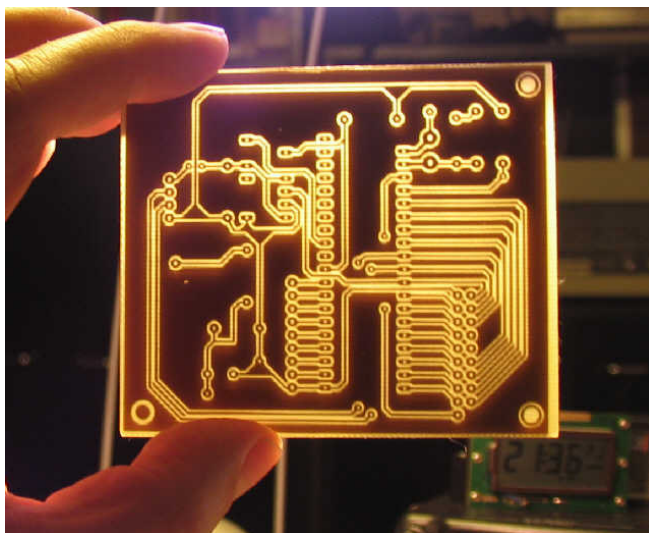


## Ein Highspeed-Controller für Sascha Ittner's ScopeClock

von Erhard Schemainda



Die Version 1.0 der Highspeed-Controller Platine

### Inhalt

- [Vorwort](#)
- [Die Schaltung](#)
- [Der Aufbau](#)
- [Das RS232-Interface](#)
- [Programmänderungen](#)
- [Die Programmierung](#)
- [Hardwareänderungen/-tuning](#)
- [Änderungsprotokoll](#)
- [Quellenverzeichnis](#)

### Vorwort

Die [ScopeClock](#) von Sascha Ittner aus [Jogis Röhrenbude](#) ist sicher ein interessantes Projekt. Nicht um eine weitere Variante einer Gebrauchsuhr herzustellen, sondern weil hier moderne Technik mit Röhrentechnik miteinander verbunden werden. Dies macht den besonderen Reiz dieses Projektes aus. Für diese Arbeit gebührt Sascha ein großer Dank. Seine Arbeit animierte auch mich, da auch bei mir seit über 20 Jahren eine kleine Oszillographenröhre (2BP1) samt zugehörigem HV-Trafo und HV-Platine herumlagen, diese zu neuem Leben zu erwecken. Der sehr klare und übersichtliche Aufbau der Schaltung macht eine Anpassung an andere Röhrenschaltungen leicht und garantiert eine hohe Nachbausicherheit.

Im Forum waren jedoch von einigen Nachbauern mehr oder weniger enttäuschte Stimmen zu vernehmen, die das 25 Hz Flimmern des Zifferblattes als störend empfanden. Dies reizte mich, mich mit der ScopeClock näher zu befassen. Nicht um etwas besser zu machen war meine Intention, denn das Bessere ist des Guten Feind, sondern einfach die Faszination dieser Technik.

Die Verfolgung des Bildaufbaues am Oszilloskop zeigte, daß für das Schreiben des Zifferblattes ca. 33 - 35 ms benötigt werden. D.h., daß nur nach jedem 2. externen Interrupt das Zifferblatt von Neuem geschrieben wird. Dies ergibt eben das 25 Hz Flimmern, das um so deutlicher wahrnehmbar wird, je heller man den Strahl aussteuert. Um hier Abhilfe zu schaffen, müßte man das Schreiben des Zifferblattes so verkürzen, daß es zwischen zwei externen Interrupts, also innerhalb von 20 ms vollständig abgeschlossen wird. Damit käme man dann auf eine Bildfolgefrequenz von 50 Hz, entsprechend der noch üblichen Fernseh-Norm.

Um dieses Ziel zu erreichen bieten sich grundsätzlich zwei Lösungswege an:

- das Programm so zu verändern, daß weniger Bildpunkte geschrieben werden, oder
- einen schnelleren Prozessor zu verwenden.

Der erste Weg bringt sicherlich eine deutliche Verschlechterung der Darstellungsqualität mit sich und ist in der Realisierung schwer abschätzbar und zeitaufwendig. Ich habe mich daher für den zweiten Weg entschlossen, da er mir einfacher und überschaubarer erschien.

Die Mindestanforderungen, die ein geeigneter Prozessor erfüllen sollte, waren:

- weitgehende 8051 Kompatibilität, um mit wenig Programmänderungen auszukommen,
- etwa 2 - 3-fach höhere Geschwindigkeit gegenüber dem AT89C4051,
- möglichst wenig Hardware-Aufwand zum Programmieren (Brennen) des Prozessors.

Nach relativ kurzer Recherche stieß ich bei meiner Suche auf den [DS89C420](#) von Dallas-Maxim, der alle Bedingungen erfüllte bzw.

die Forderung nach höherer Geschwindigkeit noch reichlich übertraf. Mit quasi der 12-fachen Taktgeschwindigkeit gegenüber dem 4051 ist er etwa 10 mal schneller als dieser. Damit hatte ich meine Wahl getroffen, ohne noch weitere Nachforschungen anzustellen. Ich habe mich für die 40pol Ausführung des DS89C420 entschieden, mit der genauen Bezeichnung: DS89C420-MCL. Zusätzlich bietet dieser Prozessor, u.a. mit 16K Programmspeicher (Flash) und zwei weiteren Ports, genügend Reserven für zusätzliche Erweiterungen (z.B.: Datumsanzeige, Weckeinrichtung, ... wie dies im Forum schon diskutiert wurde). Auch eine Portierung des Programms nach "C" wäre denkbar.

Den Ersatz des 4051 habe ich so geplant, daß keine wesentlichen Änderungen an der ScopeClock-Platine erforderlich werden. Der neue Prozessor soll dazu auf eine Zusatzplatine aufgebaut und über ein 20-pol. Flachbandkabel mit DIL-Stecker an die Fassung des 4051 angeschlossen werden.

## Die Schaltung

Die Abbildung zeigt den DS89C420 in der Minimalbeschaltung, wie sie in den Applikationen von Maxim-Dallas verwendet wird. Als Vorlage diente die Schaltung des Highspeed-Controller-Boards von Elektor, das in den Ausgaben Juni und September 2002 veröffentlicht worden ist. Jedem der sich etwas näher mit dem 89C420 beschäftigen will, wird die Lektüre dieser beiden Artikel empfohlen. Man kann diese gegen eine geringe Gebühr vom [Elektor-Verlag](#) beziehen.

Der 89C420 kennt zwei Betriebszustände. Den Programmablauf-Modus und den Monitor-Modus. Im Programmablauf-Modus wird das im Programmspeicher befindliche Programm ausgeführt. Im Monitor-Modus ist das integrierte Monitorprogramm aktiv und kommuniziert über die RS232-Schnittstelle mit einem angeschlossenen PC. In diesem Modus wird u.a. das Laden eines Programms in den Programmspeicher (Brennen) durchgeführt. Die Schaltung in den gewünschten Modus erfolgt über die Pins EA (31), PSEN (29) und Reset (9).

Modus	Reset	$\overline{EA}$	$\overline{PSEN}$
Programmablauf	0	1	1
Monitor	1	0	0

Die Umschaltung zwischen den Modi erfolgt mittels des DTR Signals vom PC über die RS232-Schnittstelle, wobei das IC2 die erforderlichen Signallevel an den Pins erzeugt. Mit dem Jumper JP1 unterbricht man die DTR-Verbindung. Dies ist nur eine Vorsichtsmaßnahme, wenn man die ScopeClock bei angeschlossenem PC laufen läßt. Man verhindert damit ein ungewolltes Schalten in den Monitor-Modus.

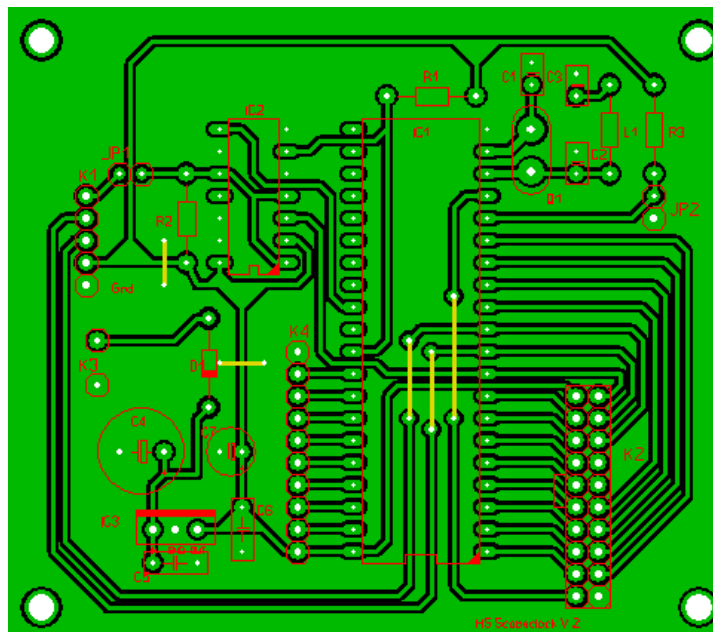
Der 89C420 Prozessor wird, wie der 89C4051, ebenfalls mit einer Taktfrequenz von 24 MHz betrieben, nur daß dabei mit jedem Takt ein Befehlszyklus ausgeführt wird. Der 4051 benötigt dagegen 12 Takte für einen Zyklus. Die Bürdekapazitäten des Quarzes (Q1) sind unkritisch und können 22 - 27 pF betragen. Es wird empfohlen, ein Grundwellenquarz zu verwenden. Man erspart sich dadurch eventuell einige Schwierigkeiten mit dem Anschwingen. Die Bauteile L1 und C3 werden nur bei Oberwellenquarzen benötigt. Es kann auch das Wertepaare L1 = 12µH und C3 = 33pF verwendet werden.

Die integrierte Stromversorgung wird nur benötigt, wenn der Highspeed-Controller nicht an der Scopeclock-Platine angeschlossen ist. Dies könnte der Fall sein, wenn man den Prozessor separat von der ScopeClock programmieren will. D1 dient als Schutz bei falscher Polung der Eingangsspannung.

Für eventuelle Funktionserweiterungen ist der Pin P3.6 (beim 4051 nicht vorhanden) als Taster/Schalter (JP2) geschaltet (geschlossen = 0, geöffnet = 1). Ebenso ist der gesamte Port PO an K4 zugänglich, zusätzlich +5V und Gnd.

## Der Aufbau

Der Aufbau der Schaltung erfolgt auf einer 82 x 72 mm großen einseitigen Leiterplatte. Leider ließen sich Brücken auf der Bestückungsseite nicht ganz vermeiden. Drei Brücken kreuzen die Leiterbahnen unterhalb des Prozessors. Sie sollten verlegt werden, bevor der 40pol-Sockel eingelötet wird. Zwei kurze Brücken dienen der Verbindung isolierter Masseflächen zu der Gesamtmasse. Der Abstand von Leiterbahnen und Lötäugen zu der umgebenden Massefläche ist sehr gering. Daher muß beim Lötvorgang sorgfältig darauf geachtet werden, daß sich keine Lötbrücken bilden. Es ist anzuraten, mit einem spitzen LötKolben zu arbeiten und den LötKolben während des Lötvorganges möglichst senkrecht auf die Kupferfläche anzusetzen. Die Seite der Lötspitze darf auf keinen Fall die benachbarte Massefläche berühren. Diese Vorsichtsmaßnahmen erübrigen sich natürlich beim Verlöten der Masseverbindungen.



Die gelben Leiterbahnen bezeichnen die Drahtbrücken auf der Bestückungsseite

Nach dem Herstellen einer überschaubaren Gruppe von Lötstellen, diese mit dem Ohmmeter/Durchgangsprüfer auf Masseschluß überprüfen. Sollten sich Lötbrücken gebildet haben, diese vollständig mit Entlötlitze absaugen, den Isolationskanal von Lötzinntropfchen säubern und erneut ansetzen.

Die fertig bestückte HS-Controller Platine

### Die Stückliste:

Bezeichnung	Wert	Bemerkung
R1	10k	Metallschicht
R2	10k	Metallschicht
R3	10k	Metallschicht
C1	27p	Keramik
C2	27p	Keramik
C3	39p	Keramik
C4	470µ/25V	Elektrolyt
C5	100n	Vielschicht Keramik
C6	100n	Vielschicht Keramik
C7	47µ/16V	Elektrolyt
IC1	DS89C420-MCL	DIL 40pol
IC2	74HCT125	
IC3	7805	TO220
D1	BY133, 1N400. o.ä.	
Q1	24MHz	Quarz
L1	10µH	Festinduktivität
K1	5pol	Steckverbinder, gerade
K2	20pol	Wannenstecker, gerade
K3	2pol	RIA Anschlußklemme, RM 5mm
K4	10pol	Stifteleiste 1-reihig oder Steckverbinder
JP1	2pol	Stifteleiste 1-reihig
JP2	2pol	Stifteleiste 1-reihig
	40pol	IC-Fassung, gedreht
	14pol	IC-Fassung, gedreht
	20pol	Flachbandkabel, 15 - 20 cm
	20pol	Steckverbinder, Dual-Inline
	20pol	Pfostensteckverbinder

Die Verbindung zwischen der Highspeed-Controller Platine und der ScopeClock-Platine wird mit einem Stück 20pol Flachbandkabel hergestellt, das an einem Ende einen 20pol Pfostensteckverbinder und auf der anderen Seite einen Dual-Inline Steckverbinder eingeschnitten bekommt. Der Dual-Inline Stecker wird anstelle des 4051 auf die ScopeClock Platine gesteckt der Pfostensteckverbinder kommt in den Wannenstecker K2 auf der HS-Controller Platine.

Das Verbindungskabel, der Stecker und Verbinder sind wie abgebildet in das Kabel zu schneiden

Schaltung und Platine des HS-Controllers sind mit den Design-Produkten der Firma [ABACOM](#) "sPlan" und "Sprint Layout" erstellt. Die mit diesen Werkzeugen erzeugten Dateien sind zusammen mit dem Quellprogramm und der Hex-Datei in dem Verzeichnis \hscontrl\ enthalten, das dem ScopeClock-Package von Sascha hinzugefügt wurde. Das Platinen-Layout sowie die Schaltung können mit den bei ABACOM kostenlos herunterzuladenden Viewer angesehen und maßstabsgerecht ausgedruckt werden. Das modifizierte Package erhält man mit dem nachstehenden Download:

### [Download ScopeClock HS-Controller Package](#)

#### Das RS232-Interface

Zum Programmieren muß der HS-Controller an einen PC angeschlossen werden. Dies erfolgt über den 5-poligen RS232-Anschluß K1. Da an diesem Anschluß jedoch TTL-Pegel erforderlich sind, wird ein RS232 - Konverter erforderlich, der die V.24-Pegel des PC auf TTL-Pegel bringt. Der Konverter wurde nicht mit auf die Controller-Platine gesetzt, da er nur für das Laden des ScopeClock - Programms erforderlich ist und für den weiteren Betrieb der Uhr keine Bedeutung hat. Darüber hinaus, wenn man mehrere 89C420 Boards besitzt, z.B. noch ein Entwicklungsboard, braucht man den RS232-Konverter nur einmal aufzubauen.

Die Schaltung ist standardmäßig mit dem MAX232 bestückt, jedoch mit einer Ausnahme. Das DTR-Signal wird "seitenverkehrt" angeschlossen, d.h.: die V.24 - Seite geht an den Controller und die TTL - Seite an den PC. Weitere Erläuterungen hierzu sind in dem bereits zitierten Elektor-Artikel nachzulesen. Bausätze und Platinen für RS232 - Konverter gibt es reichlich, so daß auf einen Platinenentwurf an dieser Stelle verzichtet wird. In der folgenden Abbildung soll die erforderliche Änderung für das DTR-Signal anhand des ELV-Bausatzes gezeigt werden (zu beziehen über [ELV Elektronik AG](#)).

#### Modifikation des RS232-Konverters

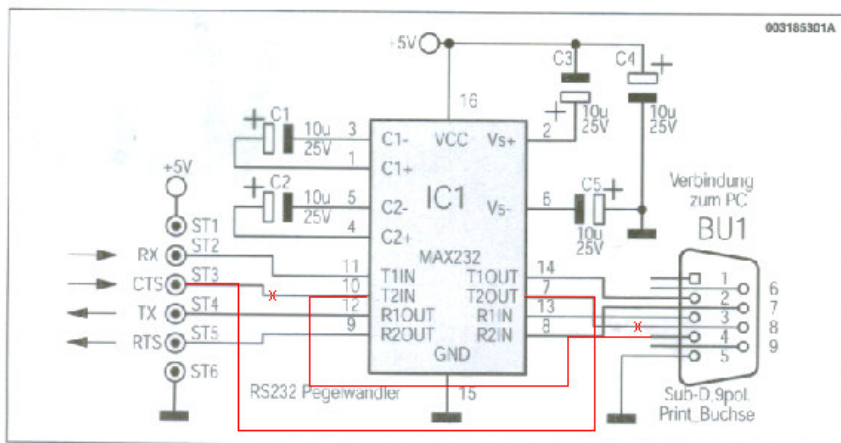


Bild 1: Schaltung des TTL- nach RS-232-Umsetzers

X = Unterbrechen — = Neu verbinden

Die Verbindung der RS232-Platine zum RS232-Anschluß auf der HS-Controller-Platine ist wie folgt vorzunehmen : (sinngemäße Anschaltung bei anderen Ausführungen)

RS232-Platine	K1
ST1	+5V
ST2	Tx
ST3	DTR
ST4	Rx
ST6	Gnd

#### Programmänderungen

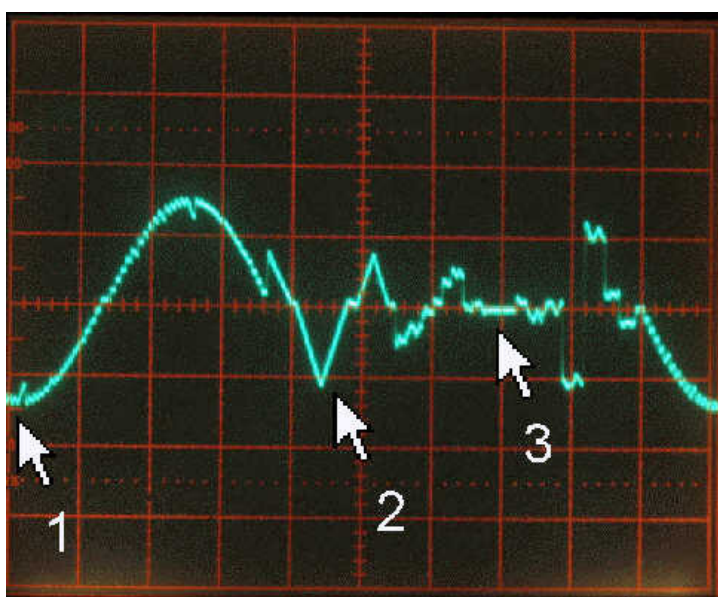
Grundsätzlich läuft die ScopeClock mit angeschlossenen HS-Controller auch mit dem unveränderten Originalprogramm. Die Darstellung ist jedoch nicht sehr ansprechend. Das rührt daher, daß das ganze Ziffernblatt jetzt in ca. 5 ms geschrieben wird. Das ist gegenüber der Originalbestückung mit dem 4051 etwa die 7-fache Geschwindigkeit. Damit reicht die Bandbreite der X-, Y- und

Z-Verstärker nicht mehr aus, um ein sauberes Bild zu erzeugen. Die Folge davon ist die Bildung von Schlieren, wenn der Strahl von einem Symbol zum anderen springt. Die Vektoren werden nicht mehr vollständig ausgeschrieben, da die Rechteckpulse zu spitzen Dreieckspulsen deformiert werden und ineinander verlaufen. Es ist daher erforderlich, um den vorherigen Zustand wieder einigermaßen herzustellen, den Bildaufbau gleichmäßig auf die zur Verfügung stehende Zeit von 20 ms zu verteilen. Zwar dürfen die 20 ms nicht ganz erreicht werden, da die Dauer des Bildaufbaues nicht konstant ist, sondern von der Konstellation der Uhrzeit in der Ziffernanzeige abhängt. Die Anzahl der zu schreibenden Punkte bzw. Vektoren der verschiedenen Ziffern ist nämlich nicht konstant.

Um auf die erforderliche Verzögerung des Bildaufbaues zu kommen, wurden im wesentlichen folgende zwei Programmänderungen durchgeführt:

- Die im Programm enthaltenen Konstanten BEAM\_ON\_DLY und BEAM\_OFF\_DLY, die die Verzögerung beim Ein- bzw. Ausschalten des Katodenstrahl bestimmen, wurden verzehnfacht (ca. 75  $\mu$ s). Da ein Byte für die Aufnahme des 12-fachen Wertes jedoch nicht mehr ausreicht, wurden diese Werte in zwei Bytes aufgeteilt und an den entsprechenden Stellen jeweils eine zusätzliche Verzögerungsschleife eingeführt.
- Eine weitere Verzögerungsschleife von ca. 6  $\mu$ s wurde in die Routine DrawLine eingefügt, die jeweils zwischen dem Schreiben der einzelnen Vektoren wirksam wird.

Das Ergebnis dieser Änderungen zeigt die folgende Abbildung, die den zeitlichen Verlauf beim Schreiben des Zifferblattes über 20 ms am Beispiel der X-Auslenkung der Scope-Röhre wiedergibt.



X-Achse - Zeitbasis 2 ms / Teilstrich

Die Kurve zeigt bei (1) das Schreiben der Teilstriche. Die kleine Auslenkung wird durch den etwas längeren Strich bei der 9 und entgegengesetzt auf dem Scheitelpunkt der Sinuskurve durch den entsprechenden Strich bei der 3 bewirkt. Das Schreiben der Striche bei der 6 und der 12 ist nur durch einen etwas breiteren Punkt beim Kreuzen der Mittellinie angedeutet (fast keine X-Ablenkung). Danach erfolgt das Zeichnen der Zeiger (2). Die drei Zacken stellen, von links nach rechts, den Sekunden-, den Minuten- und den Stundenzeiger dar. Die treppenförmige Kurve zeigt die Zifferndarstellung der Uhrzeit und anschließend das Empfangssymbol. Damit ist der logische Zyklus des Schreibens des Ziffernblattes abgeschlossen. Der kurze waagrechte Strich auf der Mittellinie (3) wird durch die Wartezeit (ca. 200 - 300  $\mu$ s) auf den nächsten Interrupt gebildet. Dann beginnt der Schreibvorgang von Neuem mit dem Schreiben der 12, 6, 3 und 9, gefolgt von den Doppelpunkten in der Zifferndarstellung. Anschließend beginnt wieder die Sinuskurve, die das Schreiben der Teilstriche zeigt, begonnen mit dem etwas längeren Strich bei der 12.

Das geänderte Programm ist in dem vorhergehend referenziertem Download-Package enthalten. In Übereinstimmung mit den Bestimmungen der GNU General Public License (GPL) wird es zusammen mit der originalen Programm-Version der ScopeClock veröffentlicht. Die nachstehend aufgeführten Dateien sind in dem zugefügten Verzeichnis \hscontrl\ enthalten:

DS89C420_Platine_MP.lay	Platinen-Layout
DS89C420_schaltung.spl	Schaltung des HS-Controllers
ScopeClk.asm	Das angepaßte Quellprogramm
scopeclk.hex	Die ladefähige Version des Programms im Intel-Format
scopeclk.lst	Das Assembler-Listing des Quellprogramms

Zur einfacheren Handhabung bei der Assemblierung wurden die Vektor-Daten in die Programmquelle eingebunden (Das Include wird in Paul's Version des AS31 - Assemblers leider nicht unterstützt). Wer das Quellprogramm neu assemblieren möchte, kann dies mit dem Befehl "as31 -l scopeclk.asm" tun. Das Ergebnis ist die ladefähige .hex - Datei sowie ein Programm-Listing.

## Die Flash-Programmierung

Zum Laden des Programms muß der HS-Controller über das RS232-Interface an einen PC sowie an eine Spannungsquelle angeschlossen werden. Letzteres kann über die Flachkabel-Verbindung zur ScopeClock-Platine oder ein Steckernetzteil erfolgen.

Der Jumper JP1 muß in jedem Fall gesteckt sein. Das Programm wird im Intel-Format an den Prozessor gesendet. Hierfür sind zwar grundsätzlich die meisten Terminalprogramme geeignet, bequemer geht es aber mit dem speziellen Programm MTK2 von Dallas-Maxim, das über eine graphische Oberfläche die erforderlichen Monitorbefehle und Daten per Maus-Bedienung an den Prozessor schickt.

Der HS-Controller mit RS232-Interface und externer Stromversorgung

Das Programm MTK2 kann von der FTP-Site von Maxim-Dallas heruntergeladen werden.

### [Download MTK2](#)

Nach der Standard-Installation des Programms kann es über das Icon auf dem Desktop gestartet werden. Es meldet sich zunächst mit einem Pop-Up-Menue zur Auswahl des Prozessors. Nach der Auswahl des DS89C420 erscheint ein übliches Programmfenster, wie wir es von Windows gewohnt sind. Zunächst ist die serielle Schnittstelle zu konfigurieren.

Über

[Options > Configure Serial Port](#)

öffnet sich ein kleines Fenster, in dem man den Com-Port (Port) und die Übertragungsgeschwindigkeit (Speed) wählt. Bei der Wahl der Übertragungsgeschwindigkeit ist man nicht ganz frei, denn der Monitor macht ein sogenanntes Autobaud und versucht die Geschwindigkeit durch Probieren mit verschiedenen Zeitkonstanten selbst herauszufinden. Leider passen die meisten dieser Konstanten nicht zu der Taktgeschwindigkeit von 24 MHz. 9600 Baud haben sich aber bewährt und die sollte man auswählen. Mit

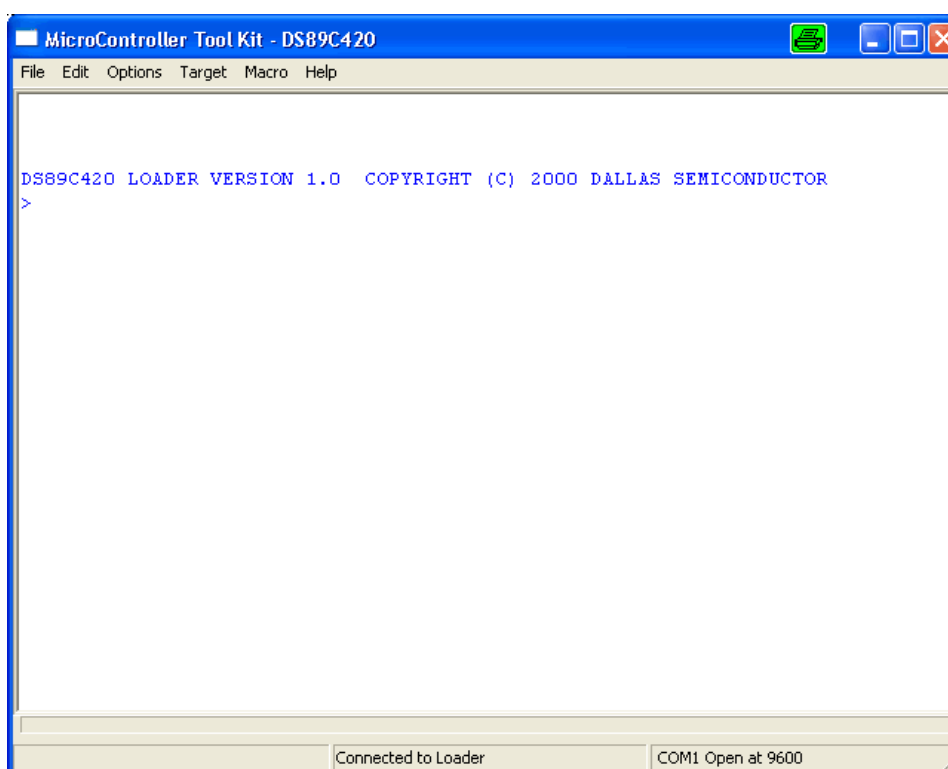
[Target > Open COMx at 9600 baud](#)

Öffnet man die serielle Schnittstelle, und mit

[Target > Connect to Loader](#)

gelangt man in den Monitor-Modus.

Der Prozessor begrüßt einen jetzt (falls alles korrekt gelaufen ist) mit einer Nachricht, wie auf nachfolgendem Screenshot zu sehen ist.



Nun ist der Monitor bereit, das ScopeClock-Programm zu laden. Hierzu geht man wieder in die Menueleiste und selektiert:

[File > Load Flash](#)

Es öffnet sich jetzt ein Open-Dialog mit dem man die Hex-Datei des ScopeClock-Programms (scopeclk.hex) auswählt. Nach Betätigung des <Öffnen> - Buttons beginnt der Ladevorgang, der auf dem Monitorschirm protokolliert wird. Anschließend führt man auf gleiche Weise ein Verify durch, um sich die Richtigkeit des Ladevorganges bestätigen zu lassen.

Mit

[File > Verify Flash](#)

wird der Verify-Vorgang initiiert. Wenn alles richtig gelaufen ist, sollte der Monitorschirm wie folgt aussehen:

```

MicroController Tool Kit - DS89C420
File Edit Options Target Macro Help
DS89C420 LOADER VERSION 1.0 COPYRIGHT (C) 2000 DALLAS SEMICONDUCTOR
>
Loading File G:\develop\C51\Projects\ScopeClock\scopeclk.hex

Erasing Flash ==> Completed

Loading Hex file...

.....

Load complete.

>
Verifying File G:\develop\C51\Projects\ScopeClock\scopeclk.hex

.....

Verify complete.

> |

Connected to Loader          COM1 Open at 9600

```

Damit ist der Prozessor programmiert und man kann nun das Programm MTK2 verlassen. Hierzu beendet man den Monitor-Modus und schließt die serielle Schnittstelle mit der Auswahl:

Target > Disconnect from Loader

Target > Close COMx

Anschließend kann man das Programm mit:

File > Exit

beenden.

Hatte man die ScopeClock-Platine mit der HS-Controller-Platine während des Programmiervorganges verbunden, wird nach der Ausführung von "Disconnect from Loader" ein Reset durchgeführt und die ScopeClock gestartet. Man sollte nun durch Ziehen des Jumpers JP1, das DTR-Signal des PC von der ScopeClock trennen, um Störungen zu vermeiden. Falls man den HS-Controller über die externe Stromversorgung angeschlossen hatte, trennt man diese und verbindet den HS-Controller mit der ScopeClock. Nach Einschalten der Betriebsspannung startet die ScopeClock nun in gewohnter Weise ohne Flimmern.

Video

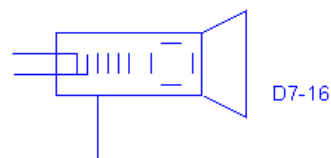
Die ScopeClock mit Highspeed-Controller "in action"

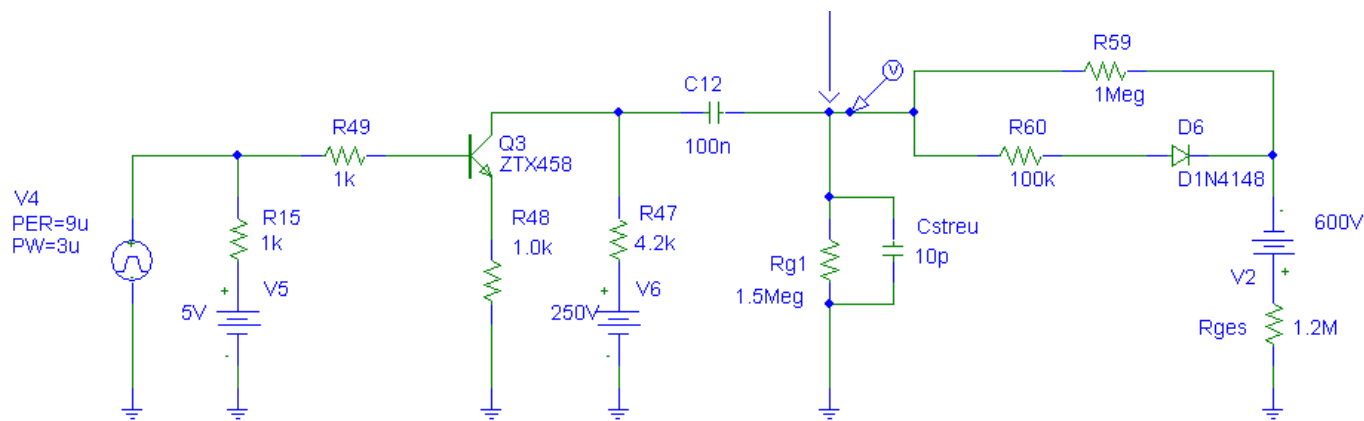
## Hardwareänderungen/-tuning

Wenn das HS-Controller Board auf Dauer mit der ScopeClock Platine verbunden bleiben soll, empfiehlt es sich, die Pull-Up Widerstände R4 und R5 zu entfernen oder einseitig auszulöten. Da sie parallel zu den internen Pull-Up Widerständen liegen, vergrößern sie unnötigerweise den Strom über die Pins P1.0 und P1.1. Beim DS89420 sind im Gegensatz zum AT89C4051 alle Pins des Ports 1 mit internen Pull-Up Widerständen versehen, so daß die externen ersatzlos entfallen können.

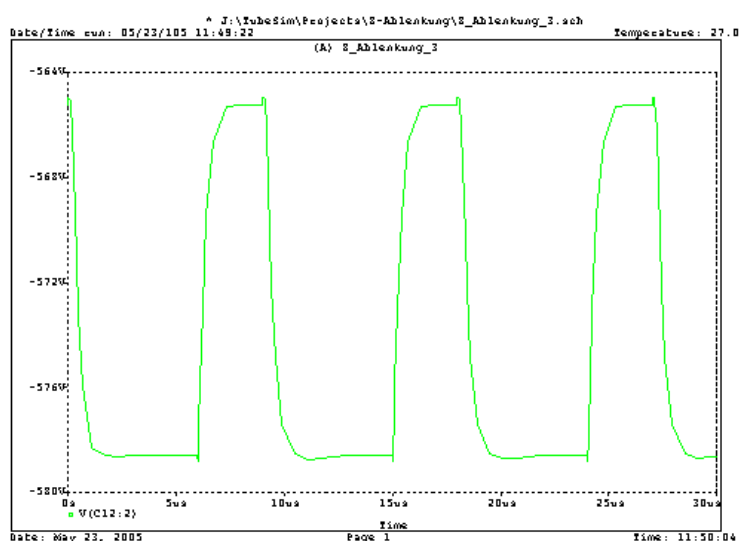
Mit der Reduzierung der Zeit für das Schreiben des Zifferblattes von etwa 35 ms auf 20 ms liegt die Schreibgeschwindigkeit mit dem HS-Controller Board um den Faktor 1,75 höher als bei der originalen ScopeClock. Um die Konturschärfe des Zifferblattes weiter zu erhöhen, wurde versucht, die Bandbreite, zunächst des Z-Verstärkers, zu verbessern. Auf Vorschlag von [Gerd Reinhöfer](#) wurde der Z-Verstärker niederohmiger gemacht (R47 = 4,7k; R48 = 1k) und in PSpice simuliert. Hier die Ergebnisse der Simulation:

ScopeClock - Z-Verstärker

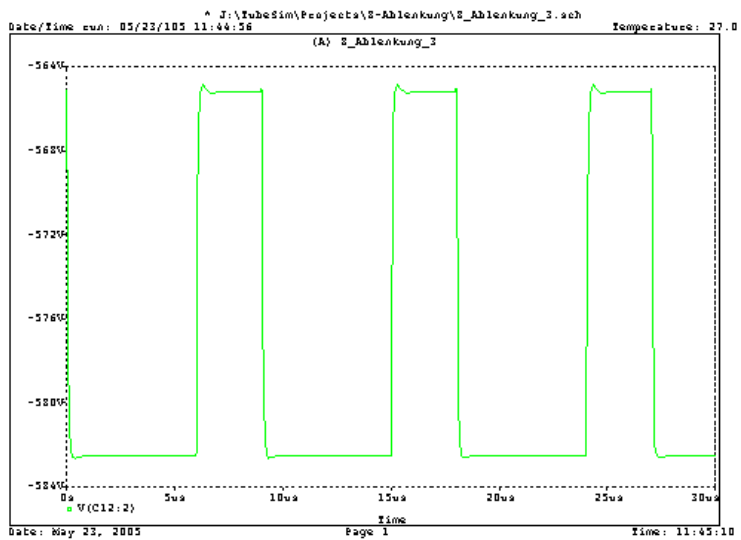




Der Z-Verstärker in PSpice



Das Ergebnis mit der Originaldimensionierung (R47 = 47k; R48 = 10k)



Und hier mit der geänderten Dimensionierung (R47 = 4,2k; R48 = 1k)

Es war nicht die Absicht, für die Simulation ein genaues Modell der Schaltung zu schaffen. Es sollte lediglich der Trend aufgezeigt werden, wenn man die Schaltung niederohmiger dimensioniert. Die Simulation wurde gewählt, da ohne Speicheroszilloskop sich der Kurvenverlauf an der Originaluhr nicht darstellen läßt. Es wurde eine Pulsfrequenz von 111 kHz (9 µs) mit einem Tastverhältnis von 1 : 2 zugrundegelegt. Das entspricht einer Auflösung beim Schreiben des Ziffernblattes von etwa 2200 Punkten oder genauer von 2200 Elementarvektoren (Bresenham-Vektoren), die der Wirklichkeit in etwa entsprechen könnte.

Wie die nachfolgenden Bilder zeigen, sind die Auswirkungen nicht dramatisch. Der größte Teil der Darstellungsoptimierung wurde bereits durch das Heruntersetzen der Schreibfrequenz erreicht, d.h., durch die kontinuierliche Dehnung des Schreibvorganges über die 20 ms. Während der Experimentierphase mit nicht optimaler Gesamtverzögerung, trat der Unterschied wesentlich



deutlicher in Erscheinung.



Die Darstellung mit R47 = 47k; R48 = 10k



Die Darstellung mit R47 = 1k; R48 = 42k

Wie man sieht, wird die Schlierenbildung etwas mehr sichtbar. Dies rührt wohl daher, daß der Schreibvorgang abrupt aufhört und der Strahl zur nächsten Position schlagartig abgelenkt wird, während ohne die Änderung, der Strahl schon in der Schreibphase schwächer (s. rechte Flanken in der PSpice-Darstellung) und sanfter zur nächsten Position verschoben wird. Will man diese Änderung nachvollziehen, ist es nicht erforderlich die Widerstände R47, R48 auszulöten. Durch Parallelschalten von 4,7k und 1,2k erreicht man die Werte 4,2k und 1k. Es empfiehlt sich dann jedoch, dem T7 einen kleinen Kühlkörper zu gönnen.

Dieses Kapitel ist sicherlich noch nicht abgeschlossen und birgt genügend Potential für weitere Experimente, wie z.B. Versuche, auch die Bandbreite der X- und Y-Verstärker zu erhöhen.

### Änderungsprotokoll

- 06.11.2005 Aufgrund eines Hinweises von Herrn Günther Zöppel wurde die Zuordnungstabelle "RS232 Platine/K1" im Abschnitt "RS232-Interface" berichtigt. ST2 muß Tx und ST4 Rx zugeordnet werden.

Harry Dronkert meldete: "Connect zum PC wird nicht ausgeführt". Dies ist eine Folge der o.a. Vertauschung von Tx und Rx.

## Quellenverzeichnis

- Sascha Ittner, [ScopeClock](#) mit D7-16 und ATMEL-Controller aus [Jogis Röhrenbude](#)
- Dallas-Maxim, Datenblatt und Applikationen [DS89C420](#)
- Elektor 6 und 9/2002, Highspeed-Controller-Board, [Elektor-Verlag](#)
- [ABACOM](#) "sPlan" und "Sprint Layout"
- E. Schemainda, ScopeClock Highspeed-Controller [Download Package](#)
- [ELV Elektronik AG](#), RS232 Interface
- Dallas-Maxim, [Download MTK2](#)
- Dallas-Maxim, DS89C420 [Ultra High-Speed Microcontroller User's Guide](#)

Stand: 16.11.2005