

embedded



Voransichts- Version

Für Bezug des Originals
siehe FAQ auf
www.embeddedFORTH.de

- 1
- 2
- 4
- 5 Januarsener Tag
- 6 Unix-Zeit
- 8 LCD-Display
- 10 Touch Memory
- 12 Intel-Hex & Motorola S
- 13 HDLC-Rahmen
- 15 Linearer Code

1

Rafael Deliano
Steinbergstr.37
82110 Germering
Tel 089/8418317

j_r_d@t-online.de

V1.0 (Papier) : 9. März 98
V2.0 (pdf) : 18. Feb. 01
V2.1 (pdf) : 28. April 04
V2.2 (pdf) : 14. Jan 07

READ . ME

In diesem Heft ist hauptsächlich Material zusammengefaßt das von Claus Vogts VD übriggeblieben ist, für sie nicht mehr rechtzeitig fertig wurde, oder für das nicht durchführbare VD-Sonderheft „embedded Systems“ gedacht war.

Die übrigen Beiträge bilden den Schwerpunkt dieses Heftes: „Zeit“. Dieses Thema ist damit nicht vollständig abgehandelt, weil z.B. eine Beschreibung von DCF77 fehlt. Kommt in einer späteren Ausgabe.

Man braucht Kalendersoftware sicher nicht oft auf Einplatinencomputern. Ich bin erst über das UNIX-Zeitformat gestolpert, als ich ZMODEM zu implementieren anfang. Irgendwann benötigt man solche Algorithmen also doch und man möchte dann gerne auf fertige und dokumentierte Routinen zurückgreifen.

Die Listings sind in nanoFORTH geschrieben und in der ursprünglichen Version von Heft 1 war deshalb ein Artikel enthalten, der die Konvertierung auf andere FORTHS beschreibt. Dieser Beitrag wurde überarbeitet ins nanoFORTH-Manual übernommen.

Datum und Normalzeit

Eine kurze Zusammenfassung der teilweise etwas komplizierten Gegebenheiten scheint angemessen, bevor man sich mit Kalenderalgorithmen oder DCF77 befaßt. Insbesondere erhält man dadurch Einblick in die Einschränkungen von Genauigkeit und Geltungsbereich bei solchen Systemen. Der „year-2000-bug“ war weder der erste noch der letzte Fehler derartiger Software.

Der Kalender war der Beginn der historischen Zeitmessung. Er ist auf astronomische Gegebenheiten abgestimmt. Leider eiert die Erde unordentlich um die Sonne, so daß diese kompliziert sind. Für den praktischen Gebrauch muß ein Kalender aber einfach zu handhaben sein. Das Kernproblem ist bis heute geblieben, einen Kompromiß zwischen Komplexität und Genauigkeit zu finden.

Tabelle 1: Tage pro Jahr

365,2421988...	tropisches Jahr (anno 1900)
365,25	Julianisches Jahr
365,2425	Gregorianisches Jahr

Jahr

Bezogen auf die Sonnenumlaufzeit gibt es das „siderische“ Jahr zu 365,2564. Bezogen auf Frühlingsanfang ist das „tropische“ Jahr jedoch 365,2422... Tage lang. Man hat also schon bei den astronomischen Grundwerten etwas Auswahl. Kalender versuchen wegen der historischen Bedeutung für die Landwirtschaft das tropische Jahr anzunähern. Nur Astronomen interessieren sich für das siderische Jahr.

Julius Cäsar führte, als er gerade nicht mit der Belagerung eines kleinen gallischen Dorfes beschäftigt war, 45vChr den Julianischen Kalender ein. Der altrömische Kalender lief damals bereits um 2 Monate falsch. Den Namen „julianisch“ hat er erst später erhalten und er ist eigentlich über Ägypten aus Arabien gekommen. Die Annäherung an das astronomische Jahr war schon recht gut (Tabelle 1). Da man keine Viertel-tage realisieren kann, gab es auch hier bereits Schaltjahre. Also 3 kurze Jahre zu 365 Tagen und ein langes Jahr zu 366 Tagen. Ergibt eine Periode von 4

Jahren. Der römische Kalender wurde bis ins Mittelalter verwendet. Da zeigte sich aber immer mehr, daß die Approximation doch noch zu ungenau war. Der Kalender hatte sich bereits auf über 10 Tage gegen die Laufzeit der Gestirne verschoben. Papst Gregor XIII führte daraufhin 1582 den gregorianischen Kalender ein, der sich langsam über die ganze Welt verbreitete.

Tabelle 2: gregorianische Ausnahmen

	/4 teilbar	/400 teilbar	Tage/Jahr
1600	y	y	366
1700	y	n	365
1800	y	n	365
1900	y	n	365
2000	y	y	366
2100	y	n	365
2200	y	n	365
2300	y	n	365
2400	y	y	366
2500	y	n	365
2600	y	n	365
2700	y	n	365
2800	y	y	366
2900	y	n	365
3000	y	n	365
3100	y	n	365
3200	y	y	366

Schaltjahr

Die Periode ist nun 400 Jahre lang, in denen 97 Schalttage verteilt sind. Kurze Jahre sind wieder 365 Tage lang. Für die langen Jahre mit 366 Tagen gilt erstens wie im julianischen Kalender: wenn die Jahreszahl durch 4 teilbar ist, liegt ein Schaltjahr vor. Zweitens gelten jedoch die „gregorianischen Ausnahmen“: ist das Jahr nicht durch 400 teilbar, ist es trotzdem kein Schaltjahr (Tabelle 2). Man kann also im Bereich 1901 - 2099 ohne die der

Annahmefähigkeit, ohne, was für eine Approximation genügt.

Der gregorianische Kalender ergibt natürlich wieder keine perfekte Anpassung an die astronomischen Gegebenheiten. Nach 3300 Jahren treten wieder nur einen 60-jährigen Eingewöhnungszeitraum. Auch der Zustand, daß sich die Erde um ein Jahr um etwa 20 Mikrosekunden verlangsamt, macht die Kalendertage nicht mehr genau auf die Jahre zentriert.

Monat

Die Bildung des Jahres in 12 Teile sollte ursprünglich die Mondphasen umhaken. Die tatsächliche Fülle der Monate von 28 bis 31 Tagen geht auf die Römer zurück und ist nicht vollständig richtig. Aus Sicht der 20. Jahre, es sollte eingetrag. Bereits im julianischen Kalender waren die Schalttage im Februar eingeschoben, was zu einer unregelmäßigen

Tage

Tage sind nicht nur nach der Länge, sondern auch nach der Zeit, die die Sonne im Zenit steht und kommt. Die Sonne wird von Osten nach Westen in 24 Stunden (1 = 24h) festgeschrieben. Heute stellt man fest, daß die Länge der Tage eigentlich schwankt (Tabelle 3).

Die als „Zerlegung“ bekannte Kurve befinden sich im Verhältnis 1:1 zu den anderen Kurven. Die

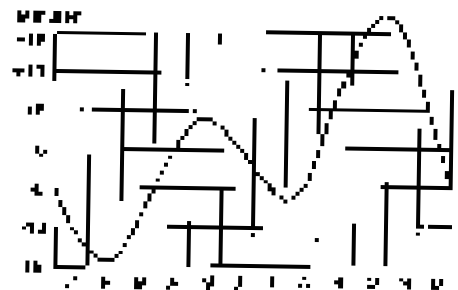


Bild 1: Zeitverlauf

Die Kurve zeigt den Verlauf der Tageslänge über ein Jahr. Die y-Achse zeigt die Tageslänge in Stunden, die x-Achse die Monate. Die Kurve ist sinusförmig und zeigt die jahreszeitliche Schwankung der Tageslänge.

Man hat jedoch auch eine UTC definiert, die die geographischen Störungen der Erdverwarpung berücksichtigt. Eine UTC die zusätzliche Jahreszeiten-Effekte mag nicht. UTC sind kaum benutzt, aber UTC ist die Zeit die die Meereszeit UTC ausdient.

Sekunde

Wenn man die Sekunden mit einem mechanischen Pendel misst, wird man gut mit dem Wert 86400 Sekunden leben und keine Probleme mit der Länge der Sekunde haben. Aber physikalische Einheit kann man auch mit einem Atomkern messen. Nach 1950 hat man mit einem Atomaufbau die Sekunde auf einen Atomkern festgelegt. 1974 schließlich wurde die heutige moderne Sekunde (die zweite) definiert. Die Coordinated Universal Time (UTC) ist die Zeit die die Weltzeit in die Sekunde ist. Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist. Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

$$LTC = LTC + DUT$$

DUT: bedeutet die Differenz zwischen der Weltzeit und der Zeit die die Weltzeit in die Sekunde ist. Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Positive Schaltsekunde:

- am 21. Dezember 1972 um 23:59:59
- am 21. Juni 1981 um 23:59:59
- am 21. Juni 1982 um 23:59:59

Negative Schaltsekunde:

- am 31. Dezember 1971 um 23:59:59
- am 31. Dezember 1972 um 23:59:59

Wenn man die Zeit die die Weltzeit in die Sekunde ist, wird man gut mit dem Wert 86400 Sekunden leben und keine Probleme mit der Länge der Sekunde haben.

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Tabelle 3: Schaltsekunden bis 1985

Jahr	Veränderung
1972	0
1973	+
1974	+
1975	+
1976	+
1977	+
1978	0
1979	+
1980	-
1981	-
1982	-
1983	-
1984	-
1985	+

Zeitzone

Die Zeitzone UTC besteht aus geographischen und die Zeit die die Weltzeit in die Sekunde ist. Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Stummzeit

Bei Sommerzeit (DST) wird die Zeit um eine Stunde vorverlegt. Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

Tabelle 4: Sommerzeit in Deutschland

Jahr	Beginn	Ende
1960	21.06	29.09
1961	26.06	27.09
1962	21.06	26.09
1963	26.06	26.09
1964	21.06	26.09
1965	26.06	26.09
1966	21.06	26.09
1967	26.06	26.09
1968	21.06	26.09
1969	26.06	26.09
1970	21.06	26.09
1971	26.06	26.09
1972	21.06	26.09
1973	26.06	26.09
1974	21.06	26.09
1975	26.06	26.09
1976	21.06	26.09
1977	26.06	26.09
1978	21.06	26.09
1979	26.06	26.09
1980	21.06	26.09
1981	26.06	26.09
1982	21.06	26.09
1983	26.06	26.09
1984	21.06	26.09
1985	26.06	26.09
1986	21.06	26.09
1987	26.06	26.09
1988	21.06	26.09
1989	26.06	26.09
1990	21.06	26.09
1991	26.06	26.09
1992	21.06	26.09
1993	26.06	26.09
1994	21.06	26.09
1995	26.06	26.09
1996	21.06	26.09
1997	26.06	26.09
1998	21.06	26.09
1999	26.06	26.09
2000	21.06	26.09
2001	26.06	26.09
2002	21.06	26.09
2003	26.06	26.09
2004	21.06	26.09
2005	26.06	26.09
2006	21.06	26.09
2007	26.06	26.09
2008	21.06	26.09
2009	26.06	26.09
2010	21.06	26.09
2011	26.06	26.09
2012	21.06	26.09
2013	26.06	26.09
2014	21.06	26.09
2015	26.06	26.09
2016	21.06	26.09
2017	26.06	26.09
2018	21.06	26.09
2019	26.06	26.09
2020	21.06	26.09
2021	26.06	26.09
2022	21.06	26.09
2023	26.06	26.09
2024	21.06	26.09
2025	26.06	26.09
2026	21.06	26.09
2027	26.06	26.09
2028	21.06	26.09
2029	26.06	26.09
2030	21.06	26.09
2031	26.06	26.09
2032	21.06	26.09
2033	26.06	26.09
2034	21.06	26.09
2035	26.06	26.09
2036	21.06	26.09
2037	26.06	26.09
2038	21.06	26.09
2039	26.06	26.09
2040	21.06	26.09
2041	26.06	26.09
2042	21.06	26.09
2043	26.06	26.09
2044	21.06	26.09
2045	26.06	26.09
2046	21.06	26.09
2047	26.06	26.09
2048	21.06	26.09
2049	26.06	26.09
2050	21.06	26.09
2051	26.06	26.09
2052	21.06	26.09
2053	26.06	26.09
2054	21.06	26.09
2055	26.06	26.09
2056	21.06	26.09
2057	26.06	26.09
2058	21.06	26.09
2059	26.06	26.09
2060	21.06	26.09
2061	26.06	26.09
2062	21.06	26.09
2063	26.06	26.09
2064	21.06	26.09
2065	26.06	26.09
2066	21.06	26.09
2067	26.06	26.09
2068	21.06	26.09
2069	26.06	26.09
2070	21.06	26.09
2071	26.06	26.09
2072	21.06	26.09
2073	26.06	26.09
2074	21.06	26.09
2075	26.06	26.09
2076	21.06	26.09
2077	26.06	26.09
2078	21.06	26.09
2079	26.06	26.09
2080	21.06	26.09
2081	26.06	26.09
2082	21.06	26.09
2083	26.06	26.09
2084	21.06	26.09
2085	26.06	26.09
2086	21.06	26.09
2087	26.06	26.09
2088	21.06	26.09
2089	26.06	26.09
2090	21.06	26.09
2091	26.06	26.09
2092	21.06	26.09
2093	26.06	26.09
2094	21.06	26.09
2095	26.06	26.09
2096	21.06	26.09
2097	26.06	26.09
2098	21.06	26.09
2099	26.06	26.09
2100	21.06	26.09

Die Zeit ist in der Sekunde, die durch Schwingungen der Schwingungszahl der Cäsium-133-Atomkerne bestimmt wird. Die Sekunde ist die Zeit die die Weltzeit in die Sekunde ist.

der Wissenschaftler. Unter anderem werden angeschafft, werden keine publikationsabhängigen Themen sind. Einmalen durch die schwarzen möge denjenigen, die andersdenkenden politischen Richtungen und über längere Zeiträume der Jahre insgesamt auf Linien der Welt der Zeit algorithmisch.

Zeitdarstellung

Wenn man die Zeit konstante, es heißt, sollte man sie auch konstant darstellen. Man muss wissen, was die Anwendungswörter. Es haben sich auch in USA und Europa eine sehr schnelle Traditionen entwickelt (Tabelle 5)

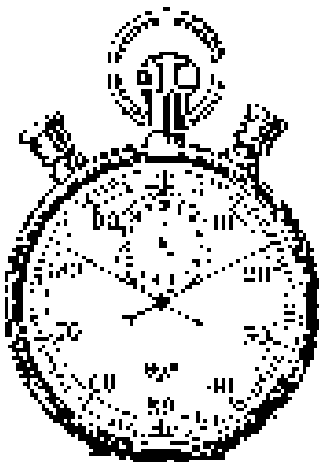
	USA	Deutschland
Wochentag	Sonntag	Montag
Datum	Mar 12, 1984	12. Mar 1984
Uhrzeit	5:32 pm	17:32

Tabelle 5: Zeitdarstellung in USA

Die Physik und Astronomie sind, um mindestens zwei für Sekunden und Minuten die Einheit der Äquivalenz zu sein.

2040' 20' = 2040m 10'

Mit dieser Rate jedoch, was die Augen der Kleinfachler können.



Marszeit

In seinem 1956 erschienen Buch macht Levitt einen kleinen aber feinen Beitrag zum Weltraumprogramm: Kalender und Uhr für die Astronauten einer Marskolonie. Sie brauchen Armbanduhren die Marszeit anzeigen und nur bedarfsweise Erdzeit umschaltbar sind. Denn der Mars tag ist mit 24h 39m 35,16s um etwa 4% länger als der Erdtag. Wollte man um 12 Uhr mittag essen und würde immer Erduhren verwenden, würde sich dieser Zeitpunkt jeden Tag um 40 Minuten weiter in den Vormittag verschieben.

Der Unterschied im Marsjahr ist noch ausgeprägter. Dieses hat doppelt soviel Tage wie das Erdjahr, nämlich 668,599051 Tage. Levitt sieht einen 5 Jahres Zyklus vor, indem das erste und 4. Jahr 668 Tage haben, die anderen 669 Tage. Diese Approximation ist für den Mars wesentlich besser als die gregorianische für die Erde. Der Schalttag wird im Dezember angehängt. Es sind 12 Monate vorgesehen, die die Namen der Erdmonate haben. Die ersten beiden Monate des Quartals haben jedoch 56, der dritte 55 Tage. Die Woche ist wieder konventionell 7 Tage lang.



er das System, das nur eine kleine Anzahl von Jahren verwendet. Je nach der Realitäts-Toleranz der Marsjahre kann man jedoch eher annehmen, dass das System 1 das Jahr der ersten Sonnenenergie ist.

Es gelang Levi, einen Uhrhersteller dazu zu gewinnen eine mechanische Uhr als Prototyp seiner (mit) Feiertage, was auf die Entwicklung der Uhrzeitentwicklung für den Mikroprozessor führt.

Wie man sich, man kann man den Kalender für den Mars, man kann man nicht sehen.



Scaligers Julianischer Tag

Das Umrechnen von Datumsangaben mit dem gregorianischen Kalender ist extrem unhandlich. Scalingers Kalender ist eine Kette aus Tagen ohne Jahre und Tage, der sich auszeichnet für Rechnungen eignet. Obwohl er schon Jahrhunderte alt ist, findet man ihn immer noch in Chronologie, Astronomie und Raumfahrt.

Josef Scaliger lebte im 16. Jahrhundert und war einer der führenden Kalenderexperten seiner Zeit. Er hatte vergleichende Studien europäischer und arabischer Kalender gemacht und seine neue Darstellungsweise als wirksames Hilfsmittel dafür entwickelt. Die gregorianische Reform war von Protestanten wie Scaliger noch nicht akzeptiert worden. Das Jahr basiert bei ihm weiterhin auf dem einfacheren julianischen Kalender der Römer mit seinen 365,25 Tagen. Also wieder 3 Jahre zu 365 Tage, gefolgt von einem Schaltjahr zu 366 Tagen. Wobei jedes durch 4 teilbare Jahr das Schaltjahr ist.

Das System wird deshalb oft auch julianischer Kalender genannt. Es ist aber nicht mit dem historischen römischen Kalender identisch. Weshalb man die Bezeichnung tunlichst vermeiden sollte. Der Scaliger-Zyklus beginnt mit dem 1. Jan 4713 vChr historischer Zeitrechnung. Dabei ist erstens anzumerken, daß die historische Zeitrechnung das Jahr 0 nicht verwendet, was die mathematische Handhabung erschwert. Es ist deshalb für negative Jahre eine astronomische Zählweise entstanden, die mathematisch korrekt ist, aber damit von der gängigen Zeitrechnung um ein Jahr abweicht.

-4713 ... -3 -2 -1 +1 +2 historische Zr,
-4712 ... -2 -1 0 +1 +2 astronomische

Zweitens wird im Zusammenhang mit Scaligers System vor der gregorianischen Kalenderreform normalerweise nach dem römischen julianischen Kalender gerechnet. Die Bezeichnung der Tage entspricht dabei jedoch dem heutigen Gebrauch und ist damit historisch nicht korrekt. Name und Dauer der Monate, auch des Februars mit seinem Schalttag sind richtig. Die alten Römer haben die Tage im Monat jedoch nicht einfach nur durchnummeriert und auch den Schalttag im Februar anders angeordnet. Der letzte Tag nach altem System ist der 4. Okt 1582. Auf



*So bin ich ein Astronomus/
Erkenn zukünftig Finsternuß/
An Sonn vnd Mond/durch das Gestirn
Darauf kan ich denn practieirn/
Ob künfftig kom ein fruchtbar jar
Oder Theurung vnd Krieghsgefahr/
Vnd sonst manicherley Kranckheit/
Milesius den anfang geit.*

Der Astronomus. Holzschnitt von Jost Amman, Vers von Hans Sachs

ihn folgt als nächster Tag der 15. Okt 1582 nach gregorianischem System.

Der Scaligerzyklus endet nach 7980 julianischen Jahren mit dem 31. Dez 3267 nChr julianischer Zeitrechnung. Im Gregorianischen Kalender ist das der 22. Jan 3268. Da der Zyklus damit die gesamte interessante Zeit der Weltgeschichte abdeckt und einige rechentechnische Vorteile bietet, avancierte er zum bevorzugten System der Chronologie und Astronomie. Er ist deshalb heute immer noch in Gebrauch.

Julianischer Tag

Tatsächlich rechnet man heute in diesem Kalender nicht mit Jahren, sondern sein Rückgrat ist der Julianische Tag, „Julian Date“, JD. Die Tage laufen ab der Stunde Null kontinuierlich hoch. Z.B.:

1. Jan 1900 = JD 2415020

Die Zahl ist ein wenig größer, wenn man sich aber nur mit der astronomischen Zeitrechnung beschäftigt, ist die Zahl nicht so wichtig. Auch die Stunden, Minuten, Sekunden, die man mag, sind nicht so wichtig, wie die Werte, die man als Kalenderstelle angedenkt. Z.B.

1900 = Mitternacht des 1. Jan
1901 = Mittag des 1. Jan

Es gibt aber eine astronomische Annahme, die an der Definition gleich ist. Sie lautet, die Stunde Nr. 1000000000 der Epoche verschieben, damit sie an einem bestimmten Zeitpunkt im Universum beginnt. Das ist die Epoche der Weltzeitrechnung.

1900 + 1000 = 1901

Und es haben ihren eigenen Namen. Die Zahl 1000000000 ist die Epoche der Weltzeitrechnung. Sie kann daher auch als Epoche bezeichnet werden. Die Epoche ist die Epoche der Weltzeitrechnung. Die Epoche ist die Epoche der Weltzeitrechnung.

Modifizierter Julianischer Tag

Mit anderen Worten, die Epoche ist die Epoche der Weltzeitrechnung. Die Epoche ist die Epoche der Weltzeitrechnung. Die Epoche ist die Epoche der Weltzeitrechnung.

1900 = JD 2415020
1901 = JD 2415021

Die angegebenen Werte sind die JD der Epoche. Die JD der Epoche ist die JD der Epoche. Die JD der Epoche ist die JD der Epoche.

erweitert sein. Nullpunkt zum 1. Jan 1970
gelöst:

ANSI C Tag - 11: 24439735

Implementierung

Bei der Implementierung für die Wandlung von Kalendertangaben in UTC wurde auf die unvollständigen Daten vor die 15. Jan 1970 (Dawn) verzichtet und Jan 1. 1970 als Startzeit genommen. Die Zeitangaben sind in Sekunden gegeben.

Die Routine `SEC2CAL` und `CAL2SEC` wandeln die Zeitangaben in `SEC` in die `YEAR`, `MONTH` und `DAY` des Julianischen Datums um. Die Variable `DAY` ist in Sekunden gegeben.

Die Angaben auch gegeben als Julianischer Kalender und ist für die Angaben gegeben. In der 15. Jan 1970 zu verwenden. Die Variable `DAY` ist in Sekunden gegeben. Die Zeitangaben sind in Sekunden gegeben.

Die Variable `YEAR` ist in Sekunden gegeben. Die Variable `MONTH` ist in Sekunden gegeben. Die Variable `DAY` ist in Sekunden gegeben. Die Variable `YEAR` ist in Sekunden gegeben. Die Variable `MONTH` ist in Sekunden gegeben. Die Variable `DAY` ist in Sekunden gegeben.

Die Variable `YEAR` ist in Sekunden gegeben. Die Variable `MONTH` ist in Sekunden gegeben. Die Variable `DAY` ist in Sekunden gegeben. Die Variable `YEAR` ist in Sekunden gegeben. Die Variable `MONTH` ist in Sekunden gegeben. Die Variable `DAY` ist in Sekunden gegeben.

man im üblichen 16 Bit FORTH keine 64 Bit Divisionsbefehle. Zweitens haben die Jahre unterschiedliche Länge, Division wäre also unmittelbar nicht verwendbar. Man behilft sich mit einer Primitivdivision durch kontinuierliche Subtraktion. Für das Sekundenjahr das man jeweils abzieht, kann man so auch passend die Länge variieren, wenn es sich um ein Schaltjahr handelt. Nächstes Problem ist das Erkennen des Unterlaufs, wenn man in FORTH kein Carryflag hat. Für Jahre bis 2038 ist das oberste Bit des 32 Bit Wertes normalerweise Null. Man kann Unterlauf dann leicht erkennen, wenn man das oberste Bit nach der Subtraktion darauf testet, ob es gesetzt ist. Damit ist das Resultat negativ und diese Subtraktion war schon zuviel. Man verwirft das neue Resultat und der vorhergehende Wert ist das Endergebnis.

[1] von Meyer, „Computerische Algorithmen“ Verlag North 1984

UNIX-Zeit

Dieses computerinterne Zeitformat dürfte nicht nur die größte Verbreitung haben, sondern es ist auch sehr effizient. Man muß nur einen 32 Bit Zähler jede Sekunde inkrementieren. Leider ist die Umwandlung daraus ins „menschliche“ Format etwas mühsam. Deshalb nicht günstig um daraus die Uhrzeit auf einem Display anzuzeigen. Aber sicher optimal für Zeitstempel.

Die 32 Bit Zahl gibt die Sekunden an, die seit dem 1. Jan 1970 00:00:00 UTC vergangen sind. Durch UTC wird man von Zeitzonen unabhängig. Dafür fehlt natürlich der sichere Bezug zur örtlichen Zeit. Gewisse Feinheiten wie Schaltsekunden werden dabei auch nicht berücksichtigt. Beispiel:

00000000h = 1. 1.1970 0: 0: 0
30000000h = 9. 7.1995 16:12:48
7FFFFFFFh = 19. 1.2038 3:14: 7
FFFFFFFh = 7. 2.2106 6:28:15

Vorteilhaft ist die Kompaktheit, man belegt nur 4 Bytes. Diese ist wünschenswert, wenn man im Kopf eines Files abspeichern will, wenn es das letztmal modifiziert wurde. Gleichzeitig kann man diese Zeitangabe sehr simpel erzeugen, wenn man als Systemtick 1sec wählt. Man muß nur einen 32 Bit Zähler inkrementieren. Trotz des platzsparenden Formats wird ein Zeitraum von über

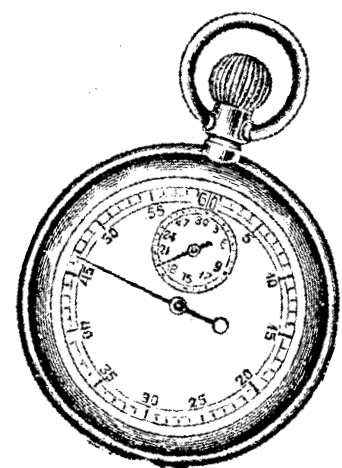
100 Jahren auf die Sekunde genau aufgelöst.

Gewisse Nachteile ergeben sich natürlich auch. Man kann kein Datum vor 1970 darstellen, denn negative Zeiten mittels des 32. Bits sollte man nicht definieren. Sonst reicht die Wortbreite nur bis zum Jahr 2038, und das ist heute schon etwas knapp. Hauptproblem ist aber vor allem die zähe Umwandlung ins „menschliche“ Format.

Datum decodieren

Die Routine `DATE!` erhält die codierte Zahl auf dem Stack und legt das Ergebnis in den Variablen `DAY`, `MONTH`, `YEAR`, `HOURL`, `MIN`, `SEC` ab (Listing UT1). Für diese genügt ein Byte. Nur `YEAR` muß 16 Bit breit sein.

Die Extraktion der Jahre erfolgt nominell durch „Division“ mit der Sekundendauer eines Jahres. Damit fangen dann schon die Probleme an. Erstens hat



man im üblichen 16 Bit FORTH keine 64 Bit Divisionsbefehle. Zweitens haben die Jahre unterschiedliche Länge, Division wäre also unmittelbar nicht verwendbar. Man behilft sich mit einer Primitivdivision durch kontinuierliche Subtraktion. Für das Sekundenjahr das man jeweils abzieht, kann man so auch passend die Länge variieren, wenn es sich um ein Schaltjahr handelt. Nächstes Problem ist das Erkennen des Unterlaufs, wenn man in FORTH kein Carryflag hat. Für Jahre bis 2038 ist das oberste Bit des 32 Bit Wertes normalerweise Null. Man kann Unterlauf dann leicht erkennen, wenn man das oberste Bit nach der Subtraktion darauf testet, ob es gesetzt ist. Damit ist das Resultat negativ und diese Subtraktion war schon zuviel. Man verwirft das neue Resultat und der vorhergehende Wert ist das Endergebnis.

Treiber für LCD-Display

Trotz der Vielfalt an Herstellern und Bauformen haben Text-LCD-Module eine Gemeinsamkeit: sie verwenden den HD44780 als LCD-Controller. Auch die Pinbelegung der Schnittstelle ist relativ einheitlich. Die hier dargestellte Hardware und Software wird deshalb zwar nicht für alle Typen sofort passen. Sie stellt aber einen guten Ausgangspunkt dar. Die nötigen Änderungen sind meist minimal.

Hardware

Es gibt zwei verschiedene Möglichkeiten das LCD anzusteuern. Entweder man hängt es an einen Port, oder direkt an den Bus der CPU. Letzteres hat einige verkappte Nachteile. Erstens wird das LCD aus mechanischen Gründen meist über ein Flachbandkabel verbunden. Dessen Koppelkapazität führt zu Übersprechen und verschlechtert so die elektrischen Signale auf dem Datenbus. Zweitens ist das LCD mit fast 300nsec Zugriffszeit recht langsam. Entweder man taktet die CPU niedrig, oder man muß eine Wait-state-Logik einbauen.

Hier wird deshalb Zugriff über einen Port dargestellt (Bild 1). Dafür wird der LCD-Controller dann mit 4 Bit und nicht mit 8 Bit Datenbusbreite angesteuert. Da der LCD-Controller Pullups hat, bleiben die untere Hälfte seines Datenbusses unbeschaltet. Beim Port des Controllers muß die Richtung bitweise umschaltbar sein, da man Daten rücklesen muß. Die drei Steuerleitungen sind Outputs. R/W steuert die Richtung, A0 selektiert zwei Adressen, CS wird für Datenübertragung kurz high gepulst (Bild 2). Das freie Bit 7 am Port kann als Input verwendet werden.

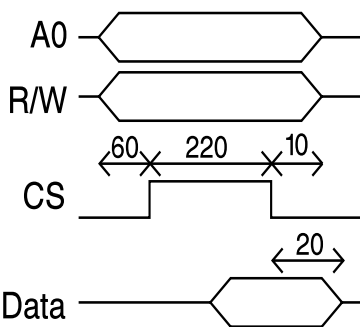


Bild2: Timing

Bias

Die Spannung am Bias-Pin muß entsprechend der Umgebungstemperatur nachgeführt werden, damit der Kontrast und damit die Lesbarkeit erhalten bleibt. Der genaue Spannungsverlauf hängt vom Typ, Betriebsart (Refresh) und Bauteilstreuung ab (Bild 4). Typisch variiert die Spannung in einem Bereich von 0 - 1V (manchmal 2V). Der Lastwiderstand hat einen Wert von ca. 10kOhm und ist nur sehr lose spezifiziert.

Am einfachsten realisierbar ist manuelle Einstellung mit einem Poti. Die in Bild 5 gezeigte simple Schaltung ist nur optimal, wenn man den hohen Querstrom durch das Poti akzeptiert. Mit einem Transistor als invertierendem Verstärker optimiert man Querstrom und Linearität. Beide Schaltungen sind ausreichend, wenn das Gerät nur bei Zimmertemperatur betrieben wird, oder der Benutzer das Poti nachstellt.

Letzteres ist in der Praxis aber meist nicht erwünscht. Für weiten Temperaturbereich empfiehlt sich deshalb eine automatische Regelung. Im gezeigten Beispiel (Bild 6) wird ein Siliziumtemperatursensor im kompakten Diodengehäuse verwendet. Er sollte auf dem Blechrahmen des LCD-Moduls montiert werden. An den OP werden besondere Anforderungen bezüglich der maximalen Eingangsspannung (CMV) gestellt, die hier 4,2V beträgt. Der ICL7611 ist als Alternative zum LM4250 geeignet. Das Poti ermöglicht die Feinkorrektur um Bauteilstreuung zu korrigieren.

ESD

Gute LCDs haben eine durchsichtige Plastikkappe die das ganze Glas

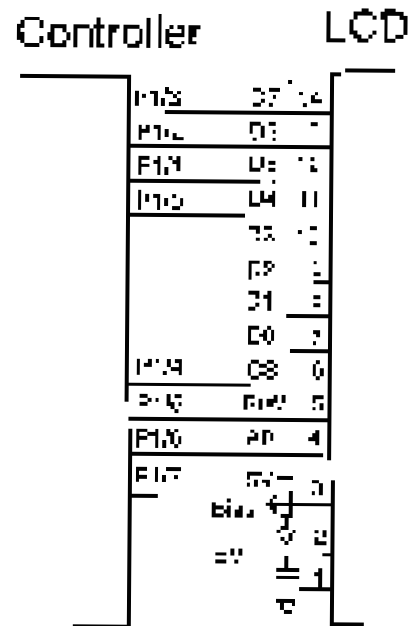
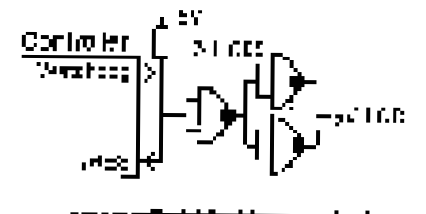


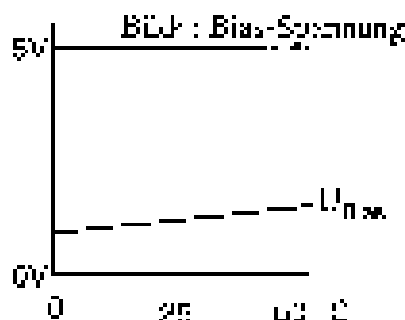
Bild 4: Pinbelegung

... die in der Praxis aber meist nicht erwünscht. Für weiten Temperaturbereich empfiehlt sich deshalb eine automatische Regelung. Im gezeigten Beispiel (Bild 6) wird ein Siliziumtemperatursensor im kompakten Diodengehäuse verwendet. Er sollte auf dem Blechrahmen des LCD-Moduls montiert werden. An den OP werden besondere Anforderungen bezüglich der maximalen Eingangsspannung (CMV) gestellt, die hier 4,2V beträgt. Der ICL7611 ist als Alternative zum LM4250 geeignet. Das Poti ermöglicht die Feinkorrektur um Bauteilstreuung zu korrigieren.

Zeichen & Zeichen

... die in der Praxis aber meist nicht erwünscht. Für weiten Temperaturbereich empfiehlt sich deshalb eine automatische Regelung. Im gezeigten Beispiel (Bild 6) wird ein Siliziumtemperatursensor im kompakten Diodengehäuse verwendet. Er sollte auf dem Blechrahmen des LCD-Moduls montiert werden. An den OP werden besondere Anforderungen bezüglich der maximalen Eingangsspannung (CMV) gestellt, die hier 4,2V beträgt. Der ICL7611 ist als Alternative zum LM4250 geeignet. Das Poti ermöglicht die Feinkorrektur um Bauteilstreuung zu korrigieren.





Software

Auf unterster Ebene muß man die Befehle `LDI` und `LDD` definieren, um Daten zum Betriebssystem und umgekehrt zum `UIC` (UART) zu verschieben. Wenn `UIC2` dargestellt ist, gibt es zwei Kanäle für die Übertragung. In der ersten Instanz sind diese Kanäle in einer Sprache vorhanden, man kann sie aber auf `ASCII`

umschalten. Der `LCD`-Controller wird auch als `Bestandteil` des `UIC` (UART) angesehen. Diese Lösung ist nicht mehr proprietär. Die `UIC`-Controller sind in einem `UART` (UART) integriert. Der Zugriff auf die Daten ist von `UART` und `UIC` (UART) über den `UART` Controller mit dem `UART` verbunden. Die Daten können über den `UART` (UART) Controller übertragen werden. Für weitere Informationen über die `UART` (UART) Controller besuchen Sie die Website www.freescale.com.

In dieser Implementierung wird die Daten über den `UART` (UART) Controller im `UART` (UART) Controller. Die Daten werden über den `UART` (UART) Controller übertragen. Diese Methode verwendet die Parallelität und die Flexibilität der `UART` (UART) Controller. Wenn man nur wenige Zeichen verwendet, werden die Daten über den `UART` (UART) Controller übertragen. Die Daten werden über den `UART` (UART) Controller übertragen. Die Daten werden über den `UART` (UART) Controller übertragen. Die Daten werden über den `UART` (UART) Controller übertragen.

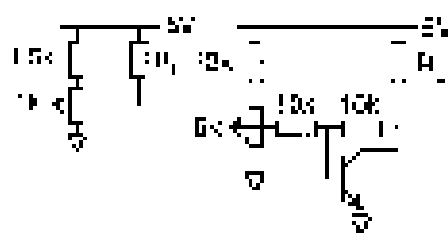


Bild 3: manuelle Biasregulierung

Um den Text in RAM aufzuzeichnen und Einstellungen zu erhalten, stehen dem Anwender folgende Befehle zur Verfügung:

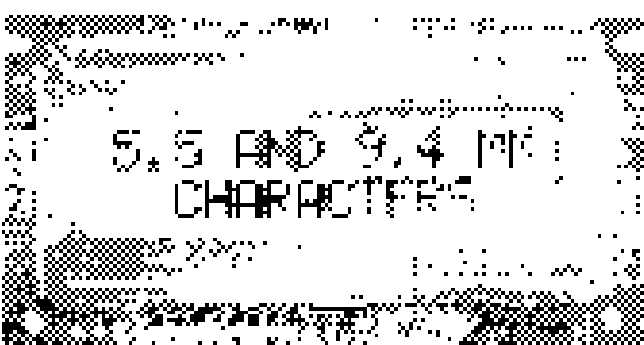
- `WRITE` - Zeile des Displays
- `MOVE` - verschieben des Cursors nach links/rechts
- `UP` - Cursor auf Zeile nach oben
- `DN` - Cursor eine Zeile abwärts
- `CLR` - löscht Zeichen im `UART`
- `PRINT` - `UART` und `UART`
- `WRITE` - `UART` Cursor sichtbar
- `WRITE` - `UART` Cursor unsichtbar
- `WRITE` - `UART` Display abschalten

Die Buchstaben die maximal `16` Zeichen pro Zeile sind mit `8` Bit kodiert. Der empfangene `UART` (UART) des `LCD` liefert ein Byte von `00` - `7F` die entsprechenden `ASCII`-Zeichen. Man kann die Zeichen in `RAM` definieren. Ein `UART` (UART) gespeichert werden. Im `UART` (UART) der `UART` (UART) Controller. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.

Wenn man den Cursor steuern möchte, sollte man die zweite Hälfte des Befehls `UART` (UART) verwenden.

Eigene Zeilen

Während der Entwicklung des Displays sind die `UART` (UART) Controller und die `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.



die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.

Modifikation

Das `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.

Man kann die `UART` (UART) Controller mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.

Das `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden. Die `UART` (UART) Controller sind mit dem `UART` (UART) Controller verbunden.

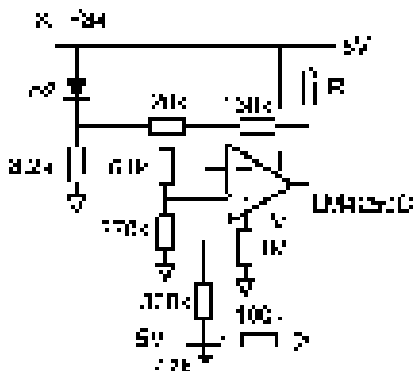


Bild 4: Halbbrücke Leistungszug

Treiber für Touch Memory

Die Firma Dallas Semiconductor hat einen seriellen Bus eingeführt mit dem man über nur zwei Kontakte auf Peripherie-ICs zugreifen kann. Auf Leiterplattenebene ergeben sich dabei kaum Vorteile gegenüber gängigen Verfahren wie dem I2C-Bus. Doch kann man damit ICs in den robusten Knopfzellegehäusen unterbringen, die von Batterien bekannt sind. Im einfachsten Fall enthält der Chip nur eine individuelle Seriennummer im ROM. Für komplexere Anwendung ist auch lithiumgepuffertes RAM verfügbar. Damit ist das Touch Memory (TM) als billiger Transponder einsetzbar, da es in vielen Fällen kein Nachteil ist, daß es mechanisch kontaktiert werden muß.

Hier wird die Software für die Ansteuerung der ältesten Version, den DS1991 dargestellt. Dieses TM verfügt über eine Seriennummer im ROM und 192 Byte lithiumgepuffertes RAM. Man kann mehrere TMs in einem Bussystem parallel auf eine Leitung schalten. Was jedoch die Software verkompliziert und von der Applikation meist nicht benötigt wird. Hier ist deshalb nur die Ansteuerung eines einzelnen TMs vorgesehen.

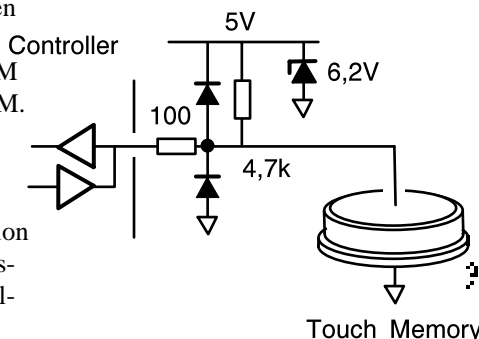
Hardware

Man braucht am Controller einen Portpin den man bidirektional umschalten kann, sowie einen Pullup-Widerstand (Bild 1). Da das TM jedoch vom Anwender wechselbar sein soll, ist normalerweise noch etwas ESD-Beschaltung benötigt (Bild 2).

Software

Auf unterster Ebene sind Befehle definiert die einen Reset auslösen, Bits lesen und schreiben. Im Gegensatz zum I2C-Bus muß hier ein festes Timing eingehalten werden, weshalb man die Verzögerungsschleifen auf den Takt der

Bild2: robustere Beschaltung



CPU abstimmen muß. Interrupts sind während dieser Befehle gegebenenfalls zu sperren. Das bedeutet eine Sperrung von 70 usec während eines Bits. Und nominell 1msec für den Reset, jedoch kommt man hier mit etwas komplizierterer Programmierung auf 80 usec herunter.

Der Reset des TM wird vom Controller durch einen langen Lowpuls ausgelöst (Bild 3). Das TM antwortet mit einem kurzen Quittungspuls. Der Befehl RESET-TM liefert deshalb ein entsprechendes Flag zurück. Soll das Einstecken eines TM automatisch erkannt werden, muß der Controller die Leitung etwa alle 500msec mit einem

Bild3: Timing

Reset

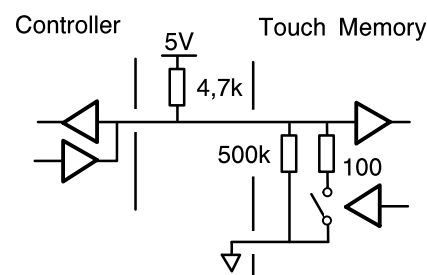
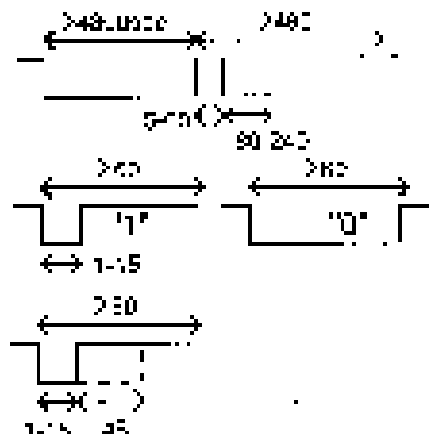


Bild1: simple Beschaltung

RESET-TM liefert und prüfen, ob eine Quittung kommt. Die Uhrzeitgleichung wird durch einen kurzen Stamp des vom Controller gesendeten. Zwischen den Bits können notwendig einige Verzögerungen sein. Die vom Frontkollimator gemessene Zeit ist ein bisschen länger als die im Datenblatt angegebene. Die Verzögerung ist durch die Länge der Leitung und die Qualität der Verbindung bedingt. Die Verzögerung ist durch die Länge der Leitung und die Qualität der Verbindung bedingt.

Zugriff aufs ROM

Jede Operation beginnt mit einem Reset. Bei jedem Zugriff aufs ROM. Um einen manuellen Zugriff auf das Teil (Skip ROM) oder auch über das ROM einen Bit zu lesen oder zu schreiben. Beim DS1991 ist dieses Byte immer 0x00. Beim folgenden 8 Bit lange Seriennummer. Die bei jedem Zugriff zu erhalten. Die Seriennummer ist programmiert. Die Seriennummer ist ein CRC-Byte. Zwischen die Typnummer und die Seriennummer. Die Seriennummer liefert auf dem ersten Bit die Seriennummer zurück. Es gibt eine Variable Typnummer als die des DS1991. In der ersten der des Typnummers. Die Seriennummer liefert auf dem ersten Bit die Seriennummer zurück. Es gibt eine Variable Typnummer als die des DS1991. In der ersten der des Typnummers.

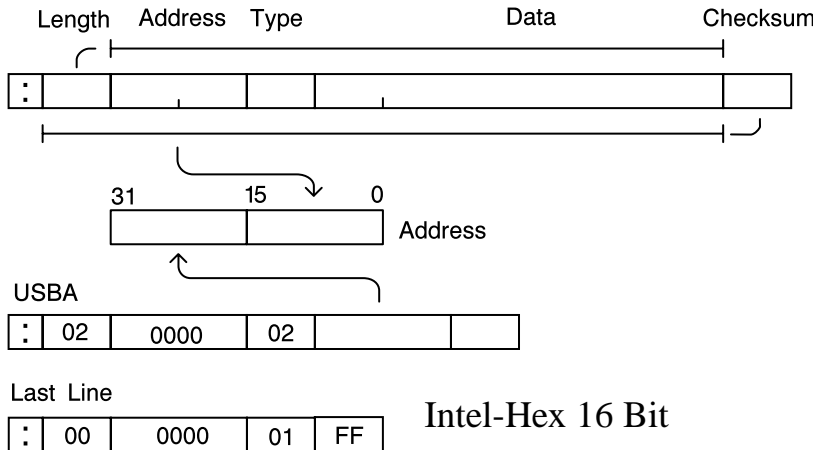
Zugriff aufs RAM

Das RAM besteht aus 192 Bytes. Die Seriennummer liefert auf dem ersten Bit die Seriennummer zurück. Es gibt eine Variable Typnummer als die des DS1991. In der ersten der des Typnummers. Die Seriennummer liefert auf dem ersten Bit die Seriennummer zurück. Es gibt eine Variable Typnummer als die des DS1991. In der ersten der des Typnummers.

Intel-Hex und Motorola-S-Files

Auf einem Einplatinencomputer auf dem FORTH installiert ist, kompiliert man Software ins RAM und testet sie dort. Anschließend will man das Programm dauerhaft in ein EPROM übertragen. Für die Übertragung des Binärfiles aufs Programmiergerät gibt es spezielle ASCII-Formate. Die gängigsten sind Intel-Hex und Motorola-Hex, beide haben einige Varianten.

Intel-Hex



Gemeinsam ist beiden Formaten, daß jede Zeile eine in sich abgeschlossene Einheit ist. Bytes werden als Hexzahlen zu 2 Ziffern übertragen. Adressen oder 16 Bit Zahlen werden genauso zu 4 oder mehr Ziffern formatiert, wobei die höchste Ziffer links und die niedrigste rechts steht.

Intel Hex 8 Bit

Der Aufbau der Zeile ist in Bild 1 dargestellt. Jede Zeile hat als Startzeichen einen Doppelpunkt. Das „Length“-Feld (4 Buchstaben) enthält die Zahl der Bytes die im Datenfeld übertragen werden. „Address“ ist die Adresse des ersten Datenbytes der Zeile. Das Feld „Type“ hat den Wert „00“ in einer Datenzeile und „01“ in der Endzeile. Für „Checksum“ werden alle Bytes des Length-, Address-, Type- und Datenfelds addiert und dann daraus das 2er-Komplement gebildet. Nur das untere Byte davon wird jedoch gesendet.

Ein File besteht aus beliebig vielen Datenzeilen und einer Endzeile als Abschluß.

Intel-Hex 16 Bit

Die Daten sind hier keine Bytes, sondern 16 Bit Werte. „Length“ zählt nun keine Bytes, sondern 16 Bit Worte. Das Feld „Type“ ist aber immer noch „00“.

USBA

Durch eine USBA-Zeile wird eine Adreßerweiterung von 16 Bit auf 32 Bit durchgeführt. Damit überwindet man die 64k Adreßbegrenzung. Die USBA-Zeile ist optional. Hat der Empfänger keine USBA empfangen, nimmt er sie als 0000 an. Und dann wurde Intel-HEX nochmal für 32 Bit Daten aufgebohrt. Dazu wird eine Startzeile mit dem „Type“-Feld 03 geschickt.

Motorola-Hex

Der Aufbau der Zeile findet sich in Bild 2. „Record Type“ gibt den Zeilentyp an. „S0“ ist eine Startzeile. Es sind mehrere S0-Zeilen am Anfang eines Files möglich. Das Adreßfeld wird hier fest mit der Adresse 0000 gesendet. Das Datenfeld kann Zusatzinformationen enthalten. Aber da ASCII-

Informationen enthalten werden ASCII-Zeichen in Hex codiert und meistens sind die Kommandozeilen mit einem normalen Editor nicht lesbar, was die Nutzbarkeit einschränkt. Diese Zeilen sind optional und werden nicht übertragen. S0-Feld: 32-Bit Adreßbegrenzung mit unterschiedlicher Adreßlänge. Der Typ „S0“ kann außerdem die Checksum Bit (in Endzeile) darstellen. Adreßfeld: Adreß spezifiziert die Startadresse der Software übertragen werden. Das Feld „Length“ enthält die Zahl der im Adreßfeld und Checksum übertragen Bytes. Die Zahl der Bytes der Zahl der hexadecimale Übertragung ASCII-Zeichen. Die Summe wird hexadecimale dargestellt.

Adreßfeld enthält die Adreß im 24-Bit Typ (in 16-Bit Adreß) gewesen. 30-Bit in 32-Bit und 16-Bit im 32-Bit Typ. Die 32-Bit spezifiziert zum 64k Adreßbereich. 30-Bit Controller. Der 32-Bit enthält die Daten verwendet immer als Bytes dargestellt. Checksum enthält die Summe der Bytes der Summe vor hexadecimale Adreß und Daten. Die Zahl der Bytes wird über den hexadecimale ASCII-Zeichen dargestellt. Die 32-Bit spezifiziert die Zahl der Bytes der hexadecimale Übertragung ASCII-Zeichen.

Dump

Die Programme sind in den Listing 111EX, 112EX und 113EX dargestellt. Speicheradressen beträgt 0x0000 bis 0x0000. Mit dem ASCII-Zeichen werden die ASCII-Zeichen dargestellt. Die Zahl der Bytes der hexadecimale Übertragung ASCII-Zeichen. Die Zahl der Bytes der hexadecimale Übertragung ASCII-Zeichen.

Die Zahl der Bytes der hexadecimale Übertragung ASCII-Zeichen. Die Zahl der Bytes der hexadecimale Übertragung ASCII-Zeichen.

Um zu vermeiden, daß die Kommunikation bei Übertragung der Adressen nicht abgebrochen wird, muß in jedem Byte ein 1er- und ein 0er-Bit vorkommen. Dies wird durch Bit Stuffing erreicht.

Load

Das Listing DUMP zeigt das Ergebnis der Übertragung von 1000 Bytes. Die Übertragung wurde durch den Controller gestoppt, weil die Übertragungsgeschwindigkeit zu langsam war. Die Übertragung wurde durch den Controller gestoppt, weil die Übertragungsgeschwindigkeit zu langsam war. Die Übertragung wurde durch den Controller gestoppt, weil die Übertragungsgeschwindigkeit zu langsam war.

Motorola-S-Records

Type	Length	Address	Data	Checksum
31	4	0000	0000	0000
32	4	0000	0000	0000
33	4	0000	0000	0000

Das Listing zeigt die Übertragung von 1000 Bytes. Die Übertragung wurde durch den Controller gestoppt, weil die Übertragungsgeschwindigkeit zu langsam war. Die Übertragung wurde durch den Controller gestoppt, weil die Übertragungsgeschwindigkeit zu langsam war.

HDLC-Rahmen in Software

„High Level Data Link Control“ ist ein sehr verbreitetes Format zur Übertragung von Datenpaketen. Es arbeitet auf synchronen Kanälen. Im Gegensatz zur asynchronen Übertragung, wie bei V24, sind hier keine Start- und Stopbits nötig, wodurch der Durchsatz höher liegt. Normalerweise findet die Auswertung von HDLC-Rahmen und CRC in einem Kommunikations-IC, wie z.B. der Z80 SIO, in Hardware statt. Bei hoher Übertragungsgeschwindigkeit und großen Datenmengen wird man immer auf solche ICs zurückgreifen müssen. Bei Anwendungen mit niedriger Datenrate, wie dem D-Kanal von ISDN mit 15kBit/sec, und ausreichender Rechenleistung des Controllers, kann man diese exotischen Bausteine vermeiden, indem man die Bits in Software verarbeitet.

könnte eine beliebig krumme Zahl von Bits lang sein. Es ist aber vorgeschrieben, daß die Zahl der Bits durch 8 teilbar ist, die kleinste Einheit sind also Bytes oder ein Vielfaches davon. Da das Ende durch ein Stopzeichen markiert wird, gibt für das Paket keine Längenbeschränkung. Meistens begrenzt man es auf maximal 4k Byte, weil die Sicherheit der Fehlererkennung durch die 16 Bit CRC sonst zusehr nachläßt. Außerdem ist die Wiederholung der Übertragung mit kürzeren Paketen effizienter.

Flag

Start des Blocks (Bild 1) ist das „Flag“, ein Byte mit dem Wert 01111110b. Es ist das heilige Zeichen des Protokolls und darf bis zum Ende des Blocks nicht mehr im Datenpaket auftreten. Das wird durch „bit stuffing“ zwangsweise sichergestellt. Bit stuffing bedeutet, daß wenn in den Daten mehr als 5 Einsen hintereinander vorkommen, der Sender nach der fünften Eins eine Null einfügt (Bild 2). Der Empfänger entfernt dieses Bit wieder. Außerdem überprüft der Empfänger, ob 7 aufeinanderfolgende Einsen ankommen. Das ist offensichtlich illegal und wird in

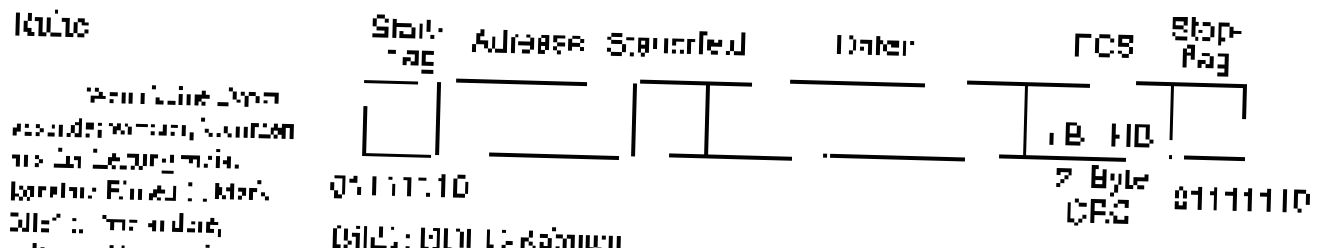
manchen Systemen als „abort sequence“ dazu verwendet, die Übertragung eines Pakets vorzeitig abbrechen. Das Ende-Flag eines Pakets kann gleichzeitig das Start-Flag des nächsten Blocks sein.

Datenblock

Der Datenteil besteht nominell aus Adreß-, Steuer-, und Datenfeld. Deren Aufbau ist in den verschiedenen Anwendungsvarianten unterschiedlich. Weshalb dieser Teil auch nicht von den Hardwarecontrollern sondern der Software behandelt wird. Der Datenblock

CRC

Die CRC gehört eigentlich in den Datenteil, wird aber traditionell vom Hardwarecontroller miterledigt. Weshalb man sie in der Praxis dem Rahmen zuordnet. Sie wird über alle Datenbytes, nicht aber das Start-Flag gebildet. Verwendet wird das CCITT-Polynom mit der Übertragung LSB zuerst. Eine genauere Beschreibung der Anomalien für HDLC findet sich in [1].



Das 1-Mark-Datum wird konstant mit dem Byte 01111110 gegeben. Durch dieses Zeichen wird der Empfänger auf die Verbindung hingelenkt. Danach wird die Länge der Daten angegeben, gefolgt von den Daten selbst. Danach kommt das 1-Mark-Datum.

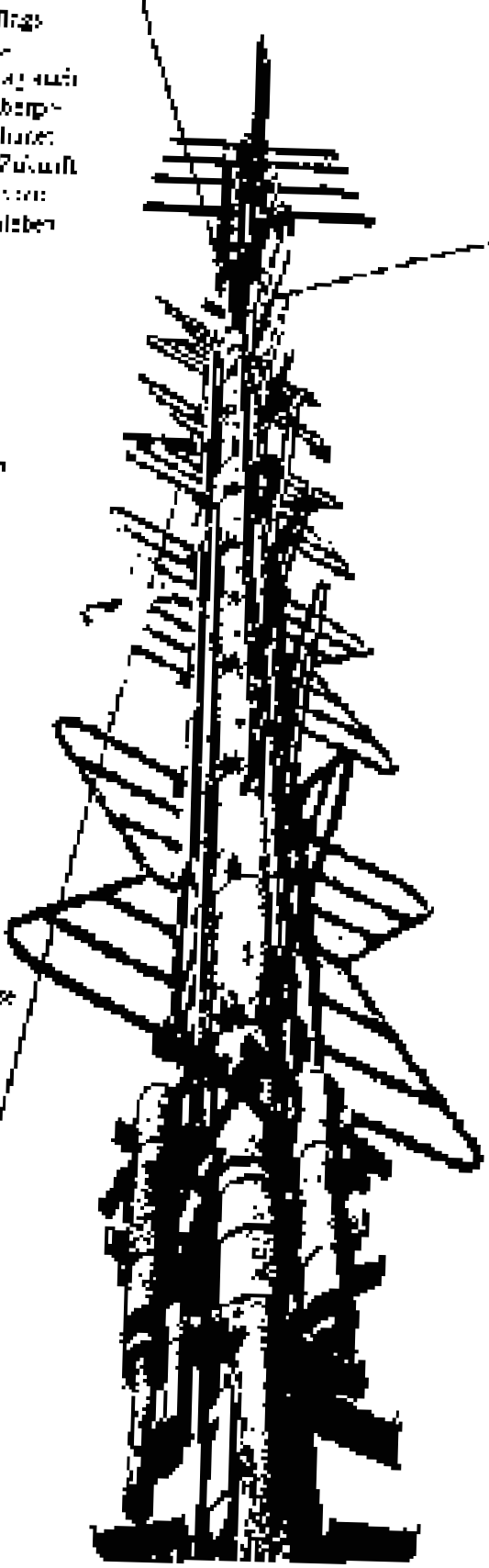
Das 1-Mark-Datum wird konstant mit dem Byte 01111110 gegeben. Durch dieses Zeichen wird der Empfänger auf die Verbindung hingelenkt. Danach wird die Länge der Daten angegeben, gefolgt von den Daten selbst. Danach kommt das 1-Mark-Datum.

Coden

Synchronisation: Die Leitung beginnt mit dem Sendersync-Mark, das die Datenposition markiert. Danach wird die Adresse angegeben. Danach folgt das Steuerfeld, das die Länge der Daten angibt. Danach folgen die Daten selbst. Danach folgt das FCS-Feld. Danach folgt das Stop-Mark.

Datensatz

Der Datensatz besteht aus einem Header, der die Länge der Daten angibt, gefolgt von den Daten selbst. Danach folgt das FCS-Feld. Danach folgt das Stop-Mark.



Beispiel 2: Bistufig

