MP35-5
80535 Controller Modul
mit Flash-Memory

1. FUNKTION

1.1. Datenblatt

1.1.1. Anwendung

Universelles Controller-Modul mit integriertem RS232 Interface, 5 x 8 Bit Ports, 3 Kanal Timer und 8 x 10 Bit ADC.

Verwendung für eigenständige Messdatenerfassung und Steuerung.

1.1.2. Daten

Parameter	Wert	Bemerkung
Controller	80C535	
RAM	32KB	..64KB erw.bar
Flash-EPROM	128KB	
Systemclock	12MHz	(bzw. 11,0592MHz)
Befehlszeiten	>=1µs	meiste Befehle

1.1.3. Besonderheiten

- Vordekodierung von 8 Select Leitungen.
- PAL zur flexiblen Einstellung von Steuerleitungen
- Status LED
- 4 Setups (Pages) programmierbar

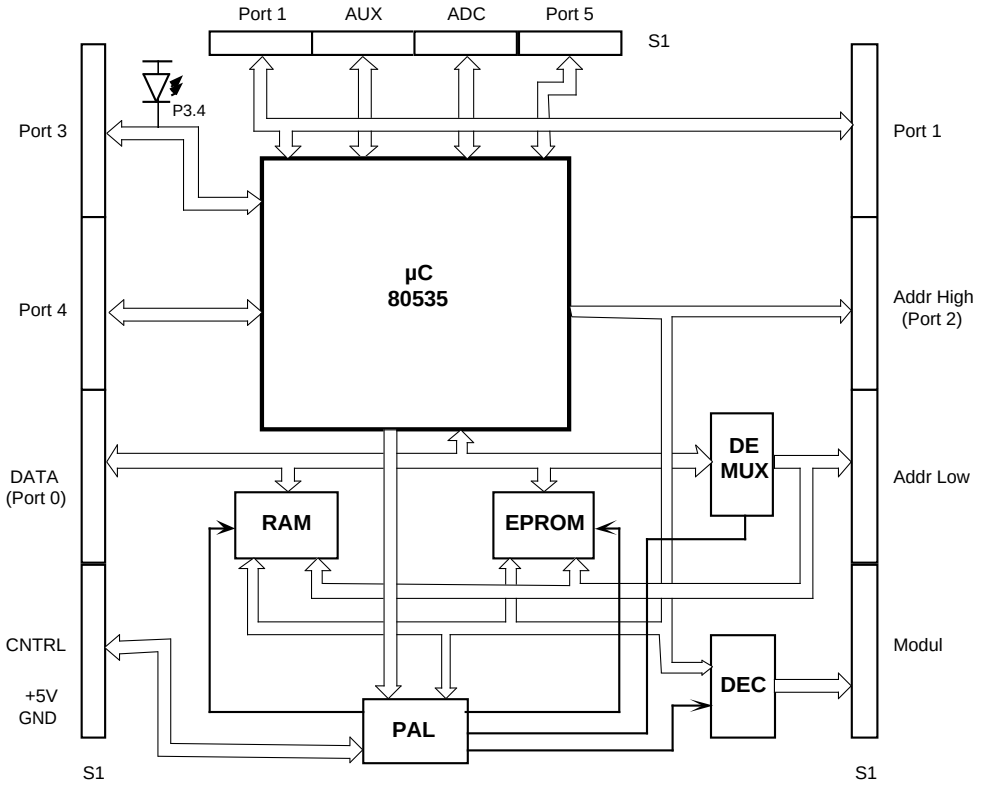
1.1.4. Aufbau

Scheckkartengröße 53 * 83 mm mit seitlichen Busstiften. Frontstecker für zusätzliche Signale.

1.1.5. Stromversorgung

Spannung	Strom	Leistung
+5V	ca. 120mA	600mW
Gesamt		600mW

1.2. Blockdiagramm



1.3. Beschreibung

Ein **Single-Chip-Controller** (80535) aus der 8051 Familie stellt alle grundsätzlichen I/O Funktionen zur Verfügung. Ein statisches **RAM** realisiert einen 32KByte großen Arbeitsspeicher. Die Programme sowie weitere Daten (nichtflüchtig) können in einem **Flash-EPROM** untergebracht werden.

Alle Ports (8 Bits) des Controllers sind an externen Pins zur allgemeinen Verwendung herausgeführt:

Port 0: (bidirektional 8 TTLS). **Datenbus** (D0..7) und **LowAdressbus** (A0..7) gemultiplext. Auf dem Board befindet sich ein Demultiplexer (U3) um für normale Speicher auch diese Adressen separat zur Verfügung zu haben.

Port 1: (bidirektional 4 TTLS). Frei verwendbar oder alternativ Spezialfunktionen für Timer und Interrupt.

Port 2: (bidirektional 4 TTLS). **HighAdressbus** (A8..15).

Port 3: (bidirektional 4 TTLS).
Steuersignale RD (P3.7), WR (P3.6).
Serielle Schnittstelle TxD (P3.1), RxD (P3.0).
Setup (P3.2), (P3.4). Bei Beschränkung auf 1 Setup (anderes PAL) frei verwendbar.
Timer (P3.5) und **Interrupt** (P3.3). Frei verwendbar.

Port 4: (bidirektional 4 TTLS). Frei verwendbar.

Port 5: (bidirektional 4 TTLS). Frei verwendbar.

ADC: 8 Kanal ADC mit 8 Bit (durch Steuerung der Referenz bis 10Bit, C515C direkt 10Bit) Auflösung.

AUX: weitere Steuersignale des Controllerbausteins.

MODUL: Durch einen **Decoder** werden aus dem 32KB Addressbereich 8*4KByte große Fenster vordekodiert und als Select-Leitungen (Low active) zur Verfügung gestellt.

CNTRL: Hier liegen die Pins zur Stromversorgung (GND, +5V) sowie Signale zur Systemsteuerung (RES,INT). 4 weitere Signale (C0..C2, FA16) können über ein **PAL** programmiert und zur komplexeren Ansteuerung weiterer Komponenten verwendet werden.

Eine frei programmierbare **Status-LED** (Pin 3.3) auf dem Board kann z.B. zur Fehleranzeige verwendet werden.

1.4. Anschlußsignale

PIN	Symbol	Eing./Ausg	Funktion	
1-3 5-9	P4.0-4.2 P4.3-4.7	E/A	Port 4 ist bitadressierbar, 8 Bit und bidirektional. Er kann 4 TTLS treiben.	
4	VPD		Ruhestromversorgung für 40-Byte-RAM (58H bis 7FH). Ist VCC > VPD, wird dieser Bereich von VCC versorgt.	
10	_RESET	E	Liegt dieser Pin während 2 Maschinenzyklen auf LOW, wird der 80515 rückgesetzt.	
11	VAREF		5V Referenzspannung des A/D-Wandlers	
12	VAGND		0V Referenzspannung des A/D-Wandlers	
13-20	AN7-AN0	E	Gemultiplexte Analog-Eingänge	
21-28	P3.0-3.7	_E/_A	Port 3 ist bitadressierbar, 8 Bit und bidirektional. Er kann 4 TTLS treiben. Jedem Pin ist eine alternative Funktion zugeordnet. Wird diese benötigt, ist vorher das Ausgangs-Latch auf HIGH zu programmieren. Die Alternativfunktionen sind:	
21	P3.0	E/A	RXD:	serieller Port-Daten-Eingang (async), Ein-/Ausgang (sync).
22	P3.1	E/A	TXD:	serieller Port-Daten-Ausgang (async), Takt Ausgang (sync).
23	P3.2	E/A	_INT0:	Interrupt 0 Eingang, Timer 0 Gate-Eingang.
24	P3.3	E/A	_INT1:	Interrupt 1 Eingang, Timer 1 Gate-Eingang.
25	P3.4	E/A	T0:	Zähler 0 Eingang
26	P3.5	E/A	T1:	Zähler 1 Eingang
27	P3.6	E/A	_WR:	Steuersignal für externe Datenspeicher, es übergibt das Datenbyte an Port 0 zum externen Speicher.
28	P3.7	E/A	_RD:	Steuersignal für externe Speicher. Es übernimmt das Datenbyte in Port 0.
29-36	P1.7-1.0	E/A	Port 1 ist bitadressierbar, bidirektional und 8 Bit breit. Er kann 4 TTLS treiben. Jedem Pin ist eine alternative Funktion zugeordnet. Wird diese benötigt, ist vorher das Ausgangs-Latch auf HIGH zu programmieren. Die Alternativfunktionen sind :	

29 30 31 32 33 34 35 36	P1.7 P1.6 P1.5 P1.4 P1.3 P1.2 P1.1 P1.0	E/A E/A E/A E/A E/A E/A E/A E/A	T2: CLKOUT: T2EX: INT2: INT6/CC3: INT5/CC2: INT4/CC1: _INT3/CC0:	Timer 2 Eingang. Systemtakt Ausgang. externer Reload-Eingang. externer Interrupt 2. Interrupt 6- Capture 3 Eingang, Compare 3 Ausgang. Interrupt 5- Capture 2 Eingang, Compare 2 Ausgang. Interrupt 4- Capture 1 Eingang, Compare 1 Ausgang. Interrupt 3- Capture 0 Eingang, Compare 0 Ausgang.
37	VBB		Substrat Pin. Dieser Pin muß über eine Kapazität (10 bis 100 nF) an GND liegen.	
39	XTAL2	E/A	Ausgang des Oszillatorverstärkers. Es ist ein Quarz, Keramikschwinger oder eine externe Taktquelle verwendbar.	
40	XTAL1	E	Eingang des Oszillatorverstärkers. Der Pin muß auf Masse liegen, wenn an XTAL2 ein externes Taktsignal eingespeist wird.	
41-48	P2.0-2.7	E/A	Port 2 ist bitadressierbar, 8 Bit breit und bidirektional. Er kann bis 4 TTLS treiben. Beim Anschluß externer Speicher gibt P2 das höherwertige Adreßbyte aus.	
49	_PSEN	A	Ext. Programmspeicher-Enable Ausgang.	
50	ALE	A	Address-Latch-Enable. Mit diesem Signal wird das niederwertige Adreßbyte extern zwischengespeichert.	
51	EA	E	Bei HIGH führt die CPU Befehle vom internen Befehlsspeicher aus, wenn der Inhalt des PC kleiner als 8191 (1FFFH) ist. Bei LOW werden Befehle nur vom externen Befehlsspeicher ausgeführt. EA muß beim 80535 auf LOW gelegt werden.	
52-59	P0.0-0.7	E/A	Port 0 ist bitadressierbar, 8 Bit breit und bidirektional. Er kann bis 8 TTLS treiben. Beim Anschluß externer Speicher gibt P0 zuerst das niederwertige Adreßbyte aus, dann folgt das Einlesen des Befehlsbyte bzw. das Ein- oder Ausgeben des Datenbyte.	
60-67	P5.7-5.0	E/A	Port 5 ist bitadressierbar, bidirektional und 8 Bit breit. Er treibt 4 TTLS.	
68	VCC		Spannungsversorgung (+5V)	
38	VSS		Masse (0V)	

2. BETRIEB

2.1. Konfigurierung

2.1.1. Jumper (fett=vorverdrahtet):

Betriebsart	Jumper	Bemerkung
freie Verwendung	J1: 1-2	Eingang an PAL I6=ALE
freie Verwendung	**J1: 2-3**	Eingang an PAL I6=P3.5
Externes ROM benutzen	**J2: 1-2**	EA low für 80535 (normal!)
Internes ROM benutzen	J2: 2-3	EA für 80515 (ROM-Version)
Ruhestromversorgung f. 40B RAM	J3: 1-2	VPD an Vcc (Versorgung d. Vcc)
Ruhestromversorgung f. 40B RAM	J3: 2-3	VPD an GND (Versorgung d. Vcc)
freie Verwendung	**J4: 1-2**	Eingang an PAL I1=A14
freie Verwendung	J4: 2-3	Eingang an PAL I1=P3.3

2.2. Speicherbelegung

Das Flash-EPROM ist in 8 Segmente à 16KB (**F0..F7**) aufgeteilt.

Das RAM ist in 2 Segmente à 16KB (**R0..R1**) aufgeteilt.

Je nach Setup (**S0..S3**) sind die einzelnen Segmente im Memory unterschiedlich lokalisiert. Die Setups werden durch die PortBits [P3.4, P3.2] eingestellt.

2.2.1. Setup 0 (S0)

Start	CODE (PSEN) nur lesen	DATA (RD/WR) lesen&schreiben
$C000	F3	M4..7
$8000	F2	M0..3
$4000	F1	R1
$0000	F0	R0

Flash F0…F3 enthält das Programm und/oder fixe Daten.

RAM R0…R1 enthält variable Daten.

Module M0…M7 mit Vordekodierung.

2.2.2. Setup 1 (S1)

Start	CODE (PSEN)	DATA (RD/WR)
	nur lesen	lesen&schreiben
$C000	F3	F5
$8000	F2	F4
$4000	R1	R1
$0000	F0	R0

Flash F0, F2, F3 für Programme und/oder fixe Daten.
Flash F4…F5 für Programmierung (Beschreiben) und Daten.
RAM R0 für variable Daten.
RAM R1 für Daten und Programme (vonNeumann).

2.2.3. Setup 2 (S2)

Start	CODE (PSEN)	DATA (RD/WR)
	nur lesen	lesen&schreiben
$C000	F7	M4..7
$8000	F6	M0..3
$4000	R1	R1
$0000	R0	R0

Flash F6, F7 für Programme und/oder fixe Daten (typisch MONITOR).
RAM R0, R1 für Daten und Programme (vonNeumann).
Module M0…M7 mit Vordekodierung.

2.2.4. Setup 3 (S3)

Start	CODE (PSEN)	DATA (RD/WR)
	nur lesen	lesen&schreiben
$C000	F7	F1
$8000	F6	F0
$4000	R1	R1
$0000	F6	R0

Flash F6, F7 für Programme und/oder fixe Daten (typisch MONITOR).

Flash F0…F1 für Programmierung (Beschreiben) und Daten (Applikation).

RAM R0 für variable Daten.

RAM R1 für Daten und Programme (vonNeumann).

2.3. Bedienung

2.3.1. Reset

Beim **Reset** ist immer der **Setup 3** eingeschaltet und ein Programm wird an der Code-Speicherstelle **$0000** (F6) gestartet.

2.3.2. Monitor MON51

Zur Entwicklung von Programmen und zum weiteren Laden von Applikationen ist ein Monitor (**MON51**) im F6, F7-Segment bereits vorhanden, der bei Reset somit gestartet wird.

Als erstes überprüft MON51 ob die Leitung **INT** im LOW-Zustand ist. In diesem Fall wird immer in MON51 fortgesetzt.

Andernfalls überprüft MON51 dann die Adresse **$0000** in **F0**. Falls die Daten dort ungleich **$FF** sind, wird dort der Start eines lauffähigen Programms angenommen, welches nach Umschalten in den **Setup 0** gestartet wird. Das dort befindliche Programm übernimmt dann die Kontrolle (Autostart).

ACHTUNG: Da der Monitor im Augenblick der Setupumschaltung nicht mehr existent wäre, wird ein kleines (temporäres) Hilfsprogramm in das Segment R1 geladen, welches zum Umschalten und Starten des Hauptprogrammes verwendet wird.

Andernfalls wird in MON51 fortgesetzt. Der Monitor schaltet anschließend auf den **Setup 2** um, der dem typischen Anwenderbereich mit den Datensegmenten R0,R1, M0..M7 entspricht. R0, R1 sind hier auch in den Codebereich gespiegelt um das Laden und Ausführen (Testen) von Programmen dort zu ermöglichen. In F6, F7 läuft nachwievor der Monitor **MON51.** Es kann aber im Monitor auch jederzeit in den **Setup 3** umgeschaltet werden um z.B. die Programmierung der Flash-Segmente F0, F1 durchzuführen.

Zum Start auf dem Terminal dann mehrmals die **SPACE**-Taste drücken, bis der Controller z.B. mit der Systemmeldung:

```
--- MON51 5.0 (9600 Baud @ 11,0592MHz) ---
```

antwortet. Damit erfolgt eine automatische Adaptierung an die verwendete Baudrate. Unterstützt wird mit Systemclock:

12MHz: 1200, 2400, 4800 Baud

11,0592 MHz: 1200, 2400, 4800, 9600, 19200 Baud

Alle Kommandos sind in ASCII und werden bereits durch das Senden eines entsprechenden Zeichens (Buchstaben) ohne <CR> ausgelöst.

Parameter (Zahlen) werden vorangestellt und sind durch das Kommando abgeschlossen! Spezielle Kommandos (. , ;) erlauben das Setzen bestimmter Parameter (Addr, Data, Size). Ansonsten ergibt sich die Bedeutung des Parameters aus dem Kommando. Die Eingabe kann auch in Hex mit vorgestelltem $ erfolgen (z.B: $A1 =161). Gesendete Parameter werden immer mit einem <CR> beendet.

Eingegebene Zeichen werden nur in "Verbose=ON" zurückgesendet (Echo).

Der Monitor beansprucht keinerlei RAM. Lediglich im Falle der Programmierung des Flash-Speichers wird ein kleiner 256 Byte großer Bereich (Scratch: normalerweise $7F00…$7FFF, einstellbar mit &) beansprucht.

Mit dem Befehl "?" können die verfügbaren Kommandos und eine kurze Erläuterung aufgelistet werden, zB:

```
--- MP46 5.0 (9600 Baud @ 10MHz) ---
Author: vWalter 301100
?        HELP (this screen)
: ...    HEX load
.        set,get Addr
,        set,get Data
;        set,get Size
;+/-     add/subtr Data; to/from Addr.
.G/.g    GO to Addr./CDATA read from Addr.
,X/.x    XDATA write Data,/read from Addr.
,I/.i    IDATA write Data,/read from Addr.
,J/.j    DATA  write Data,/read from Addr.
,P/.p    PORT  write Data,/read from Port.
,O/,o    OUT  OR/AND Data, to  (1-5) Port.
,M/.m    MOVE  Flash/block Size; from. to,
.Z/.z    ZERO(FF)/INFO MemorySegm at Addr.
,S/s     SETUP (0..3) set/get
.&       set,get Scratch RAM
V/v      Verbose On/Off
```

Beschreibung:

?	Ausgabe aller Befehle in Kurzform
:	Lädt HEX listing und setzt ADDR , DATA, SIZE automatisch. Somit kann mit dem unmittelbar anschließendem Befehl "M" der eingelesene Speicherbereich automatisch nach in F0 und F1 ($8000..) geladen werden.
a.	Setzt interne Adressvariable ADDR auf a (z.B.: $1000.)
.	Ausgabe ADDR
d,	Setzt interne Datenvariable DATA auf d (z.B.: $FF,)
,	Ausgabe DATA
s;	Setzt interne Grössenvariable SIZE auf s (z.B.: 20;)
;	Ausgabe SIZE
a+/-	Addiert/Subtrahiert a zu ADDR (z.B.: 1+)
aG	Startet Programm an ADDR
ag:	Liest Daten aus dem CODE Bereich (PSEN) von Addr a (0..$FFFF)
dX	Schreibt DATA d an ADDR (0..$FFFF) in den externen RAM Bereich (WR)
ax	Liest Daten von ADDR a (0..$FFFF) aus dem externen RAM Bereich (RD)

dI	Schreibt DATA d an ADDR (0..$FF) in den internen RAM Bereich (Reg.Ind.Addressing)
ai	Liest Daten von ADDR a (0..$FF) aus dem internen RAM Bereich (Reg.Ind.Addressing)
dJ	Schreibt DATA d an ADDR (0..$FF) in den internen SFR Bereich (Dir. Byte Addressing)
aj	Liest Daten von ADDR a (0..$FF) aus dem internen SFR Bereich (Dir. Byte Addressing)
dP	Setzt Port ADDR (0..5) auf DATA d (z.B.: 2.$FFP)
ap	Liest Daten von Port ADDR a (0..5) (zB.: 2p)
dO	Verknüpft DATA d mit Portzustand (OR) und setzt Port ADDR (0..5)
do	Verknüpft DATA mit Portzustand (AND) und setzt Port ADDR (0..5)
dM	Schaltet immer in Setup=3 und kopiert Daten mit der Länge SIZE von ADDR nach DATA d. Wird zur automatischen Programmierung der Flash-Segmente F0…F1 verwendet. Anschließend wird in den Ausgangssetup zurückgeschaltet.
am	kopiert Daten mit der Länge SIZE von ADDR nach DATA d (ohne automatische Setup-Umschaltung).
aZ	Löscht (=$FF) Segment an ADDR a (0..$FFFF) im Datasegment. Ein Segment ist jeweils 16kB groß, und ist durch eine beliebige Adresse in diesem Bereich gekennzeichnet. Um das Flash-Segment (F0,F1) zu löschen, muß dies durch Setup 3 in den Databereich gebracht werden!
az	Falls das Segment durch ein Flash-Memory belegt ist wird hier die Bausteininformation (Manufacturer, Device) ausgelesen. Bei einem RAM-Bereich wird der Inhalt mit "H" (alle Bytes=$FF) oder "x" gekennzeichnet.
nS	Setzt auf Setup n=0..3. Bei Setup 0 und 1 wird der Monitor verlassen und ein Programm bei $0000 gestartet.
s	gibt den aktuellen Setup an. Es wird zusätzlich Information über die vorhandenen Segmente (R,F,M) ausgegeben. Falls ein Segement Daten (<>$FF) enthält, wird dies durch einen Kleinbuchstaben (r,f) angezeigt.
a&	Setzt Speicherbereich (ca. 100 Bytes) im RAM ab Addresse a für Flash-Hilfsroutinen.
&	Ausgabe Startadresse des Speicherbereichs.
V	Verbose=ON (default). Alle Ausgaben über die serielle Schnittstelle werden ausführlich (mit vollem ECHO und z.T. in HEX) gesendet. Durch Angabe einer SIZE-Länge in den Befehlen g,x,i,j, werden ganze Blöcke mit Adressangaben ausgegeben.
v	Verbose=OFF. Hier wird der Text auf die nur unbedingt notwendige Ausgabe beschränkt (KEIN Echo der eingebenen Daten und Ausgabe typisch in Dezimal).

2.3.3. Autostart

Programme, die in F0.. geladen sind, werden bei Vorhandensein von **MON51** automatisch bei Reset gestartet.

ACHTUNG: Falls **MON51** nicht existiert, können Programme bei F0.. nicht gestartet werden, da der Controller immer im Setup 3 startet und somit ein Programm bei F6 erwartet wird. In diesem Fall muß das Hauptprogramm in F6, F7 gespeichert sein!

2.4. Programmierung und Ausführung

2.4.1. Extern

Entsprechend **Setup 3** muß immer zunächst ein Programm im Flash-Segment F6, F7 (Speicheradresse Flash $18000..) mit einer maximalen Größe von 32KB liegen. Das Flash-EPROM wird hier auf einem Programmiergerät programmiert. Typischerweise ist dies der Monitor **MON51**. Durch Setup-Umschaltung kann dann ein weiteres Programm mit einer maximalen Größe von 64KB (F0..F3) gestartet werden.

2.4.2. Download

Programme, die nach F0, F1 geladen werden sollen, werden zunächst in den RAM-Bereich R0, R1 geladen und müssen anschließend mit dem Befehl **M** in die Flash-Segmente F0…F1 kopiert werden. Anschließend werden sie bei Reset durch den Monitor automatisch nach Umschaltung in **Setup 0** gestartet.

ACHTUNG: Es ist hier zu beachten, daß in diesem Fall ein kleiner RAM-Bereich (Scratch: normalerweise $7F00…$7FFF einstellbar mit &) für Hilfsprogramme zur Programmierung des Flash-Speichers verwendet wird.

2.4.3. Temporär im RAM

Im **Setup 2** ist der ganze Bereich von $0000…$7FFF (in **Setup 3** nur der Bereich $4000…$7FFF) als kombinierter DATEN&CODE Bereich (vonNeumann) verfügbar. Es können dort durch **MON51** Programme geladen und unmittelbar ausgeführt werden (Befehl **G**).

3. FERTIGUNG

3.1. Mechanik

3.1.1. Stecker (S1)

Vierkant-Stifte: Zu empfehlen bei Einbau in Geräten ohne Sandwich-Aufbau.

Wire-Wrap Buchsen/Stifte (Fischer: BL9): Sandwich-Aufbau

3.2. Elektronik

3.2.1. Schaltbild

(Anhang)

3.2.2. Bestückungsplan

(Anhang)

3.2.3. Stücklisten

(Anhang)

3.2.4. Kondensatoren für Powerup-Reset

10uF: Anstiegszeit 15ms; Abfallzeit 300ms

100uF: Anstiegszeit 20ms; Abfallzeit 300ms

1000uF: Anstiegszeit 150ms; Abfallzeit 300ms

3.3. PLDs

3.3.1. PAL-Listing MP35_5

```
module MP35_05
title 'vwa241100';

"19.07.99: I6 in Setup3 schaltet zwischen Flash und IO um
"24.11.00: New Setups (ohne I6)


"inputs
MS,A14,P34,A15,PSEN,P32,I6,RES,RD,WR     PIN 1,2,3,4,5,6,7,8,9,11;

"outputs
FA15,OE                 PIN 12,19;
C2,C1,C0                PIN 13,14,15;
CSB,FA16,CSA            PIN 16,18,17;

x = .X.;
Setup   = [P34,P32];
S0  = (Setup==0);
S1  = (Setup==1);
S2  = (Setup==2);
S3  = (Setup==3);

FSelect = [FA16,FA15];
F01         = [0,0];
F23         = [0,1];
F45         = [1,0];
F67         = [1,1];

ASelect = [A15,A14];
A0          = (ASelect==0);
A1          = (ASelect==1);
A2          = (ASelect==2);
A3          = (ASelect==3);
A01         = !A15;
A23         = A15;

CODE    = !PSEN;
DATA    = !RD # !WR;
CSRam   = CSA;
CSRom   = CSB;              "Flash

Equations
!C0     =   !PSEN # !RD;
!C1     =   !WR;
C2                  =       S0 & A23 & DATA # S2 & A23 & DATA;   "for LCD Display!
```

```
!CSRam =        S0 & DATA & A01
                # S1 & CODE & A1  # S1 & DATA & A01
                # S2 & CODE & A01 # S2 & DATA & A01
                # S3 & CODE & A1  # S3 & DATA & A01;

!CSRom =        S0 & CODE
                # S1 & CODE & A23 # S1 & DATA & A23
                # S2 & CODE & A23
                # S3 & CODE & A0  # S3 & CODE & A23 # S3 & DATA & A23;

!OE    =   S0 & DATA & A23 # S2 & DATA & A23;

FSelect =       F01 & (S0 & CODE & A01 # S3 & DATA & A23)
                # F23 & (S0 & CODE & A23 # S1 & CODE & A23)
                # F45 & (S1 & DATA & A23)
                # F67 & (S2 & CODE & A23 # S3 & CODE & A23 # S3 & CODE & A0);

end
```

4. MODIFIKATIONEN

4.1. Versionen

4.1.1. Vers. 0:

1. Versuchsmuster
   verschiedene Leitungen von Hand gepatched!

4.1.2. Vers. 1:

1. Layout entsprechend Vers. 0 geändert.

4.1.3. Vers. 2:

2. Setup (in PAL) so geändert, daß Programme bei $0000 starten!

4.1.4. Vers. 3:

4.1.4. Vers. 4:

4.1.4. Vers. 5:

2. Setup (in PAL) so geändert, daß Programme sowohl in F0,F1 wie auch in R0,R1 bei $0000 gestartet (und getestet) werden können!

5. ANHANG

5.1. Bausteinunterlagen

5.1.1. Schaltbild

5.1.2. Stückliste

5.1.3. Bestückungsplan

5.1.4. Controller 80C535

5.1.5. Memory 62256

5.1.6. Flash EPROM Am29F010