32K/32K-Modul

Hinweis: von E. Müller gibt es ebenfalls ein 32K-Modul 2x32k Speicher

Norbert stellt hier seinen TINY nebst 32K-Modul vor:

JU+TE TINY Modul 32K/32K mit kompletter Dekodierung des Tastaturzugriffs entsprechend JU+TE 01/1988

Einleitung

Aus 2007 steht noch die Doku meiner Lösung bzgl. des 32K Moduls aus, gleichfalls der 4K-Monitor auf einem 2732. 64K ROM ist bis auf Monitor, pxl1000, Copy und RAM-Test noch leer. Die Schaltungs- und meine allerersten (nach über 20 Jahren!) Tiny-Bilder (hab ich nicht besser hinbekommen) ebenfalls anbei:

Hier stelle ich meine Ergänzung des Original-TINY mit einem 32K-RAM/32K-ROM Modul (Modul32) mit vollständiger Dekodierung des Tastaturzugriffs vor. Auf eine komplette Schaltung und Layout verzichte ich, da die kleine Leiterplatte in „Fädeltechnik“ frei verdrahtet ist.

Meinen Tiny hab ich mit Erscheinen der Beiträge in der JU+TE 1988 parallel zum bereits laufenden Projekt AC1 aufgebaut, um die Wartezeit zu verkürzen.

Die Darstellung des DB0…7 sowie AB 0…14 zu den Speichern sowie R/W zum RAM und die Stromversorgung setze ich als bekannt voraus und habe dieses demzufolge nicht separat aufgenommen. Wichtig ist allerdings die Steuerlogik, auf die ich nachfolgend noch detailliert eingehe.

Damit sollt dann auch der nicht ganz so geübte Bastler den Nachbau leicht bewerkstelligen

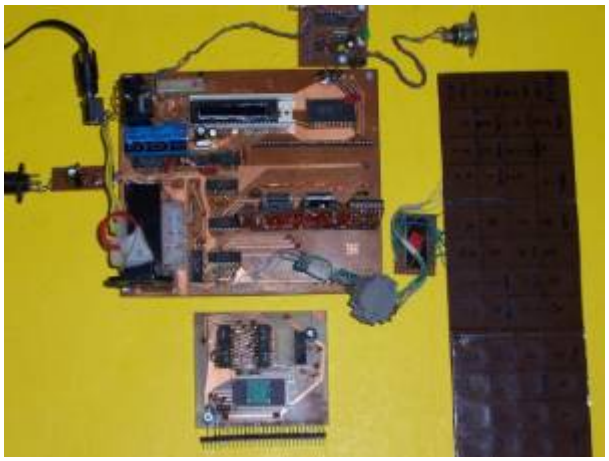
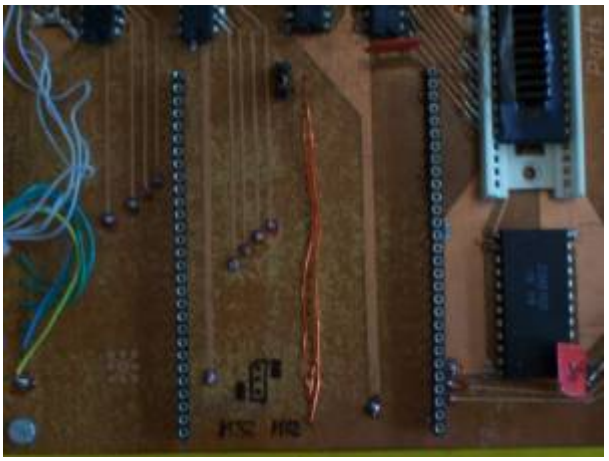
Ausgangspunkt der ganzen Geschichte war die Überlegung Mitte 2007, den maximalen Speicherausbau zu gewährleisten und insbesondere den ROM im Bereich der ROM-Ausblendung wegen des Tastaturzugriffs beim Original ebenfalls zu nutzen. Die Steuerung blendet ja immer 6000h…7FFFh aus. Das sind immerhin 8 KB (1/8el des normalen Adressraumes).

Entsprechende Diskussion zum TINY gab es seinerzeit im Forum von robotrontechnik sowie Beiträge diesbezüglich auf dieser Homepage. Dabei ging es auch um die Implementierung des 4 KB Monitors auf einem ´2732. (Dazu jedoch etwas mehr ganz unten nach der Hauptsache.) Wegen verschiedener Umstände komme ich erst jetzt, im Mai 2009 dazu das kleine Projekt abschließen.

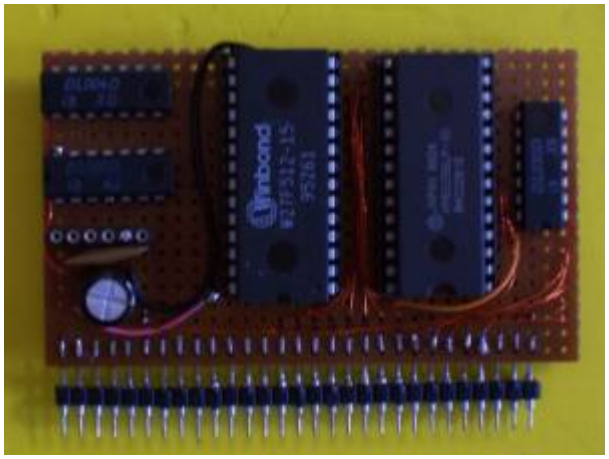
In der JU+TE 01/1988 S. 70ff. wird eine vollständige Tastaturdekodierung beschrieben. Allerdings wird hier natürlich auf die ursprüngliche 8 KB Segmentierung Bezug genommen. Ferner wird dabei die Bankumschaltung mittels P34 einbezogen.

Auf Letzteres habe ich schlussendlich verzichtet, weil, sobald ich P34 (invertiert) auf A15 des 64 KB ROM gelegt habe, das System zwar sauber lief, die Speicheranzeige im Bereich des gesamten ROM nur FFh lieferte. ROM A15 statisch auf LOW (bzw. HIGH) funktioniert jedoch problemlos. (P34 direkt an ROM A15 geht logischerweise auch nicht, sofern die obere ROM-Hälfte nicht mit dem Monitor

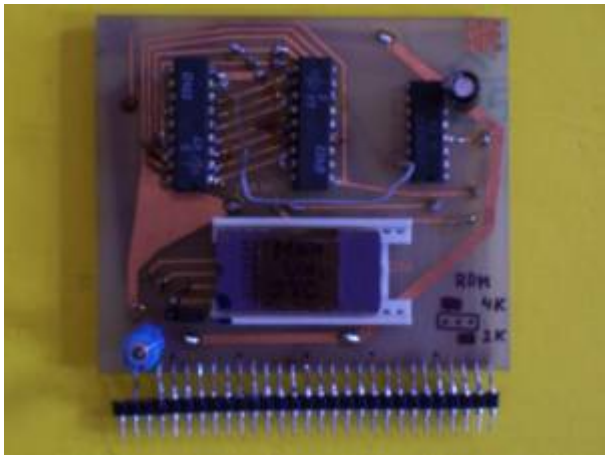
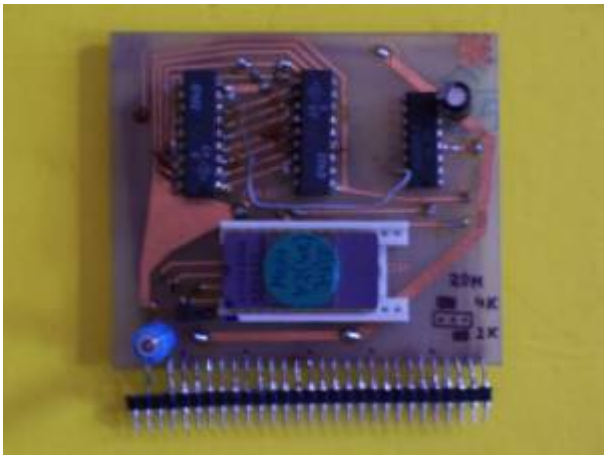
versehen ist

TINY gesamt. Modul-Selektion.



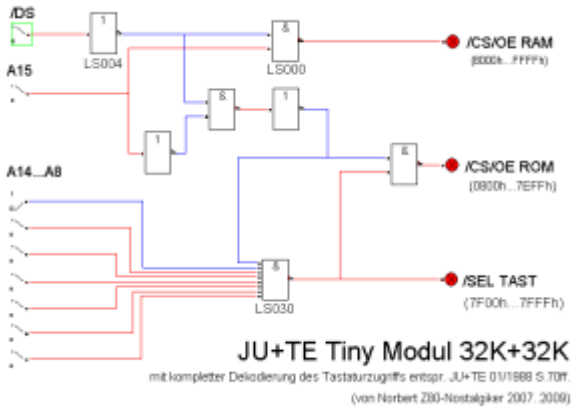

Das ganze Modul32 steckt auf Modulplatz 3. Speicheranzeige am Bildschirm.

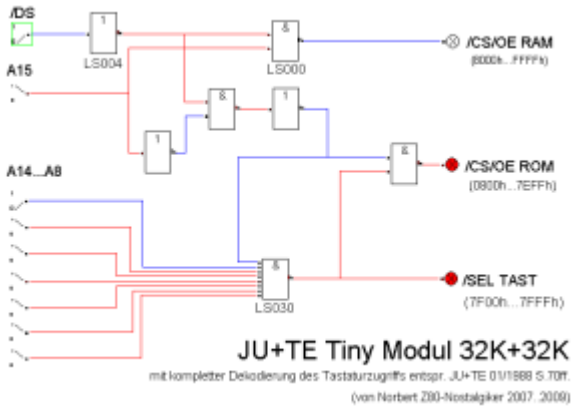
modifiziertes Originalmodul mit 2K-ROM. modifiziertes Originalmodul mit 4K-ROM (Monitor EWS1988).

Nun zur Lösung

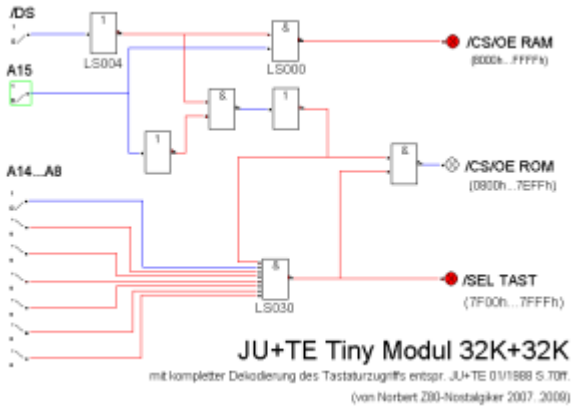
In den Bildern wird schematisch der Schaltungsaufbau deutlich. Die relevanten BUS-Leitungen werden mit H/L-Pegel (rot/ blau) dargestellt. Die lustigen roten Lämpchen repäsentieren H-Pegel am entsprechenden Selekt-Ausgang, aktiv (LOW) sind sie weiß. Die Bilder habe ich mit dem Digitalsimulator von Andreas Herz (FreeGroup.de) erstellt.



Wenn /DS nicht aktiv ist, wird natürlich gar kein Speicher angesteuert.



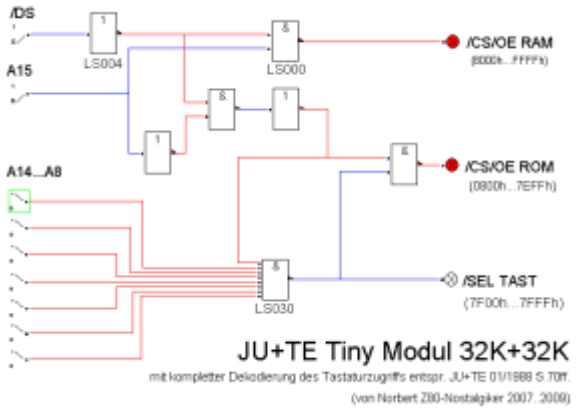
Das ändert sich sofort mit /DS=L, hier wird mit A15=H der RAM ab 8000h eingeblendet und kann benutzt werden.



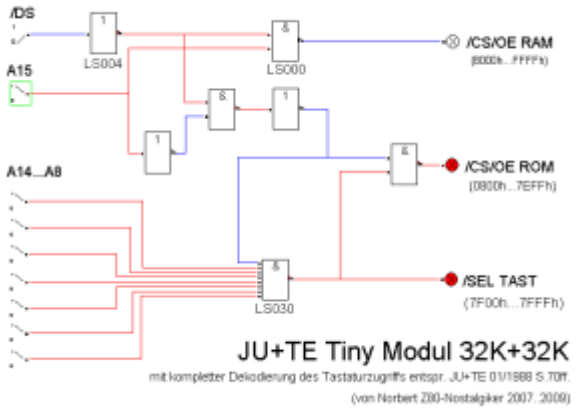
Mit A15=L greifen wir auf den ROM unterhalb 8000h zu, sofern die Tastatursteuerung nicht aktiv ist. Im Original werden A15..A13 zur Bildung des CSB4-Signal (/CS03 - Pin 12 ´8205) herangezogen:

```
A15     A14    A13    A12    A11    A10    A09    A08
L   H    H    X    X    X    X    X    entspricht 6000h...7FFFh
L   H    H    H    H    H    H    H    entspricht 7F00h...7FFFh
```

In der veränderten Version werden nur noch 256 Byte anstelle von 8192 Byte des Adressraums für den Tastaturzugriff ausgeblendet. Der Tastaturdekoder benutzt A3..A0 zur Spaltenselektion der angeschlossenen Tastatur. Über A7..A4 wird in der Dokumentation keine Aussage getroffen. (Zumindest habe ich nichts gefunden, hier könnte man noch mal „gigantische“ 240 Byte herausholen )



Sobald im Bereich der Tastatursteuerung selektiert wird, ist der ROM inaktiv.



Gibt es jetzt einen Zugriff oberhalb 7FFFh ist wieder RAM eingeblendet.

Damit ist die ganze Angelegenheit samt schaltungsmäßiger Ausführung nach meiner Meinung hinreichend und detailliert klargestellt. Bei mir sieht das Modul32 folgendermaßen aus (s.o.; kein Schönheitspreis, aber es erfüllt seinen Zweck :)

Die fünf Steckkontakte unter dem DL000 sind für die nicht auf dem Modulstecker vorhandenen Signale vorgesehen (A15..A13, /DS, /SEL TAST). Hier kommt ein kleiner kodierter Stecker hinein. Letztlich habe ich aber nach mehrmaligem Abreißen der Drähte beim Ein- und Ausstecken auf das zusätzliche Drahtgewirr verzichtet und die selten benutzen Leitungen (P34, U1, U2) der Grundplatte analog der Lösung von Enrico Müller für A15..A13 missbraucht, /DS kommt auf /CSB3 und /SEL TAST auf /CSA3. (/CSA3 und /CSB3 werden zum Dekoder hin unterbrochen!)

Auf der Grundleiterplatte ist weiter die Leitung von Pin 12 / ´8205 zum V40098 zu unterbrechen und

statt dessen /SEL TAST einzuspeisen. Ich habe einen kleinen Jumper neben dem nicht genutzten Modulplatz 2 eingebaut, um die originale Konfiguration wieder herstellen zu können. Außerdem sind ein paar Adressleitungen durch die vorhandenen Löcher verlegt.

```
       Jumper G 1 2 3
Pin 12  o--------o o o--------o /CSA3 (/SEL TAST)
DS8205             !             vom Modul32
                   !
      Pin 1 V40098 o     (JPG 1-2 Originalbetrieb)
                         (JPG 2-3 mit Modul32)
```

Und Voila! .... der TINY zeigt jetzt den ROM bis 7EFFh (da steht im Moment FFh drin, den muss ich noch weiter füllen). Ab 7F00h zählt er, was vorher bereits bei 6000h losging (s.o.)

Wenn Modul32 nicht auf Platz 3 steckt, kann ich das originale 2K/2K Modul auf Platz 1 stecken. Jetzt funktioniert der TINY natürlich nur mit (JPG 1-2), weil sonst keine Tastatureingabe möglich ist 

Und jetzt kommt der Bonustrack: Das originale Modul (Bild 09) habe ich auch etwas umgebaut, um wahlweise den ursprünglichen 2K-Monitor auf ´2716 oder den 4K-Monitor (EWS1988) auf ´2732 zu nutzen. Dazu wird einfach der ROM gewechselt und natürlich wieder entsprechend gejumpert (Bilder s.o.).

Wie geht das nun? In 2007 habe ich ebenfalls mit der Anpassung der oberen 2K (2000h..27FFh) experimentiert, wie andere auch, aber leider kein lauffähiges Ergebnis erzielt. Dann kam im robotrontechnik - Forum der Vorschlag, einfach die Adresslogik zu ändern. Hier ist meine Lösung dazu:

In den ´2732 wird fortlaufend der 4K Monitor gebrannt. Die oberen 2K werden jedoch nicht über A11 sondern A13 selektiert. Der TINY merkt das nicht, da er ohnehin bei (nicht vollständiger) Selektion innerhalb des ganzen 8K Segments die Bereiche spiegelt und mittels A13 folglich auf 2000h..2FFFh zugreifen kann. Dafür müssen auf der Grundleiterplatte die Signale /CSB1 und /CSB2 verknüpft werden, ich hab einfach zwei Schottky-Dioden 1N5819 benutzt (Ge-Dioden haben nicht funktioniert, hatte aber bloß GAZ17). Auch hier für Unterbrechung der ursprünglichen Leitungen zu Pin 15 und 14 des ´8205 sorgen. Die Unterbrechung wird auf der Leiterseite mittels der Dioden (Kathode zum Dekoder) gebrückt:

```
Pin 15 o----!&lt; -------o------&gt;!----o Pin 14
DS8205    1N5819      !    1N5819    DS8205
(/CSB1)               !              (/CSB2)
                      o
           /CSB1 vom Modul-Platz 1
```

Auf dem Original-Modul habe ich den Jumper auf die freie Stelle links neben den ROM gesetzt und frei verdrahtet:

```
             Jumper 1 2 3
A13         o--------o o o--------o 5P
Modulplatz             !
                       !     JP 1-2 4K-ROM
                       o     JP 2-3 2K-ROM
```

```
                    A11/PR ROM
                      (Pin 21)
```

So, das war schon alles. Wem es zu langatmig war, möge mir das nachsehen. Die bunten Bildchen hab ich vor allem benutzt, weil ich mittlerweile Logiktabellen über mehr als zwei Ebenen aufschreiben muss. Hinweise usw. sind jederzeit willkommen. Möglicherweise dauert es aber etwas bis ich antworten kann.

Viel Erfolg wünscht euch

Norbert Z80-Nostalgiker(ät)eMail.de

Download

Im Download-Paket 32k_norbert.zip enthalten sind:

- die Schaltung im Eagle-Format
- der (bisherige) ROM-Inhalt mit der Variante pxl1000:

Tiny-MP-Basic	0000h
Monitor ES1988/1	0800h
PXL1000	1000h
Monitor ES1988/2	2000h
Copy	4000h
Speichertest	4300h

- sowie dem auf pxl1000 angepassten Mal-Fix.