten Primär min. 1mm<sup>2</sup> Draht und Sekundär 0,8mm<sup>2</sup> Draht.
6. Eine übliche Wechselspannung von 50 Hz reicht leider nicht aus um genügend Strom auf die Sekundärspule zu übertragen. Es ist daher zu empfehlen eine Wechselspannung mit höherer Frequenz zu generieren. Je weniger Windungen die Primärspule besitzt, desto höher kann die Frequenz sein. Bei oben genannten Richtwerten haben sich Frequenzen zwischen 12kHz und 20kHz als optimal erwiesen. Darunter und darüber sinkt der Wirkungsgrad, was man leicht durch Messen der Spannung an der Sekundärspule (am besten bei Belastung) feststellen kann.

## Beispielaufbau

Das Bild zeigt unseren Beispielaufbau. Auf die Nabe des Lüfters wurde die Sekundärspule (40mm Rohr) aufgebracht und daran die Platine angeschraubt. Außen herum wurde die Primärspule auf ein etwas größeres Rohr gewickelt. Dieses Rohr ist am Rahmen angeklebt. Im Bild sieht man also nur die Primärspule. Der Rahmen des Ventilators ist dann mit Gummiringen als Schwingungsdämpfung an einer Acrylglasplatte, die als Ständer dient, befestigt. Letzteres ist nicht unbedingt nötig da unser Lüfter ohnehin sehr ruhig läuft und kaum eine Unwucht aufzeigt. Man muss genau darauf achten das man die Rohre und Platine möglichst genau mittig platziert, sonst nerven die Vibrationen / die Unwucht schon erheblich. Das empfand ich auch als größte Schwierigkeit bei dem ganzen Projekt. Das Wickeln der Spule hingegen ist nicht sonderlich schwierig. Es ist nicht weiter schlimm wenn die Windungen mal nicht ganz genau nebeneinander liegen und die Windungszahl nicht 100% genau stimmt. Damit die Windungen bei wickeln nicht verrutschen kann übrigens etwas doppelseitiges Klebeband auf dem Rohr (Spulenkörper) sehr hilfreich sein.



## Wechselspannung generieren

Kommen wir zurück zur Wechselspannung die für die primäre Spule nötig ist. Am einfachsten lässt sich diese ebenfalls über einen kleinen Microcontroller erzeugen. Das hat auch den Vorteil das man sehr leicht verschiedene Frequenzen programmieren und ausprobieren kann. Ein weiterer Vorteil besteht darin, das man über die Betriebsspannung auch Daten an die Propellerplatine übertragen könnte, aber dazu später.

Zur Erzeugung der Wechselspannung verwendet man am besten einfach einen kleinen Motortreiber. Dieser sollte allerdings mindestens 2A aushalten. Es ginge zum Beispiel mit dem IC L298, allerdings wird dieser schon mächtig warm bei dem benötigten Strom. Ideal ist der neue Motortreiber „**HighPower Motortreiber36V9**“ welcher auch im Shop von [www.robotikhardware.de](http://www.robotikhardware.de) angeboten wird. Er hat soviel Leistungsreserven das er sich quasi noch nicht mal erwärmt. Er hat noch einen anderen Vorteil, nämlich die Ansteuerung mittels **DIR** Eingang. Zur Erzeugung einer optimalen Wechselspannung legt man einfach den PWM-Eingang dauerhaft auf High und versorgt den DIR-Eingang mit einer Frequenz die man per Timer generiert. Das Software-Programm kann dann besonders einfach ausfallen. Verwendet man doch den L298 so müsste man in einer Interrupt Routine stets die Eingänge IN1 und IN2 umpolen was einen erheblich größeren Rechenaufwand bedeutet. Der Controller könnte dann bei hohen Frequenzen nicht mehr allzu viel anderes machen. Ginge aber auch!

Durch den Richtungswechsel ergibt sich am Ausgang eine optimale Rechteck-Wechselspannung die man direkt auf die Primärspule leiten kann. Im Test ergab das optimale Ergebnisse. Man kann die Wechselspannung auch generieren indem man nur die Spannung ein- und ausschaltet, jedoch hat man dann sekundär auch nur die halbe Spannung. Daher ist eine H-Bridge und Richtungsumschaltung wesentlich günstiger.



## Das dazu passende Beispielprogramm für einen Mega128

```

#####
' PowerMotorDriverFrequenzgenerator
' für
' Controllerboard Board mit Mega128 für
' die Serie RN-Modular, dem modularen
' AVR Entwicklungssystem
'
'Aufgabe:
' Ansteuerung des Pololu Power Motor Driver 36v9
' Erzeugt eine Konstante Frequenz am Ausgang OC1A (Kein PWM)
'Autor: Frank
'Infos:http://www.Roboternetz.de oder robotikhardware.de
#####

$programmer = 12                                'MCS USB (Zeile weglassen wenn anderer Programmier)

' -----RN-Mega128 übliche -----

$regfile = "m128def.dat" ' Bei Mega 128 "
$framesize = 64
$swstack = 64
$hwstack = 64

$crystal = 16000000                            'Quarzfrequenz

Config Pine.3 = Output
Motorboard_richtung Alias Porte.3

Config Pina.1 = Output
Motorboard_reset Alias Porta.1

Config Pine.4 = Output
Motorboard_pwm Alias Porte.4

Config Timer3 = Timer , Prescale = 1 , Compare A = Toggle , Clear Timer = 1

Motorboard_reset = 1
Motorboard_pwm = 1
Motorboard_richtung = 1

Enable Timer3
Start Timer3
' Compare3a = 100                                '79,2 khz
' Compare3a = 200                                '39,7 khz
' Compare3a = 300                                '26,5 khz
' Compare3a = 400                                '20 khz
' Compare3a = 500                                '16 khz
' Compare3a = 600                                '13,3 khz
' Compare3a = 700                                '11,4 khz
' Compare3a = 800                                '10 khz
' Compare3a = 900                                '8,9 khz
' Compare3a = 1000                               '8 khz
' Compare3a = 2000                               '4 khz
' Compare3a = 3000                               '2,6 khz
' Compare3a = 4000                               '2 khz
' Compare3a = 10000                              '800 Hz
' Compare3a = 30000                              '266 Hz
' Compare3a = 50000                              '160 Hz
' Compare3a = 65000                              '123 Hz

Do
Loop

```

## Schaltungsbeschreibung der Propellerclock

Unsere Propellerclock verfügt über 50 LED's, wobei 32 blaue auf der einen Seite und 16 rote auf der anderen Seite angeordnet sind. Eine grüne und gelbe befindet sich im Bereich der Mitte.

Die Hauptaufgabe sollen eigentlich die 32 blauen Leds übernehmen, sie reichen aus um zum Beispiel eine Bildauflösung von ca. 200-220 Pixel x 32 Pixel darzustellen. Sie sind so angeordnet dass vom Mittelpunkt bis zum äußeren Ende der „Rotorplatine“ keine Lücke entsteht. Dies ist wichtig wenn man zum Beispiel eine Uhr mit Zeiger darstellen möchte.

Die anderen Leds können genutzt werden um etwas mehr Farbe in die Grafik zu bringen.

Bestückt werden wahlweise Standard SMD Leds (1206 Bauform) oder etwas hellere Osram LED's (siehe Bauteileliste). Wichtig ist nur das diese nur 20mA verbrauchen, die Durchlassspannung spielt keine Rolle. Leds mit anderen Strömen sind zwar auch möglich, jedoch müsste man dann für R1,R6,R10 einen anderen Wert nehmen.

Angesteuert werden die Leds von einem hochmodernen LED-Treiber TLC59116. Drei solche Treiberbauteile sind vorhanden, jeder steuert 16 LED's an. Der LED-Treiber sorgt automatisch dafür das jede LED den korrekten Strom von 20 mA erhält. Angesteuert werden die LED-Treiber ganz bequem über den I2C-Bus. Der I2C-Bus darf dabei sogar maximal mit 1 Mhz Takt betrieben werden, dadurch ist eine sehr schnelle Ansteuerung aller Leds möglich. Jede kann beliebig ein oder ausgeschaltet werden. Eigentlich ist damit der TLC59116 noch unterfordert, denn er kann noch mehr. Er kann auch die Helligkeit jeder einzelnen LED verändern, dazu muss lediglich das Ansteuerprogramm geändert werden. Allerdings wird diese Möglichkeit in unserem ersten Beispielprogramm nicht genutzt. Würde man diese Fähigkeit nutzen so müsste man auch die Bildwiederholungsrate sehr zurücknehmen oder nur wenige Leds zur Anzeige nutzen weil sonst einfach das Timing zu knapp wird, der Controller wird es dann kaum noch schaffen die Leds rechtzeitig zu aktualisieren da ja mit den Helligkeitswerten viel mehr Daten übertragen werden müssen.

Aus diesem Grunde empfehlen die enormen Fähigkeiten des TLC59116 ruhig mal alle zu testen damit man sie vielleicht auch bei anderen Projekten anwenden kann. Jedoch dann bei der Propelleruhr nur den Mode zum Ein- und Ausschalten der Leds zu nutzen. Im Bascom Beispielprogramm, das mit der Platine geliefert wird, sieht man genau wie dies funktioniert. Ein Datenblatt des TLC59116 mit all seinen I2C Befehlen liegt auf CD ebenfalls bei. Der Chip ist derzeit auch über [Robotikhardware.de](http://Robotikhardware.de) erhältlich.

Als Hauptcontroller verwendet unsere Propellerclock einen Mega644. Es könnte aber auch jeder andere kompatible AVR Controller mit gleicher Bauform verwendet werden. Er sollte nur 20 Mhz verkraften, da unser verwendeter Oszillator 20 Mhz Takt liefert. Die Entscheidung für den Mega644 viel wegen des Taktes und den 4kByte Ram sowie 65KB Flash über die er verfügt. So haben wir genügend Speicher um sogar kleine Animationen einzuprogrammieren.

### **Datenübertragung vorgesehen**

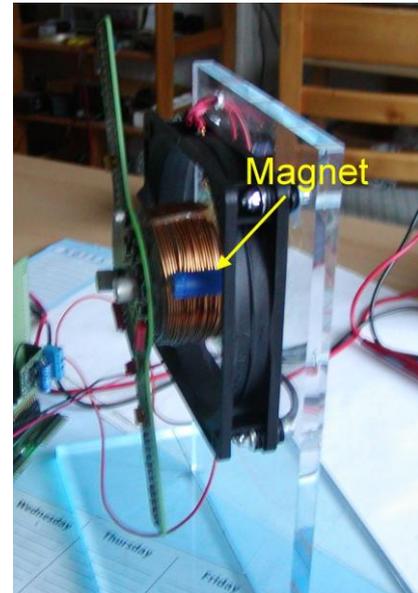
Ich habe auch noch einen zweiten CoController (Mega168 20 Mhz)vorgesehen. Dieser muss nicht unbedingt bestückt werden. Er ist nur dann sinnvoll wenn Daten während des Rotierens

auf die Propellerplatine übertragen werden sollen, zum Beispiel Texte, Uhrzeit oder Bilder die angezeigt werden sollen. Für diesen Zweck führt eine Halbwelle der Wechselspannung auf PD2 (INT0) und PB0 (ICP1). Also auf einen Interrupt Eingang und einen Timer Capture Eingang. Dadurch kann der CoController jederzeit genau messen wie lange eine Halbwelle dauert (woraus er die Frequenz errechnen kann).

Wenn nun die Frequenz der Wechselspannung der Primärspannung leicht geändert wird, erkennt dies somit der CoController. Dies kann genutzt werden indem man die Frequenz im binären Rhythmus ändert, also bei 1 eine andere Frequenz als bei 0. Somit könnte der CoController Binärzahlen decodieren, also Daten empfangen. Synchronisieren müsste man das ganze dann noch, am einfachsten mittels Umdrehung. Immer wenn eine Umdrehung rum ist, wird eine 1 oder 0 übertragen.

Wann eine Umdrehung rum ist, wird durch einen Hallsensor ermittelt. Der Hallsensor reagiert auf einen Magneten an dem er bei jeder Runde vorbeikommt.

Aber auch der Controller der die Wechselspannung generiert muss erkennen wann eine Runde vorbei ist. Dazu kann eine IR-LED (D7) zu Hilfe genommen werden. Sie kann ein Lichtsignal senden das z.B. von einem Fototransistor empfangen wird. Alternativ kann natürlich auch eine Reflexlichtschranke einfach das Rotorblatt erfassen, das kann jeder nach Belieben umsetzen.



Die Daten die der CoController empfängt, könnte er dann per RS232 zum Hauptcontroller senden. Achtung, hier muss beim CoController der RX-Pin zum Senden verwendet werden, man muss also die Software RS232 von Bascom zum Senden nutzen. Dies liegt daran das RX und TX beider Controller parallel geschaltet auf einen 4 poligen Stecker führen. An diesem Stecker kann man während der Programmierung ein (über Pegelwandler Max232) PC mit Terminalprogramm anschließen um mit den Controllern zu kommunizieren. Zum Beispiel um Variableninhalte zu debuggen. Achtung, nie bei beiden Controllern gleichzeitig den Hardware Uart (RS232) nutzen.

Leider bin ich nicht mehr dazu gekommen um auch ein Beispielprogramm für die Datenübertragung zu schreiben, zumal ich es selbst für meine Anwendung nicht brauche. Aber vielleicht habe ich später noch Lust und zeit dazu oder aber ein anderer Bastler stellt ein passendes ins Roboternetz.de. Gerne packe ich es dann noch auf CD.

Erwähnen möchte ich noch dass selbst ohne CoController die Datenübertragung möglich ist, denn die Halbwelle wird nicht nur auf den CoController sondern auch auf den Capture Eingang PD6(ICP) des Hauptcontrollers geleitet. Allerdings hat der Hauptcontroller mit der Anzeige soviel zu tun, das er Daten vermutlich nur dann empfangen kann wenn er die Leds nicht gleichzeitig noch aktualisieren muss!

Erwähnenswert sind noch die zwei Fototransistoren auf der Platine. Eine befindet sich auf der Foderseite und eine auf der Rückseite. Sie führen zu einem analogen Eingang. Sie können wahlweise beide oder auch nur einer von ihnen bestückt werden. Sie können

verschiedene Aufgaben erfüllen, zum Beispiel einfach um die Zimmerhelligkeit zu messen und die LED Helligkeit anzupassen. Denkbar wäre aber auch das nur der auf der Rückseite bestückt wird. Er könnte dann Rückmeldungen einer IR-Led registrieren. Denkbar wäre so eventuell auch eine Datenübertragung über diesen Fototransistor, ganz ohne Spulenfrequenzen. Auch dieser Empfang könnte mit der Umrundung synchronisiert werden. Aber das ist nur so ein Gedanke, ich habe es nicht ausprobiert, aber halt mal für Leute vorgesehen die gerne an solchen Timing-Software-Dingen basteln.

Zur Programmierung ist noch zu sagen das für den Hauptcontroller und den CoController jeweils eine 10 poliger ISP-Buchse auf der Platine vorhanden ist. Die Belegung entspricht dem üblichen ISP-Standard sowie es in den RN-Definitionen beschrieben ist. Allerdings wurde eine kleinere Buchse verwendet, da die großen Wannenbuchsen einfach weniger gut für die rotierende Platine geeignet waren. Dennoch können die üblichen 10 poligen Programmieradapter wie (ISP, Bascom USB-ISP usw.) zur Programmierung verwendet werden. Es muss lediglich noch ein zusätzlicher kleiner roter Stecker (Reichelt Best.Nr. MM SL 10SK) an das übliche ISP Kabel angepresst werden (siehe Foto).



Standard 10 poliger ISP, jedoch kleinerer Micromatch Stecker

So das waren eigentlich die wichtigsten Dinge die zur Schaltung zu sagen sind. Die anderen Eigenschaften und Merkmale könnt ihr aus dem Schalt- und Bestückungsplan auf den letzten Seiten dieser Doku entnehmen. Ihr findet dort auch eine Beschreibung aller verwendeten Bauteile und deren Bezugsquelle.

Somit steht einem Bau einer PropellerClock nix mehr im Wege. Mit etwas Elektronik Grundkenntnissen und SMD Lötverfahren sollte das ganze nicht weiter schlimm sein. Aber etwas SMD Lötverfahren sollte schon vorhanden sein. Die Platine von Robotikhardware.de hat eine sogenannte hochwertige Chemisch Gold-Verzinnung, dadurch lassen sich übrigens SMD Bauteile noch besser löten als mit herkömmlichen Verzinnungen. Mit genügend Flußmittel lassen sich die kleinen Bauteile übrigens besser löten als man herkömmlich denkt.



**Viel Spaß und gutes Gelingen!**

## Beispielprogramm

Zum Abschluss noch ein kleines Testprogramm für den Hauptcontroller der Propelleruhr. Das Programm gibt die Texte aus die in den Fotos auf den vorherigen Seiten zu sehen sind. Dazu wurde eine Unterfunktion gebastelt, welcher lediglich Text und Zeile übergeben wird. Diese Funktion macht dann alles Weitere. Sie zentriert den Text entsprechend und überträgt die Zeichenmatrix des Textes in den Bildspeicher. Verwendet wird ein 8x8 Pixel Zeichensatz welcher in Form von DATA-Anweisungen in den Flash Speicher übernommen wird. Alternativ könnte man bei Programmänderung natürlich auch jede andere Zeichenmatrix nutzen, auch verschiedene gleichzeitig. Alles nur Sache der Programmierung.

Erwähnen möchte ich noch dass das Programm keine Analyse der Umdrehungszeit vornimmt. Sobald ein Rundenwechsel erkannt wird, gibt das Programm im festen Timer-Rhythmus die 200 Spalten aus. Da unser Lüfter sehr konstante Umdrehungsgeschwindigkeiten zeigt, reicht dieses Verfahren völlig für ein stehendes Bild aus. Rechnet man dagegen mit wechselnden Drehzahlen, so könnte man natürlich pro Runde die Timer Intervalle pro Spalte anpassen um dies auszugleichen.

Sollte euer Motor zu langsam drehen, so erkennt ihr das daran dass die Schrift nicht mehr den ganzen 360 Grad Bereich ausnutzt. Dreht er zu schnell, so erkennt ihr dass daran das die Schrift nicht komplett in den Bereich passt und überläuft. Dann entweder Programm ändern (Spalten reduzieren) oder Geschwindigkeit des Motors herabsetzen (weniger Spannung).

```
'#####
'rn_propeller_main.bas
'Dieses Programm ist für das Projekt RN-PropellerClock
'50 drehende Led's erzeugen ein Bild
'Boardbeschreibung Roboternetz.de oder robotikhardware.de
'Verwendeter Compiler Bascom V 1.11.9.3
'
'Aufgabe:
'Programmiert werden vornehmlich drei Chips vom Typ TLC59116
'
'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de bzw. robotikhardware.de
'#####

$programmer = 12                                'MCS USB (Zeile weglassen wenn
anderer Programmer)
$regfile = "m644def.dat"
$crystal = 20000000                             'Quarzfrequenz
$baud = 9600
$hwstack = 80
$framesize = 80
$swstack = 80

$lib "I2C_TWI.LBX"
Config Twi = 1100000                             ' wanted clock frequency
Config Scl = Portc.0                             'Ports fuer IIC-Bus
Config Sda = Portc.1

Declare Sub Printzeichenoben(byval Zeichen As Byte , Byval Zeile As Byte)
Declare Sub Printzeichenunten(byval Zeichen As Byte , Byval Zeile As Byte)
Declare Sub Printstring(byval Text As String , Byval Zeile As Byte)
Declare Sub Clearbild()

Const An = 0
Const Aus = 1
```

```

'I2C Bus Slave Adressen
Const Tlc1 = &B11000010
Const Tlc2 = &B11000000
Const Tlc3 = &B11000100

'TLC Register
Const Ledout0 = &H14 Or 128

I2cinit

Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc

Config Int1 = Falling                                'Rundenstart duch Hallsensor ermitteln

Config Pind.4 = Output
Led49_gruen Alias Portd.4

Config Pind.7 = Output
Led50_gelb Alias Portd.7

Config Pind.5 = Output
Irled Alias Portd.5

Config Pind.3 = Input
Hallsensor Alias Pind.3

Config Pind.6 = Input
Spannung Alias Pind.6

Dim Helligkeit As Word
Dim Stopzeit As Word
Dim Runden As Word
Dim Timervorgabe As Word
Dim Spalte As Byte

Dim Bildspeicher(1608) As Byte
Dim Pbildbyte As Word
Dim Aktpos As Word

'Lokale Variablen bei Zeichensatzauslesung
Dim Iz As Word
Dim Dmatrix As Byte
Dim It As Byte
Dim Zz As Byte
Dim Text As String * 100
Dim Temptext As String * 100

'TLC einschalten (Sleep out)
I2cstart                                'start condition
I2cwbyte Tlc1
I2cwbyte 0                                'Register
I2cwbyte 0
I2cstop

I2cstart                                'start condition
I2cwbyte Tlc2
I2cwbyte 0                                'Register
I2cwbyte 0
I2cstop

I2cstart                                'start condition
I2cwbyte Tlc1
I2cwbyte Ledout0                            'Register
I2cwbyte &B01010101

I2cstart                                'start condition
I2cwbyte Tlc2

```

```

I2cwrite Ledout0                                'Register
I2cwrite &B01010101
I2cwrite &B01010101
I2cwrite &B01010101
I2cwrite &B01010101
I2cstop

On Int1 Rundenstart

Config Timer1 = Timer , Prescale = 1
Timervorgabe = 65036 'reduzierung erhöht die zeit für Spaltenwechselausgabe
On Timer1 Spalten_irq
Disable Timer1

Enable Int1
Enable Interrupts

Do
Led50_gelb = An

Clearbild

Text = "Die"
Printstring Text , 1

Text = "Elektronik"
Printstring Text , 2

Text = "Community"
Printstring Text , 3

Text = "Roboternetz.de"
Printstring Text , 4

Wait 1
Clearbild

Text = "Test der"
Printstring Text , 1

Text = "Propeller"
Printstring Text , 2

Text = "Darstellung"
Printstring Text , 3

Text = "50 LED`s" "
Printstring Text , 4

Wait 1
Clearbild

Text = "Platine"
Printstring Text , 1

Text = "auf Anfrage"
Printstring Text , 2

Text = "Robotik"
Printstring Text , 3

Text = "hardware.de"
Printstring Text , 4

Wait 1
Clearbild
Toggle Led49_gruen
Loop
    
```

```

' *** Interrupt Routine wird aktiviert wenn Hallsensor Magnet erkennt
' *** also wenn Runde startet
Rundenstart:

    Pbildbyte = 1
    Spalte = 1
    Timer1 = Timervorgabe
    Enable Timer1
    Disable Int1
Return

Spalten_irq:
    Timer1 = Timervorgabe

    I2cstart                                'start condition
    I2cwbyte Tlc2
    I2cwbyte Ledout0                        'Register
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte

    I2cstart                                'start condition
    I2cwbyte Tlc1
    I2cwbyte Ledout0                        'Register
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cwbyte Bildspeicher(pbildbyte)
    Incr Pbildbyte
    I2cstop

    If Spalte > 200 Then
        Disable Timer1

        I2cstart                                'start condition
        I2cwbyte Tlc1
        I2cwbyte Ledout0                    'Register
        I2cwbyte 0
        I2cwbyte 0
        I2cwbyte 0
        I2cwbyte 0

        I2cstart                                'start condition
        I2cwbyte Tlc2
        I2cwbyte Ledout0                    'Register
        I2cwbyte 0
        I2cwbyte 0
        I2cwbyte 0
        I2cwbyte 0

        I2cstop
        Enable Int1
    End If
    Incr Spalte
Return

Sub Clearbild()                                'Bildspeicher loeschen
Local Le As Word
    For Le = 1 To 1608
        Bildspeicher(le) = 0
    Next

```

```

End Sub

'Gibt einen String aus Zeile 1 und 2 ist oben
'und Zeile 3 und 4 unten
Sub Printstring(byval Text As String , Byval Zeile As Byte) 'Eine Art Print Befehl zum
Ausgeben von Text
Local Le As Word
Local Zei As Byte
Local S As String * 1
Local Ls As Word

    If Zeile = 1 Or Zeile = 2 Then
        Aktpos = 1601
    Else
        Aktpos = 1
    End If

    Temptext = ""
    Ls = 0
    Le = Len(text)
    If Le < 12 Then
        Ls = 12 - Le
        Ls = Ls / 2
    End If

    If Ls > 0 Then
        Temptext = Space(Ls) + Text
    Else
        Temptext = Text
    End If

    Le = Len(temptext)
    For Zz = 1 To Le
        S = Mid(temptext , Zz , 1)
        Zei = Asc(s)
        If Zeile = 1 Or Zeile = 2 Then
            Printzeichenoben Zei , Zeile
        Else
            Printzeichenunten Zei , Zeile
        End If
        Print S
    Next Zz

End Sub

Sub Printzeichenunten(byval Zeichen As Byte , Byval Zeile As Byte)
Iz = Zeichen
Iz = Iz * 8

If Zeile = 4 Then
    For It = 1 To 8
        Dmatrix = Lookup(iz , Zeichensatz)
        Bildspeicher(aktpos) = 0
        If Dmatrix.3 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.2 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
        End If
        If Dmatrix.1 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
        End If
        If Dmatrix.0 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
        End If

        Incr Aktpos
        Bildspeicher(aktpos) = 0

        If Dmatrix.7 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.6 = 1 Then

```

```

        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
    End If
    If Dmatrix.5 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
    End If
    If Dmatrix.4 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
    End If
    Aktpos = Aktpos + 7
    Incr Iz
Next It
End If

If Zeile = 3 Then
    For It = 1 To 8
        Dmatrix = Lookup(iz , Zeichensatz)
        Incr Aktpos
        Incr Aktpos
        Bildspeicher(aktpos) = 0
        If Dmatrix.3 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.2 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
        End If
        If Dmatrix.1 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
        End If
        If Dmatrix.0 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
        End If

        Incr Aktpos
        Bildspeicher(aktpos) = 0

        If Dmatrix.7 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.6 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
        End If
        If Dmatrix.5 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
        End If
        If Dmatrix.4 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
        End If
        Aktpos = Aktpos + 5
        Incr Iz
    Next It
End If

    Aktpos = Aktpos + 8
End Sub

Sub Printzeichenoben(byval Zeichen As Byte , Byval Zeile As Byte)
    Iz = Zeichen
    Iz = Iz * 8

    If Zeile = 1 Then
        For It = 1 To 8
            Dmatrix = Lookup(iz , Zeichensatz)
            Bildspeicher(aktpos) = 0
            If Dmatrix.4 = 1 Then
                Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
            End If
            If Dmatrix.5 = 1 Then
                Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
            End If
            If Dmatrix.6 = 1 Then
                Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
            End If
            If Dmatrix.7 = 1 Then
                Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
            End If
        Next It
    End If

```

```

    Incr Aktpos
    Bildspeicher(aktpos) = 0

    If Dmatrix.0 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
    End If
    If Dmatrix.1 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
    End If
    If Dmatrix.2 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
    End If
    If Dmatrix.3 = 1 Then
        Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
    End If

    Aktpos = Aktpos - 9
    Incr Iz
Next It
End If

If Zeile = 2 Then
    For It = 1 To 8
        Dmatrix = Lookup(iz , Zeichensatz)
        Incr Aktpos
        Incr Aktpos
        Bildspeicher(aktpos) = 0
        If Dmatrix.4 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.5 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
        End If
        If Dmatrix.6 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
        End If
        If Dmatrix.7 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
        End If

        Incr Aktpos
        Bildspeicher(aktpos) = 0

        If Dmatrix.0 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B01000000
        End If
        If Dmatrix.1 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00010000
        End If
        If Dmatrix.2 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000100
        End If
        If Dmatrix.3 = 1 Then
            Bildspeicher(aktpos) = Bildspeicher(aktpos) Or &B00000001
        End If

        Aktpos = Aktpos - 11
        Incr Iz
    Next It
End If

    Aktpos = Aktpos - 8
End Sub

'Zeichensatz 8x8 vertikal
Zeichensatz:
Data &H00 , &H00 ' &H00
Data &H7E , &H81 , &HA9 , &H8D , &H8D , &HA9 , &H81 , &H7E ' &H01
Data &H7E , &HFF , &HD7 , &HF3 , &HF3 , &HD7 , &HFF , &H7E ' &H02
Data &H70 , &HF8 , &HFC , &H7E , &HFC , &HF8 , &H70 , &H00 ' &H03
Data &H10 , &H38 , &H7C , &HFE , &H7C , &H38 , &H10 , &H00 ' &H04
Data &H1C , &H5C , &HF9 , &HFF , &HF9 , &H5C , &H1C , &H00 ' &H05
Data &H08 , &H1C , &H3D , &HFF , &H3D , &H1C , &H08 , &H00 ' &H06

```

Data	&H00	, &H00	' &H07						
Data	&H00	, &H00	' &H08						
Data	&H00	, &H00	' &H09						
Data	&H00	, &H00	' &H0a						
Data	&H0E	, &H1F	, &H11	, &H11	, &HBF	, &HFE	, &HE0	, &HF0	' &H0b
Data	&H00	, &H72	, &HFA	, &H8F	, &H8F	, &HFA	, &H72	, &H00	' &H0c
Data	&H00	, &H00	' &H0d						
Data	&H03	, &HFF	, &HFE	, &HA0	, &HA0	, &HA6	, &HFE	, &HFC	' &H0e
Data	&H99	, &H5A	, &H3C	, &HE7	, &HE7	, &H3C	, &H5A	, &H99	' &H0f
Data	&HFE	, &H7C	, &H7C	, &H38	, &H38	, &H10	, &H10	, &H00	' &H10
Data	&H10	, &H10	, &H38	, &H38	, &H7C	, &H7C	, &HFE	, &H00	' &H11
Data	&H00	, &H24	, &H66	, &HFF	, &HFF	, &H66	, &H24	, &H00	' &H12
Data	&H00	, &HFA	, &HFA	, &H00	, &H00	, &HFA	, &HFA	, &H00	' &H13
Data	&H60	, &HF0	, &H90	, &HFE	, &HFE	, &H80	, &HFE	, &HFE	' &H14
Data	&H5B	, &HFD	, &HA5	, &HA5	, &HBF	, &H9A	, &HC0	, &H40	' &H15
Data	&H00	, &H0E	, &H00	' &H16					
Data	&H01	, &H29	, &H6D	, &HFF	, &HFF	, &H6D	, &H29	, &H01	' &H17
Data	&H00	, &H20	, &H60	, &HFE	, &HFE	, &H60	, &H20	, &H00	' &H18
Data	&H00	, &H08	, &H0C	, &HFE	, &HFE	, &H0C	, &H08	, &H00	' &H19
Data	&H10	, &H10	, &H10	, &H54	, &H7C	, &H38	, &H10	, &H00	' &H1a
Data	&H10	, &H38	, &H7C	, &H54	, &H10	, &H10	, &H10	, &H00	' &H1b
Data	&H3C	, &H3C	, &H04	, &H00	' &H1c				
Data	&H10	, &H38	, &H7C	, &H10	, &H10	, &H7C	, &H38	, &H10	' &H1d
Data	&H0C	, &H1C	, &H3C	, &H7C	, &H7C	, &H3C	, &H1C	, &H0C	' &H1e
Data	&H60	, &H70	, &H78	, &H7C	, &H7C	, &H78	, &H70	, &H60	' &H1f
Data	&H00	, &H00	' &H20						
Data	&H00	, &H60	, &HFA	, &HFA	, &H60	, &H00	, &H00	, &H00	' &H21
Data	&H00	, &HE0	, &HE0	, &H00	, &HE0	, &HE0	, &H00	, &H00	' &H22
Data	&H28	, &HFE	, &HFE	, &H28	, &HFE	, &HFE	, &H28	, &H00	' &H23
Data	&H24	, &H74	, &HD6	, &HD6	, &H5C	, &H48	, &H00	, &H00	' &H24
Data	&H62	, &H66	, &H0C	, &H18	, &H30	, &H66	, &H46	, &H00	' &H25
Data	&H0C	, &H5E	, &HF2	, &HBA	, &HEC	, &H5E	, &H12	, &H00	' &H26
Data	&H20	, &HE0	, &HC0	, &H00	' &H27				
Data	&H00	, &H38	, &H7C	, &HC6	, &H82	, &H00	, &H00	, &H00	' &H28
Data	&H00	, &H82	, &HC6	, &H7C	, &H38	, &H00	, &H00	, &H00	' &H29
Data	&H10	, &H54	, &H7C	, &H38	, &H38	, &H7C	, &H54	, &H10	' &H2a
Data	&H10	, &H10	, &H7C	, &H7C	, &H10	, &H10	, &H00	, &H00	' &H2b
Data	&H00	, &H05	, &H07	, &H06	, &H00	, &H00	, &H00	, &H00	' &H2c
Data	&H10	, &H00	, &H00	' &H2d					
Data	&H00	, &H00	, &H06	, &H06	, &H00	, &H00	, &H00	, &H00	' &H2e
Data	&H06	, &H0C	, &H18	, &H30	, &H60	, &HC0	, &H80	, &H00	' &H2f
Data	&H7C	, &HFE	, &H9A	, &HB2	, &HFE	, &H7C	, &H00	, &H00	' &H30
Data	&H42	, &H42	, &HFE	, &HFE	, &H02	, &H02	, &H00	, &H00	' &H31
Data	&H46	, &HCE	, &H9A	, &H92	, &HF6	, &H66	, &H00	, &H00	' &H32
Data	&H44	, &HC6	, &H92	, &H92	, &HFE	, &H6C	, &H00	, &H00	' &H33
Data	&H18	, &H38	, &H68	, &HC8	, &HFE	, &HFE	, &H08	, &H00	' &H34
Data	&HE4	, &HE6	, &HA2	, &HA2	, &HBE	, &H9C	, &H00	, &H00	' &H35
Data	&H3C	, &H7E	, &HD2	, &H92	, &H9E	, &HC0	, &H00	, &H00	' &H36
Data	&HC0	, &HC6	, &H8E	, &H98	, &HF0	, &HE0	, &H00	, &H00	' &H37
Data	&H6C	, &HFE	, &H92	, &H92	, &HFE	, &H6C	, &H00	, &H00	' &H38
Data	&H60	, &HF2	, &H92	, &H96	, &HFC	, &H78	, &H00	, &H00	' &H39
Data	&H00	, &H00	, &H36	, &H36	, &H00	, &H00	, &H00	, &H00	' &H3a
Data	&H00	, &H05	, &H37	, &H36	, &H00	, &H00	, &H00	, &H00	' &H3b
Data	&H10	, &H38	, &H6C	, &HC6	, &H82	, &H00	, &H00	, &H00	' &H3c
Data	&H28	, &H00	, &H00	' &H3d					
Data	&H00	, &H82	, &HC6	, &H6C	, &H38	, &H10	, &H00	, &H00	' &H3e
Data	&H40	, &HC0	, &H8A	, &H9A	, &HF0	, &H60	, &H00	, &H00	' &H3f
Data	&H7C	, &HFE	, &H82	, &HBA	, &HBA	, &HF8	, &H78	, &H00	' &H40
Data	&H3E	, &H7E	, &HC8	, &HC8	, &H7E	, &H3E	, &H00	, &H00	' &H41
Data	&H82	, &HFE	, &HFE	, &H92	, &H92	, &HFE	, &H6C	, &H00	' &H42
Data	&H38	, &H7C	, &HC6	, &H82	, &H82	, &HC6	, &H44	, &H00	' &H43
Data	&H82	, &HFE	, &HFE	, &H82	, &HC6	, &HFE	, &H38	, &H00	' &H44
Data	&H82	, &HFE	, &HFE	, &H92	, &HBA	, &H82	, &HC6	, &H00	' &H45
Data	&H82	, &HFE	, &HFE	, &H92	, &HB8	, &H80	, &HC0	, &H00	' &H46
Data	&H38	, &H7C	, &HC6	, &H82	, &H8A	, &HCE	, &H4E	, &H00	' &H47
Data	&HFE	, &HFE	, &H10	, &H10	, &HFE	, &HFE	, &H00	, &H00	' &H48
Data	&H00	, &H82	, &HFE	, &HFE	, &H82	, &H00	, &H00	, &H00	' &H49
Data	&H0C	, &H0E	, &H02	, &H82	, &HFE	, &HFC	, &H80	, &H00	' &H4a
Data	&H82	, &HFE	, &HFE	, &H10	, &H38	, &HEE	, &HC6	, &H00	' &H4b
Data	&H82	, &HFE	, &HFE	, &H82	, &H02	, &H06	, &H0E	, &H00	' &H4c
Data	&HFE	, &HFE	, &H60	, &H30	, &H60	, &HFE	, &HFE	, &H00	' &H4d
Data	&HFE	, &HFE	, &H60	, &H30	, &H18	, &HFE	, &HFE	, &H00	' &H4e
Data	&H38	, &H7C	, &HC6	, &H82	, &HC6	, &H7C	, &H38	, &H00	' &H4f
Data	&H82	, &HFE	, &HFE	, &H92	, &H90	, &HF0	, &H60	, &H00	' &H50
Data	&H78	, &HFC	, &H84	, &H8E	, &HFE	, &H7A	, &H00	, &H00	' &H51
Data	&H82	, &HFE	, &HFE	, &H98	, &H9C	, &HF6	, &H62	, &H00	' &H52
Data	&H64	, &HE6	, &HB2	, &H9A	, &HDE	, &H4C	, &H00	, &H00	' &H53

Data	&HC0	,	&H82	,	&HFE	,	&HFE	,	&H82	,	&HC0	,	&H00	,	&H00	'	&H54
Data	&HFE	,	&HFE	,	&H02	,	&H02	,	&HFE	,	&HFE	,	&H00	,	&H00	'	&H55
Data	&HF8	,	&HFC	,	&H06	,	&H06	,	&HFC	,	&HF8	,	&H00	,	&H00	'	&H56
Data	&HFE	,	&HFE	,	&H0C	,	&H18	,	&H0C	,	&HFE	,	&HFE	,	&H00	'	&H57
Data	&HC6	,	&HEE	,	&H38	,	&H10	,	&H38	,	&HEE	,	&HC6	,	&H00	'	&H58
Data	&HE0	,	&HF2	,	&H1E	,	&H1E	,	&HF2	,	&HE0	,	&H00	,	&H00	'	&H59
Data	&HE6	,	&HCE	,	&H9A	,	&HB2	,	&HE2	,	&HC6	,	&H8E	,	&H00	'	&H5a
Data	&H00	,	&HFE	,	&HFE	,	&H82	,	&H82	,	&H00	,	&H00	,	&H00	'	&H5b
Data	&H80	,	&HC0	,	&H60	,	&H30	,	&H18	,	&H0C	,	&H06	,	&H00	'	&H5c
Data	&H00	,	&H82	,	&H82	,	&HFE	,	&HFE	,	&H00	,	&H00	,	&H00	'	&H5d
Data	&H10	,	&H30	,	&H60	,	&HC0	,	&H60	,	&H30	,	&H10	,	&H00	'	&H5e
Data	&H01	,	&H01	'	&H5f												
Data	&H00	,	&H00	,	&HC0	,	&HE0	,	&H20	,	&H00	,	&H00	,	&H00	'	&H60
Data	&H04	,	&H2E	,	&H2A	,	&H2A	,	&H3C	,	&H1E	,	&H02	,	&H00	'	&H61
Data	&H82	,	&HFC	,	&HFE	,	&H22	,	&H22	,	&H3E	,	&H1C	,	&H00	'	&H62
Data	&H1C	,	&H3E	,	&H22	,	&H22	,	&H36	,	&H14	,	&H00	,	&H00	'	&H63
Data	&H0C	,	&H1E	,	&H12	,	&H92	,	&HFC	,	&HFE	,	&H02	,	&H00	'	&H64
Data	&H1C	,	&H3E	,	&H2A	,	&H2A	,	&H3A	,	&H18	,	&H00	,	&H00	'	&H65
Data	&H12	,	&H7E	,	&HFE	,	&H92	,	&HC0	,	&H40	,	&H00	,	&H00	'	&H66
Data	&H19	,	&H3D	,	&H25	,	&H25	,	&H1F	,	&H3E	,	&H20	,	&H00	'	&H67
Data	&H82	,	&HFE	,	&HFE	,	&H10	,	&H20	,	&H3E	,	&H1E	,	&H00	'	&H68
Data	&H00	,	&H22	,	&HBE	,	&HBE	,	&H02	,	&H00	,	&H00	,	&H00	'	&H69
Data	&H02	,	&H23	,	&H21	,	&HBF	,	&HBE	,	&H00	,	&H00	,	&H00	'	&H6a
Data	&H82	,	&HFE	,	&HFE	,	&H08	,	&H1C	,	&H36	,	&H22	,	&H00	'	&H6b
Data	&H00	,	&H82	,	&HFE	,	&HFE	,	&H02	,	&H00	,	&H00	,	&H00	'	&H6c
Data	&H3E	,	&H3E	,	&H30	,	&H18	,	&H30	,	&H3E	,	&H1E	,	&H00	'	&H6d
Data	&H3E	,	&H3E	,	&H20	,	&H20	,	&H3E	,	&H1E	,	&H00	,	&H00	'	&H6e
Data	&H1C	,	&H3E	,	&H22	,	&H22	,	&H3E	,	&H1C	,	&H00	,	&H00	'	&H6f
Data	&H21	,	&H3F	,	&H1F	,	&H25	,	&H24	,	&H3C	,	&H18	,	&H00	'	&H70
Data	&H18	,	&H3C	,	&H24	,	&H25	,	&H1F	,	&H3F	,	&H21	,	&H00	'	&H71
Data	&H22	,	&H3E	,	&H1E	,	&H22	,	&H38	,	&H18	,	&H00	,	&H00	'	&H72
Data	&H12	,	&H3A	,	&H2A	,	&H2A	,	&H2E	,	&H24	,	&H00	,	&H00	'	&H73
Data	&H00	,	&H20	,	&H7C	,	&HFE	,	&H22	,	&H24	,	&H00	,	&H00	'	&H74
Data	&H3C	,	&H3E	,	&H02	,	&H02	,	&H3C	,	&H3E	,	&H02	,	&H00	'	&H75
Data	&H38	,	&H3C	,	&H06	,	&H06	,	&H3C	,	&H38	,	&H00	,	&H00	'	&H76
Data	&H3C	,	&H3E	,	&H06	,	&H0C	,	&H06	,	&H3E	,	&H3C	,	&H00	'	&H77
Data	&H22	,	&H36	,	&H1C	,	&H08	,	&H1C	,	&H36	,	&H22	,	&H00	'	&H78
Data	&H39	,	&H3D	,	&H05	,	&H05	,	&H3F	,	&H3E	,	&H00	,	&H00	'	&H79
Data	&H32	,	&H26	,	&H2E	,	&H3A	,	&H32	,	&H26	,	&H00	,	&H00	'	&H7a
Data	&H10	,	&H10	,	&H7C	,	&HEE	,	&H82	,	&H82	,	&H00	,	&H00	'	&H7b
Data	&H00	,	&H00	,	&H00	,	&HEE	,	&HEE	,	&H00	,	&H00	,	&H00	'	&H7c
Data	&H82	,	&H82	,	&HEE	,	&H7C	,	&H10	,	&H10	,	&H00	,	&H00	'	&H7d
Data	&H40	,	&HC0	,	&H80	,	&HC0	,	&H40	,	&HC0	,	&H80	,	&H00	'	&H7e
Data	&H1E	,	&H3E	,	&H62	,	&HC2	,	&H62	,	&H3E	,	&H1E	,	&H00	'	&H7f
Data	&H78	,	&HFD	,	&H87	,	&H86	,	&HCC	,	&H48	,	&H00	,	&H00	'	&H80
Data	&H5C	,	&H5E	,	&H02	,	&H02	,	&H5E	,	&H5E	,	&H02	,	&H00	'	&H81
Data	&H1C	,	&H3E	,	&H6A	,	&HEA	,	&HBA	,	&H18	,	&H00	,	&H00	'	&H82
Data	&H40	,	&HC4	,	&HAE	,	&HAA	,	&HAA	,	&HBE	,	&HDE	,	&H42	'	&H83
Data	&H84	,	&HAE	,	&H2A	,	&H2A	,	&HBE	,	&H9E	,	&H02	,	&H00	'	&H84
Data	&H04	,	&HAE	,	&HEA	,	&H6A	,	&H3E	,	&H1E	,	&H02	,	&H00	'	&H85
Data	&H00	,	&H44	,	&HEE	,	&HAA	,	&HAA	,	&HFE	,	&H5E	,	&H02	'	&H86
Data	&H38	,	&H7D	,	&H47	,	&H46	,	&H6C	,	&H28	,	&H00	,	&H00	'	&H87
Data	&H40	,	&HDC	,	&HBE	,	&HAA	,	&HBA	,	&HBA	,	&HD8	,	&H40	'	&H88
Data	&H9C	,	&HBE	,	&H2A	,	&H2A	,	&HBA	,	&H98	,	&H00	,	&H00	'	&H89
Data	&H1C	,	&HBE	,	&HEA	,	&H6A	,	&H3A	,	&H18	,	&H00	,	&H00	'	&H8a
Data	&H80	,	&HA2	,	&H3E	,	&H3E	,	&H82	,	&H80	,	&H00	,	&H00	'	&H8b
Data	&H40	,	&HC0	,	&HA2	,	&HBE	,	&HBE	,	&HC2	,	&H40	,	&H00	'	&H8c
Data	&H00	,	&HA2	,	&HFE	,	&H7E	,	&H02	,	&H00	,	&H00	,	&H00	'	&H8d
Data	&H9E	,	&HBE	,	&H64	,	&H64	,	&HBE	,	&H9E	,	&H00	,	&H00	'	&H8e
Data	&H0E	,	&H5E	,	&HB4	,	&HB4	,	&H5E	,	&H0E	,	&H00	,	&H00	'	&H8f
Data	&H22	,	&H3E	,	&H7E	,	&HEA	,	&HAA	,	&H22	,	&H00	,	&H00	'	&H90
Data	&H04	,	&H2E	,	&H2A	,	&H2A	,	&H3E	,	&H3E	,	&H2A	,	&H2A	'	&H91
Data	&H3E	,	&H7E	,	&HD0	,	&H90	,	&HFE	,	&HFE	,	&H92	,	&H00	'	&H92
Data	&H4C	,	&HDE	,	&H92	,	&H92	,	&HDE	,	&H4C	,	&H00	,	&H00	'	&H93
Data	&H4C	,	&H5E	,	&H12	,	&H12	,	&H5E	,	&H4C	,	&H00	,	&H00	'	&H94
Data	&H0C	,	&H9E	,	&HD2	,	&H52	,	&H1E	,	&H0C	,	&H00	,	&H00	'	&H95
Data	&H5C	,	&HDE	,	&H82	,	&H82	,	&HDE	,	&H5E	,	&H02	,	&H00	'	&H96
Data	&H1C	,	&H9E	,	&HC2	,	&H42	,	&H1E	,	&H1E	,	&H02	,	&H00	'	&H97
Data	&H5D	,	&H5D	,	&H05	,	&H05	,	&H5F	,	&H5E	,	&H00	,	&H00	'	&H98
Data	&H9C	,	&HBE	,	&H22	,	&H22	,	&H22	,	&HBE	,	&H9C	,	&H00	'	&H99
Data	&HBC	,	&HBE	,	&H02	,	&H02	,	&HBE	,	&HBC	,	&H00	,	&H00	'	&H9a
Data	&H1C	,	&H3E	,	&H26	,	&H2A	,	&H32	,	&H3E	,	&H1C	,	&H00	'	&H9b
Data	&H16	,	&H7E	,	&HFE	,	&H92	,	&HC2	,	&H6E	,	&H04	,	&H00	'	&H9c
Data	&H3A	,	&H7C	,	&HCE	,	&H92	,	&HE6	,	&H7C	,	&HB8	,	&H00	'	&H9d
Data	&H22	,	&H36	,	&H1C	,	&H1C	,	&H36	,	&H22	,	&H00	,	&H00	'	&H9e
Data	&H02	,	&H13	,	&H11	,	&H7F	,	&HFE	,	&H90	,	&HD0	,	&H40	'	&H9f
Data	&H04	,	&H2E	,	&H6A	,	&HEA	,	&HBE	,	&H1E	,	&H02	,	&H00	'	&Ha0

Data	&H00	,	&H22	,	&H7E	,	&HFE	,	&H82	,	&H00	'	&Ha1								
Data	&H0C	,	&H1E	,	&H12	,	&H52	,	&HDE	,	&H8C	,	&H00	,	&H00	,	&H00	,	&H00	'	&Ha2
Data	&H1C	,	&H1E	,	&H02	,	&H42	,	&HDE	,	&H9E	,	&H02	,	&H00	,	&H00	,	&H00	'	&Ha3
Data	&H5E	,	&HDE	,	&H90	,	&HD0	,	&H5E	,	&HCE	,	&H80	,	&H00	,	&H00	,	&H00	'	&Ha4
Data	&H5E	,	&HDE	,	&H98	,	&HCC	,	&H5E	,	&HDE	,	&H80	,	&H00	,	&H00	,	&H00	'	&Ha5
Data	&H00	,	&H64	,	&HF4	,	&H94	,	&HF4	,	&HF4	,	&H14	,	&H00	,	&H00	,	&H00	'	&Ha6
Data	&H00	,	&H64	,	&HF4	,	&H94	,	&H94	,	&HF4	,	&H64	,	&H00	,	&H00	,	&H00	'	&Ha7
Data	&H0C	,	&H1E	,	&HB2	,	&HA2	,	&H06	,	&H04	,	&H00	,	&H00	,	&H00	,	&H00	'	&Ha8
Data	&H38	,	&H44	,	&HBE	,	&HD2	,	&HDA	,	&HA6	,	&H44	,	&H38	,	&H00	,	&H00	'	&Ha9
Data	&H10	,	&H10	,	&H10	,	&H10	,	&H1C	,	&H1C	,	&H00	,	&H00	,	&H00	,	&H00	'	&Haa
Data	&H86	,	&HFC	,	&HF8	,	&H33	,	&H77	,	&HD5	,	&H9D	,	&H09	,	&H00	,	&H00	'	&Hab
Data	&H86	,	&HFC	,	&HF8	,	&H32	,	&H66	,	&HCE	,	&H9B	,	&H1F	,	&H00	,	&H00	'	&Hac
Data	&H00	,	&H00	,	&H06	,	&H5F	,	&H5F	,	&H06	,	&H00	,	&H00	,	&H00	,	&H00	'	&Had
Data	&H10	,	&H38	,	&H6C	,	&H44	,	&H10	,	&H38	,	&H6C	,	&H44	,	&H00	,	&H00	'	&Hae
Data	&H44	,	&H6C	,	&H38	,	&H10	,	&H44	,	&H6C	,	&H38	,	&H10	,	&H00	,	&H00	'	&Haf
Data	&H55	,	&H00	,	&HAA	,	&H00	,	&H55	,	&H00	,	&HAA	,	&H00	,	&H00	,	&H00	'	&Hb0
Data	&H55	,	&HAA	,	&H00	,	&H00	'	&Hb1												
Data	&HAA	,	&HFF	,	&H55	,	&HFF	,	&HAA	,	&HFF	,	&H55	,	&HFF	,	&H00	,	&H00	'	&Hb2
Data	&H00	,	&H00	,	&H00	,	&HFF	,	&HFF	,	&H00	'	&Hb3								
Data	&H08	,	&H08	,	&H08	,	&HFF	,	&HFF	,	&H00	'	&Hb4								
Data	&H0E	,	&H1E	,	&H34	,	&H74	,	&HDE	,	&H8E	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hb5
Data	&H4E	,	&H9E	,	&HB4	,	&HB4	,	&H9E	,	&H4E	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hb6
Data	&H8E	,	&HDE	,	&H74	,	&H34	,	&H1E	,	&H0E	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hb7
Data	&H38	,	&H44	,	&HBA	,	&HAA	,	&HAA	,	&H82	,	&H44	,	&H38	,	&H00	,	&H00	'	&Hb8
Data	&H28	,	&H28	,	&HEF	,	&HEF	,	&H00	,	&HFF	,	&HFF	,	&H00	,	&H00	,	&H00	'	&Hb9
Data	&H00	,	&H00	,	&HFF	,	&HFF	,	&H00	,	&HFF	,	&HFF	,	&H00	,	&H00	,	&H00	'	&Hba
Data	&H28	,	&H28	,	&H2F	,	&H2F	,	&H20	,	&H3F	,	&H3F	,	&H00	,	&H00	,	&H00	'	&Hbb
Data	&H28	,	&H28	,	&HE8	,	&HE8	,	&H08	,	&HF8	,	&HF8	,	&H00	,	&H00	,	&H00	'	&Hbc
Data	&H18	,	&H3C	,	&H24	,	&HE7	,	&HE7	,	&H24	,	&H24	,	&H00	,	&H00	,	&H00	'	&Hbd
Data	&HD4	,	&HF4	,	&H3F	,	&H3F	,	&HF4	,	&HD4	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hbe
Data	&H08	,	&H08	,	&H08	,	&H0F	,	&H0F	,	&H00	'	&Hbf								
Data	&H00	,	&H00	,	&H00	,	&HF8	,	&HF8	,	&H08	'	&Hc0								
Data	&H08	,	&H08	,	&H08	,	&HF8	,	&HF8	,	&H08	'	&Hc1								
Data	&H08	,	&H08	,	&H08	,	&H0F	,	&H0F	,	&H08	'	&Hc2								
Data	&H00	,	&H00	,	&H00	,	&HFF	,	&HFF	,	&H08	'	&Hc3								
Data	&H08	,	&H08	'	&Hc4																
Data	&H08	,	&H08	,	&H08	,	&HFF	,	&HFF	,	&H08	'	&Hc5								
Data	&H44	,	&HEE	,	&HAA	,	&HEA	,	&H7E	,	&HDE	,	&H82	,	&H00	,	&H00	,	&H00	'	&Hc6
Data	&H4E	,	&HDE	,	&HB4	,	&HF4	,	&H5E	,	&HCE	,	&H80	,	&H00	,	&H00	,	&H00	'	&Hc7
Data	&H00	,	&H00	,	&HF8	,	&HF8	,	&H08	,	&HE8	,	&HE8	,	&H28	,	&H00	,	&H00	'	&Hc8
Data	&H00	,	&H00	,	&H3F	,	&H3F	,	&H20	,	&H2F	,	&H2F	,	&H28	,	&H00	,	&H00	'	&Hc9
Data	&H28	,	&H28	,	&HE8	,	&HE8	,	&H08	,	&HE8	,	&HE8	,	&H28	,	&H00	,	&H00	'	&Hca
Data	&H28	,	&H28	,	&H2F	,	&H2F	,	&H20	,	&H2F	,	&H2F	,	&H28	,	&H00	,	&H00	'	&Hcb
Data	&H00	,	&H00	,	&HFF	,	&HFF	,	&H00	,	&HEF	,	&HEF	,	&H28	,	&H00	,	&H00	'	&Hcc
Data	&H28	,	&H00	,	&H00	'	&Hcd														
Data	&H28	,	&H28	,	&HEF	,	&HEF	,	&H00	,	&HEF	,	&HEF	,	&H28	,	&H00	,	&H00	'	&Hce
Data	&H66	,	&H3C	,	&H3C	,	&H24	,	&H3C	,	&H3C	,	&H66	,	&H00	,	&H00	,	&H00	'	&Hcf
Data	&HA0	,	&HE4	,	&H4E	,	&HEA	,	&HBE	,	&H1C	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd0
Data	&H92	,	&HFE	,	&HFE	,	&H92	,	&HC6	,	&HFE	,	&H38	,	&H00	,	&H00	,	&H00	'	&Hd1
Data	&H62	,	&HBE	,	&HBE	,	&HAA	,	&HAA	,	&H62	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd2
Data	&HA2	,	&HBE	,	&H3E	,	&H2A	,	&HAA	,	&HA2	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd3
Data	&H22	,	&HBE	,	&HFE	,	&H6A	,	&H2A	,	&H22	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd4
Data	&H50	,	&H70	,	&H10	,	&H00	'	&Hd5												
Data	&H00	,	&H22	,	&H7E	,	&HFE	,	&HA2	,	&H00	'	&Hd6								
Data	&H40	,	&HA2	,	&HBE	,	&HBE	,	&HA2	,	&H40	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd7
Data	&H80	,	&HA2	,	&H3E	,	&H3E	,	&HA2	,	&H80	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hd8
Data	&H08	,	&H08	,	&H08	,	&HF8	,	&HF8	,	&H00	'	&Hd9								
Data	&H00	,	&H00	,	&H00	,	&H0F	,	&H0F	,	&H08	'	&Hda								
Data	&HFF	,	&H00	,	&H00	'	&Hdb														
Data	&H0F	,	&H00	,	&H00	'	&Hdc														
Data	&H00	,	&H00	,	&H00	,	&HEE	,	&HEE	,	&H00	'	&Hdd								
Data	&H00	,	&HA2	,	&HFE	,	&H7E	,	&H22	,	&H00	'	&Hde								
Data	&HF0	,	&H00	,	&H00	'	&Hdf														
Data	&H1C	,	&H3E	,	&H62	,	&HE2	,	&HA2	,	&H3E	,	&H1C	,	&H00	,	&H00	,	&H00	'	&He0
Data	&H3F	,	&H7F	,	&H54	,	&H54	,	&H7C	,	&H28	,	&H00	,	&H00	,	&H00	,	&H00	'	&He1
Data	&H5C	,	&HBE	,	&HA2	,	&HA2	,	&HA2	,	&HBE	,	&H5C	,	&H00	,	&H00	,	&H00	'	&He2
Data	&H1C	,	&H3E	,	&HA2	,	&HE2	,	&H62	,	&H3E	,	&H1C	,	&H00	,	&H00	,	&H00	'	&He3
Data	&H4C	,	&HDE	,	&H92	,	&HD2	,	&H5E	,	&HCC	,	&H80	,	&H00	,	&H00	,	&H00	'	&He4
Data	&H5C	,	&HFE	,	&HA2	,	&HE2	,	&H62	,	&HFE	,	&H9C	,	&H00	,	&H00	,	&H00	'	&He5
Data	&H01	,	&H7F	,	&H7E	,	&H04	,	&H04	,	&H7C	,	&H78	,	&H00	,	&H00	,	&H00	'	&He6
Data	&H42	,	&H7E	,	&H7E	,	&H2A	,	&H38	,	&H10	,	&H00	,	&H00	,	&H00	,	&H00	'	&He7
Data	&H82	,	&HFE	,	&HFE	,	&HAA	,	&H28	,	&H38	,	&H10	,	&H00	,	&H00	,	&H00	'	&He8
Data	&H3C	,	&H3E	,	&H42	,	&HC2	,	&HBE	,	&H3C	,	&H00	,	&H00	,	&H00	,	&H00	'	&He9
Data	&H5C	,	&H9E	,	&H82	,	&H82	,	&H9E	,	&H5C	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hea
Data	&H3C	,	&HBE	,	&HC2	,	&H42	,	&H3E	,	&H3C	,	&H00	,	&H00	,	&H00	,	&H00	'	&Heb
Data	&H1D	,	&H1D	,	&H45	,	&HC5	,	&H9F	,	&H1E	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hec
Data	&H30	,	&H3A	,	&H4E	,	&HCE	,	&HBA	,	&H30	,	&H00	,	&H00	,	&H00	,	&H00	'	&Hed

```
Data &H40 , &H40 , &H40 , &H40 , &H40 , &H40 , &H00 , &H00 ' &Hee
Data &H00 , &H00 , &H40 , &HC0 , &H80 , &H00 , &H00 , &H00 ' &Hef
Data &H08 , &H08 , &H08 , &H08 , &H08 , &H08 , &H00 , &H00 ' &Hf0
Data &H22 , &H22 , &HFA , &HFA , &H22 , &H22 , &H00 , &H00 ' &Hf1
Data &H14 , &H14 , &H14 , &H14 , &H14 , &H14 , &H00 , &H00 ' &Hf2
Data &H8E , &HAC , &HF8 , &H32 , &H66 , &HCE , &H9B , &H1F ' &Hf3
Data &H60 , &HF0 , &H90 , &HFE , &HFE , &H80 , &HFE , &HFE ' &Hf4
Data &H5B , &HFD , &HA5 , &HA5 , &HBF , &H9A , &HC0 , &H40 ' &Hf5
Data &H10 , &H10 , &HD6 , &HD6 , &H10 , &H10 , &H00 , &H00 ' &Hf6
Data &H00 , &H01 , &H03 , &H02 , &H00 , &H00 , &H00 , &H00 ' &Hf7
Data &H00 , &H60 , &HF0 , &H90 , &HF0 , &H60 , &H00 , &H00 ' &Hf8
Data &H40 , &H40 , &H00 , &H00 , &H40 , &H40 , &H00 , &H00 ' &Hf9
Data &H00 , &H00 , &H00 , &H08 , &H08 , &H00 , &H00 , &H00 ' &Hfa
Data &H00 , &H48 , &HC8 , &HF8 , &HF8 , &H08 , &H08 , &H00 ' &Hfb
Data &H00 , &H88 , &HA8 , &HA8 , &HF8 , &HF8 , &H50 , &H00 ' &Hfc
Data &H00 , &H98 , &HB8 , &HA8 , &HE8 , &H48 , &H00 , &H00 ' &Hfd
Data &H00 , &H00 , &H3C , &H3C , &H3C , &H3C , &H00 , &H00 ' &Hfe
Data &H00 , &H00 ' &Hff Test
```

**Wenn ihr weitere schöne Programme oder Animationen für die Propellerclock geschrieben habt, sendet sie mir bitte. Ich packe sie dann mit auf die CD, andere Bastler werden sich freuen.**

**Sendet eure Programme an [support@robotikhardware.de](mailto:support@robotikhardware.de) oder postet Sie einfach frei im Roboternetz.de.**

## Video

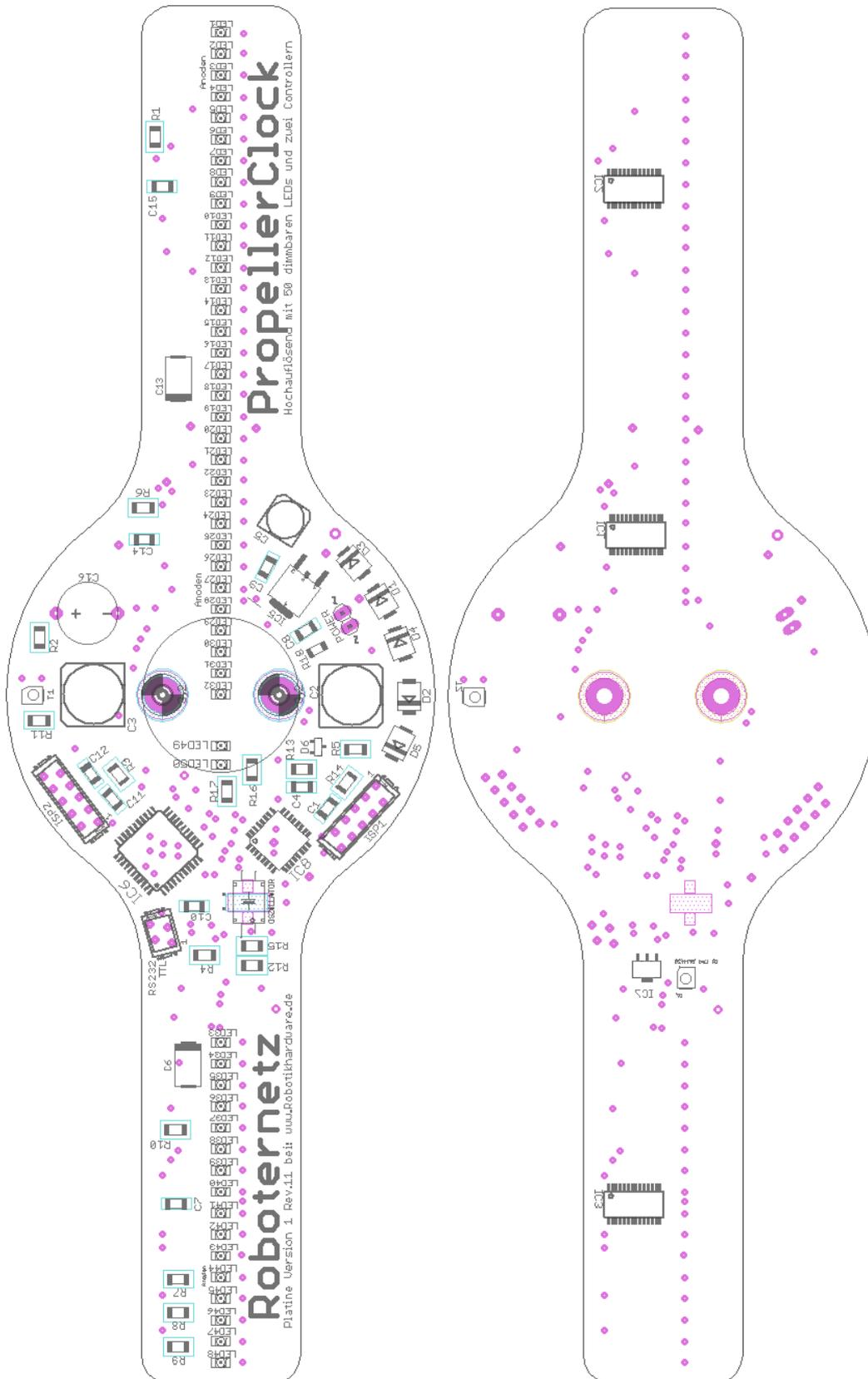
**Ein Video zu unserer Propellerclock mit diesem Beispielprogramm findet ihr hier:**

<http://www.youtube.com/watch?v=VHujWnEULzI>

bzw. hier:

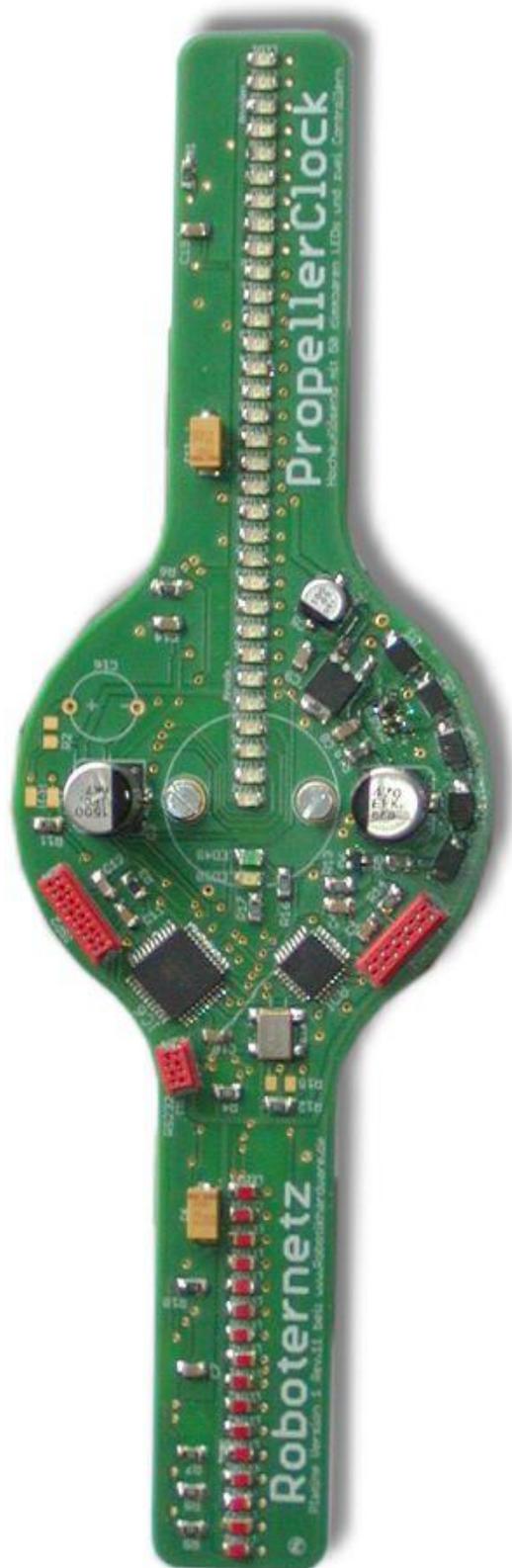
<http://www.youtube.com/user/Roboternetz>

## Der Bestückungsplan

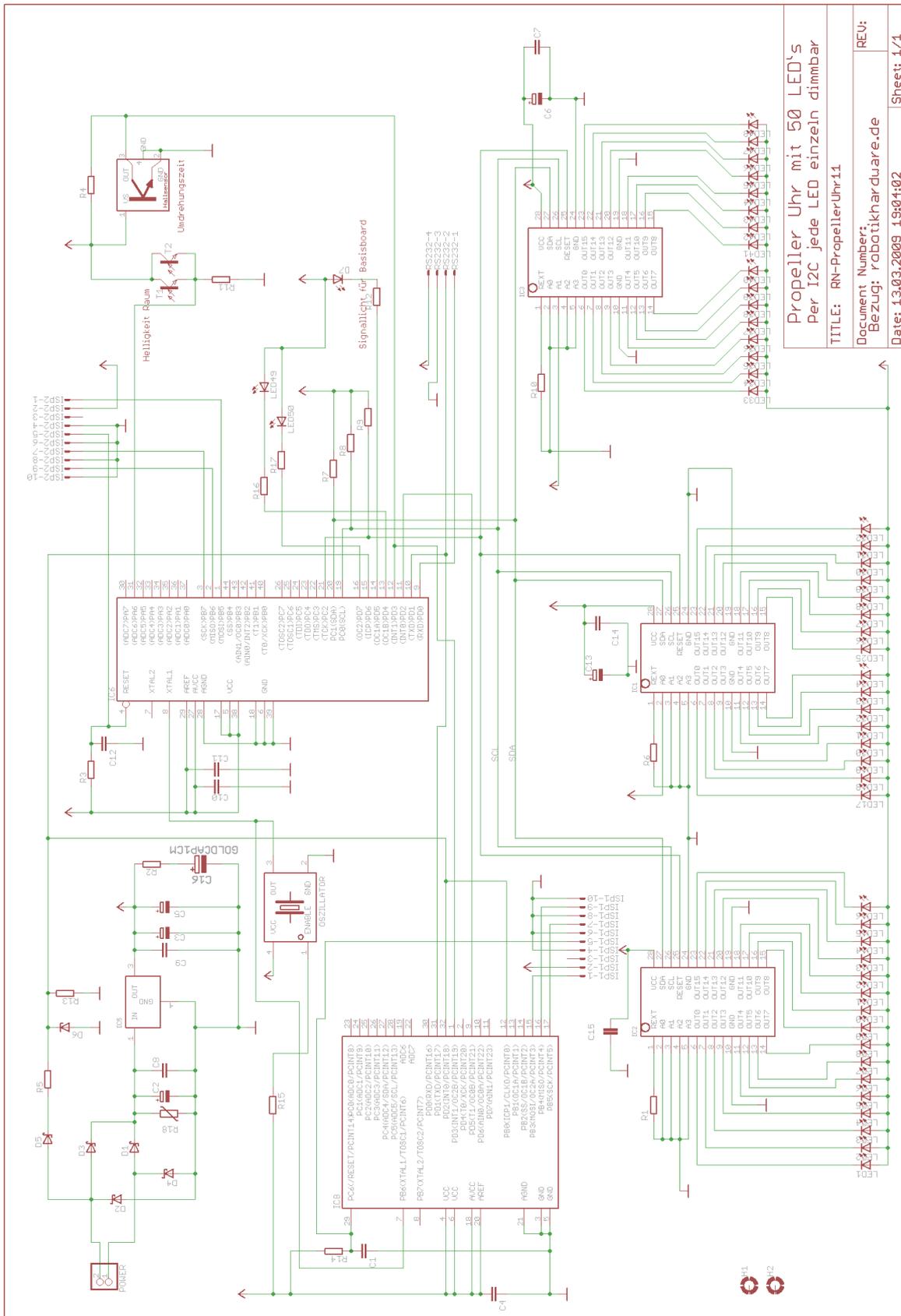


## Manuell bestückte Platine

Hier mit normalen LED's bestückt. Man kann auch hellere LED's bestücken!



## Der Schaltplan



Propeller Uhr mit 50 LED's  
 Per I2C jede LED einzeln dimmbar  
 TITLE: RN-PropellerUhr11  
 Document Number:  
 Bezug: robotikhardware.de  
 Date: 13.03.2009 19:04:02  
 REV:  
 Sheet: 1/1

Auf CD ist der Schaltplan noch einmal in größerer Darstellung als Bild vorhanden

## Bauteileliste und Bezugsquellen

Menge	Wert	Device	Bauteile	Anmerkung
32		LEDSMT1206	LED1 bis LED32	Blaue LED`s möglichst hell
		Entweder normale LED`s (1206 Bauform) oder OSRAM TOPLED, 224 mcd, blau (beide Bauformen passen)		Reichelt.de Artikel-Nr.: LB T67C
16		LEDSMT1206	LED33 bis LED48	Orange LED`s möglichst hell
		Entweder normale LED`s (1206 Bauform) oder OSRAM TOPLED, 355 mcd, orange (beide Bauformen passen)		Reichelt.de Artikel-Nr.: LO T676
1		LEDSMT1206	LED49	Gelbe LED möglichst hell
		Entweder normale LED`s (1206 Bauform) oder OSRAM TOPLED, 355 mcd, gelb (beide Bauformen passen)		Reichelt.de Artikel-Nr.: LY T676
1		LEDSMT1206	LED50	Rote LED möglichst hell
		Entweder normale LED`s (1206 Bauform) oder OSRAM TOPLED, 224 mcd, superrot (beide Bauformen passen)		Reichelt.de Artikel-Nr.: LS T67B
1		OSZILLATOR_AQ07050	OSZILLATOR	20 Mhz SMD Oszillator Erhältlich bei <a href="http://Robotikhardware.de">Robotikhardware.de</a>
1	4,7V zehnerdiode	ZENER-DIODE SOT23	D6	Reichelt
7	10k	R-EU_M1206	R3, R4, R7, R8, R9, R14, R15	Reichelt
1	20k	R-EU_M1206	R11	Reichelt
1	22k	R-EU_M1206	R5	Reichelt
1	100k	R-EU_M1206	R13	Reichelt
10	100n	C-EUC1206	C1, C4, C7, C8, C9, C10, C11, C12, C14, C15	Reichelt
1	100uF/6,3V	CPOL-EU153CLV-0605	C5	Elko Reichelt SMD Elko 100/6.3
3	200	R-EU_M1206	R12, R16, R17	Reichelt
1	470uF/25V	CPOL-EU153CLV-1014	C2	Elko Reichelt VF470/25 K-G
3	910 Ohm	R-EU_M1206	R1, R6, R10	Reichelt
1	1500uF/6,3V	CPOL-EU153CLV-1014	C3	Elko Reichelt VF 1500/6.3C-G
1	IR Led SFH420	SFH420_IRLED	D7	Fotodiode IR Reichelt
5	MBRS240	MBRS240	D1, D2, D3, D4, D5	Reichelt: MBRS 240 SMD
1	MEGA48/88/168-AU	MEGA168-AU	IC8	Atmel Mega168 20Mhz
1	MEGA644	MEGA644 20Mhz	IC6	Atmel Mega644 20Mhz
1	MICROMATCH-4	MICROMATCH-4	RS232	Micromatch Federleiste gerade 4p Reichelt MM FL 4G
2	MICROMATCH-10	MICROMATCH-10	ISP1, ISP2	Micromatch Federleiste gerade 10p Reichelt MM FL 10G
1	SFH320	SFH320	T2	Fototransistor Reichelt
2	SMD TAN.100/6,3	CPOL-EUSMCD	C6, C13	Tantal 100uF Reichelt Reichelt SMD TAN 100/6.3
3	TLC59116	TLC59116	IC1, IC2, IC3	<a href="http://TLC59116IPWR.Robotikhardware.de">TLC59116IPWR Robotikhardware.de</a>
1	TLE4905	TLE4905SOT89	IC7	Hallsensor <a href="http://TLE4905G.Robotikhardware.de">TLE4905G Robotikhardware.de</a> Erhältlich bei <a href="http://Robotikhardware.de">Robotikhardware.de</a>
1	VARISTORCN1206	VARISTORCN1206	R18	SMD Varistor 30V Reichelt VC 1206-30
1	mc7805	MC7805BDTG	IC5	SMD 5V/1A Spannungsregler Erhältlich bei <a href="http://robotikhardware.de">robotikhardware.de</a>
1		Platine		<a href="http://robotikhardware.de">robotikhardware.de</a>

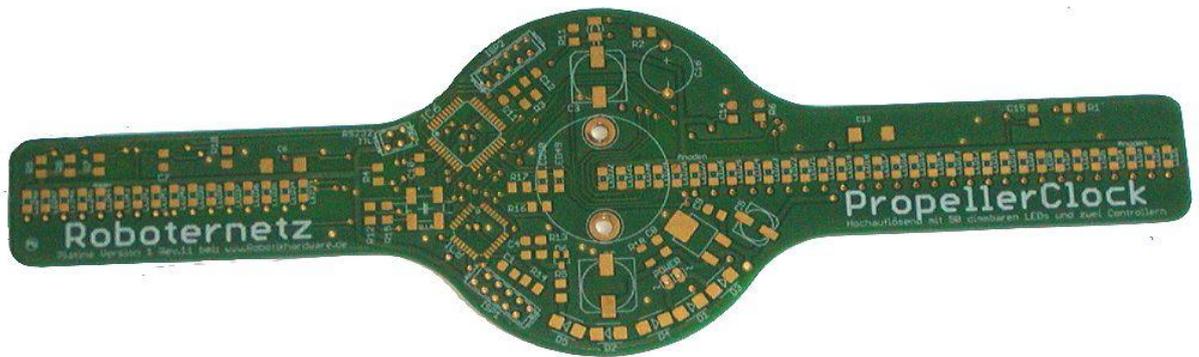
**Unbestückt lassen:**

- POWER - Hier kann wahlweise auch eine zweipolige Stiftleiste angelötet werden. Über Power Wird das Board mit Spannung versort (Gleich- oder Wechselspannung 7 - 15V)
- C16 Vorgesehen für Goldcap - falls man Board über Schleifer versorgt
- R2 Vorwiderstand für das optionale Goldcap
- T1 Der Fototransistor ist nur notwendig wenn die Raumhelligkeit gemessen werden soll Dies kann unter umständen jedoch auch über T2 auf der Rückseite erfolgen

Bezugsquelle für Platine:

[www.Robotikhardware.de](http://www.Robotikhardware.de)

Bei uns mit CD und Bascom Beispielprogrammen



## Video

<http://www.youtube.com/watch?v=VHujWnEULzI>

## Hinweise zur Haftung

Diese Dokumentation versteht sich nicht als genaue Bauanleitung sondern lediglich als Hilfestellung beim Bau einer Propelleruhr oder Propelleranzeige. Die komplette Mechanik, der Antrieb sowie die Art der Spannungsübertragung liegt im Ermessen des Bastlers der diese Schaltung aufbaut. Hierzu werden nur praktische Erfahrungswerte in der Doku genannt die den Aufbau erleichtern sollen. Aus diesem Grund kann keinerlei Haftung für Schäden übernommen werden, welche durch einen falschen oder ungeeigneten Aufbau entstehen. Wir garantieren lediglich das wie die Platine in ordnungsgemäßen und getesteten Zustand entsprechend dem Schaltplan ausliefern, solange Vorrat besteht! Alle anderen Ansprüche schließen wir aus. Fehlerhaft aufgebaute Boards unterliegen nicht der Gewährleistung und können leider nicht von uns repariert werden. Alle Platinen werden vor der Auslieferung elektronisch getestet!