

Der Open-Access-Publikationsserver der ZBW – Leibniz-Informationzentrum Wirtschaft
The Open Access Publication Server of the ZBW – Leibniz Information Centre for Economics

Mumm, Harald

Working Paper

Der vollständige Aufbau eines einfachen Fahrradcomputers

Wismarer Diskussionspapiere, No. 06/2005

Provided in cooperation with:

Hochschule Wismar

Suggested citation: Mumm, Harald (2005) : Der vollständige Aufbau eines
einfachen Fahrradcomputers, Wismarer Diskussionspapiere, No. 06/2005, <http://hdl.handle.net/10419/23315>

Nutzungsbedingungen:

Die ZBW räumt Ihnen als Nutzerin/Nutzer das unentgeltliche, räumlich unbeschränkte und zeitlich auf die Dauer des Schutzrechts beschränkte einfache Recht ein, das ausgewählte Werk im Rahmen der unter

→ <http://www.econstor.eu/dspace/Nutzungsbedingungen>
nachzulesenden vollständigen Nutzungsbedingungen zu vervielfältigen, mit denen die Nutzerin/der Nutzer sich durch die erste Nutzung einverstanden erklärt.

Terms of use:

The ZBW grants you, the user, the non-exclusive right to use the selected work free of charge, territorially unrestricted and within the time limit of the term of the property rights according to the terms specified at

→ <http://www.econstor.eu/dspace/Nutzungsbedingungen>
By the first use of the selected work the user agrees and declares to comply with these terms of use.



Hochschule Wismar

University of Technology, Business and Design

Fachbereich Wirtschaft



Hochschule Wismar

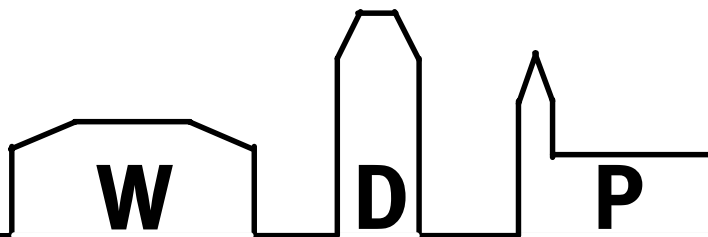
University of Technology, Business and Design

Faculty of Business

Harald Mumm

Der vollständige Aufbau eines einfachen
Fahrradcomputers

Heft 06 / 2005



Wismarer Diskussionspapiere / Wismar Discussion Papers

Der Fachbereich Wirtschaft der Hochschule Wismar, University of Technology, Business and Design bietet die Präsenzstudiengänge Betriebswirtschaft, Management sozialer Dienstleistungen, Wirtschaftsinformatik und Wirtschaftsrecht sowie die Fernstudiengänge Betriebswirtschaft, International Management, Krankenhaus-Management und Wirtschaftsinformatik an. Gegenstand der Ausbildung sind die verschiedenen Aspekte des Wirtschaftens in der Unternehmung, der modernen Verwaltungstätigkeit im sozialen Bereich, der Verbindung von angewandter Informatik und Wirtschaftswissenschaften sowie des Rechts im Bereich der Wirtschaft.

Nähere Informationen zu Studienangebot, Forschung und Ansprechpartnern finden Sie auf unserer Homepage im World Wide Web (WWW): <http://www.wi.hs-wismar.de/>.

Die Wismarer Diskussionspapiere/Wismar Discussion Papers sind urheberrechtlich geschützt. Eine Vervielfältigung ganz oder in Teilen, ihre Speicherung sowie jede Form der Weiterverbreitung bedürfen der vorherigen Genehmigung durch den Herausgeber.

Herausgeber: Prof. Dr. Jost W. Kramer
Fachbereich Wirtschaft
Hochschule Wismar
University of Technology, Business and Design
Philipp-Müller-Straße
Postfach 12 10
D – 23966 Wismar
Telefon: ++49/(0)3841/753 441
Fax: ++49/(0)3841/753 131
e-mail: j.kramer@wi.hs-wismar.de

Vertrieb: HWS-Hochschule Wismar Service GmbH
Phillipp-Müller-Straße
Postfach 12 10
23952 Wismar
Telefon:++49/(0)3841/753-574
Fax: ++49/(0)3841/753-575
e-mail: info@hws-startupfuture.de
Homepage: www.hws-startupfuture.de

ISSN 1612-0884
ISBN 3-910102-62-X

JEL-Klassifikation Z00

Alle Rechte vorbehalten.

© Hochschule Wismar, Fachbereich Wirtschaft, 2005.
Printed in Germany

Inhaltsverzeichnis

1. Einleitung	4
2. Der logische Aufbau der Hardware	5
2.1. Der Datenfluss in den Anzeigegeräten	6
2.2. Der Aufbau des Unterbrechungsprozessors	8
3. Die Verwaltersoftware	9
4. Das Zusammenspiel von Soft- und Hardware bei der Anzeige von Puls und Geschwindigkeit	9
5. Die Hauptspeicheraufteilung	11
6. Fazit	12
Anhang	13
Literatur	15
Autorenangaben	15

1. Einleitung

Der Ausgangspunkt dieser Arbeit ist die im Heft 11 der Wismarer Diskussionspapiere (2004) (s. [3]) vorgestellte Soft- und Hardware für einen einfachen Fahrradcomputer. Im Gegensatz zu der Arbeit wird die Software hier vereinfacht dargestellt, indem auf Systemaufrufe verzichtet wird. Die Software ist in der neuen Version auch vollständiger, weil Puls und Geschwindigkeit berechnet und angezeigt werden. Dabei wird die vergangene Zeit seit der letzten Unterbrechung vom Pulsfühler bzw. von der Nabe ausgewertet und mit entsprechenden Formeln in den aktuellen Puls bzw. in die Geschwindigkeit (km pro Stunde) umgerechnet. Wurde in [3] der Puls erst nach ca. einer Minute (Laufzeit des Leerlauf-Prozesses) angezeigt, erfolgt die Berechnung hier nach jeder Unterbrechung. Hier wären Mischvarianten denkbar, z. B. die Anzeige des Pulses nach mehreren Herzschlägen, die jedoch nicht Gegenstand dieser Arbeit sein sollen.

Die Hardware, insbesondere das Anzeigegerät und der Unterbrechungsprozessor werden detaillierter dargestellt als in [3]. Erstmals soll in dieser Arbeit die Speicheraufteilung für die beteiligten Prozesse vorgestellt und das Problem des Bootens, also des Ladens der Software in den Hauptspeicher, diskutiert werden.

Es wird hier davon ausgegangen, dass die Unterbrechungsbehandlung vollständig von der Verwaltersoftware durchgeführt wird und es lediglich einen Leerlaufprozess gibt. Dieser ist unbedingt notwendig, weil die Verwaltersoftware nicht unterbrechbar ist. In [3] wurde für jede Ziffernstelle eine extra Gerätenummer reserviert. Im folgenden gibt es nur noch die Geräteummern „1“ zur Anzeige des Pulses und „2“ zur Anzeige der Geschwindigkeit.

Neben den Anzeigegeräten war in [3] auch der Aufbau des Unterbrechungsprozessors offen gelassen worden. Diese Lücke wird hier geschlossen. Sowohl für die Anzeigegeräte als auch für den Unterbrechungsprozessor kommt der Steuersignalprozessor „Mikrospatz1321“ aus [4] zum Einsatz. Dieser Steuersignalprozessor arbeitet mit einem Mikroprogramm, das hier für die Anzeigegeräte und den Unterbrechungsprozessor offengelegt wird.

Bei jedem Computer sind die Speicheraufteilung und die Frage von Interesse, wie die Software in den Hauptspeicher des Computers geladen wird (Booten). Bei dem hier vorgestellten Fahrradcomputer auf der Basis der Neumann-Maschine „Paula“ aus [4] wird der Hauptspeicher in einen ROM- (read only memory) und einen sich anschließenden RAM-Bereich (read access memory) aufgeteilt, wobei davon ausgegangen wird, dass die gesamte Software im ROM-Bereich vorliegt, so dass auf ein Nachladen von Software vollständig verzichtet werden kann.

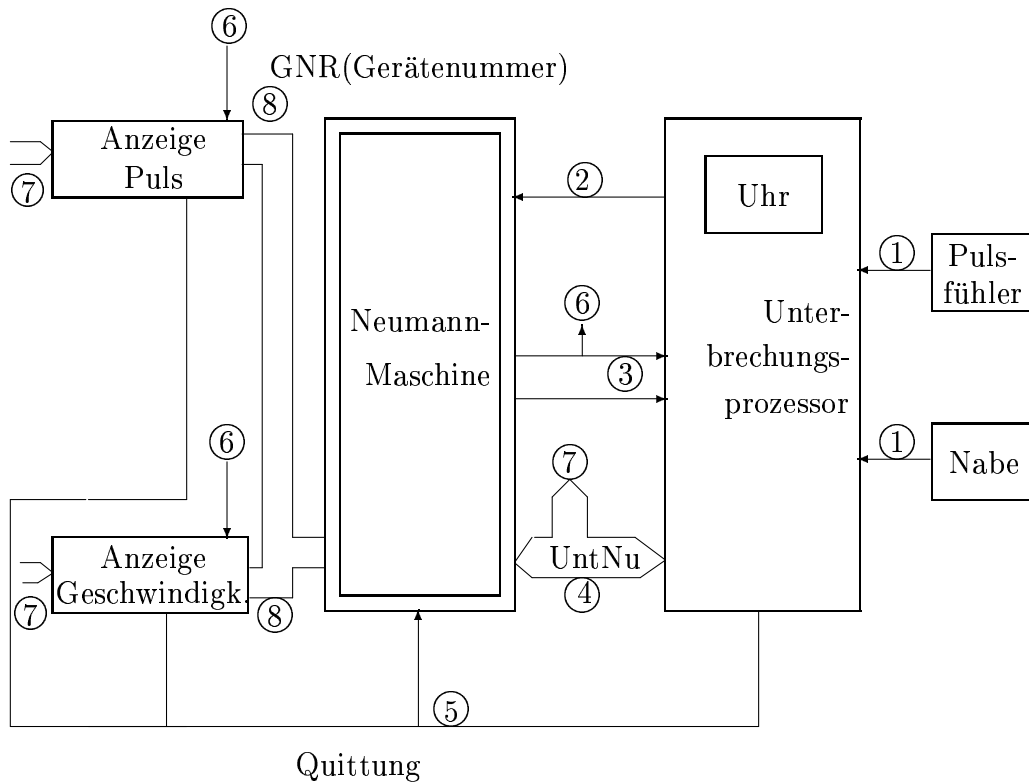
Die zur Steuerung der Unterbrechungsbehandlung mit der Neumann-Maschine „Paula“ erforderlichen Zahlenkonstanten am Ende des Hauptspeichers werden über einen kleinen Trick mit Hilfe von Maschinenanweisungen dort gespeichert bevor die eigentliche Arbeit beginnt.

Die Adresse <http://www.wi.hs-wismar.de/~mumm/ergoapp2/Ergo.html> zum Testen der Software behält ihre Gültigkeit.

2. Der logische Aufbau der Hardware

In [3] wurde die Anzeige der berechneten Werte nur symbolisch dargestellt, ohne zwischen der Anzeige von Puls und Geschwindigkeit zu differenzieren. Dieser Nachteil wird in der folgenden Abbildung beseitigt.

Abbildung 1: Die erweiterte schematische Darstellung der Hardware



Legende: UntNu ist die Nummer der Unterbrechung und kann den Wert 50, 51 oder 52 annehmen, je nachdem ob die Uhr, der Pulsfühler oder die Nabe eine Unterbrechung signalisiert haben.

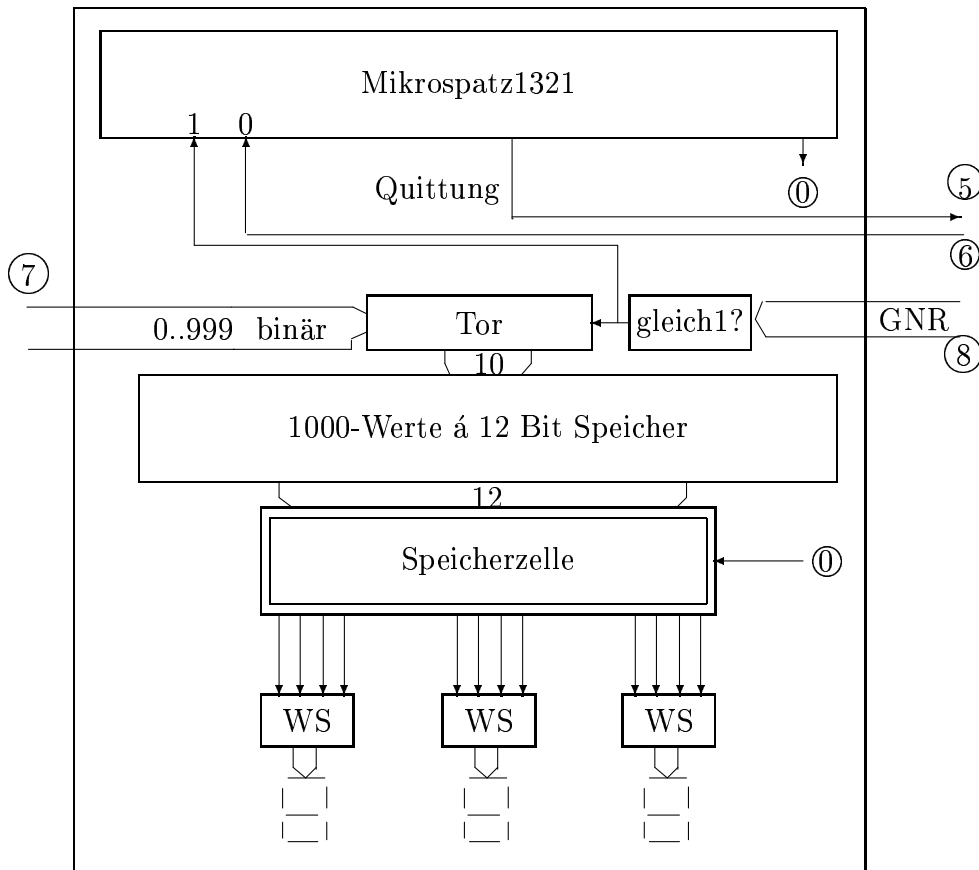
Das Auslösen von Unterbrechungen durch externe Geräte wurde in [3] wie folgt beschrieben: *Externe Geräte wie Pulsfühler und Nabe sind über einen Unterbrechungsprozessor, der eine Uhr enthält, am Rechner (Neumann-Maschine) angeschlossen. Erfolgt ein Unterbrechungswunsch (1 und 2) signalisiert der Rechner über zwei Bits (3), dass er eine Eingabe vom Unterbrechungsprozessor erwartet (die Nummer der Unterbrechung zu ihrer Identifizierung). Der Unterbrechungsprozessor stellt diese Nummer bereit (4) und signalisiert die Bereitstellung mit einem Bit (5), das hier Quittung heißt.*

Dieser Mechanismus bleibt hier bestehen. Neu kommt die Bereitstellung einer Gerätenummer (8) für die beiden Anzeigegeräte hinzu und der auszugebende Wert (7). Das Achtungsbit (6) der Neumann-Maschine wird jetzt auch an beide Anzeigegeräte geschickt, die ebenfalls ein Quittungsbit (5) senden, nachdem die Anzeige erfolgt ist.

2.1. Der Datenfluss in den Anzeigegeräten

Zur Anzeige von Puls und Geschwindigkeit (siehe Abbildung 1) werden Geräte mit dem logischen Aufbau nach Abbildung 2 verwendet.

Abbildung 2: Der Aufbau eines Anzeigegerätes



Legende: WS–Wertespeicher, GNR–Gerätenummer (1 oder 2), die Konnektoren 5 bis 8 beziehen sich auf die Konnektoren der Abbildung 1.

Die Geräte zur Anzeige von Puls und Geschwindigkeit unterscheiden sich äußerlich nicht, weil die Werte für Puls und Geschwindigkeit maximal dreistellige Zahlen sein können. (Eine Geschwindigkeit über 100 km pro Stunde auf dem Fahrrad ist aber schon eher unwahrscheinlich.)

Im Inneren unterscheiden sie sich nur bei der Schaltung „gleich1“, die den Wert „1“ liefert, falls ihre Eingabebits die Zahl Eins binär darstellen, ansonsten den Wert „0“. Für die Anzeige des Geschwindigkeitswertes wird ein zweites Anzeigegerät mit fast identischem Aufbau verwendet, bei dem die Schaltung „gleich1“ durch die Schaltung „gleich2“ ersetzt wurde, die den Wert „1“ liefert, falls ihre Eingabebits die Zahl Zwei binär darstellen, ansonsten den Wert „0“. Das Ausgabebit dieser beiden Schaltungen wird sowohl als Torwächter für die Eingabebitreihe der anzuzeigenden Zahl als auch als Testbit für den Steuersignalprozessor verwendet. Im folgenden Mikroprogramm des Steuersignalprozessors wird dieses Testbit in Mikroprogrammschritt Nr. 0 ausgewertet und zu Schritt Nr.

1 verzweigt, falls es gleich Eins ist. Im Schritt Nr. 1 wird das Achtungsbit der Neumann-Maschine (Konnektor 6) auf Eins getestet. Ist es gleich Eins, wird im Schritt Nr. 2 das Impulssignal Nr. 0 dazu verwendet, die Bitreihe aus dem „1000-Werte á 12 Bit Speicher“ dauerhaft der Speicherzelle zuzuweisen. Dadurch werden die richtigen Ziffern auf der LCD-Anzeige dargestellt, weil im Wertespeicher WS für alle zehn Ziffern Balkenansteuerungen gespeichert sind. Im Schritt Nr. 3 gibt das Steuerbit Nr. 9 der Neumann-Maschine das Quittungsbit, worauf sie ihre Arbeit fortsetzen kann (siehe Konnektor 5 in den Abbildungen 1 und 2).

Hier nun das vollständige Mikroprogramm eines Anzeigerätes:

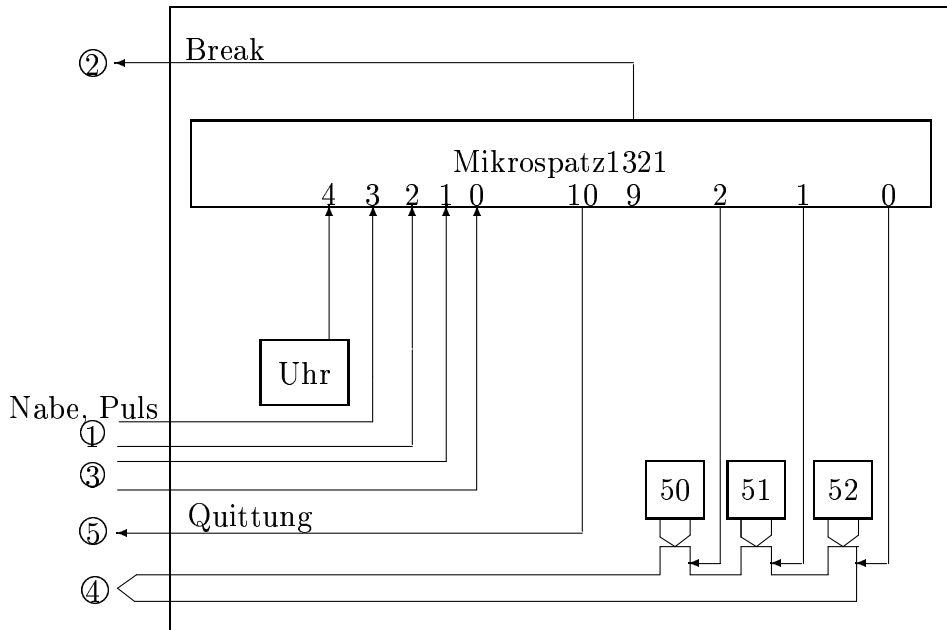
Tabelle 1: Das Mikroprogramm eines Anzeigerätes

Nr	Wahl	dann	sonst	Steuerbits
0	0	1	0	
1	1	2	0	
2	0	3	3	0
3	0	3	0	9 (Quittung)

2.2. Der Aufbau des Unterbrechungsprozessors

Die Hauptaufgabe des Unterbrechungsprozessors, wie er in der Abbildung 3 dargestellt ist, besteht im Herauslassen der Unterbrechungs-codes (hier 50, 51, 52) für die Neumann-Maschine, nachdem ein Unterbrechungssignal eingetroffen ist.

Abbildung 3: Der logische Aufbau des verwendeten Unterbrechungsprozessors



Legende: Die Konnektoren beziehen sich auf die Konnektoren der Abbildung 1.

Dieser Unterbrechungsprozessor verfügt über einen internen Steuersignalprozessor, der mit dem Mikroprogramm aus Tabelle 2 arbeitet. Die Geräte Uhr, Nabe und Pulsfühler liefern ihren Unterbrechungswunsch an die Testbits Nr. 4, 3, und 2 des internen Steuersignalprozessors. Die Uhr nimmt insofern eine Sonderstellung ein, als dass sie kein externes Gerät sondern ein Bestandteil des Unterbrechungsprozessors ist.

Wenn diese Testbits wahr (gleich Eins) sind, werden die Mikroprogrammschritte Nr. 3, 4 oder 5 aktiviert, deren Steuerbits Nr. 0, 1 oder 2 als Buswächter einer Busschaltung fungieren, die den Wert des Unterbrechungs-codes in den Bus hineinlassen. Der Ausgang dieses Busses wird zum Eingang der Neumann-Maschine (Konnektor 4 in Abbildung 1), wenn sie diesen Unterbrechungs-codes einlesen will. Gleichzeitig wird in den Schritten Nr. 3 bis 5 über das Steuersignal Nr. 9 der Unterbrechungswunsch an die Neumann-Maschine weitergeleitet (Konnektor 2 in den Abbildungen 1 und 3).

Das vollständige Mikroprogramm des Unterbrechungsprozessors hat den folgenden Aufbau:

Tabelle 2: Das Mikroprogramm des Unterbrechungsprozessors

Nr	Beschreibung	Wahl	dann	sonst	Steuerbits
0	Pulsbit	2	3	1	
1	Nabenbit	3	4	2	
2	Uhrbit	4	5	0	
3	Break-Wunsch 51	0	6	3	9,1
4	Break-Wunsch 52	0	6	4	9,0
5	Break-Wunsch 50	0	6	5	9,2
6	Bit 10 auch?	1	7	6	
7	Quittung geben	0	7	0	10

3. Die Verwaltersoftware

In der neuen Version gibt es im Vergleich zu [3] keine Systemaufrufe mehr. Die Anzeige der errechneten Werte erfolgt problemorientiert mit den beiden Anweisungen `gibAus(1,puls)` bzw. `gibAus(2,gesch)`. Der Quelltext im Anhang ist bis auf die beiden Funktionen `Produkt` und `Quotient` vollständig. Diese Funktionen sind notwendig, weil die Neumann-Maschine „Paula“ weder multiplizieren noch dividieren kann. Die mit `**` gekennzeichneten Anweisungen dienen lediglich der Generierung von Dummy-Maschinenanweisungen. Die generierten Maschinenanweisungen werden später per Hand so umgeschrieben, dass durch sie die erforderlichen Zahlenkonstanten am Ende des Hauptspeichers gesetzt werden können und dadurch eine klare Trennung von ROM- und RAM-Speicher möglich wird. Das ist notwendig, weil die Software nicht von externen Geräten, wie z. B. von einer Festplatte, geladen werden kann.

4. Das Zusammenspiel von Soft- und Hardware bei der Anzeige von Puls und Geschwindigkeit

Nach jeder Unterbrechung des Leerlaufprozesses durch die Geräte „Pulsfühler“ und „Nabe“ werden die aktuellen Werte von Puls und Geschwindigkeit durch die Anweisungen `gibAus(1,puls)` bzw. `gibAus(2,gesch)` zur Anzeige gebracht (siehe Quelltext im Anhang).

Die maschinensprachliche Umsetzung dieser beiden Anweisungen ergibt u.a. die Maschinenanweisungen `gibAusAn 1` für den Puls und `gibAusAn 2` für die Geschwindigkeit, durch die jeweils der zuvor in das Akkumulator-Register geladene Wert wie folgt zur Anzeige gebracht wird. (Das Anzeigegerät für den Puls hat die Nummer 1 und das Anzeigegerät für die Geschwindigkeit hat die Nummer 2.)

Bei der Interpretation dieser beiden Maschinenanweisungen durch das Mikroprogramm der Neumann-Maschine signalisiert ein Ausgabebit (Nr. 9) der Neumann-Maschine (Konnektor 6) beiden Geräten, dass eine Zahl zur Anzeige im Datenbus (Konnektor 7) bereit steht. Das Mikroprogramm der Anzeigegeräte

wechselt dadurch vom Mikroprogrammschritt 0 zum Mikroprogrammschritt 1.

Über die Schaltungen „gleich1?“ bzw. „gleich2?“ wird das Testbit Nr. 1 desjenigen Gerätes auf Eins gesetzt, dessen Nummer im GNR-Bus (Konnektor 8) vorliegt. Das Ausgabebit dieser beiden Schaltungen fungiert gleichzeitig als Torwächter für ein Tor der darzustellenden Zahl im Binärformat. Wird diese Zahl durch das Tor gelassen, übernimmt sie die Rolle der Adresse einer Reihe von 12 Bits im Wertespeicher von 1000 Werten, die zur Anzeige der Zahlen von „000“ bis „999“ mit einer LCD-Anzeige benötigt werden.

Wenn das Testbit Nr. 1 (des Steuersignalprozessors im Anzeigegerät) gleich Eins ist, wird der Mikroprogrammschritt Nr. 2 ausgeführt. Er bewirkt, dass durch ein Impulssignal (Konnektor 0) der Wert aus dem „1000-Werte a 12 Bit Speicher“ dauerhaft in einer Speicherzelle gespeichert wird. In den drei Wertespeichern vor den LCD-Anzeigen werden Bitreihen á 7 Bits zur Ansteuerung der Balken in der LCD-Anzeige vorgehalten, z. B. 1111111 für die Ziffer 8. Die Adresse einer derartigen Bitreihe, für das Beispiel ist sie gleich 1000, ergibt sich aus dem Wert in der Speicherzelle, d. h. jeweils vier Bits stellen eine derartige Adresse dar.

Zum Schluss wird durch den Mikroprogrammschritt Nr. 3 des Anzeigegerätes solange ein Quittungsbit auf Eins gesetzt, bis die Neumann-Maschine ihr Ausgabebit Nr. 9 zurückgesetzt hat (Konnektor 5).

5. Die Hauptspeicheraufteilung

Die maschinensprachliche Umsetzung des obigen Verwalterprogrammes **Fahrrad** belegt die Hauptspeicheradressen 0 bis 398. Das Leerlaufprogramm benötigt lediglich fünf Maschinenworte. Um Lücken im Hauptspeicher zu vermeiden, wird das Leerlaufprogramm auf den Plätzen mit den Adressen 399 bis 403 abgelegt. Der übliche Basisabstand des Verwalterprogrammes von 399 muss dazu auf 404 hochgesetzt werden. Das Leerlaufprogramm verwaltet selbst keine Daten. Am Ende des gesamten Hauptspeichers auf den Adressen 16777208 bis 16777215 benötigt die Neumann-Maschine Paula gewisse Daten zur richtigen Verwaltung des Prozesswechsels.

Tabelle 3: Die Speicherbelegung beim Fahrradcomputer

Adresse	Belegung	Bedeutung
0...398	Fahrradcode	Verwalterprogramm
399...403	Leerlaufcode	Unterbrechbarer Code
404...	Daten des Fahrradproz.	Keller und Halde
16777208	399	Adresse der ersten Leerlaufinstruktion
16777209	399	Adresse der nächsten Instruktion
16777210	404	Startadresse des Kellers (Leerlaufprozess)
16777211	0	Wert des Akkumulatorregisters
16777212	0	Adresse der ersten Fahrradinstruktion
16777213	227	Adresse der nächsten Instruktion
16777214	404	Startadresse des Kellers (Fahrradprozess)
16777215	0	Wert des Akkumulatorregisters

Die im obigen Programm **Fahrrad** mit ****** gekennzeichneten Anweisungen führen bei der maschinensprachlichen Umsetzung zu Maschinenanweisungen, die nachträglich vom Menschen auf Maschinenanweisungen zum Setzen der benötigten Konstanten auf den Adressen 16777208 bis 16777215 geändert werden müssen. Dadurch ist es möglich, die erforderliche Software auf einem ROM-Bereich (Adressen von 0 bis 404) auszuliefern. Der sich anschließende Teil des Hauptspeichers kann als RAM ausgeliefert werden.

Tabelle 4: Die Aufteilung des Hauptspeichers in RAM und ROM

Adresse	Art des Speichers
0...403	ROM
404...16777215	RAM

6. Fazit

Die Unterbrechungsbehandlung und der Prozesswechsel sind zwei sehr schwer vermittelbare Themen in der Ausbildung von Wirtschaftsinformatikerinnen und Wirtschaftsinformatikern. Dem Autor sind dazu keine Referenzen bekannt. Mit den Ausführungen in dieser Arbeit soll diese Lücke, zumindest was die Unterbrechungsbehandlung angeht, geschlossen werden. Ein echter Prozesswechsel war beim gewählten Demonstrationsbeispiel leider nicht möglich, weil es mit nur einem Anwendungsprozess, dem Leerlaufprozess, auskommt. Anders als noch in [3], wird die Unterbrechungsbehandlung hier vom nicht unterbrechbaren Verwalter vorgenommen. Eine Unterbrechungsbehandlung durch Kundenprozesse hätte den Nachteil, dass schon während ihrer Arbeit erneut Unterbrechungen auftreten könnten, die wiederum behandelt werden müssten. Als Ausweg ist hier eine Warteschlange von Unterbrechungen im Unterbrechungsprozessor denkbar, wodurch dieses Gerät aber seine Einfachheit wie in Abbildung 3 verlöre.

Bei der hier vorgestellten Lösung ist es auch möglich, dass Unterbrechungen während der Behandlung älterer Unterbrechungen auftreten. Dadurch würde die vorgestellte Lösung aber nicht ins Chaos stürzen, sondern diese neuen Unterbrechungen würden einfach nicht registriert werden. Dieser Fall kann besonders bei der Simulation auftreten, weil die simulierte Neumann-Maschine lediglich eine Taktfrequenz von maximal 3000 Hertz erreicht.

Als weiterer Nachteil der vorgestellten Lösung soll erwähnt werden, dass es nur einen banalen Prozess, den Leerlaufprozess gibt. Um diesen Nachteil beseitigen zu können, müsste man einen zweiten Prozess, z. B. den eines Schachspieles installieren. Das führte jedoch zu einer gewaltigen Ausdehnung der Eingabe- und Ausgabeschnittstelle, was dem eigentlichen Anliegen der Arbeit, die **vollständige** Beschreibung einer einfachen Anwendung, widersprechen würde.

Unerwähnt bleibt in dieser Arbeit auch die Behandlung von Messfehlern. Sie können z. B. durch eine falsche Verwendung des Pulsfühlers entstehen. Dadurch kann es zur Anzeige von medizinisch unmöglichen Pulswerten kommen, die den Nutzer beunruhigen könnten. Bei kommerziellen Fahrradcomputern wird in derartigen Fällen ein spezielles Herzpiktogramm anstelle eines Pulswertes angezeigt. Die Erweiterung der vorgestellten Lösung auf diesen Spezialfall ist nicht besonders aufwendig und könnte als Übungsaufgabe gestellt werden (siehe Anhang).

Anhang 1: Der Quelltext der Software

```
Programm Fahrrad;
Konstante OffsetEpa=3;
Konstante vonUhr =50;
Konstante vonHerz=51;
Konstante vonNabe=52;
Konstante KLeer = 16777211;
Konstante tps = 10;//Ticks p.Sek.
Konstante Radumfang = 200; //cm

Typ Prozess = Klasse (
    Epa    : Adresse;
    Nia    : Adresse;
    Basis  : Adresse;
    Akkuw  : ganzzahl;

Methode werdeAktiv:ganzzahlImAkku
Beginn
    aktiviere;
Ende;

);

Typ Betrieb = Klasse (
    bereit: Prozess;
    ticks1 : ganzzahl;
    ticks2 : ganzzahl;

Methode holeBereit : Prozess;
Beginn
    liefereAlsFunktionswert(bereit);
Ende;

Methode hole1ti: ganzzahl;
Beginn
    liefereAlsFunktionswert(ticks1);
Ende;

Methode hole2ti: ganzzahl;
Beginn
    liefereAlsFunktionswert(ticks2);
Ende;

Methode setze1ti: ganzzahl;
Beginn
    ticks1:= 0;
    liefereAlsFunktionswert(0);
Ende;

Methode setze2ti: ganzzahl;
Beginn
    ticks2 := 0;
    liefereAlsFunktionswert(0);
Ende;

Methode initBD: ganzzahl;
Beginn
    ticks1 := 0;
    ticks2 := 0;
    bereit := KLeer;
    liefereAlsFunktionswert(0);
Ende;

Methode erhoeheTicks: ganzzahl;
Beginn
    ticks1 := ticks1 + 1;
    ticks2 := ticks2 + 1;
    liefereAlsFunktionswert(0);
Ende;

);

Variable Kunde: Prozess;
Variable bv: Betrieb;
Variable dummy: Reihe[1..3] von
ganzzahl;
Variable rc: ganzzahl;
Variable Aufruf: ganzzahl;
Variable ticks : ganzzahl;
Variable puls  : ganzzahl;
Variable gesch : ganzzahl;
Variable hi1   : ganzzahl;
Variable hi2   : ganzzahl;
Variable sds1,...,sds9: Adresse;
Beginn
    sds1 := 1; sds2 := 2;//**
    sds3 := 3; sds4 := 4;//**
    sds5 := 5; sds6 := 6;//**
    sds7 := 7; sds8 := 8;//**
    sds9 := 9;          /**
    bv := & dummy; //Platz schaffen
    rc := bv.initBD;
    Kunde:= bv.holeBereit;
    solange wahr wiederhole Beginn
```

```

    Aufruf := Kunde.werdeAktiv;
    falls Aufruf = vonUhr dann
        rc := bv.erhoeheTicks
    sonst
        falls Aufruf = vonHerz dann
Beginn
ticks := bv.hole1ti;
rc := bv.setze1ti;
puls := Quotient(600,ticks);
//600=60*tps
gibAus(1,puls) ;
Ende

    sonst
        falls Aufruf = vonNabe dann
Beginn
ticks := bv.hole2ti;
rc := bv.setze2ti;

hi1 := Produkt(Radumfang,360);
//360=36*tps
hi2 := Produkt(1000,ticks);
gesch := Quotient(hi1,hi2);

gibAus(2,gesch);
Ende;

Ende; //solange
Ende! //Fahrrad

```

Anhang 2: Übungsaufgaben

Der vorgestellte Fahrradcomputer soll derart geändert werden, dass bei sehr kleinen Pulswerten (unter 30) anstelle des Wertes eine andere Anzeige, der Einfachheit halber z. B. „999“ erfolgen soll. Welche Änderungen müssten dazu an welcher Stelle vorgenommen werden? Wie müsste das Anzeigegerät für den Puls verändert werden, damit in der Komfortvariante sogar ein spezielles Herzpiktogramm für derartige Fälle zur Verfügung stünde?

Der vorgestellte Fahrradcomputer soll derart geändert werden, dass die Anzeige des Pulses nicht nach jedem Pulsschlag, sondern z. B. erst nach fünf Pulsschlägen erfolgt. Um welche Variablen und Methoden müsste dazu die Klasse „Betrieb“ erweitert werden? Welche Änderungen wären weiterhin im Programm „Fahrrad“ vorzunehmen? Wie müsste die Hardware verändert werden, damit zusätzlich bei jedem Pulsschlag ein Herzpiktogramm erscheinen kann und erst nach mehreren Pulsschlägen ein neuer Pulswert angezeigt wird?

Literatur

- [1] **Mumm**, Harald (2003): Entwurf und Implementierung einer objektorientierten Programmiersprache für die Paula-Virtuelle-Maschine, Hochschule Wismar, Fachbereich Wirtschaft 2003, WDP – Wismarer Diskussionspapiere Heft 8/2003, ISBN 3-910102-32-8.
- [2] **Mumm**, Harald (2004): Die Wirkungsweise von Betriebssystemen am Beispiel der Tastatur-Eingabe, Hochschule Wismar, Fachbereich Wirtschaft 2004, WDP – Wismarer Diskussionspapiere Heft 2/2004, ISBN 3-910102-43-3.
- [3] **Mumm**, Harald (2004): Unterbrechungsgesteuerte Informationsverarbeitung, Hochschule Wismar, Fachbereich Wirtschaft 2004, WDP – Wismarer Diskussionspapiere Heft 11/2004, ISBN 3-910102-54-9.
- [4] **Röck**, Hans (2002): Arbeitsmaterial Neumann-Maschine, Universität Rostock, Sommersemester 2002.
- [5] **Röck**, Hans (2003): Arbeitsmaterial Betriebssysteme, Universität Rostock, Wintersemester 2003.

Autorenangaben

Prof. Dr. rer. nat. Harald Mumm
Fachbereich Wirtschaft
Hochschule Wismar
Philipp-Müller-Straße 14
Postfach 12 10
D – 23966 Wismar
Telefon: ++49 / (0)3841 / 753 450
Fax: ++49 / (0)3841 / 753 131
E-mail: harald.mumm@wi.hs-wismar.de

WDP - Wismarer Diskussionspapiere / Wismar Discussion Papers

- Heft 01/2003 Jost W. Kramer: Fortschrittsfähigkeit gefragt: Haben die Kreditgenossenschaften als Genossenschaften eine Zukunft?
- Heft 02/2003 Julia Neumann-Szyszka: Einsatzmöglichkeiten der Balanced Scorecard in mittelständischen (Fertigungs-)Unternehmen
- Heft 03/2003 Melanie Pippig: Möglichkeiten und Grenzen der Messung von Kundenzufriedenheit in einem Krankenhaus
- Heft 04/2003 Jost W. Kramer: Entwicklung und Perspektiven der produktivgenossenschaftlichen Unternehmensform
- Heft 05/2003 Jost W. Kramer: Produktivgenossenschaften als Instrument der Arbeitsmarktpolitik. Anmerkungen zum Berliner Förderungskonzept
- Heft 06/2003 Herbert Neunteufel/Gottfried Rössel/Uwe Sassenberg: Das Marketingniveau in der Kunststoffbranche Westmecklenburgs
- Heft 07/2003 Uwe Lämmel: Data-Mining mittels künstlicher neuronaler Netze
- Heft 08/2003 Harald Mumm: Entwurf und Implementierung einer objektorientierten Programmiersprache für die Paula-Virtuelle-Maschine
- Heft 09/2003 Jost W. Kramer: Optimaler Wettbewerb – Überlegungen zur Dimensionierung von Konkurrenz
- Heft 10/2003 Jost W. Kramer: The Allocation of Property Rights within Registered Co-operatives in Germany
- Heft 11/2003 Dietrich Nöthens/Ulrike Mauritz: IT-Sicherheit an der Hochschule Wismar
- Heft 12/2003 Stefan Wissuwa: Data Mining und XML. Modularisierung und Automatisierung von Verarbeitungsschritten
- Heft 13/2003 Bodo Wiegand-Hoffmeister: Optimierung der Sozialstaatlichkeit durch Grundrechtsschutz – Analyse neuerer Tendenzen der Rechtsprechung des Bundesverfassungsgerichts zu sozialen Implikationen der Grundrechte -
- Heft 14/2003 Todor Nenov Todorov: Wirtschaftswachstum und Effektivität der Industrieunternehmen beim Übergang zu einer Marktwirtschaft in Bulgarien
- Heft 15/2003 Robert Schediwy: Wien – Wismar – Weltkulturerbe. Grundlagen, Probleme und Perspektiven
- Heft 16/2003 Jost W. Kramer: Trends und Tendenzen der Genossenschaftsentwicklung in Deutschland
- Heft 01/2004 Uwe Lämmel: Der moderne Frege
- Heft 02/2004 Harald Mumm: Die Wirkungsweise von Betriebssystemen am Beispiel der Tastatur-Eingabe
- Heft 03/2004 Jost W. Kramer: Der Einsatz strategischer Planung in der Kirche
- Heft 04/2004 Uwe Sassenberg: Stand und Möglichkeiten zur Weiterentwick-

- lung des Technologietransfers an der Hochschule Wismar
- Heft 05/2004 Thomas Gutteck: Umfrage zur Analyse der Kunden des Tourismuszentrum Mecklenburgische Ostseeküste GmbH
- Heft 06/2004: Anette Wilhelm: Probleme und Möglichkeiten zur Bestimmung der Promotions-effizienz bei konsumentengerichteten Promotions
- Heft 07/2004: Jana Otte: Personalistische Aktiengesellschaft
- Heft 08/2004 Andreas Strelow: VR-Control – Einführung eines verbundeinheitlichen Gesamtbanksteuerungskonzepts in einer kleinen Kreditgenossenschaft
- Heft 09/2004 Jost W. Kramer: Zur Eignung von Forschungsberichten als einem Instrument für die Messung der Forschungsaktivität
- Heft 10/2004 Jost W. Kramer: Geförderte Produktivgenossenschaften als Weg aus der Arbeitslosigkeit? Das Beispiel Berlin
- Heft 11/2004 Harald Mumm: Unterbrechungsgesteuerte Informationsverarbeitung
- Heft 12/2004 Jost W. Kramer: Besonderheiten beim Rating von Krankenhäusern
- Heft 01/2005 Michael Laske/Herbert Neunteufel: Vertrauen eine „Conditio sine qua non“ für Kooperationen?
- Heft 02/2005 Nicole Uhde: Rechtspraktische Probleme bei der Zwangseinziehung von GmbH-Geschäftsanteilen – Ein Beitrag zur Gestaltung von GmbH-Satzungen
- Heft 03/2005 Kathrin Kinder: Konzipierung und Einführung der Prozesskostenrechnung als eines Bestandteils des Qualitätsmanagements in der öffentlichen Verwaltung
- Heft 04/2005: Ralf Bernitt: Vergabeverfahren bei öffentlich (mit)finanzierten sozialen Dienstleistungen
- Heft 05/2005: Jost W. Kramer: Zur Forschungsaktivität von Professoren an Fachhochschulen, untersucht am Beispiel der Hochschule Wismar
- Heft 06/2005 Harald Mumm: Der vollständige Aufbau eines einfachen Fahrradcomputers