

# Z1013 PS/2 Chamäleon-Tastaturadapter V.0004

Programmdatei für ATTiny, ATmega oder Arduino Nano zur Emulation der drei Standard-tastaturen für den Z1013 oder Z1013-128:

- Brosig K7659 8x12 Tastatur
- Original Robotron 2.02 8x4 Folienflachtastatur
- 8x8 .A2 Monitor

Der eigentliche Ansatz war, dass meine A2 Tastatur sowas von selbst gebaut ist, dass sie durch etwas besseres ersetzt werden musste, wenn man mit dem Rechner vernünftig programmieren möchte.

Die Anschlüsse sollen nicht nur auf ein PS/2 Protokoll beschränkt bleiben. Andere Tastaturen, wie z.B. Amiga, serielle Konsolen oder andere Tastatur-Matrizen können auch unterstützt werden. Es funktioniert schon in Ansetzen, aber dazu später, eventuell.

Im Moment ist es eine Beta Version.

Wichtige Programmeigenschaften sind:

- automatische Monitor-Erkennung beim Einschalten, **kein Jumpen mehr**
- umschaltbares Tastaturlayout: **Wer kennt das Problem nicht, wenn Programme die Matrix direkt abfragen und man hat dann die richtige Tastatur nicht zur Hand**
- 4 MHz tauglich
- generiert /Reset und /NMI über Tastatur
- Funktionstasten der Brosig Tastatur erreichbar über F1-F9
- weitere Sonderfunktionen über ALT- erreichbar

Voraussetzung ist ein ATTiny2313A oder besser ein ATTiny4313 mit mehr Platz für spätere Updates oder weitere Tastaturtabellen. Die 2312A-Variante ist beim Z1013-128 zwingend notwendig. Ohne A funktioniert die A2 Tastatur nicht.

Weiterhin wird der Umbau wie zum Brosig Monitor angenommen, bei der die 4 Spaltenleitungen auf den Steckverbinder X2:A11, A10, A6, A4 gelegt werden. Beim Z1013-128 ist das standardmäßig der Fall, so dass dort kein Umbau vorgenommen werden muss.

Es gibt Rechner bei denen die 4 Spaltenleitungen nicht vorhanden sind und die auch nicht mehr umgelötet werden sollen. Eine Software Version, die ohne Umbau die 8 Spalten auswertet, ist möglich. Dementsprechend funktioniert dort aber die Brosig Emulation nicht.

## Sonderfunktionen der Tastatursoftware

ALT+F1: Hilfe\*

ALT+F2: Umschaltung Layout für K7656 Brosig Monitor

ALT+F3: Umschaltung Layout für A2 Monitor

ALT+F4: Umschaltung Layout für 2.02 Monitor

ALT+F5: Ausgabe des eingestellten Tastatur-Typs über die Tastatur

wenn hier "wirre" Zeichen kommen, liegt kein Fehler vor, sondern der aktuelle Monitor kann die Tastatureingaben nicht verarbeiten. Es wurde das falsche Layout eingestellt.

ALT+CTRL+BACKSPACE: Aktivierung des /RESET Signals

*\*beim ATTiny2313 aus Platzgründen nur sehr kurz mit Versionsinfo*

### Die Funktionstasten des Brosig Monitors\*

F1:RST38

F2:J100 oder 2MHz Umschaltung\*\*

F3:J200 oder 4MHz Umschaltung\*\*

F4:J300 oder 64+32 Zeichen\*\*

F5:HCOPIY ON/OFF

F6:GRAFIK ON/OFF

F7:SCRDUMP oder Zeichensatzumschaltung\*\*

F8:SLOW/FAST

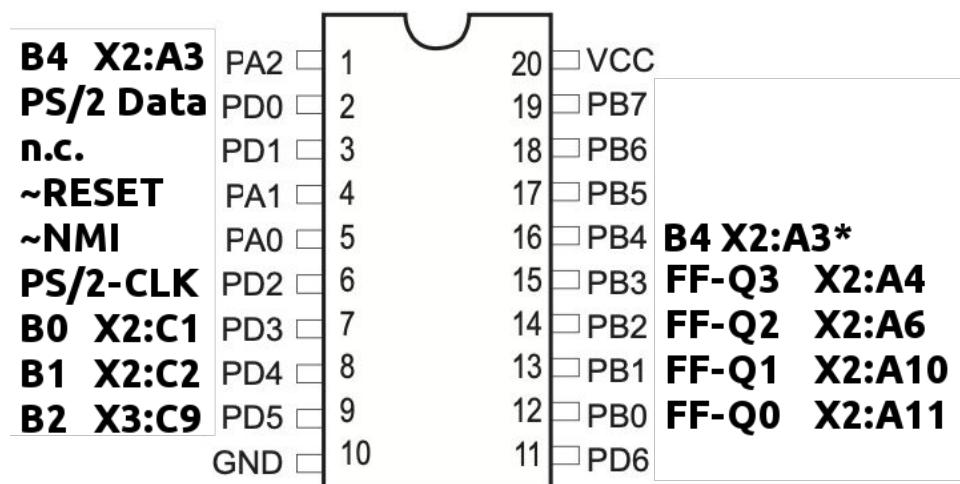
F9:BEEP ON/OFF

*\*die Funktion hängt von der Implementierung im Monitor ab*

*\*\*je nach Monitorversion zusätzlich über Shift erreichbar*

### Pinbelegung

Angefangen habe ich die Entwicklung mit dem Arduino Nano und später zum Z1013-128 gewechselt. Dort ist die Hardware schon integriert und die ROM- und Frequenzumschaltung unproblematisch, zum Testen, über Jumper möglich. Deswegen folgt die Pinbelegung auch dem Schaltplan aus dem Z1013-128. Nach ähnlichem Schema sind wohl auch die externen PS/2 Adapter aufgebaut.



\*für externe Platinen oder neue Designs sollte die B4-Leitung an Pin16, dann geht die Abfrage im Interrupt schneller. Beim Z1013-128 kann man dieses Pin leider nicht verwenden und man muss die CPU höher takten.

Für die volle Funktionalität beim Z1013-128 müssten die freien Pins PA0,PA1,PA2 mit den entsprechenden Pins von X2 verbunden werden. Im Mindestfall braucht man PA2 für die A2-Tastatur.

ATTiny2313A Fuses: LFUSE=0xE4 HFUSE=0xDE

### Implementierungsdetails

## Auszug aus dem A2 Monitor

667/ F1CE : ; Abfrage einer Spalte A2 Monitor

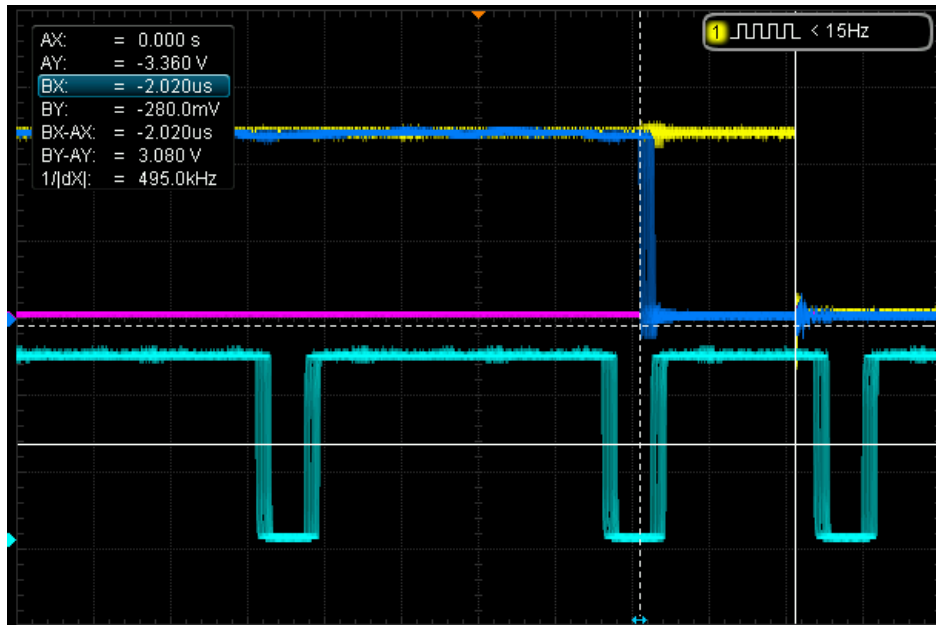
...

678/ F1D8 : CB E7 set 4, a ; PIO B4 = 1 (Zeile 4..7)

679/ F1DA : D3 02 out (PIOBD), a

680/ F1DC : DB 02 in a, (PIOBD) ; Zeilen 4..7 einlesen

Timing Diagram (magenta=IORQ, blau=PIO Ausgang, gelb=AVR Ausgang)



Die kürzeste Zeit, die zum Berechnen Ausgabe in Abhängigkeit von den Spalteninformationen zur Verfügung steht, sind etwa 9 Takte. Der IN Befehl selbst braucht 11 Takte. Die Zeit zwischen steigender und fallender Flanke der OUT/IN Folge ist dann nur noch 9 Takte. Bei 4 MHz sollte die Antwort also sicherheitshalber innerhalb  $9 \cdot 0,25 \mu\text{s} = 2,25 \mu\text{s}$  fertig sein. Das ist mit einem 8 MHz ATtiny zusammen mit der Auswertung des extra PIO-Pins B4 (X2:A3) in nicht zu schaffen. Genauer kann man in der Interrupt-Service Routine nachschauen.

Für den Auswertung des A2-Pins braucht man mindestens 2 extra Befehle, die die Pipeline ausser Acht gelassen, mit extra  $0,25 \mu\text{s}$  zu Buche schlagen. So wird die Unterstützung aller Varianten gleichzeitig schon eng.

2 extra Befehle im ISR um auf die 5. Spalte zu testen

```
sbic PINA,A2_SELECT
ori r26,0x10
```

Bei 8 MHz beträgt die Antwortzeit etwa  $2,3-2,6 \mu\text{s}$ . Beim ATmega kann man mehr optimieren, aber beim ATtiny blieb als letzter Ausweg nur eine Takterhöhung bzw. den Wert von OSCCAL zu erhöhen, so dass die CPU jetzt mit etwa 10 MHz läuft und die Antwortzeit im Bereich von  $2 \mu\text{s}$  liegt.

Andreas Ziermann, 2021-08-28