KEY4

Baugruppe für den NDR-Klein-Computer zum
Anschluß einer PS/2-Tastatur und PS/2-Maus.

Stand: April 2007

Wichtiger Hinweis:

Die in dieser Anleitung wiedergegebenen Schaltungen und Verfahren werden ohne Rücksicht auf die Patentlage oder Lizenzrechte Dritter mitgeteilt. Sie sind ausschließlich für private Zwecke und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden. *)
Alle Schaltungen und technische Angaben in dieser Anleitung wurden vom Autor sorgfältig erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht aus zuschließen. Daher kann der Autor weder eine Garantie noch die juristische Verantwortung oder irgend eine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Die Rechte an Firmennamen, Logos und Warenzeichen, die in dieser Anleitung genannt werden, liegen bei den jeweiligen Inhabern.

*) Bei gewerblicher Nutzung ist vorher die Genehmigung des möglichen Lizenz- oder Rechteinhabers einzuholen.

Inhalt:

Vorwort

Die Idee zu dieser Baugruppe hatte ich, als ich Graphik-JADOS testen wollte. Dazu währe eine Maus von Vorteil, ohne bringt das nicht viel. Eine HCOPY-Baugruppe habe ich ja, leider hat sich die dazu gehörige Atari-Maus vor 19 Jahren verabschiedet und ist wohl damals in einem Mülleimer verschwunden. Ich habe aber noch irgendwo ein Cordless Desktop mit PS/2-Anschluß.
Die Frage war jetzt: Wie bringt man den NDR-Klein-Computer dazu, daß er PS/2-Geräte versteht. Meine Lösung finden Sie auf den folgenden Seiten dieses Handbuchs.
Eine Platine werde ich aus Kostengründen erst mal nicht produzieren lassen. Meine gefädelte Variante tut's auch.

Da nichts auf dieser Welt perfekt sein kann, werden sich auch hier Fehler eingeschlichen haben. Wenn Sie einen entdecken, so teilen Sie mir dies mit einer möglichst genauen Beschreibung im NKC-Forum (siehe Literatur) mit.
Danke.

1 Kurzbeschreibung der Funktion

Die KEY4-Baugruppe stellt die Schnittstelle zwischen einer PS/2-Tastatur und einer PS/2-Maus und dem NDR-Klein-Computer (NKC) dar. Damit bleibt die Möglichkeit offen, mit heute noch erhältlichen Tastaturen und vor allem Mäusen, den NKC am Leben zu erhalten. Auf diese Weise erhielt mein NKC nach fast 19 Jahren wieder eine Maus.
Ziel der Entwicklung dieser Baugruppe war der Anschluß eines PC-Desktops (noch besser Cordless) an den NKC. Eine Lösung zum Anschluß einer PS/2-Tastatur gab es zwar schon; ich habe jedoch keinen Zugriff auf die Quelltexte. Daher habe ich das bereits existierende Programm für Tastaturcontroller nicht benutzt, sondern ein eigenes Programm dafür geschrieben. Was eigentlich nur die halbe Arbeit war, da beide Controller das gleiche PS/2-Schnittstellenmodul benutzen.

Die KEY4 ist hardwarekompatibel zu ihren Vorgängerbaugruppen, sodaß keine Anpassungen in der Software nötig sind.

Bitte beachten Sie:

Die Baugruppe wird bei Verwendung des Grundprogramm 68K erst ab der Version 6 vollständig unterstützt. In den Versionen davor wird die Baugruppe nach dem Systemstart nicht initialisiert. Dadurch funktioniert der eingebaute Tastaturpuffer nicht und es werden keine Zeichen an das Latch weitergereicht. Siehe Workaround in Kapitel 6.

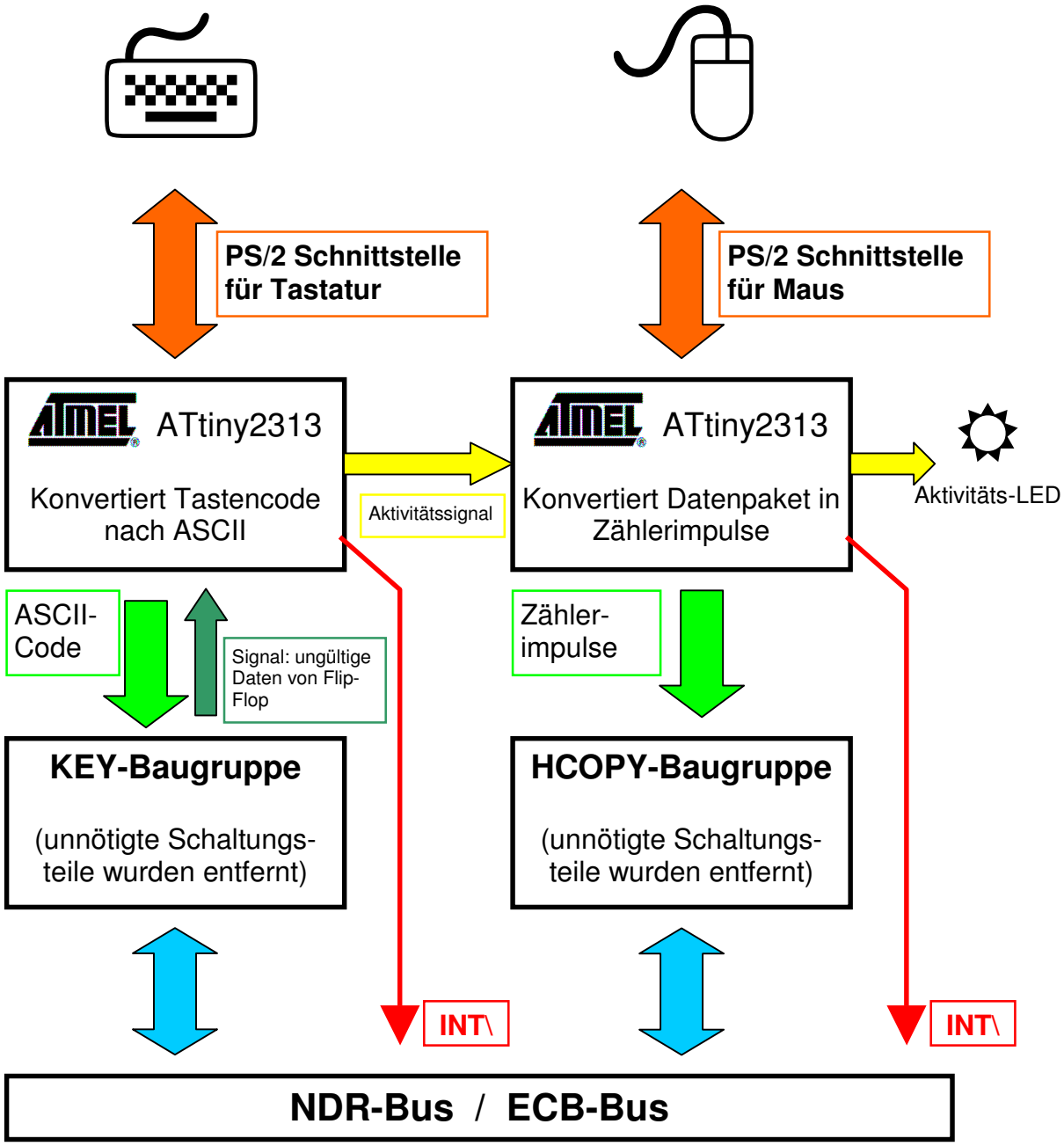
2 Technische Daten

- Europakarte 160 x 100 mm doppelseitig mit Bestückungsaufdruck
- NDR-Bus und ECB-Bus
- Stromaufnahme ca. 320 mA (Bestückung mit LS-Typen);
  zusätzlich ca. 100 – 250 mA für Tastatur und Maus, je nach Hersteller und Gerätetyp; Betriebsspannung 5V
- Mini-DIN-Buchsen zum Anschluß einer PS/2-Tastatur und einer PS/2-Maus
- Zusätzlich 2x5 polige Stiftleiste um die Mini-DIN-Buchsen wahlfrei in einem Gehäuse plazieren zu können
- Aktivitätsanzeige durch LED
- INT\ kann ausgelöst werden, wenn neue Daten von Tastatur oder Maus zur Verfügung stehen. Mittels Jumper einstellbar.
- Tastatur kann externen RESET auslösen, wenn Ctrl-Alt-Del gleichzeitig gedrückt werden.

3 Prinzipbeschreibung

Die PS/2-Schnittstellenkontrolle und Übersetzung der PS/2-Datenpakete in ein NKC-konformes Format übernehmen zwei Mikrocontroller. Der Rest der Schaltung auf der Baugruppe entspricht in leicht abgewandelter Form den bereits bekannten Schaltungen auf den alten Baugruppen KEY und HCOPY.

3.1 Blockschaltbild KEY4

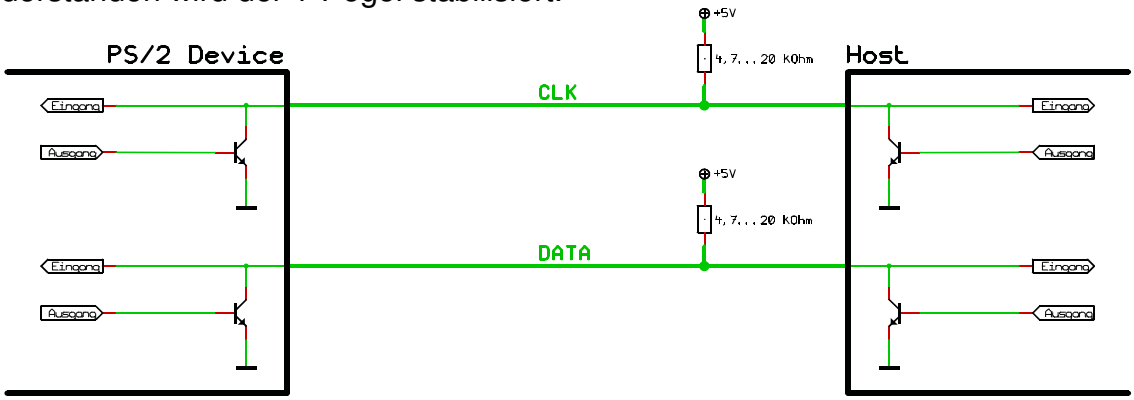


3.2 Beschreibung des Blockschaltbildes

Der Sinn der Schaltung besteht darin die von der PS/2-Tastatur bzw. PS/2-Maus gelieferten Signale so umzusetzen, daß die bisher genutzten Schaltungen und damit auch die Kompatibilität erhalten bleibt. Jede PS/2-Schnittstelle wird von einem eigenen Mikrocontroller überwacht und gesteuert. Er erledigt die eigentliche Anpassungsarbeit.
Der Tastaturcontroller liest die Tastencodes seriell von der PS/2-Schittstelle ein. Die Tastencodes werden dann in ASCII-Codes, die der NKC versteht übersetzt und in einen Puffer zwischengespeichert. Anschließend wird mittels Signalleitung (Q\ des Filp-Flop = Pin 8 am LS74) überprüft, ob das Zeichen im Latch (LS374) ungültig ist (1-Pegel). Wenn ja, dann wird das Zeichen am Ausgabeport des Mikrocontrollers ausgeben und mittels positivem Impuls am Strobe-Ausgang in das Latch geladen. Von jetzt an finden die Tastatureingaben auf bekanntem Weg, wie bei der alten KEY-Baugruppe, den Weg in den NKC.
Der Mauscontroller liest die relativen Positionen in 3-Byte-Paketen seriell von der PS/2-Schittstelle ein. Wenn ein Paket komplett übertragen wurde, aktualisiert der Mikrocontroller seine internen Zähler (Links, Rechts, Auf, Ab) und den Tastenzustand. Danach wird der Tastenzustand am Ausgabeport ausgegeben. Anschließend erzeugt der Mikrocontroller Zählimpulse um die externen Zähler (LS590) zu aktualisieren. Von jetzt an finden die Daten zur relativen Mausposition auf bekanntem Weg, wie bei der alten HCOPY-Baugruppe, den Weg in den NKC.
Jeder Mikrocontroller löst einen Interrupt (0-Impuls auf der Busleitung INT\) aus, wenn er neue Daten weitergeben hat. Diese Eigenschaft kann mittels Jumper ein- bzw. ausgeschaltet werden.
Der Tastaturcontroller kann zusätzlich noch einen externen RESET am NKC auslösen (nicht Blockschaltbild dargestellt).
Der Mauscontroller enthält noch eine weitere Spielerei. Wenn er ein neues Datenpaket von der PS/2-Schnittstelle erhalten hat, läßt er eine LED für ca. 20 ms leuchten. Der Tastaturcontroller kann die LED ebenfalls über eine Signalleitung aktivieren. --- Was die PC-Jungs können, können wir schon lange ! ---

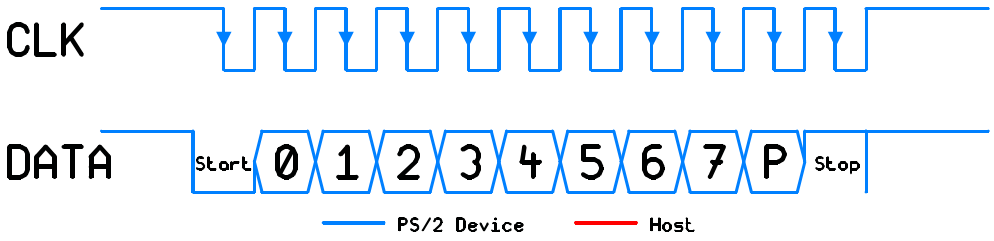
3.3 Beschreibung des PS/2-Protokolls

Das PS/2-Protokoll benötigt 2 Signalleitungen. Auf einer Signalleitung wird ein Taktsignal (CLK) und auf der anderen synchron die Datenbits (DATA) übertragen. Die Signalleitungen werden über „Open Collector“-Ausgänge gesteuert. Mit Pullup-Widerständen wird der 1-Pegel stabilisiert.



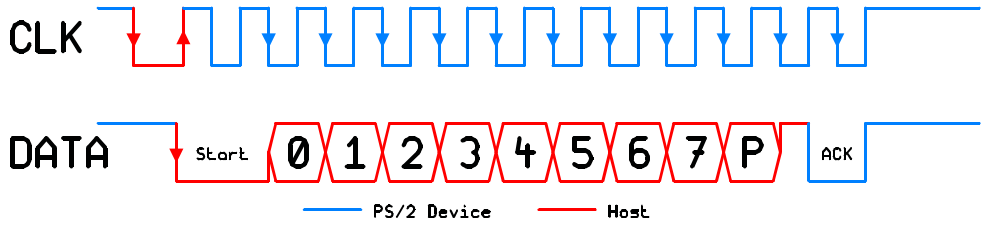
Wie in der Funktionsdarstellung ersichtlich ist die Schnittstelle bidirektional. Das PS/2- Eingabegerät sendet seine Daten (Tastencodes/Mausposition) selbständig an den Host. Der Host kann aber auch Kommandos zum PS/2-Eingabegerät zur Parametrierung oder Fehlerbehandlung schicken. Dazu legt der Host das CLK-Signals für min. 100 µs auf 0-Pegel. Das PS/2- Eingabegerät erkennt dann den Sendewunsch des Hosts und wechselt in den Empfangsmodus. Der Host hat dabei immer die höhere Priorität. Er kann damit auch das Senden des PS/2-Eingabegerätes unterbrechen.
Jedes Datenpaket ist 11 Bit lang. Zur Fehlererkennung wird ein Paritybit benutzt; Odd-Parity. Die Taktfrequenz des CLK-Signals liegt bei 20-30 kHz.

PS/2–Eingabegerät zu Host Kommunikation:



Das PS/2-Eingabegerät setzt den Pegel der Datenleitung (DATA) an der fallenden Flanke des Taktsignals (CLK). Der Host liest die Bits anschließend an der steigenden Flanke des Taktsignals ein. Die Übertragung beginnt mit einem Startbit (0-Pegel) und endet mit einem Stopbit (1-Pegel).

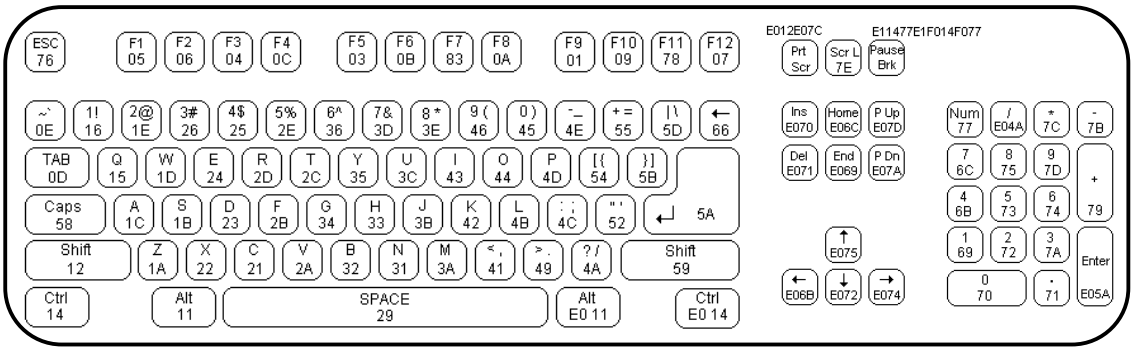
Host zu PS/2–Eingabegerät Kommunikation:



Der Host beginnt seine Datenübertragung mit dem Setzen des Taktsignals (CLK) auf 0-Pegel. Dieser Zustand wird ca. 100 µs gehalten, um sicher zu sein, daß das PS/2-Eingabegerät der Sendewunsch des Host auch erkannt hat. Dann wird die Datenleitung (DATA) auf 0-Pegel gesetzt und die Taktleitung auf 1-Pegel. Das PS/2-Eingabegerät beginnt jetzt das Taktsignal zu erzeugen. Der Host setzt nun die Datenleitung auf 0/1-Pegel, je nach Wert des zu übertragenden Bits, an der fallenden Flanke des Taktsignals. Das PS/2-Eingabegerät liest die Bits an der steigenden Flanke des Taktsignals ein. Hat der Host das Paritybit übertragen, dann quittiert das PS/2-Eingabegerät das Ende des Datenpaketes durch setzen der Datenleitung auf 0-Pegel für einen Takt.

3.4 Tastencodes und Tastaturkommandos

Das Diagramm einer PS/2-Tastatur zeigt den Tastencode, der den einzelnen Tasten zugeordnet ist. Der Tastencode ist in der unteren Hälfte der Taste in hexadezimaler Darstellung.



Normalerweise hat eine Taste einen 1 Byte Code. Die Tasten zur Cursorsteuerung sowie die Sondertasten auf „Windows“-Tastaturen haben einen 2 Byte-Code der immer mit dem „Extension“-Byte ($E0) beginnt. Nach dem Drücken einer Taste sendet die Tastatur den der Taste zugewiesenen Tastencode. Bleibt die Taste gedrückt, dann wiederholt die Tastatur nach einer Wartezeit (Delaytime) den Tastencode. Wird die Taste Wieder losgelassen sendet die Tastatur noch einmal den Tastencode; jetzt aber mit dem voran gestellten Break-Code ($F0). Dies gilt für alle Tasten.
Nach dem Einschalten der Tastatur oder einem Softwarereset wird der BAT-Complete-Code ($AA) gesendet. Dies wird genutzt um die Tastatureigenschaften für den NKC zu parametrisieren.

Hier noch kurz eine Tabelle der Kommandos die bei der Tastaturkommunikation verwendet werden (die Liste ist nicht vollständig):

Code	Beschreibung	Sender
$AA	BAT-Complete = Selbsttest der Tastatur abgeschlossen	Tast
$ED	Tastatur-LEDs und Modus ein-/ausschalten.	Host
$EE	Echo; Tastatur antwortet durch Senden von $EE	Host / Tast
$F3	Wartezeit und Tastenwiederholrate einstellen	Host
$FA	Acknoledge = Bestätigung des Erhaltes eines Kommandos	Tast
$FC	Error = Tastatur meldet einen Kommandofehler	Tast
$FE	Resend = Letztes Byte erneut senden	Host / Tast
$FF	Reset = Gerät neu starten	Host

3.5 Mausdatenpaket und Mauskommandos

Die Standard PS/2-Maus sendet Bewegungs- und Tastenzustandsinformationen in folgendem 3-Byte-Paket zum Host.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y Überlauf	X Überlauf	Y Vorz.bit	X Vorz.bit	Immer 1	Mittlere Taste	Rechte Taste	Linke Taste
Byte 2	Bewegung in X-Richtung							
Byte 3	Bewegung in Y-Richtung							

Dieses Paketformat wird als Grundeinstellung von jeder PS/2-Maus verwendet. Mäuse mit Scrollrad können durch Senden einer Codefolge auch auf ein 4-Byte-Paket umschalten, wobei das 4. Byte die Bewegung in Z-Richtung darstellt. Für den NKC, der eigentlich nur eine Atari-Maus kennt, werden solche erweiterten Paketformate nicht benötigt.
Bei jeder Mausbewegung oder einer Tastenbewegung schickt die PS/2-Maus immer ein vollständiges Datenpaket. Die maximale Paketrate (Paket pro Sekunde) kann aber eingestellt werden, damit der Mikrocontroller keinen Streß kriegt.
Bei einem Übertragungsfehler, die bei meinen Tests ab und zu mal auftraten, kann man sich das letzte Paket durch Senden des „Resend“-Kommandos ($FE) noch einmal senden lassen. Der Benutzer merkt nichts; die Mausbewegung bleibt fließend.

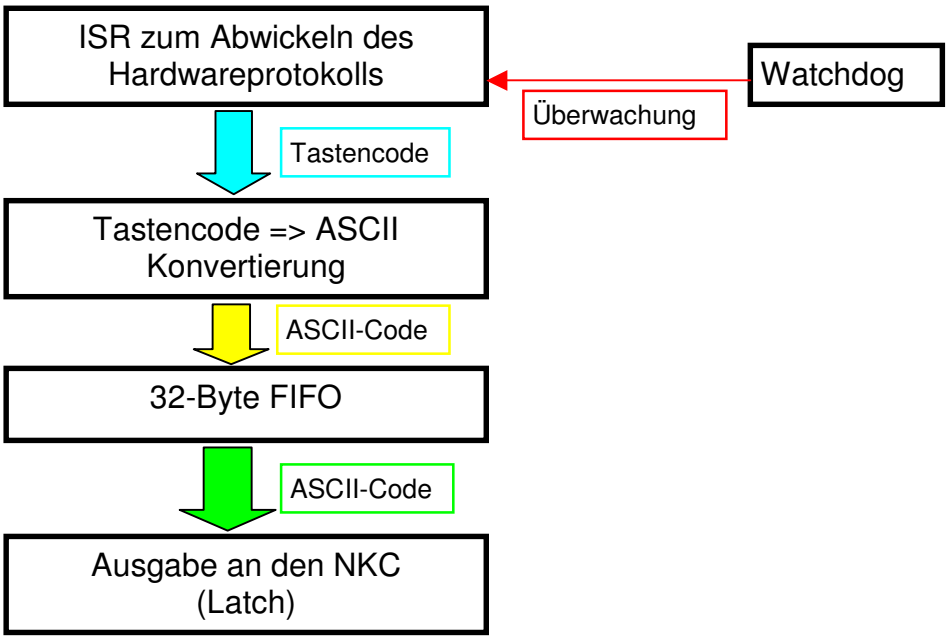
Auch hier noch kurz eine Tabelle der Kommandos die bei der Mauskommunikation verwendet werden (die Liste ist nicht vollständig):

Code	Beschreibung	Sender
$AA	BAT-Complete = Selbsttest der Maus abgeschlossen	Maus
$E8	Auflösung der Mausauszeichnung einstellen	Host
$EE	Echo; Maus antwortet durch Senden von $EE	Host / Maus
$F3	Paketrate (Pakete pro Sekunde) einstellen	Host
$F4	Mausbewegungsauswertung einschalten	Host
$FA	Acknoledge = Bestätigung des Erhaltes eines Kommandos	Maus
$FC	Error = Maus meldet einen Kommandofehler	Maus
$FE	Resend = Letztes Datenpaket erneut senden	Host / Maus
$FF	Reset = Gerät neu starten	Host

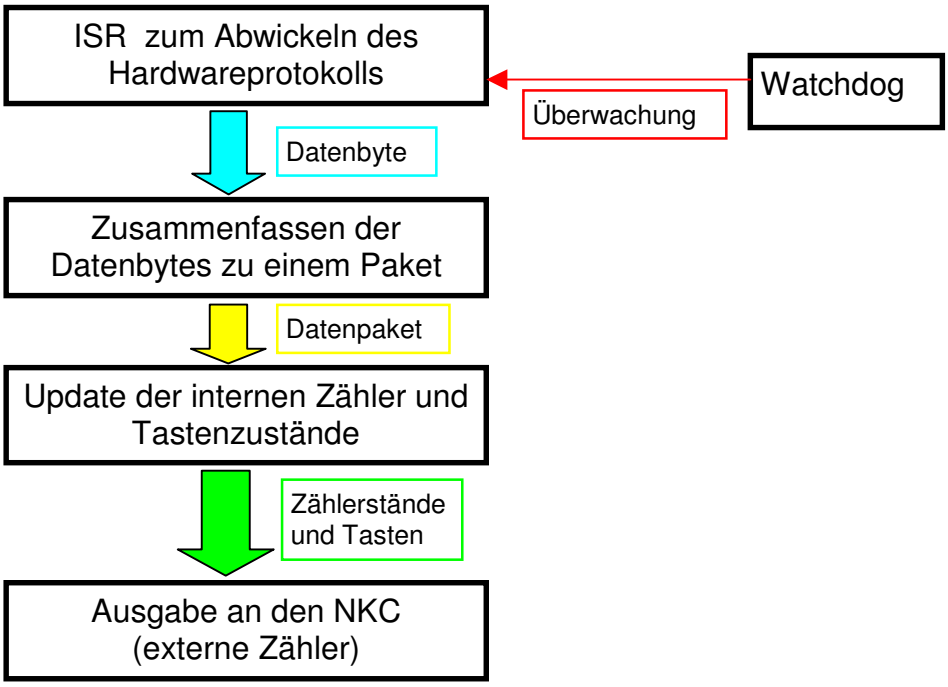
4 Software und Programmierung der Mikrocontroller

Die Software für beide Controller ist in der Programmiersprache „C“ geschrieben. Als Entwicklungsumgebung wurde das AVR-Studio (4.12.498) benutzt. Darin eingebunden ist der C-Compiler WinAVR (20060421). Zum Programmieren der Controller diente ein betagtes STK-500-Entwicklungsboard.

4.1 Blockschaltbild der Software für den Tastaturcontroller



4.2 Blockschaltbild der Software für den Mauscontroller

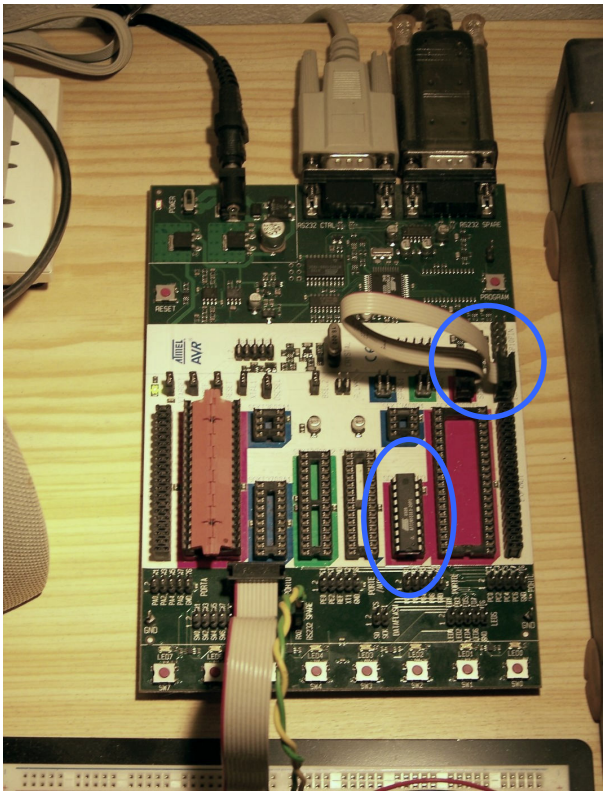


4.3 Beschreibung der ISR zur Kontrolle der PS/2-Schnittstelle

Die Kommunikation mit dem PS/2-Gerät wird im MikroController über eine Interrupt-Service-Routine (ISR) abgewickelt. Dazu muß das CLK-Signal mit dem Eingang des Pins INT0\ am Mikrocontroller verbunden werden.
Nach einem Reset des Mikrocontrollers wird der ISR-Status auf LISTEN gesetzt und auf eine fallende Flanke am CLK-Signal gewartet. Sobald diese detektiert wird, aktiviert der Mikrocontroller die ISR. Die ISR setzt alle Zählvariablen zurück und den Status auf RECEIVE. Anschließend wird auf jeder fallenden Flanke am CLK-Signal ein Bit des PS/2-Protokollrahmens eingelesen bis alle Bits angekommen sind. Danach setzt die ISR ihren Status wieder zurück auf LISTEN und teilt dem Hauptprogramm mit, daß ein Byte empfangen wurde. Das Hauptprogramm verarbeitet dieses dann weiter, während die ISR auf die nächste Sendung wartet. Sollte die Übertragung unvollständig sein und die ISR im Status RECEIVE hängen bleiben, dann greift der Watchdog-Timer ein. Er setzt die ISR zwangsweise zurück in den Anfangszustand.
Das Senden von Daten wird von Hauptprogram nach obigem Schema aktiviert (siehe Kap. 3.3). Sobald eine fallende Flanke detektiert wird, setzt alle Zählvariablen zurück und den Status auf SEND. Anschließen werden alle zu sendenden Bit und das Parity an jeder fallende Flanke am DATA-Signal gesetzt. Danach steht der Status auf ACK und ISR warte auf einen 0-Pegel des DATA-Signals auf einer fallenden Flanke. Tritt auch dieses Ereignis ein, dann wird der Status wieder zurück auf LISTEN gesetzt und dem Hauptprogramm mitgeteilt, daß das Datenbyte gesendet wurde.

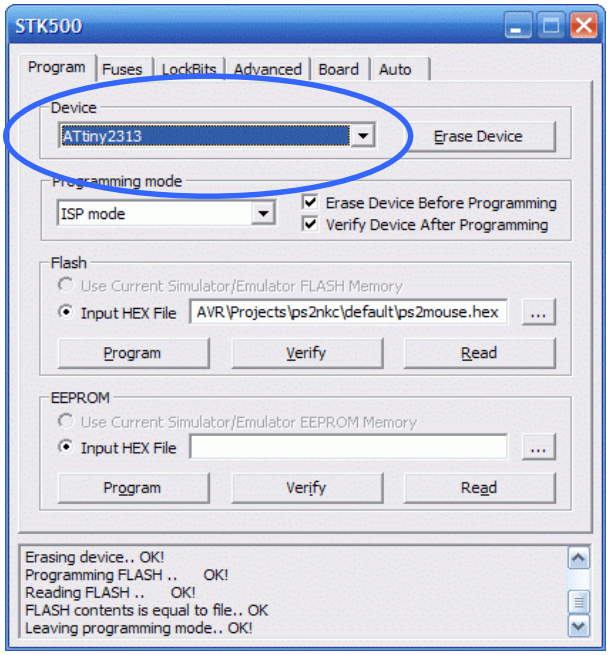
4.4 Programmierung der Mikrocontroller

Die folgende Beschreibung bezieht sich auf das Entwickungsboard STK-500 und der AVR-Studio Entwicklungssoftware von Atmel.

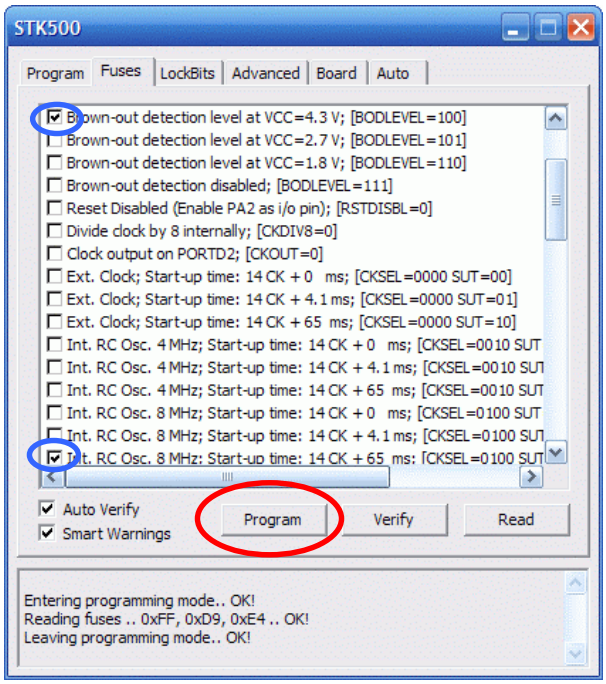


Zuerst muß der Mikrocontroller auf das Entwicklungsboard. Schalten Sie das STK-500 aus und stecken dann einen Attiny2313 in den Sockel SCKT3300D3. Achten Sie darauf, daß der ISP-Stecker in der Stiftleiste SPROG3 steckt. Das neben-stehende Bild zeigt die richtige Konfiguration. Schalten Sie jetzt das STK-500 wieder ein.

Starten Sie nun das Programm AVR-Studio. Wählen Sie im Menü „Program AVR -> Auto Connect“ aus oder klicken Sie in der Toolbar auf dieses Symbol  .



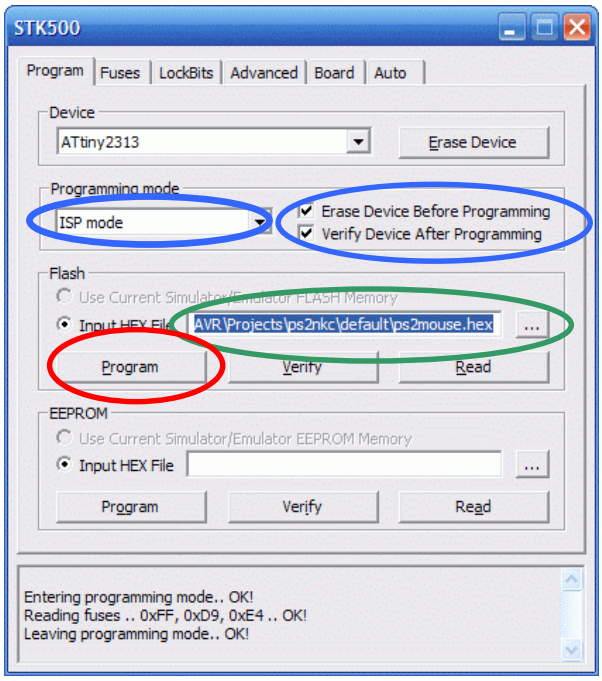
Ein Dialogfenster mit 6 Tabulatorreitern öffnet sich. Wählen Sie zuerst den zu programmierenden Controllertyp aus. Im nächsten Schritt müssen die sogenannten „Fusebits“ programmiert werden, sofern Sie das nicht schon gemacht haben. Klicken Sie dazu den Tabulatorreiter „Fuses“ an.



Sie sehen jetzt eine Tabelle mit den für den Attiny2313 möglichen Einstellungen. Für den sicheren Betrieb des Mikrocontroller benötigen werden nur zwei Optionen benötigt:

- **Brownout Detection 4,3V**
- **Int.Oscillator 8MHz (CKSEL 0100)**

Alle anderen Optionen (kein Häckchen) werden entfernt falls nötig. Klicken Sie jetzt auf die Schaltfläche „Program“. Der Mikrocontroller ist nun betreit zur Programmierung des Programm-speichers (Flash)



Kontrollieren Sie noch schnell,ob die Einstellungen des „Programming Mode“ den im nebenstehenden Bild entsprechen. Dann müssen Sie noch die Programmdatei laden (Input HEX File).

„ps2keyboard.hex“ für Tastaturcontroller
„ps2mouse.hex für Mauscontroller

Klicken Sie jetzt auf die Schaltfläche „Program“. Der Mikrocontroller ist betriebsbetreit sobald der Schreibvorgang beendet ist.

4.5 Sonderfunktionen der Tastatur

Taste	ASCII-Code für NKC	Beschreibung
Pfeil ←	^S	Cursor nach links
Pfeil →	^D	Cursor nach rechts
Pfeil ↑	^Ê	Cursor nach oben
Pfeil ↓	^X	Cursor nach unten
Bild ↑	^R	Seite zurück
Bild ↓	^C	Seite vor
Pos 1	^QS	Cursor zum Zeilenanfang
Ende	^QD	Cursor zum Zeilenende
Einfg	^V	Einfügemodus ein/aus
Entf	^G	Zeichen rechts vom Cursor löschen
F1	^J	Hilfe
F2	^KA	Assembler aufrufen
F3	^L	Suche wiederholen
F4	^QF	Suchen
F5	^KB	Blockanfangmarke setzen
F6	^KK	Blockendemarke setzen
F7	^KC	Block kopieren
F8	^KV	Block verschieben
F9	^P	Zeichenssatz umschalten
F10	^KX	Editor mit Speichern beenden
Strg - Pfeil ←	^A	Cursor ein Wort nach links
Strg - Pfeil →	^F	Cursor ein Wort nach rechts
Strg - Pfeil ↑	^Z	Eine Zeile nach oben
Strg - Pfeil ↓	^W	Eine Zeile nach unten
Strg - Bild ↑	^QE	Cursor in oberste Zeile
Strg - Bild ↓	^QX	Cursor in unterste Zeile
Strg - Pos 1	^QR	Zum Textanfang
Strg - Ende	^QC	Zum Textende
Strg - Einfg	^N	Zeile einfügen
Strg - Entf	^Y	Zeile löschen
Strg - F1	^J	Hilfe
Strg - F2	^KA	Assembler aufrufen
Strg - F3	^L	Suche wiederholen
Strg - F4	^QA	Ersetzen
Strg - F5	^KH	Blockmarken löschen
Strg - F6	^KY	Block löschen
Strg - F7	^QT	Zeile trennen
Strg - F8	^QV	Zeilen verschmelzen
Strg - F9	ESC S	Scrollart umschalten
Strg - F10	^KQ	Editor ohne Speichern beenden
Strg – Alt - Del		Externer Reset der CPU

5 Aufbauanleitung

5.1 Umgang mit IC's

CMOS-Bausteine sind hochempfindlich gegen elektrostatische Aufladung! Bewahren oder Transportieren Sie CMOS-Bausteine nur auf leitenden Schaumstoff! Alle Pins müssen kurzgeschlossen sein.
Achten Sie darauf, daß Sie Verbindung mit einer Erdungsmöglichkeit haben, bevor Sie mit diesen Bausteinen arbeiten. Geeignete ESD-Artikel gibt es im Fachhandel.

5.2 Stückliste

Anzahl	Kennung	Bauteil (klassisch)	Bauteil (alternativ)
1		Platine	
2	IC 3, 13	Atmel ATtiny2313	
2	IC 1, 8	74LS244	74HCT244
1	IC 2	74LS374	74HCT374
2	IC 4, 5	74LS688	74HCT688
2	IC 6, 7	74LS138	74HCT138
4	IC 9 - 12	74LS590	74HC590
1	IC 14	74LS74	74HCT74
1	IC 15	74LS00	74HCT00
1	IC 16	74LS32	74HCT32
1	IC 17	74LS125	74HCT125
1	C 1	Kondensator 10µF Tantal	
2	C 8, 10	Kondensator 4,7 µF Tantal	
9	C 2 – 7, 9, 11, 12	Kondensator 100nF MKS-2	
2	C 13, 14	Kondensator 1 µF Elko oder MKS-2	
3	R 1, 2, 4	Widerstand 10 kΩ	
1	R 3	Widerstand 330 Ω	
1	R 5	Widerstand 220 Ω	
1	RN 1	Widerstandsnetzwerk 8 x 4,7 kΩ	
2	RN 2, 3	Widerstandsnetzwerk 4 x 4,7 kΩ	
1	D 1	LED, standard	
1	ST 1	Messerleiste DIN 41612 Bauform C 64 pol für ECB-Bus	
1	ST 2	Stiftleiste 1 x 36 pol & 1 x 18 pol gew. für NDR-Bus	
1	ST 3	Mini-DIN-Einbaubuchse violett	
1	ST 4	Mini-DIN-Einbaubuchse grün	
1	ST 5	Stiftleiste 2 x 5 pol	
1	ST 6	Stiftleiste 2 x 2 pol	
1	ST 7	Stiftleiste 1 x 2 pol	
optional:			
1		Widerstandsnetzwerk 4 x 10 kΩ	
1		Transistor BC328	
2		Widerstand 10 kΩ	
1		Widerstand 10 kΩ	
1		Stiftleiste 1 x 3 pol	
1		Jumper	

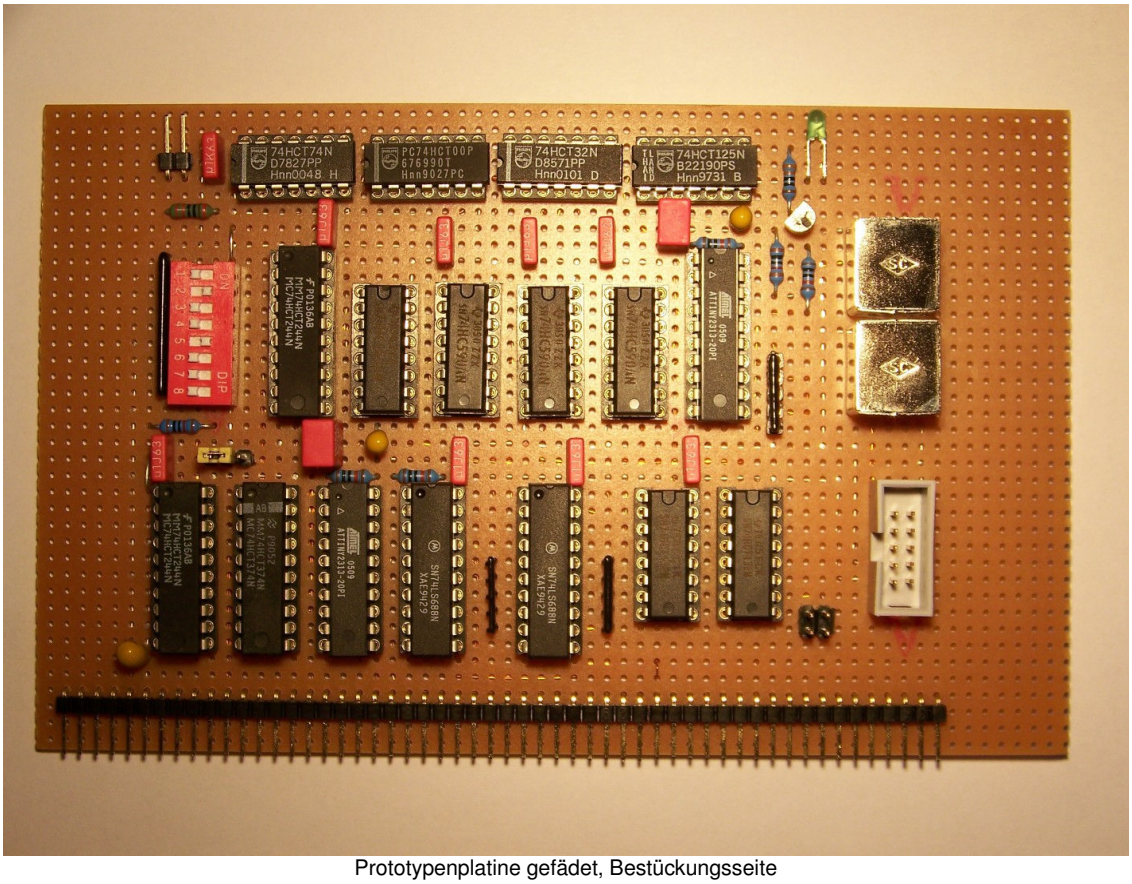
IC 3: Tastaturcontroller
IC13: Mauscontroller

5.3 Aufbau Schritt für Schritt

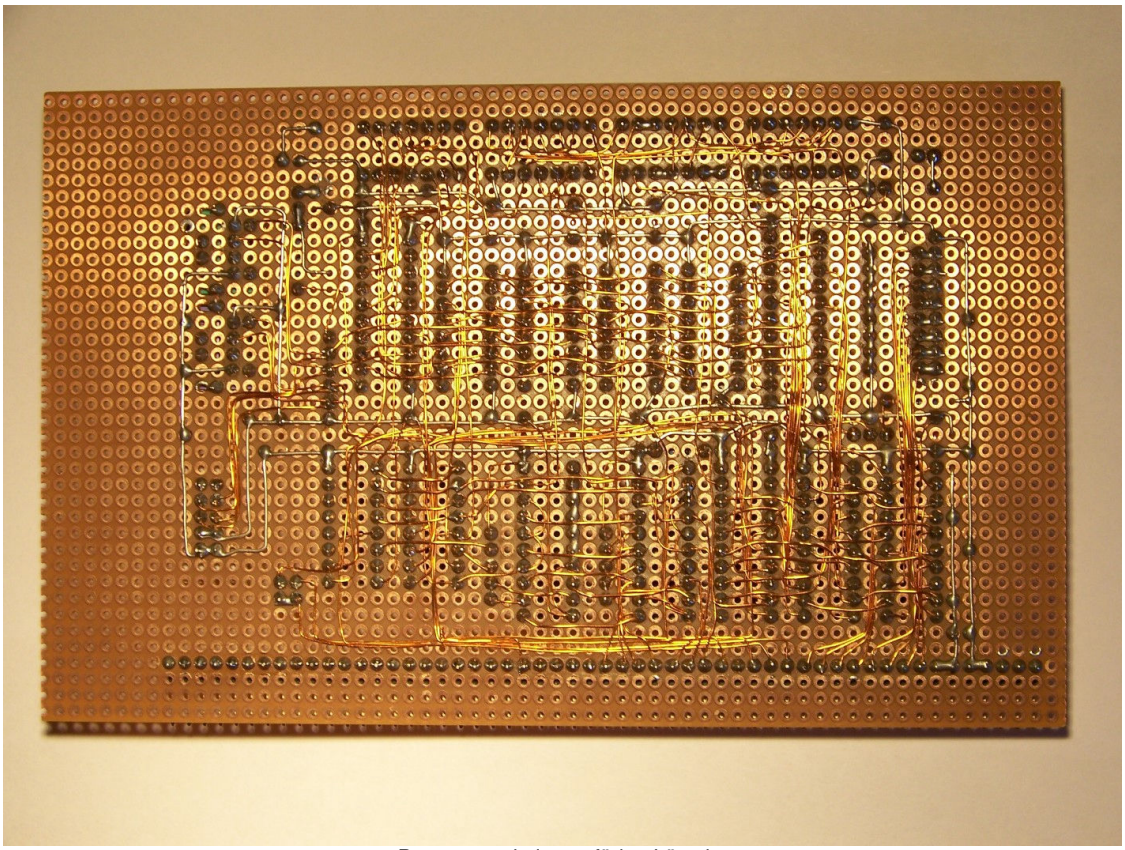
Entfällt

Eine Platine werde ich aus Kostengründen vorerst nicht produzieren lassen. Deshalb spare ich mir eine detaillierte Anleitung.

Ich habe ich mir eine Prototypenplatine möglichst layoutnah gefädelt. Als Anregung zum Nachbau die Bilder meiner Baugruppe:



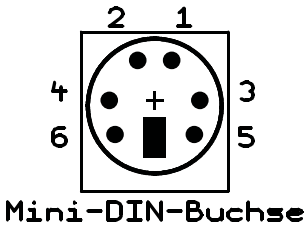
Prototypenplatine gefädet, Bestückungsseite



Prototypenplatine gefädet, Lötseite

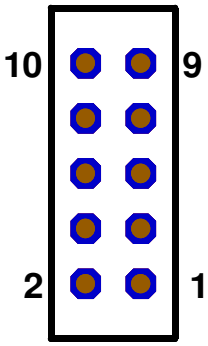
5.4 Belegung von Steckleisten, Buchsen und Jumpern

ST 3, 4:



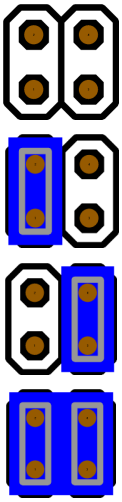
PIN	Bezeichnung
1	Data
2	nc
3	GND
4	Vcc, 5V
5	Clock
6	Nc

ST 5:



PIN	Bezeichnung
1	Vcc, 5V
2	Vcc, 5V
3	GND
4	GND
5	Tastatur Data
6	Tastatur Clock
7	Maus Data
8	Maus Clock
9	LED – Anode
10	LED – Kathode

ST 6:



Es wird kein Interrupt ausgelöst.

Es wird nur ein Interrupt vom Tastaturcontroller ausgelöst, wenn ein Tastencode von der Tastatur verarbeitet wurde.

Es wird nur ein Interrupt vom Mauscontroller ausgelöst, wenn ein Datenpaket von der Maus verarbeitet wurde.

Beide Mikrocontroller können Interrupts auslösen.

6 Neuerungen und Fehlerkorrekturen

In Laufe der Zeit haben sich leider ein paar Unzulänglichkeiten im Design der Baugruppe gezeigt. An dieser Stelle folgt eine Auflistung möglicher Fehler und deren Beseitigung bzw. Umgehung.

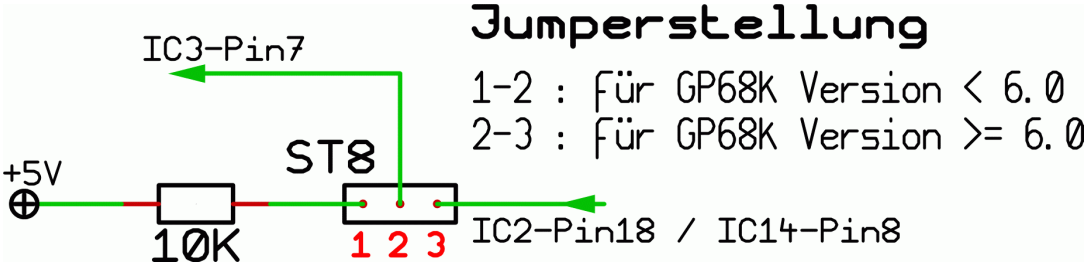
- **<Ctrl>-<Alt>-<Del> und Timer NE555**
  Im Gegensatz zu den Originalschaltungen der CPU-Baugruppen verwende ich nicht den Timer NE555 zur Auslösung eines Hardware-Reset, sondern den Spannungswächter TL7705A, wie er im Handbuch der ROA256/1M vorgeschlagen wird. Dieser entlädt den Kondensator selbst, sodaß nur ein kurzer Impuls (ca. 250 µs) nötig ist, um den Reset auszulösen. Leider reicht die Zeit nicht, um den Kondensator beim NE555 zu entladen. Bei einem Entladestrom von 25 mA muß die Leitung mindestes 10 ms gegen Masse gehalten werden, um den Kondensator (10 µF) sicher zu entladen.
  Damit die Baugruppe auch bei Verwendung des Timers NE555 einen Hardware-Reset auslösen kann, ist sowohl eine Änderung der Software des µC als auch eine Hardwareänderung nötig. Verwenden Sie das Tastaturcontroller-Programm vom 14.04.2007 oder später. Trennen Sie die Leitung zwischen IC3-Pin4 und der 2poligen Stiftleiste auf und löten Sie einen 220 Ω Widerstand dazwischen. Dieser dient zur Strombegrenzung, damit der Tastaturcontroller nicht abraucht.
  Diese Änderung ist im Schaltplan und dem Platinenlayout der Revision 4 bereits enthalten (siehe Anhang).

  Neues Platinenlayout
  Das bisherige Platinenlayout war nur professionell herstellbar. Um der Forderung nachzukommen, diese Platine auch mit semiprofessionellen Geräten herstellen zu können, habe ich das Layout komplett überarbeitet:

  - Leitungsbreite von 0,3 auf 0,4 mm erhöht
  - Leitungszwischenraum von 0,15 auf 0,25 mm erhöht
  - Kleinster Bohrungsdurchmesser von 0,3 auf 0,8 mm erhöht
  - Kleinster Paddurchmesser von 0,6 auf 1,5 mm erhöht
  - Kleinste Restringbreite von 0,15 auf 0,3 mm erhöht

Ich hoffe, daß sich der Aufwand gelohnt hat und die Arbeit eines Nachmittages nicht umsonst war. Mehr ist bei dieser Bestückungs- und Leitungsdichte nicht mehr möglich.

Verwendung des Grundprogramm 68k Version älter als 6.0
Beim Test der Funktion des Hardware-Reset mit Timer NE555 und einer original beschalteten CPU68k-Baugruppe mit Grundprogramm 68k V4.3 stellte ich fest, daß keine Zeichen vom Tastaturcontroller an das Latch weitergereicht werden. Damit kommt auch nie ein Zeichen bei der CPU an. Das liegt daran, daß die Tastatur-Baugruppe beim Start der Grundprogramms nicht initialisiert wird (Lesen der Schalterstellung des Mäuseklaviers).
Man kann sich hier mit einer Schaltungsergänzung als Workaround helfen. Leider funktioniert dann der 32 Zeichen Tastaturpuffer des Tastaturcontrollers nicht mehr, was aber nicht unbedingt funktionsentscheidend ist.



Der Test mit einer der Z80-Baugruppe steht noch aus. Zur Zeit kann ich dies nicht durchführen, da meine Z80-Baugruppen noch nicht restauriert sind.

7 Literaturhinweise und –nachweise

Bücher

- Rolf-Dieter Klein
  „Rechner modular“
  Der NDR-Klein-Computer – selbstgebaut und programmiert
  Franzis-Verlag, München. ISBN 3-7723-8721-7
- Rolf-Dieter Klein
  “Die Prozessoren 68000 und 68008“
  Rechnerarchitektur und Sprache im NDR-Klein-Computer
  Franzis-Verlag, München. ISBN 3-7723-7651-7

Datenblätter

- Atmel ATtiny2313 Prelimitary, Rev. 2543i-AVR-04/06
- Appl.Note AVR313 – Interfacing the PC AT Keyboard,
  Rev.1235b–AVR–05/02
- KEY Aufbaubauanleitung
- HCOPY Aufbauanleitung

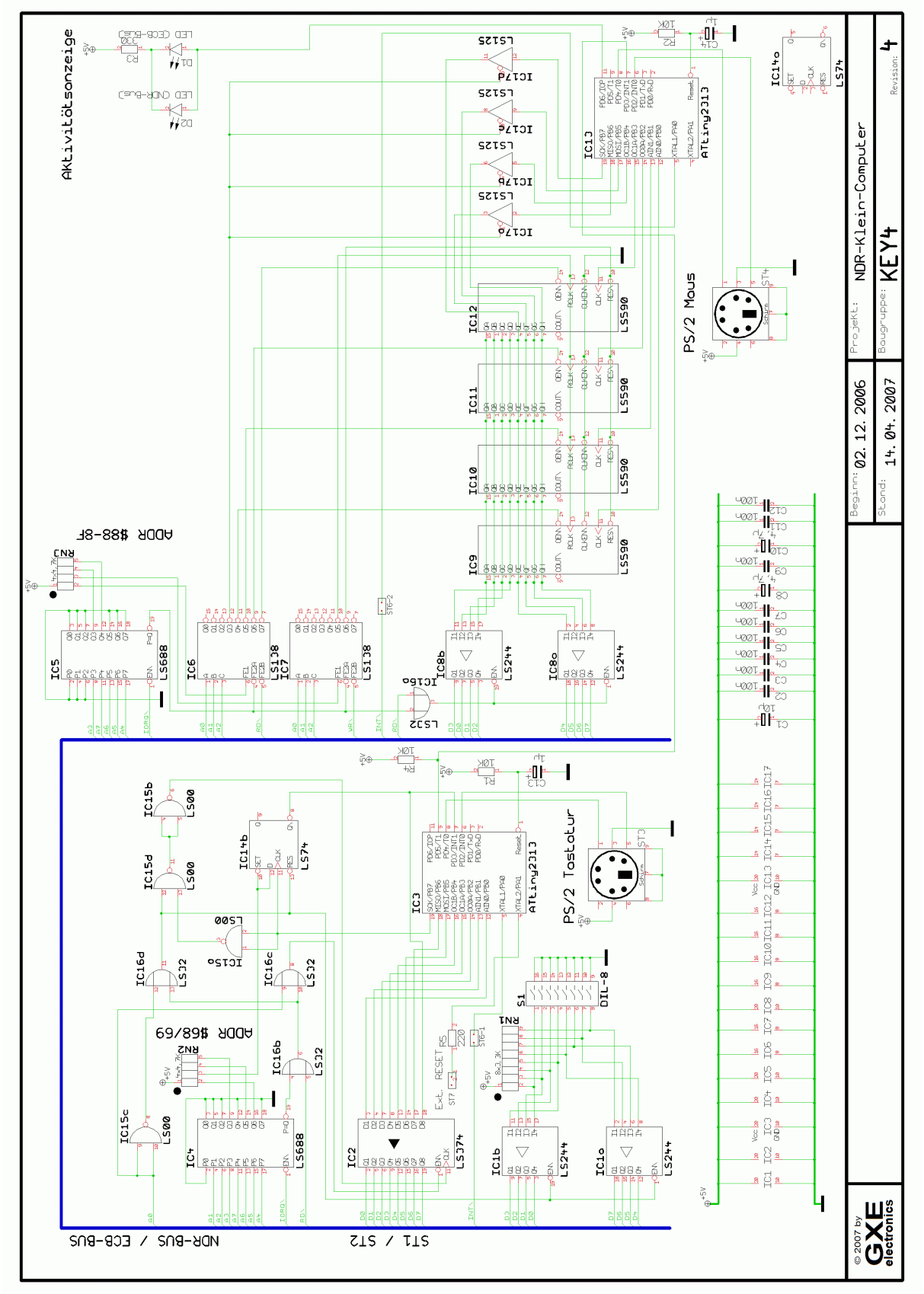
Quellen im Internet

http://www.atmel.com/avr/
http://sourceforge.net/projects/winavr/
http://www.nongnu.org/avr-libc/
http://www.computer-engineering.org
http://www.beyondlogic.org

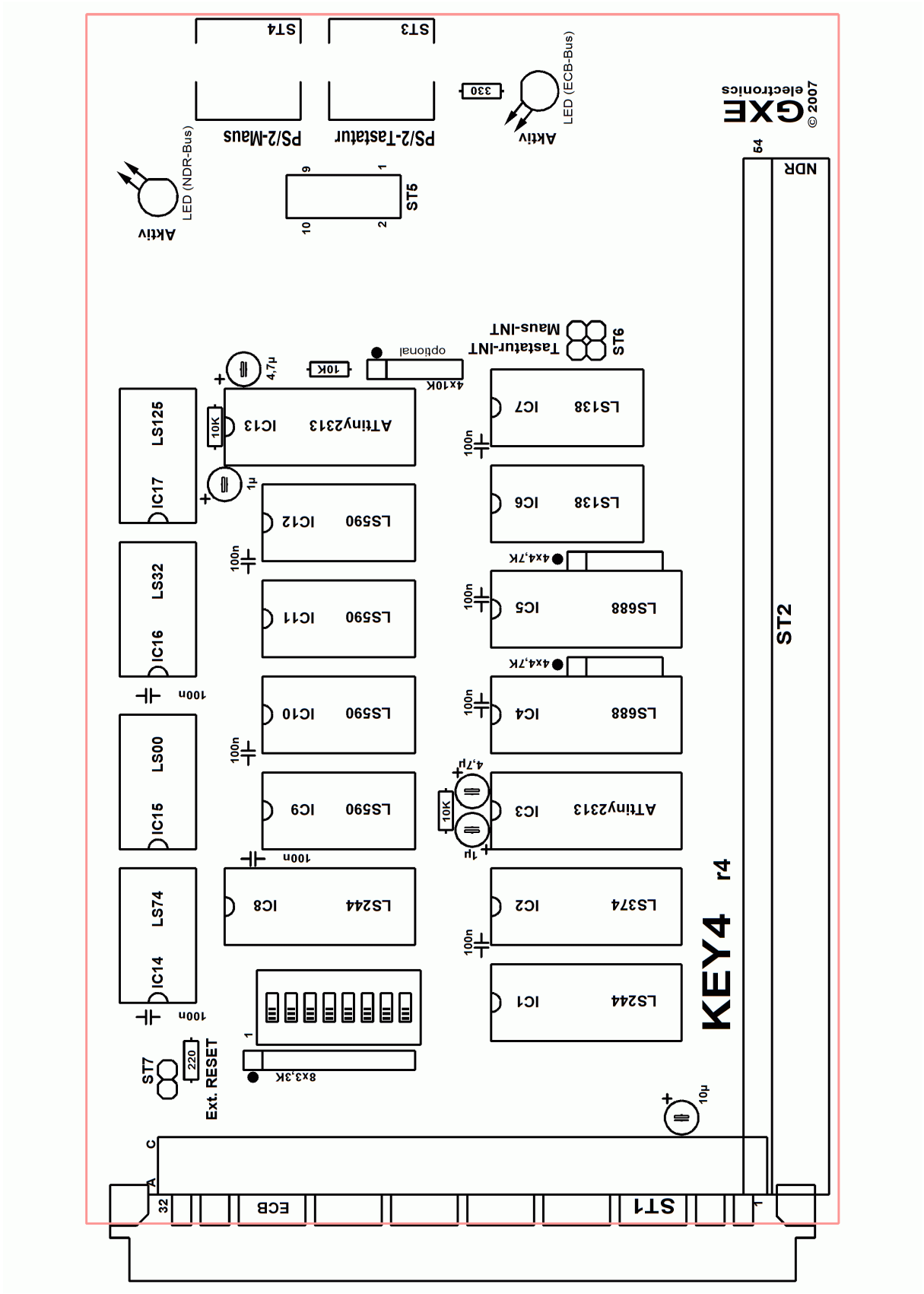
NKC-Forum

http://www.drcrazy.de/forum/

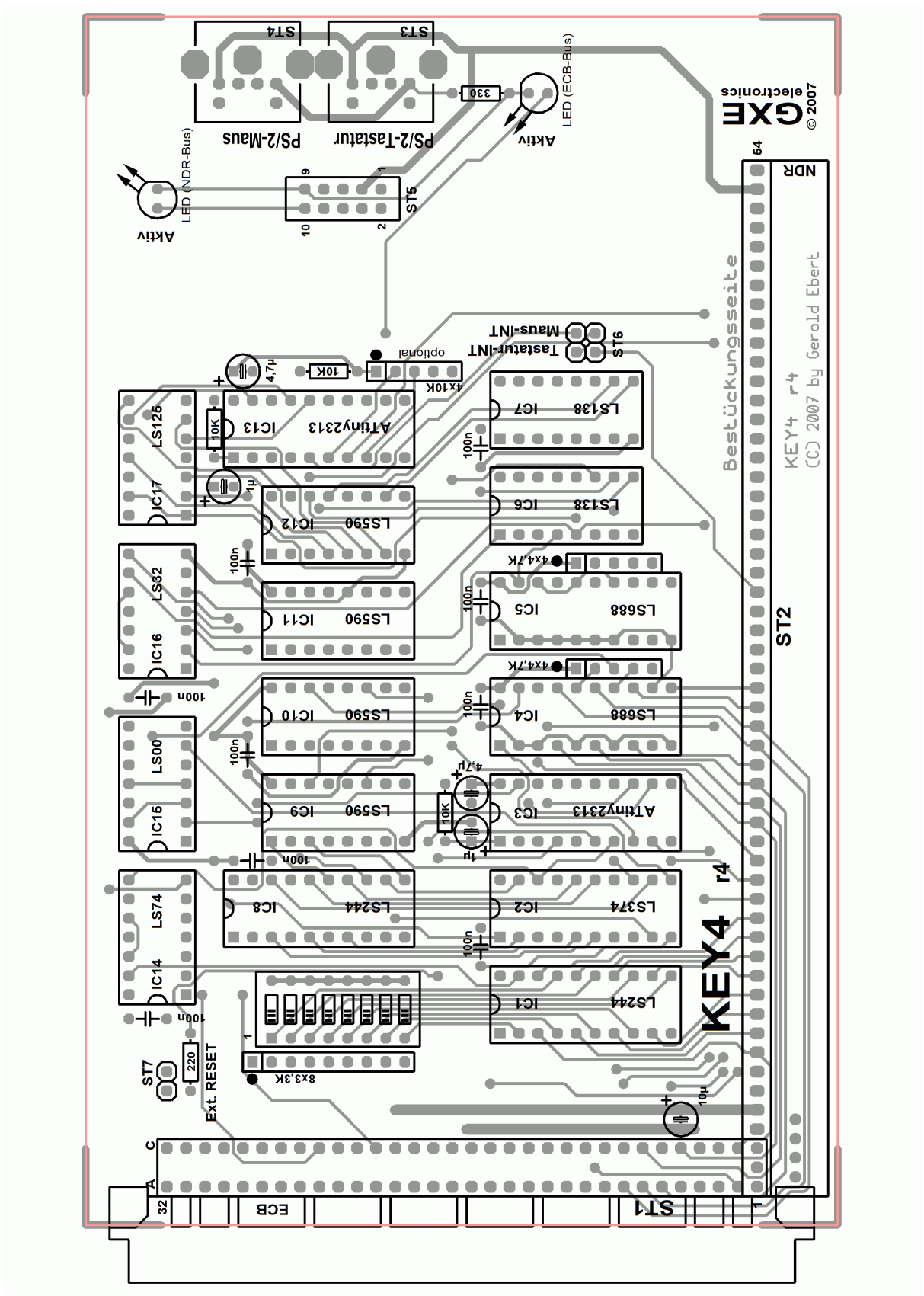
Schaltplan


Aktivitätsanzeige
PS/2 Maus
PS/2 Tastatur
ADDR $88-8F
ADDR $68/69
NDR-Klein-Computer
KEY4
02. 12. 2006
14. 04. 2007
GXE electronics
© 2007 by

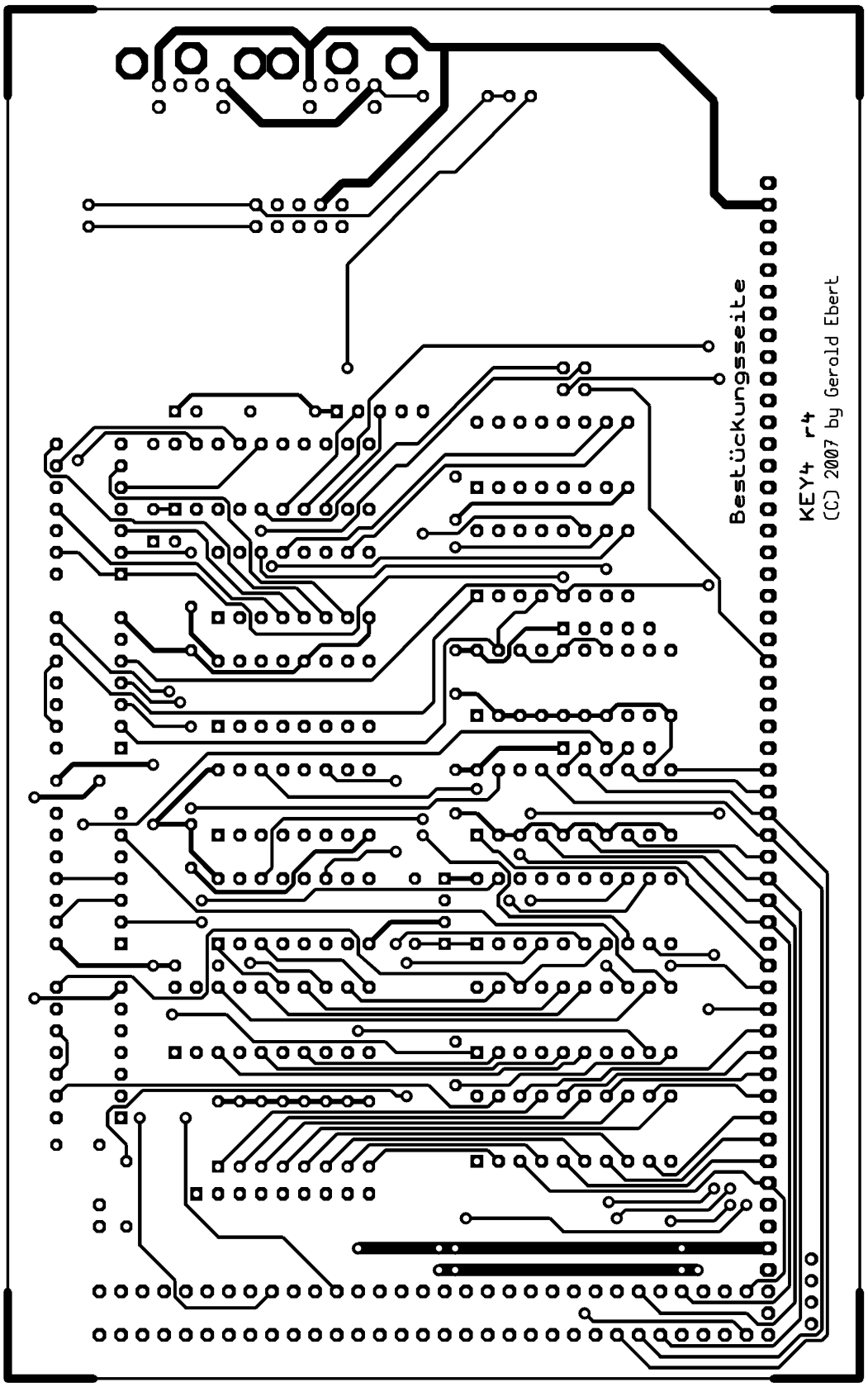
Bestückungsplan



Layout Bestückungsseite mit Aufdruck



Layout Bestückungsseite



Layout Lötseite