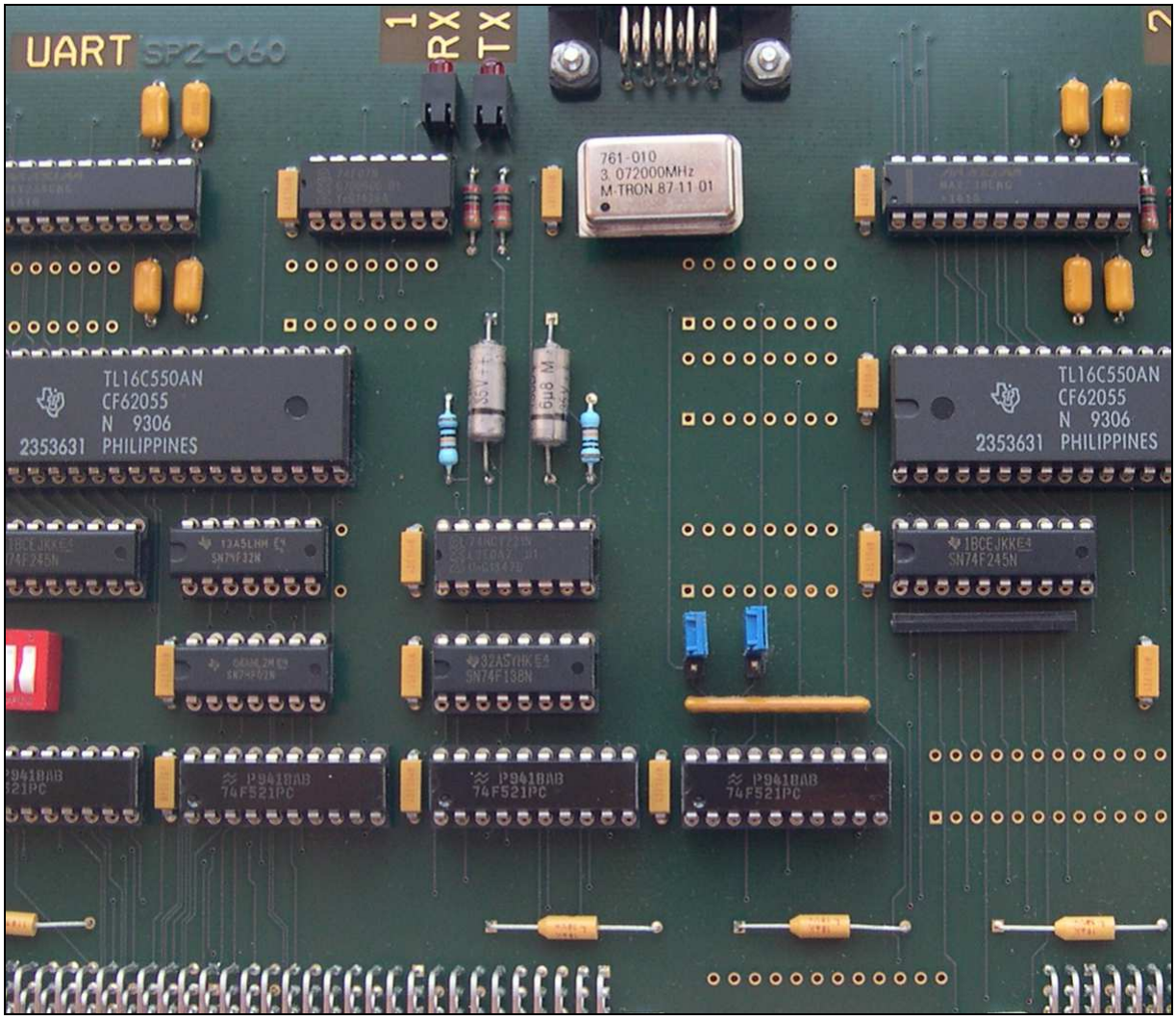
Die UART-Karte SP2-060.

Von Henry Westphal



Die Idee.

Um die CPU des SPACE AGE2 in klassischer Weise mit einem Terminal betreiben zu können, wurde eine UART-Karte konzipiert und ein historisches VT-101-Terminal beschafft.

Bei der UART-Karte wurde im Sinne einer pragmatischen Lösung das bisherige Konzept, nur Bausteine aus der 74-er Reihe und Speicherbausteine einzusetzen durchbrochen. Es wurden in diesem Fall höher integrierte UART-Bausteine mit FIFO-Puffer eingesetzt.

Die Dokumentation des Entwurfsprozesses

Die Auswahl des UART-Bausteins

Es wurde der Baustein TL16C550 mit integriertem FIFO im DIL-Gehäuse ausgewählt. Dieser, abgekündigte, Baustein wurde bei Ebay als NOS-Ware beschafft.

Die Betrachtung der zur Verfügung stehenden Zugriffszeit

Der kritischste Fall ist das Lesen aus dem UART in ein CPU-Register.

Aus der vorhandenen Timing-Analyse der SP2-CPU und der Microcode-Tabelle der SP2-CPU kann die hierfür maximal zur Verfügung stehende Zugriffszeit bestimmt werden.
Die CPU wird mit 4MHz getaktet. Ein Taktzyklus dauert dann 250ns.
Der Lesezugriff erstreckt sich über zwei Taktzyklen.

Die den Lesezugriff initiierende Taktflanke beginne bei t = 0ns
Bei t = 168ns stehen dann die Adressen und das Lesesignal am UART an.

Die Laufzeit von den Datenausgängen der UART-Karte bis zum MemoryDataRegister der CPU ist 38,4ns. Die die Übernahme ins MDR (Memory Data Register) der CPU initiierende Taktflanke erfolgt zum Zeitpunkt t = 500ns.

Damit müssen die aus dem UART gelesenen Daten zum Zeitpunkt t = 500ns – 38,4ns = 461,6ns am Busstecker der UART-Karte zur Verfügung stehen.

Damit ergibt sich eine Zugriffszeit von:
461,6ns – 168ns = 293,6ns.

Die Zugriffszeit des TL16C550 (td 10) ist 60ns max. (über Temperatur und Versorgungstoleranz)
Damit bleibt für den Decoder eine Laufzeit von 293,6ns – 60ns = 233,6ns.

Die Auswahl der Adressen

Der UART TL16C550 hat eine Datenbusbreite von 8 Bit.
Dieser Datenbus wird auf die Bits 31..24 des CPU-Datenbusses gelegt.
Damit liegt er auf den „glatten" Adressen (bezüglich der Adressierung in 4-er Schritten) (Big Endian)
Der Registersatz des UARTs umfasst 8 Adressen, die dementsprechend mit den Adressleitungen A[4..2] adressiert werden.

Mit der Adressleitung A5 wird zwischen den beiden auf der Karte vorhandenen UART-Kanälen ausgewählt.

Auf der Baugruppe SP2-001 MEM_I/O sind bereits die folgenden Adressen belegt:

FF FF FF Fx

Die UART-Karte hat einen Adress-Einstellungsschalter mit zwei Positionen.
Damit können bis zu 4 UART-Karten in einem System verwendet werden.

Der Adresschalter stellt die Wirkung der Adressleitungen A6 und A7 ein.

Damit nehmen die UART-Karten, je nach Schalterstellung, den folgenden Adressbereich ein:

FF FF FF 00 ...
FF FF FF 3F

FF FF FF 40 ...
FF FF FF 7F

FF FF FF 80 ...
FF FF FF BF

FF FF FF C0 ...
FF FF FF FF

Die Betrachtung der Logikpegel

Der TL16C550 hat TTL-kompatible Eingänge (Vih = 2,0V) und kann daher direkt von TTL-Bausteinen angesteuert werden.

Der Decoder

Der Adressbereich FF FF FF xx wird mit den 8-Bit Comparatoren U1001 bis U1003 ausgewählt.
Eine weitere Eingrenzung erfolgt mit U1004, entsprechend der Stellung des DIL-Schalters SW1001. Die UART-Karte ist dann angewählt, wenn alle Comparatoren U1001..U1004 Gleichheit erkennen, womit die Ausgänge dieser Comparatoren alle gleichzeitig auf L sind.

Die Unterscheidung zwischen den beiden UART-Kanälen auf der Karte erfolgt mit der Adressleitung A5 und dem Decoder U1005. Dieser Decoder ist nur dann freigegeben, wenn die Ausgänge aller Comparatoren U1001..U1004 gleichzeitig auf L sind. Der Ausgang des NOR-Gatters U1006A ist dann auf H, wenn beide Eingänge auf L sind, womit dann der Eingang G1 von U1005 den freigebenden H-Pegel erhält.

Es soll nun die Laufzeit durch den Decoder bestimmt werden:

74F521 (A zu Q)	11ns	max. über Vcc und Temepratur
74F02		6,5ns max. über Vcc und Temepratur
74F138 (G zu Y)	9,0ns	max. über Vcc und Temepratur
Summe:		26,5ns

Der TL16C550 kann den CPU-Datenbus nicht treiben. Daher wird noch ein Treiber 74F245 zwischen den Datenbusanschlüssen der UART-Bausteine und dem nach aussen führenden Datenbus eingefügt.

Damit ergibt sich in der Summe die folgenden Gesamt-Zugriffszeit:

Decoder:	26,5ns	
TL16C550:	60ns	
74F245: (A/B)	7ns	max. über Vcc und Temepratur
Summe:	93,5ns	

Es besteht also eine Reserve von 293,6ns – 93,5ns = 200ns.

Die Beschaltung des UART-Bausteins

Die Beschaltung erfolgt nach den Vorgaben des Datenblatts, es sind keine Besonderheiten vorhanden.

Das Signal DDIS ist dann auf H, wenn KEIN Lesezugriff auf den UART erfolgt,
In diesem Fall ist die Datenrichtung des 74F245-Treibers von A nach B. Der Bus wird also durch den 74245 nicht getrieben.

Anstelle eines Quarzes wird ein integrierter DIL-Quarzoszillator mit der Frequenz 3,072MHz vorgesehen, der aktuell nicht mehr lieferbar ist, aber bei Ebay als NOS-Ware bezogen werden konnte.

Das Interruptsignal des TL16C550 ist H-aktiv. Es wird mit U1006C/D invertiert und über den Open-Collector-Treiber U1007A/B auf die Wired-And-Interrupteingangsleitung der CPU gelegt.
Mit den Steckbrücken J1101 und J2102 ist eine Deaktivierung der Interruptanforderungen zu Testzwecken möglich.

Die Umsetzung auf RS232-Pegel erfolgt mit zwei Treiber/Empfänger-Bausteinen MAX238, die in der Standardbeschaltung betrieben werden.

Die Belegung der Steckverbinder wurde nach dem VT-100-Handbuch vorgenommen. (DEC VT-100 Technial Manual, Seiten 3-4 und 3-5)

Die Kommunikationsanzeige

Bei einer fallenden Flanke an den UART-seitigen Datenleitungen RXD und TXD (Beginn eines Zeichens) soll die dazugehörige LED für ca. 500ms leuchten. Diese Pulsverlängerung wird mit Monflops 74HCT221durchgeführt.

Die Beschaltung der Eingänge erfolgt nach der Wahrheitstabelle im Datenblatt:

TRUTH TABLE

INPUTS			OUTPUTS	
$\overline{A}$	B	$\overline{R}$	Q	$\overline{Q}$
H	X	H	L	H
X	L	H	L	H
L	↑	H	⊓	⊔
↓	H	H	⊓	⊔
X	X	L	L	H
L	H	↑	⊓ (Note 3)	⊔ (Note 3)

H = High Voltage Level, L = Low Voltage Level, X = Irrelevant, ↑ = Transition from Low to High Level, ↓ = Transition from High to Low Level, ⊓ = One High Level Pulse, ⊔ = One Low Level Pulse

Die Funktion des Reset-Eingangs /R wird nicht benötigt, der Eingang wird fest auf H-Pegel gelegt.

Das Datenblatt legt einen maximalen Wert des Timing-Widerstandes von 100kOhm nahe.
Die im Datenblatt angegebene Formel für die Pulslänge ist:
t = 0,7 x R x C => t / (0,7 x R) = C => 500ms/ (0,7 * 100kOhm) = 7,14uF.
Es wird der hinreichend nahe liegende Normwert 6,8uF vorgesehen.

Es werden LEDs mit einem Vorwärtsstrom von 20mA vorgesehen. Diese können mit den noch ungenutzt vorhandenen Treibern 74F07 angesteuert werden. Diese Treiber können einen Ausgangsstrom von bis zu 64mA aufnehmen.

Die Flussspannung der LED wird mit 1,6V angenommen. Die Restspannung des 74F07 wird mit 0,6V angenommen. Damit ergibt sich der Vorwiderstand zu: (5V – 1,6V – 0,5V) / 20mA = 145 Ohm. Es wird der Normwert 150R vorgesehen.

Die Dokumentation des Hardware-Tests für UART SP2-060

Benötigtes Testequipment:

1. Testadapter SP2-040
2. Schablone SP2-041... für Peripheriekarten für Testadapter SP2-040

3. Testkabel mit zwei IC-Pins und Serienwiderstand 100 Ohm
4. Testkabel mit zwei IC-Pins und direkter Verbindung

5. DSUB9-Buchse mit Verbindung von Pin 2 nach Pin 3

(1) Adressdecoder

Folgende Bausteine bestücken:
U1001..U1006

Die Schalter von SW1001 werden zunächst auf OFF gestellt.

Die Adressen mem_adr[31..5] werden auf H gestellt.
Das Signal #sel_UART2 (U1005/14) soll auf L sein.

Alle Bedingungen EINZELN wegnehmen. In jedem Fall soll U1005/14 auf H gehen.

Wiederholen des Tests, jedoch mit mem_adr_5 auf L
Das Signal #sel_UART2 (U1005/15) soll auf L sein.

Alle Bedingungen EINZELN wegnehmen. In jedem Fall soll U1005/15 auf H gehen.

Wiederholen des Tests,jedoch mit Schalter 1 von SW1001 auf ON und mem_adr7 auf L.
Das Signal #sel_UART2 (U1005/15) soll erneut auf L sein.

Alle Bedingungen EINZELN wegnehmen. In jedem Fall soll U1005/15 auf H gehen.

Wiederholen des Tests,jedoch mit Schalter 2 von SW1001 auf OFF und mem_adr6 auf L.
Das Signal #sel_UART2 (U1005/15) soll erneut auf L sein.

Alle Bedingungen EINZELN wegnehmen. In jedem Fall soll U1005/15 auf H gehen.

(2) Datenbus-Buffer

Folgende Bausteine zusätzlich bestücken:
U1101, U1201

Der DIR-Pin von U1101 ist nicht beschaltet.
Damit stellt sich ein H-Pegel an diesem Pin ein.

Damit ist die Datenrichtung des Bausteins vom Bus zum (noch nicht bestückten) UART.
Ausgabe von (hexadezimal) 01,02,04,08, 10, 20, 40, 80 an mem_d[31..24].
Kontrolle der Ausgänge B von U1101.
Das eingegebene Datenmuster soll sich an den Ausgängen wiederfinden.

Der DIR-Pin von U1101 wird an Masse gelegt.
(Testkabel mit zwei IC-Kontaktstiften)
Damit ist die Datenrichtung des Bausteins vom UART zum Bus.
Die Bus-Datenleitungen mem_d[31..24] auf H stellen (H ist am Bus-Simulator auch hochohmig)
Mit einem zweiten Testkabel wird jeweils einer der Eingänge B von U1101 an Masse gelegt.
Die anderen Eingänge sind auf H-Pegel, da nicht beschaltet.
Das entsprechende Datenmuster soll an der LED-Anzeige des Bus-Simulators sichtbar sein.

Identisches Vorgehen für U1201.

(3) RS232-Transceiver

Folgende Bausteine zusätzlich bestücken:
U1103, U1203, U1104, U1204, U1007

Spannungen V+ und V- kontrollieren (Sollwert ca. +/-10V)

Die TTL-Eingänge des MAX238 haben ein Pull-Up von 400kOhm.
„Auf Masse ziehen" der Eingänge auf 5V über 100 Ohm (Testkabel, eine Seite wird über Klemme temporär an GND verbunden)
Signal muss invertiert mit +/-10V an dem dazugehörigen RS232-Ausgang erscheinen.

(1_TXD, 1_RTS, 1_DTR sowie 2_TXD, 2_RTS, 2_DTR)

Bei einem H-L-Übergang des TXD-Signals (Bezug TTL-Ebene) soll die dazugehörige LED aufleuchten.
Zeitdauer des Aufleuchtens kontrollieren.

Die RS232-Eingänge des MAX238 haben ein Pull-Down von 5kOhm.
„Hochziehen" der Eingänge auf 5V über 100 Ohm (Testkabel, eine Seite wird über Klemme temporär an +5V verbunden)
Signal muss invertiert mit TTL-Pegel an dem dazugehörigen TTL-Ausgang erscheinen.

Bei einem L-H-Übergang des RXD-Signals (Bezug RS232-Ebene) soll die dazugehörige LED aufleuchten. Zeitdauer des Aufleuchtens kontrollieren.

(4) Takt

Folgende Bausteine zusätzlich bestücken:
U1008

An U1102/16 und an U1202/16 soll ein Takt 3,072MHz sein.

(5) Interrupt

In J1101 wird eine Steckbrücke gesteckt (Interrupt durchgeschleift)
Das Signal #ext int (J5101 C17) wird am Testadapter auf H (Hochohmig über Pull-Up) gestellt.

Es soll ein L-Pegel an diesem Signal angezeigt werden. (Grund: offene Eingänge 74F02)

Der Interrupt-Ausgang des UARTs (noch nicht bestückt) U1102/30 oder U1202/30 wird über ein Testkabel an Masse gelegt. Damit soll sich jeweils ein H-Pegel am Signal #ext_int einstellen.

Wiederholen des Tests mit einer Steckbrücke in J1102.

Die Steckbrücken aus J1101 und J1102 werden gezogen.
Es soll sich ein H-Pegel am Signal #ext_int einstellen.

(6) Test der UART-Funktion

Folgende Bausteine bestücken:
U1102, U1202

Registerzugriff über Bus-Simulator sinngemäß zu (2)
Auswahl der Register mit mem_adr[4..2].

Der grundsätzliche Registerzugriff wird mit Schreiben in und Lesen aus dem Scratch-Register überprüft.

Im Schnittstellen-Stecker (DSUB9) steckt ein Gegenstecker, an dem die Pins 2 und 3 gebrückt sind.

Auswählen des UARTs

mem_adr 5 auf L => UART1 angesprochen.
mem_adr 5 auf H => UART2 angesprochen.

Resetten des UARTS

Reset-Taste drücken, dann geht der UART in die Default-Settings (kein FIFO)

Test des Scratchpad-Registers

mem_adr[4..2] auf H.
#mem_oe auf H.
Zu schreibende Daten einstellen an mem_d[31..24]
Write –Taster drücken.
mem_d[31..24] auf FF stellen (Eingang)
#mem_oe auf L
Wieder ausgelesene Daten werden an mem_d[31..24] angezeigt.

Der Test wird an beiden UART mit den Daten 1,2,4...128 durchgeführt.

Baudrate einstellen / testen

Line-Control Register beschreiben.

(Schreibvorgang wie bei vorherigem Test)

mem_adr[4..2] <= [011]

zu schreibende Daten mem_d[31..24] <= [1000 0000] {*DLAB = 1, ermöglicht Zugriff auf Divisor-Latch*}

Low-Byte des Divisor-Latch beschreiben (DLL)

Wir stellen eine Baudrate von 9600 ein.
Die Werte nehmen wir aus Seite 29 des Datenblatts, Table 9

Schreibvorgang:
mem_adr[4..2] <= [000]
zu schreibende Daten mem_d[31..24] <= [0001 0100] {*20 d*}

High-Byte des Divisor-Latch beschreiben (DLH)

Wir stellen eine Baudrate von 9600 ein.
Die Werte nehmen wir aus Seite 29 des Datenblatts, Table 9

Schreibvorgang:
mem_adr[4..2] <= [001]
zu schreibende Daten mem_d[31..24] <= [0000 0000]

Kontrolle des Baudraten-Taktes

An Pin 9 des UARTs liegt ein Takt mit dem 16-fachen der Baudrate an, (Tastverhältnis unsymmetrisch) das ist 153,6 kHz.

Einstellen der Settings mit dem Line Control Register

mem_adr[4..2] <= [011]
zu schreibende Daten mem_d[31..24] <= [0000 0011]

Bit	Bedeutung
0,1	11 = Wortlänge 8 Bit.
2	0 = 1 Stopbit
3	0 = keine Parity
4	0 = even Parity (wirkungslos)
5	0 = Stick Parity (wirkungslos)
6	0 = kein Break
7	0 = Kein Schreiben in Divisor-Latch

Senden eines Zeichens

mem_adr[4..2] <= [000]
zu schreibende Daten mem_d[31..24] <= [zu sendendes Zeichen]

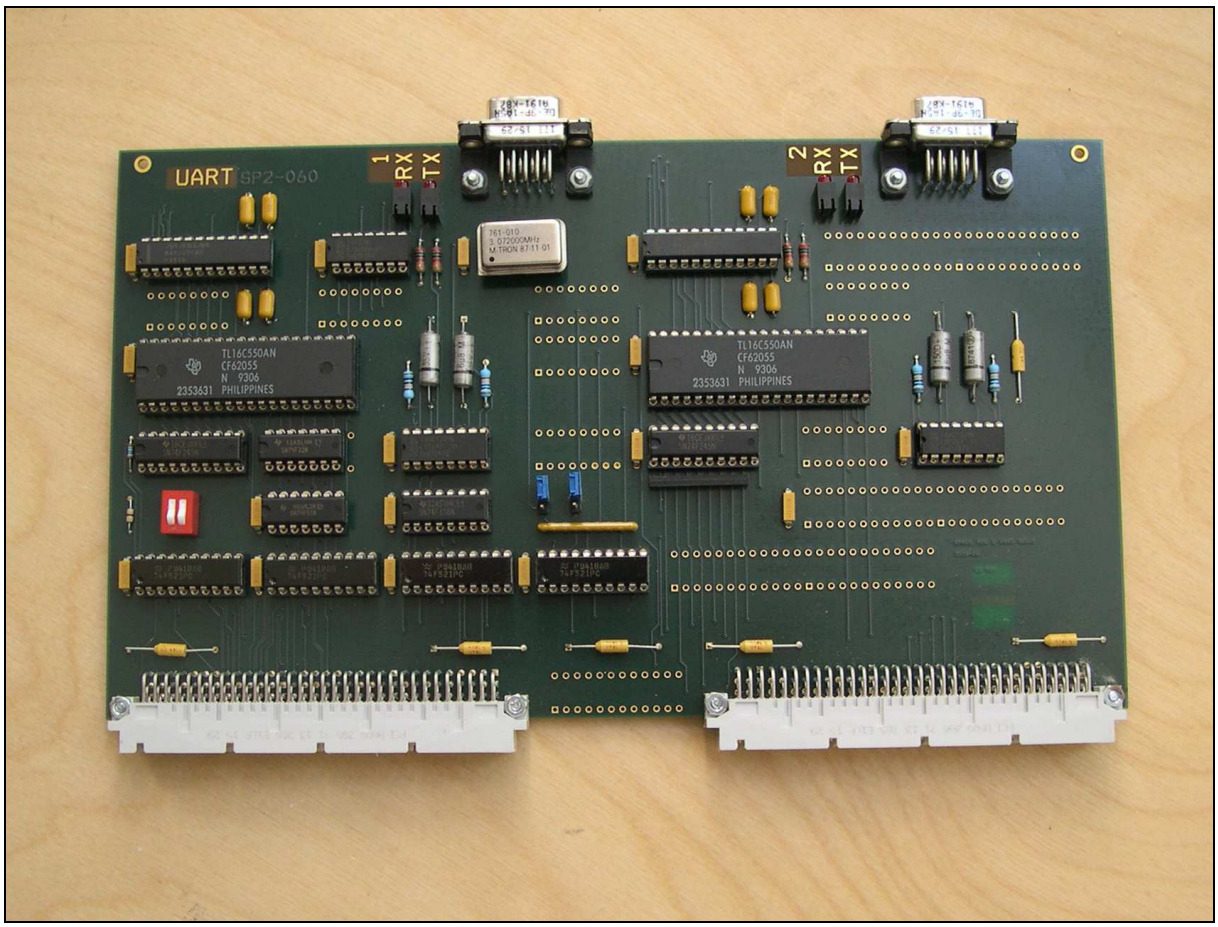
Mit dem Drücken der Write-Taste wird das Zeichen gesendet.

Kontrolle des Zeichens mit dem Oszilloskop an Pin 3 des DSUB-9-Steckers.
(Praktischerweise Bus-Funktion des DSO nutzen)

Rücklesen empfangenes Zeichen

mem_adr[4..2] <= [000]
Daten mem_d[31..24] <= [FF] {*Eingang*}
#mem_oe auf L
ausgelesenes Zeichen wird an mem_d[31..24] angezeigt.

Abbildungen



Die UART-Baugruppe



Das Terminal

Anhang: Dokumentation des Tests des VT101-Terminals

(28.11.2015)

Das VT-101-Terminal wurde zunächst unabhängig von der UART-Baugruppe mit einem PC getestet. Es wurde der PC PC-153040020002 (DELL LATITUDE3450) verwendet.

Die vorgenommenen Arbeitsschritte werden hier dokumentiert, um das spätere Testen der UART-Karte zu erleichtern.

Verbindungskabel:

Buchse D9 an EXSYS
Buchse D25 an VT100

1.1.1.1 D9			D25
2	----------	2	*Daten von VT100 zu PC*
3	----------	3	*Daten von PC zu VT100*
5	----------	7	*GND*

VT-101 seitige Arbeitschritte:

Set-Up

Der Set-Up-B des VT101 ist nicht mit dem Set-Up-B des VT100 identisch, daher müssen die spezifischen Unterlagen für das VT101 benutzt werden.

In den Set-Up B gelangt man wie folgt:

Taste SET UP drücken, dann ist man im Set-Up A
Taste „5 %" drücken, dann wechselt man zum Set-Up B

Die folgende Abbildung zeigt den Set-Up-B des VT101 mit den vorgenommenen Einstellungen:

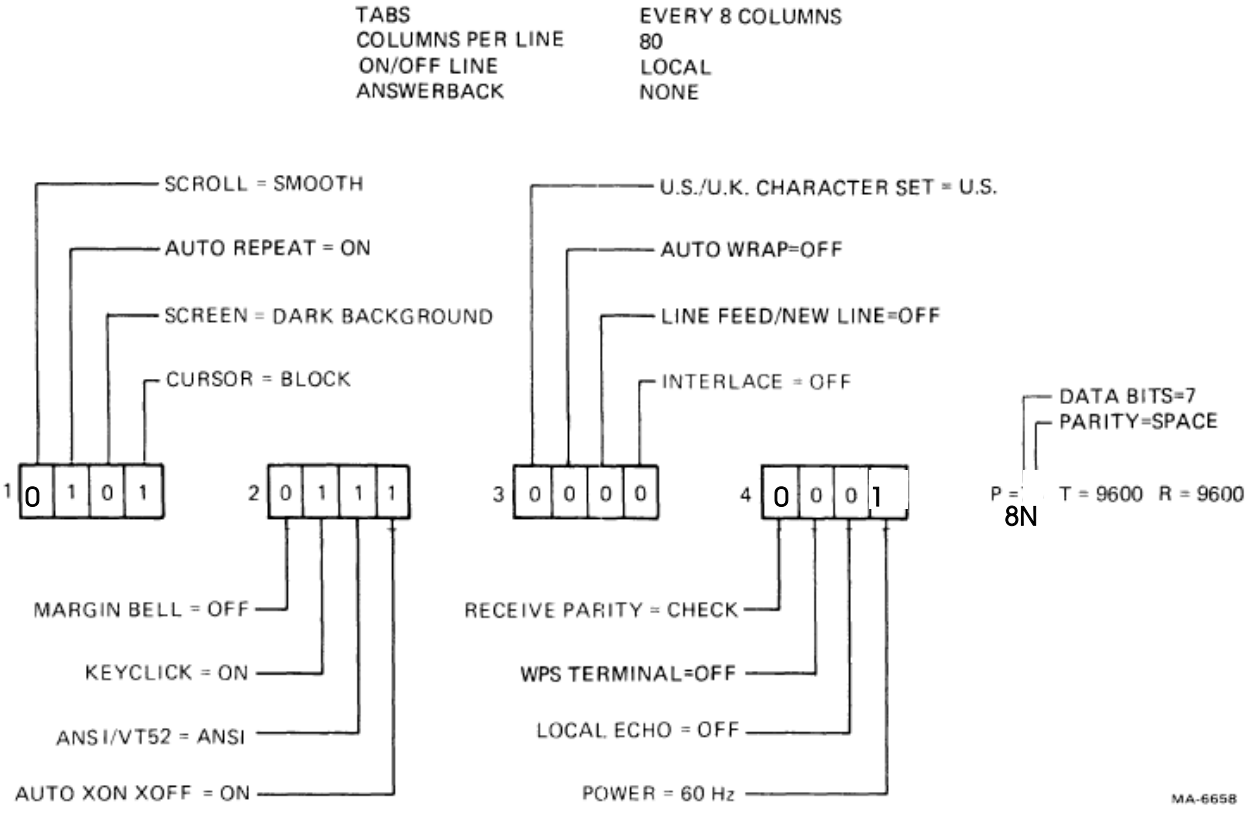


Figure 2-3 SET-UP B Default Feature Selections

SET-UP B:
1 0101 2 0111 3 0000 4 0001 P = 8N T = 9600 R = 9600

Reparaturen/ Änderungen am VT101:

Am VT101 wurden folgende Modifikationen und Reparaturen vorgenommen:

Linke Shift-Taste mit Kontaktspray behandelt, hatte keinen elektrischen Kontakt mehr.
Space-Taste und No Scroll-Taste mit Siliconöl geschmiert, Tasten blockierten häufig

Verbindungskabel von Logic-Board zu Schnittstelle 20mA gezogen, damit zunächst vorhandene Blockierung des Zeichenempfangs beseitigt . (Dieses Kabel war im Anlieferungszustand des Terminals bereits gezogen, dies wurde vom Verfasser irrtümlich für einen Transportschaden gehalten)

Das Anschlusskabel im Detail

(04.09.2016)

Buchse D9 an SP2-060 UART (Pins an Stecker auf der UART-Baugruppe)
Buchse D25 an VT100 (Pins an Stecker am VT-100-Terminal)

1.1.1.2 D9S		D25S	
2	----------	2	*Daten von VT100 zu PC*
3	----------	3	*Daten von PC zu VT100*
5	----------	7	*GND*

Kabel Farnell 170-2772 Adapter Farnell 182-5870
Nullmodemkabel
Buchse D9 auf Buchse D9 Stift D9 auf Buchse D25

1.1.1.3 D9S		D9S		D9P	D25S
2	---------- 3		3 ----------	**2**	*Daten von VT100 zu PC*
3	---------- 2		2 ----------	**3**	*Daten von PC zu VT100*
5	---------- 5		5 ----------	**7**	*GND*
4	---------- 1 und 6	1	---------- 8		
			6 ----------	6	
1 und 6	------- 4		4 ----------	20	
8	---------- 7		7 ----------	4	
7	---------- 8		8 ----------	5	
			9 ----------	22	

Danksagung

Wir danken Jörg Hoppe für die sorgfältige Restauration/Überprüfung und den günstigen Verkauf eines VT-101-Terminals aus seinem Sammlungsbestand.