

GORAT80

GOttlieb Rom And Test for System80

ralf@lisy.dev

06.01.2022 Version 0.6

Inhaltsverzeichnis

1. Einführung	3
2. Betrieb	4
2.1. Vorbereitung System 80 MPU	4
2.2. Normalbetrieb (DIP1: Closed; DIP2:Closed)	6
2.2.1. Visualisierung Reset Circuit/Einschaltverzögerung (D1 und D2).....	6
2.2.2. Visualisierung Clock (D3)	6
2.2.3. Visualisierung IRQ(D4)	6
2.2.4. Visualisierung Reset Board(D6)	7
2.3. Normalbetrieb ohne Reset (DIP1: Closed; DIP2:Open)	7
2.4. Leon Borre Testmodus (DIP1: Open; DIP2:N/A)	7
3. Hardware	8
3.1. Zusammenbau	9
3.2. Platine Details.....	11
3.3. PIC16F688.....	11
3.4. Adressdekodierung (74LS02).....	11
3.5. Eprom 27256	12
3.6. Beispiele Erstellung Image Eprom 27256.....	12
3.6.1. System80 bei zwei Game-Proms mit jeweils 1K.....	12
copy /b 4K-Free.bin + 657-1.cpu + 657-2.cpu + 2K-Free.bin + u2_80.bin + u3_80.bin + 8K-	
Free.bin + 2764af.bin gorat80_661.bin.....	13
3.6.2. System80 und 80A bei einem Game-Proms mit 2 K.....	14
3.6.3. System80B mit 2K Prom	15
3.6.4. System80B mit 4K Game-Prom (nicht direkt unterstützt mit v02)	16
4. Testen nach ‚Leon Borre‘	17
4.1. Einleitung.....	17
4.2. Start	18
4.3. Nur ein Flash.....	18
4.4. Zwei Flashes = U4 , drei Flashes = U5, und vier Flashes = U6.....	19
4.5. Über das Testprogramm.....	20

1. Einführung

GORAT80 (**G**Ottlieb **R**om **A**nd **T**est System **80**) ist eine ‚intelligente‘ Adapterplatine welche auf den ‚TC-1 Steckplatz‘ einer Gottlieb System 80 (inkl. 80A und 80B) MPU gesteckt wird und folgende Funktionalitäten bereitstellt:

- Überwachung und Visualisierung (auch im laufenden Betrieb) von
 - Reset
 - Clock
 - IRQ
- Ersatz für alle Roms (Game Rom und System ROMs Prom1, Prom2,U2 und U3)
- Ersatz für das System 80 Reset Board (inkl. Visualisierung bei Auslösung)
- Test der Hardware: CPU, RAM sowie die der drei RIOTs (Leon Borre Test ROM)



Bild 1: GORAT80 Version 02

2. Betrieb

2.1. Vorbereitung System 80 MPU

Auf der MPU muss der Steckplatz ‚TC-1‘ (in der Mitte oben auf jeder System 80 MPU) entweder gesockelt sein, um GORAT80 aufzunehmen. (Alternativ kann GORAT80 aber auch fest eingelötet werden). Dies sollte bei allen MPUs der System80 und 80A der Fall sein.

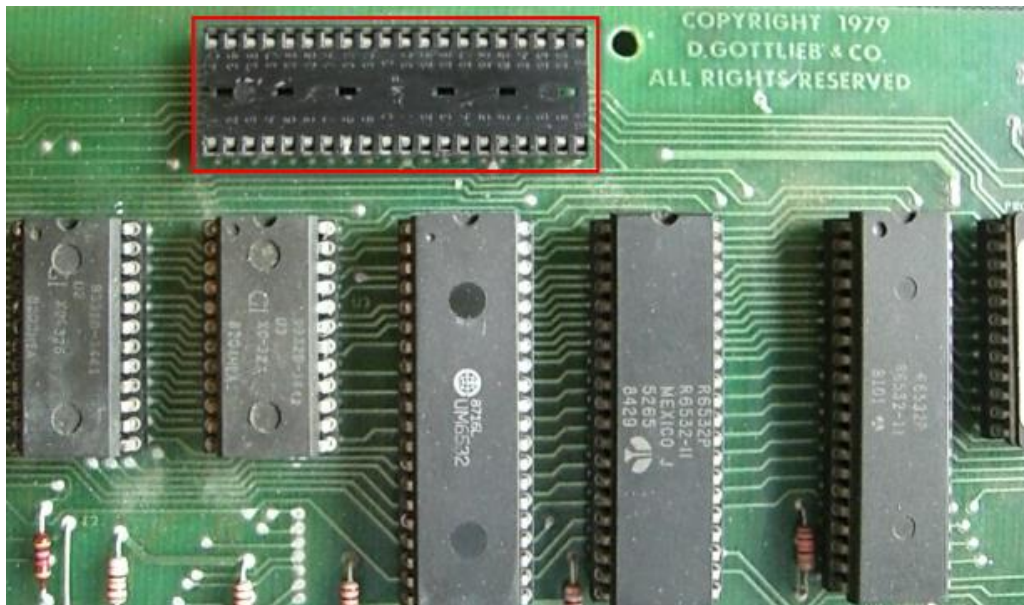


Bild 2: System80 MPU mit TC1-Sockel

Bei manchen System 80B MPUs fehlt der IC-Sockel, dann ist meistens der Anschluss für das Reset-Board dort angelötet (Pfostenstecker). In diesem Fall muss der Pfostenstecker entfernt und stattdessen ein 40-Poliger IC-Sockel eingelötet werden. HINWEIS: In diesem Fall ist auch die Diode und der Widerstand an diesen Anschlüssen zu entfernen, da diese bereits auf GORAT80 vorhanden sind! (Im Bild unten rot eingekreist)

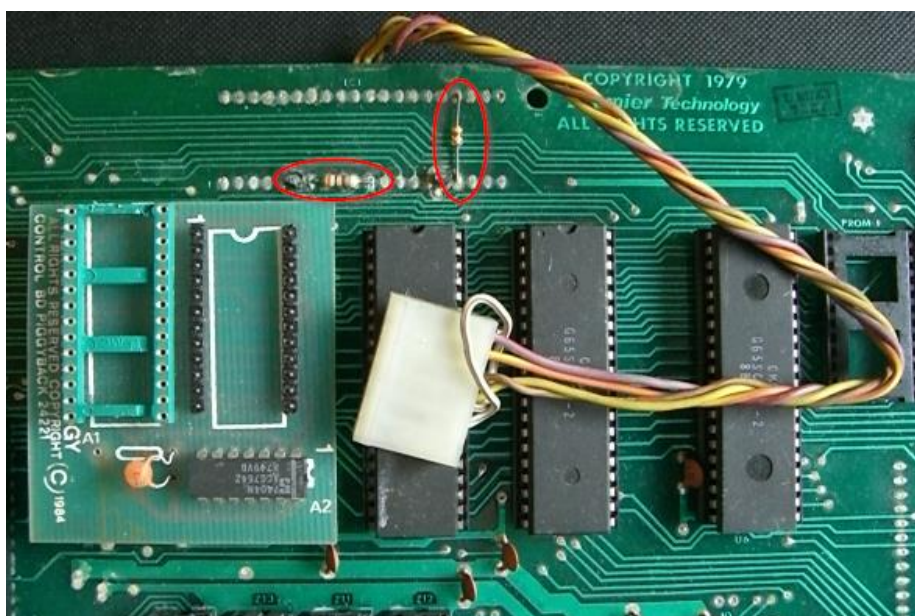


Bild3: System80B MPU ohne Sockel für TC-1

Am einfachsten entfernt man alle originalen ROMs (Game ROMs und System ROMs) damit GORAT80 die Kontrolle übernehmen kann. Bei nicht gesockelten PROMs entweder zerstörungsfrei durch auslöten, oder man einfach mittels eines kleinen Seitenschneiders indem man die ‚Beine‘ abknipst.

Will man eines der ROMs (noch) auf der Platine belassen, muss eine Brücke auf der MPU gesetzt werden um die Selektion dieser Proms zu unterbinden. Dafür wird ein Bein von Widerstand R42 auf Masse (Ground) gezogen. (Im Bild unten mit einem Pfeil gekennzeichnet).

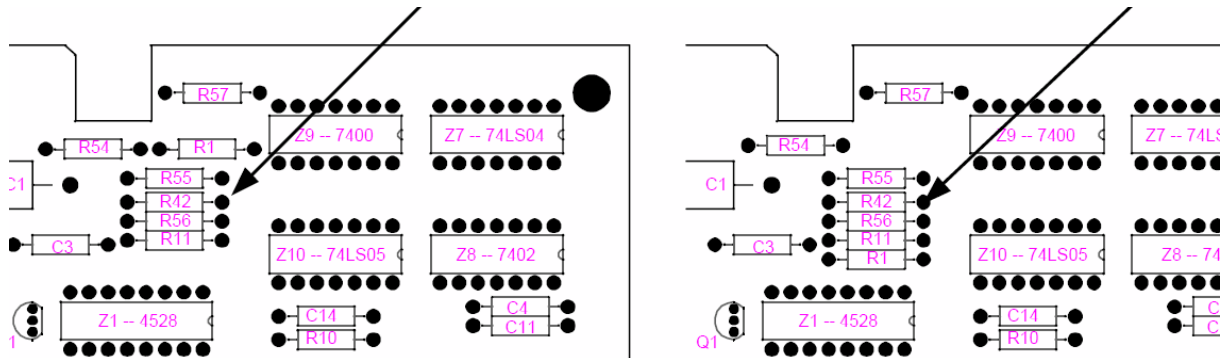


Bild 4: Anschlusspunkt für System80 MPU ‚alt‘ (links) und ‚neu‘ (rechts).

Am einfachsten nimmt man den Groundanschluss (-) am Kondensator direkt an der 5V Versorgung. Im Bild unten die gesetzte Brücke als Beispiel.

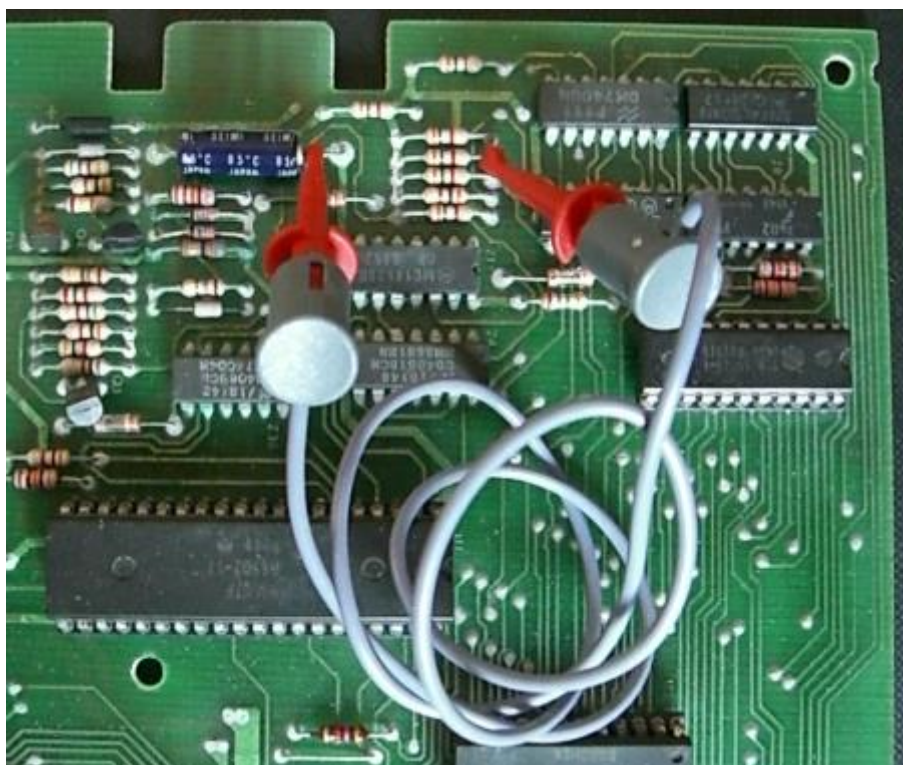
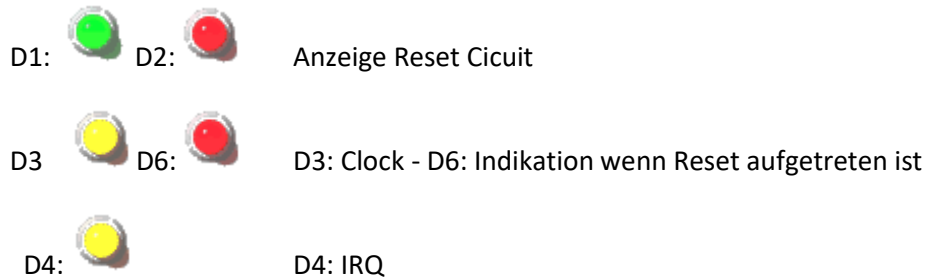


Bild 5: System80 MPU mit gesetzter Brücke

2.2. Normalbetrieb (DIP1: Closed; DIP2:Closed)

Im Normalbetrieb (beide DIP-Schalter auf closed/unten) werden der Reset sowie die Clock und IRQ visualisiert bzw. überwacht. Fallen die IRQs aus (≥ 1 Sekunde) wird das Board resettet (Funktion original Reset-Board)



Wenn alles funktioniert, ist der Status der LEDs wie auf dem Bild oben. D1 (Grün) ist an, D3 & D4 (Gelb) blinken, D2 & D6 (Rot) sind aus.

2.2.1. Visualisierung Reset Circuit/Einschaltverzögerung (D1 und D2)

Für die Funktion der System 80 MPU ist muss der Reset Circuit einwandfrei funktionieren. Er hält den/RESET Eingang der 6502-CPU nach dem Einschalten (Power-ON) für wenige Millisekunden auf Low und hindert den 6502 damit am Start. Dies ist notwendig da für den Start ein stabiler Systemzustand erforderlich ist. Der Reset Circuit ist in einer Region auf der System80 MPU die als erstes von einer eventuell ausgelaufenen Batterie in Mitleidenschaft gezogen wird und deshalb gerne defekt ist.

Bei einwandfreier Funktion sollte man direkt nach dem Einschalten ein kurzes Aufleuchten von D2 (rote LED) welches in eine dauerhaftes Leuchten von D1 (grüne LED) übergeht beobachten können. Bei neueren System80 MPUs ist der Wechsel so schnell, dass GORAT80 selber noch nicht bereit ist, und man direkt die grüne LED sieht.

Bleibt D2 (rot) an und geht nicht auf die grüne LED über, muss der Reset Circuit kontrolliert werden.

2.2.2. Visualisierung Clock (D3)

Ist die Clock (der Takt der CPU) in Ordnung, wird dies durch regelmäßiges Blinken (Sekundentakt) von D3 (erste gelbe LED) angezeigt. Das Blinken sollte sofort nach dem Einschalten anfangen und (einigermaßen) regelmäßig auftreten.

Bei Unregelmäßigkeiten oder ständigem An bzw. Aus von LED 3 ist die Clock Sektion auf der MPU zu kontrollieren. Auch diese befindet sich in einem Bereich der gerne von auslaufenden Batterien in Mitleidenschaft gezogen wird.

2.2.3. Visualisierung IRQ(D4)

Ist der IRQ (interrupt request) in Ordnung, so wird dies wenige Sekunden nach dem Einschalten durch ein Blinken von LED 4 (zweite gelbe LED) angezeigt. Das Blinken setzt bei alten System 80 Geräten etwa 3-4 Sekunden nach dem Einschalten ein, neuere System80 und 80A/80B Geräte booten wesentlich schneller, so dass hier das Blinken früher einsetzt. Die Blinkfrequenz ist abhängig davon ,was der Flipper gerade macht', sollte aber auch etwa im Sekundentakt liegen.

Ein fehlerhafter IRQ kann mehrere Ursachen haben. Zumeist liegt es daran, dass die CPU ihr Programm nicht finden kann, dann sind die PROMs, sowie die Adressdekodierung zu prüfen. Auch kann es an einem oder mehreren defekten RIOTs liegen. In diesem Fall ist der integrierte Testmodus sehr hilfreich. (siehe Abschnitt ‚Leon Borre Testmodus‘).

2.2.4. Visualisierung Reset Board(D6)

Sobald der IRQ anläuft, schaltet GORAT80 die IRQ Überwachung ‚scharf‘. Fällt der IRQ für mehr als eine Sekunde aus, wird das Board über den /Reset Eingang automatisch neu gestartet und dies über D6 (rote LED rechts) angezeigt. Auch nach dem Boot bleibt D6 an, damit im Nachhinein festgestellt werden kann ob während der Laufzeit ein Reset ausgelöst wurde. Nach eine Neustart (Power Off/On) ist die LED wieder zurückgesetzt (aus).

Diese Überwachung wurde von Gottlieb damals über das ‚Reset Board‘ eingeführt welches alle neueren System 80 Geräte haben (Das original Reset Board hat jedoch die beschriebene Visualisierung aber nicht). Hintergrund war, dass wenn die CPU in zu einem ungünstigen Zeitpunkt stoppt (blockiert) und das Gerät nicht beaufsichtigt ist, durchaus einmal die ein oder andere Spule durchbrennen kann. Mit einem sofortigen Neustart im Fehlerfall wollte man dem entgegenwirken.

2.3. Normalbetrieb ohne Reset (DIP1: Closed; DIP2:Open)

Bei geöffnetem DIP Swich 2 (Schalter oben) verhält sich GORAT80 wie unter Punkt 2. beschrieben, mit der Ausnahme, dass im Falle eines defekten IRQs dies zwar über LED 6 angezeigt, der entsprechende Reset aber nicht ausgeführt wird. Die sollte nur zu Testzwecken eingeschaltet werden oder wenn man Zweifel der einwandfreien Funktion von GORAT80 hat (zum Beispiel bei häufigem Reset, obwohl das Board ohne GORAT80 vorher einwandfrei funktioniert hat).

2.4. Leon Borre Testmodus (DIP1: Open; DIP2:N/A)

Wird DIP Swich 1 of Open (Schalter oben) gestellt, werden nicht die System 80 ROMs in den Adressraum eingeblendet sondern ein von Leon Borre entwickeltes Testprogramm. Der Zustand der einzelnen Komponenten (CPU, RAM 5101 und die drei RIOTS) wir überprüft und durch eine ‚Blinkfolge‘ von LED6 angezeigt. In diesen Modus ist die Reset Überwachung (inklusive Visualisierung) nicht eingeschaltet, somit hat die Position von Dip Switch 2 hier keine Bedeutung.

Eine Beschreibung wie dieser Test funktioniert und wie man die identifizierten Fehler beheben kann ist in Kapitel 4 ‚Test nach Leon Borre‘ beschrieben.

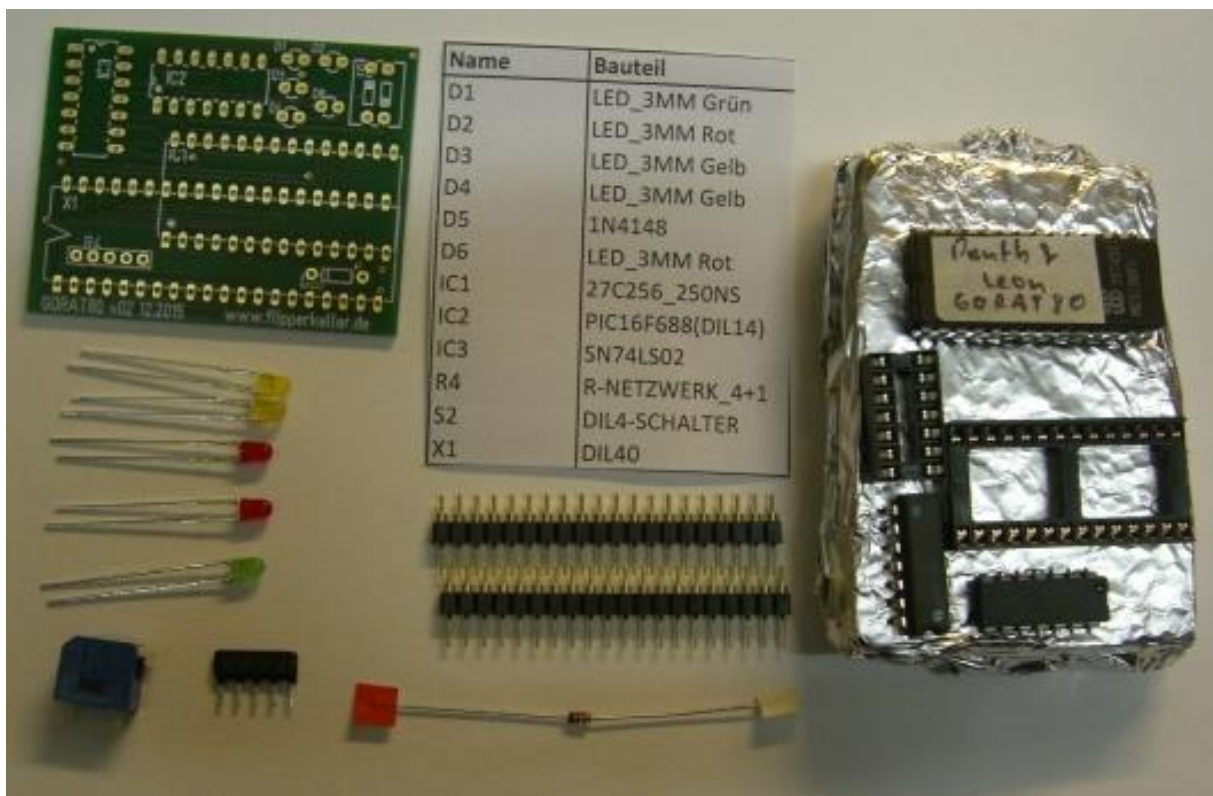
3. Hardware

Nachfolgend die komplette Stückliste sowie ein Link auf den entsprechenden Warenkorb zur direkten Bestellung bei Reichelt.

Pos	Name	Wert	Gehäuse
1	D1	L-934GD-5V	LED_3MM
2	D2	L-934GD-5V	LED_3MM
3	D3	L-934GD-5V	LED_3MM
4	D4	L-934GD-5V	LED_3MM
5	D5	1N4148	DO35
6	D6	L-934GD-5V	LED_3MM
7	IC1	27C256_250NS	DIL28
8	IC2	PIC16F688(DIL14)	DIL14
9	IC3	SN74LS02	DIL14
			R-
10	R4	4,7K DIL-SCHALTER_2-	NETZWERK_4+1
11	S2	POL.	DIL4-SCHALTER
12	X1	DIL40	DIL40

Link auf Reichelt Warenkorb:

<https://secure.reichelt.de/index.html?&ACTION=20&AWKID=1147827&PROVID=2084>

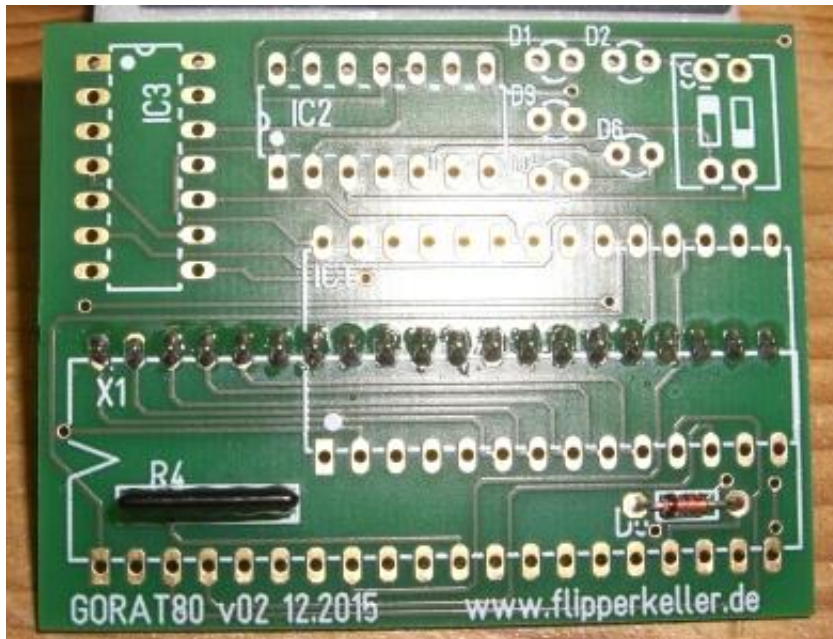


Alle Bauteile von GORAT80

3.1.Zusammenbau

Folgende Reihenfolge beim Einlöten hat sich bewährt:

1. Erster Teil des 20-Poliger Adapters(X1), hier erst einmal nur der Mittelsteg, da die zweite Reihe ansonsten beim verlöten der Fassung für IC1 im Weg ist. Damit dieser Teil gerade eingelötet wird, am besten mit einer IC-Fassung, zusammen mit dem zweiten Teil, stabilisieren.
2. Diode D5 & Widerstandsarray, hier darauf achten, dass die Kathode (der Ring) der Diode nach links zeigt und Pin1 des Widerstandsarrays (die Seite mit dem Punkt) rechts liegt.



Platine nach Schritt 2

3. Fassung für IC1
4. Zweiter Teil des Adapters (X1)



Platine nach Schritt 4

5. IC3 SN74LS02
6. Fassung für IC2
7. Leuchtdioden D1..D4 & D6. Die Kathode muss hier nach rechts zeigen. Bei einer neuen Diode erkennt man die Kathode am kürzeren der beiden Anschlusspins. Zusätzlich ist die Kathode entweder mit einer leicht abgeflachten Seite oder mit einem kleinen Punkt auf dem Gehäuse gekennzeichnet.
8. Dip-Switch S2. So einbauen, dass Dip-Switch Nummer 1 außen liegt.



Fertig aufgebaute Platine.

3.2. Platine Details

Die Verbindung zum TC-1 Adapter wird über eine Adapter hergestellt (X1 unten).

Hauptkomponenten sind die leicht überlappend installierte Fassung für das Eprom (IC1); IC2 ein ‚PIC‘ zur Steuerung und ein Standard TTL NOR (IC3) zur Adressdekodierung.

HINWEIS: Die eingesetzten LEDs haben einen integrierten Vorwiderstand, will man Standard LEDs benutzen so ist jeweils ein zusätzlicher Widerstand (1 KOhm oder mehr) in Reihe zu schalten!

3.3. PIC16F688

Der PIC16F688 ist die Steuereinheit der Platine. Er fragt die Signale ab und visualisiert sie über die angeschlossenen LEDs. Für den Ersatz der Roms wird er nicht benötigt. Der PIC wird von mir fertig programmiert zusammen mit der Platine geliefert.

3.4. Adressdekodierung (74LS02)

Um alle System80 Roms ersetzen zu können, muss der Code in die richtigen Adressbereiche einblendend werden. Nachfolgend sind die Adressbereiche eines System80 aufgelistet:

U4 Memory (RIOT) - 0x0000 - 0x007F
U5 Memory (RIOT) - 0x0080 - 0x00FF
U6 Memory (RIOT) - 0x0100 - 0x017F
Not Used - 0x0180 - 0x01FF (Test Fixture)
U4 Registers - 0x0200 - 0x027F
U5 Registers - 0x0280 - 0x02FF
U6 Registers - 0x0300 - 0x037F
Not Used - 0x0380 - 0x0FFF
Game ROM - 0x1000 - 0x17FF
Z5 (5101) - 0x1800 - 0x1FFF
System ROM - 0x2000 - 0x3FFF (U2/U3 ROMS)
Not Used - 0x4000 - 0xFFFF

Die abzudeckenden Adressbereiche für die Proms sind somit 0x1000-0x17FF und 0x2000 – 0x3FFF über die Adressen A11, A12 und A13. Der erste Bereich entspricht den Adressleitungen (~~A11~~ & A12 & ~~A13~~) der zweite der Adressleitung A13. (mit ~~A11~~ fuer ‚Nicht A11‘, also das negierte Signal). Zusammenfassend ergibt sich somit für das Select-Signal=A13 | (~~A11~~ & A12 & ~~A13~~).

3.5.Eprom 27256

Das Eprom enthält die Images der originalen System 80 ROMs (abhängig vom Flipper) sowie das Image des Test-PROMs von Leon Borre. Nachfolgend ist die Adressbelegung aufgeführt. Wahlweise kann auch ein 27512 eingesetzt werden (die zweite Hälfte wird dann nicht benutzt) oder, wenn man nur die ROMs ersetzen möchte und auf die ‚Leon Borre‘ Testfunktion verzichten kann, auch ein 27128. Will man nur die Signale Clock,IRQ und Reset überwachen kann das Eprom auch weggelassen werden.

Kbyte	address	GORAT80 usage	27C256 (32 KByte)	27C128(16KByte)
0	0	(N/A)	not used	not used
2	0x800	(N/A)	not used	not used
4	0x1000	Game ROM - 0x1000 - 0x17FF	Game Rom	Game Rom
8	0x1800	(N/A)	not used	not used
10	0x2000	System ROM - 0x2000 - 0x3FFF	System Rom	System Rom
12	0x2800	(U2/U3 ROMS)	System Rom	System Rom
14	0x3000		System Rom	System Rom
16	0x3800		System Rom	System Rom
18	0x4000	Not used	not used	
20	0x4800	Not used	not used	
22	0x5000	Not used	not used	
24	0x5800	Not used	not used	
26	0x6000	LEON	Test Rom	
28	0x6800	LEON	Test Rom	
30	0x7000	LEON	Test Rom	
32	0x7800	LEON	Test Rom	
	0x8000	Not used		

3.6.Beispiele Erstellung Image Eprom 27256

Für die Erstellung des Eprom-Images beziehe ich mich hier auf die Namenskonventionen in den ‚.zip‘ Paketen welche für Pinmame erhältlich sind. In die ‚Not used‘ Bereiche werden Dateien mit ‚0xFF‘ in der passenden Größe (1K-Free.bin, 2K-Free.bin, 4K-Free.bin, 8K-Free.bin) platziert.

Die ‚xK-Free.bin‘ dateien sowie das Leon Prom sind hier zu finden:

www.flipperkeller.de/GORAT80/roms (Die Pinmame Dateien sind dort NICHT zu finden da sie Copyright unterliegen).

3.6.1.System80 bei zwei Game-Proms mit jeweils 1K

Teilweise werde zwei 1K-Proms fuer das Game Prom verwendet. In den Zip Dateien sind diese nach der Konvention Gamename-1.cpu und - Gamename-2.cpu abgelegt

Am Beispiel der originalen ROMs für einen Gottlieb Star Race (System80 Game #657) ergibt sich damit folgendes Mapping:

GORAT80 usage	27C256 (32 KByte)
(N/A) (N/A)	4K-Free.bin
Game ROM - 0x1000 - 0x17FF	657-1.cpu+657-2.cpu
(N/A)	2K-Free.bin
System ROM - 0x2000 - 0x3FFF (U2/U3 ROMS)	u2_80.bin
	u3_80.bin
Not used Not used Not used Not used	8K-Free.bin
LEON LEON LEON LEON	2764af.bin

copy /b 4K-Free.bin + 657-1.cpu + 657-2.cpu + 2K-Free.bin + u2_80.bin + u3_80.bin + 8K-Free.bin + 2764af.bin gorat80_661.bin

3.6.2. System80 und 80A bei einem Game-Proms mit 2 K

Wird ein 2K-Prom fuer das Game Prom verwendet ergibt sich am Beispiel der originalen ROMs für einen Gottlieb Panthera (System80 Game #652) folgendes Mapping:

GORAT80 usage	27C256 (32 KByte)
(N/A)	4K-Free.bin
(N/A)	
Game ROM - 0x1000 - 0x17FF	652.cpu
(N/A)	2K-Free.bin
System ROM - 0x2000 - 0x3FFF (U2/U3 ROMS)	u2_80.bin
	u3_80.bin
Not used Not used Not used Not used	8K-Free.bin
LEON LEON LEON LEON	2764af.bin

Befehlszeile: „copy /b 4K-Free.bin + 652.cpu + 2K-Free.bin + u2_80.bin + u3_80.bin + 8K-Free.bin + 2764af.bin gorat80_652.bin“

Auch **System80A** Geräte haben ein 2K-Game Prom. Hier sind analog zu den u2_80.bin/u3_80.bin die Dateien u2_80a.bin und u3_80a.bin zu nehmen.

Befehlszeile fuer einen Devils Dare: „copy /b 4K-Free.bin + 670.cpu + 2K-Free.bin + u2_80a.bin + u3_80a.bin + 8K-Free.bin + 2764af.bin gorat80_670.bin“

3.6.3. System80B mit 2K Prom

System 80B Packages haben in der Regel zwei Dateien zwei Prom-Dateien ‚Prom1-cpu‘ (8K-Byte, das System Rom) sowie ‚Prom2-cpu‘ (2K-Byte, das Gameprom). Damit ergibt sich am Beispiel eines Tagteam (System80B Game#698) folgendes Mapping:

GORAT80 usage	27C256 (32 KByte)
(N/A)	4K-Free.bin
(N/A)	
Game ROM - 0x1000 - 0x17FF	Prom-2.cpu
(N/A)	2K-Free.bin
System ROM - 0x2000 - 0x3FFF (U2/U3 ROMS)	Prom1.cpu
Not used	8K-Free.bin
Not used	
Not used	
Not used	
LEON	2764af.bin
LEON	
LEON	
LEON	

Befehlszeile: „ copy /b 4K-Free.bin + Prom-2.cpu + 2K-Free.bin + Prom-1.cpu + 8K-Free.bin + 2764af.bin gorat80_698.bin“

3.6.4. System80B mit 4K Game-Prom (nicht direkt unterstützt mit v02)

Ab Game #715 ‚Excalibur‘ setzte Gottlieb 4K-Eproms (2732) anstelle der 2K-Proms (2716) ein. Dabei wird erstmals Adresse ‚A15‘ verwendet. **Die Prom-Simulation bei diesen Geräten wird mit GORAT80 v02 nicht unterstützt!** (Die nächste Version wird einen Dip-Switch haben). Eine manuelle Umkonfiguration, mit v02 bei Verlust der ‚Leon‘ ROM Funktionalität, kann durch auftrennen von Anschluss Pin 27 des Eproms (A14) auf der Unterseite und Verbindung mit Pin35 A15)des TC1-Adapters erfolgen. (Ungetestet!, Bild folgt).

Mapping:

GORAT80 usage	27C256 (32 KByte)
(N/A)	4K-Free.bin
(N/A)	
Game ROM - Part1	Prom-2.cpu
(N/A)	
System ROM - 0x2000 - 0x3FFF (U2/U3 ROMS)	Prom1.cpu
Not used	Prom-2.cpu
Game ROM - Part 2	
Not used	4K-Free.bin
Not used	
LEON	2764af.bin
LEON	
LEON	
LEON	

4. Testen nach ‚Leon Borre‘

Der Nachfolgende Text ist (meine) Übersetzung von Auszügen aus dem englischen Original, sowie ein paar Hinweise wo nötig bzw. angebracht

Das englische Original kann hier nachgelesen werden.

[http://pinwiki.com/wiki/index.php?title=Leon Borre Gottlieb System 80 CPU Board Repair](http://pinwiki.com/wiki/index.php?title=Leon_Borre_Gottlieb_System_80_CPU_Board_Repair)

Im Text wird immer wieder auf ein Multimeter verwiesen, ich persönlich bevorzuge einen Logik-Tester wie diesen hier:

<https://www.reichelt.de/Komponententester/LOGIK-TESTER/3/index.html?&ACTION=3&LA=2&ARTICLE=10536&GROUPID=4024&artnr=LOGIK+TESTER>

Hier kann man direkt ablesen kann ob das Signal High oder Low ist, bzw. ob das Signal ‚pulsed‘.

4.1. Einleitung

Der Test wurde dem Bally-Test nachempfunden, ohne aber einige der Nachteile dieses Tests zu implementieren. So ist die Test-LED beim Bally test PIA U10 zugeordnet, ist dieser PIA defekt, funktioniert der ganze Test nicht ... die Outputs der PIAs werden nicht getestet und der Testcode ist im Game Prom integriert, welches kaputt sein kann. Der grosse Vorteil bei Bally ist, dass die CPU und die PIAs gesockelt sind, was beiden meisten Gottlieb System80 Systemen nicht der Fall ist, im Grund mehr ein gutes Testprogramm zu haben.

Der Test funktioniert folgendermaßen. Nach dem Start wird als erstes die CPU (6202), dann das RAM(5101) und dann die drei RIOTs(6532) getestet. Für jeden erfolgreichen Test blinkt die angeschlossene LED (im Fall von GORAT80 LED6) ein Mal. Nach den ersten 5 ‚flashes‘ wird der Test fortgesetzt und testet die Ausgänge der RIOTs indem sie zyklisch auf High und Low gesetzt werden. Dieser letzte Teil des Tests ist ‚non-blocking‘, läuft also auch weiter wenn ein oder mehrere Ausgänge defekt sind.

Der geeignete Anwender kann jetzt die Ausgänge prüfen und kontrollieren ob sich Ihr Level zwischen 0 und 5Vold bewegt.

Sollte der Test eine defekte Komponente entdecken, so wird an dieser Stelle gestoppt und diese Komponente kontinuierlich geprüft. Das bedeutet die Testroutine kann innerhalb der ersten 5 ‚flashes‘ stoppen. Läuft sie jedoch durch und die LED fängt an kontinuierlich zu blinken, kann nur noch einer der Ausgänge defekt sein. Der Schwachpunkt der Bally Routine wird dadurch umgangen, dass als Output zur LED die Adressline No. 6 verwendet wird. GORAT80 monitored diese Line und gibt den Status 1:1 an LED6 weiter.

HINWEIS: Sollte der Test nach dem ersten ‚flash‘ stoppen (in diesem Fall ist das RAM5101 höchstwahrscheinlich defekt), man aber trotzdem bis zum Ende des Test durchlaufen möchte, kann der Test auch durch einen simplen Trick fortgeführt werden.

4.2.Start

Wie wird der Test durchgeführt?

Nach dem Einschalten erst einmal langsam bis vier zählen und dann anfangen die ‚flashes‘ von LED6 zu zählen.

Starten wir mit dem ‚worst case‘, KEINE flashes

In diesem Fall erst einmal den Prozessort U1 (6502) inspizieren. Dazu GORAT80 und alle eventuell vorhandenen ROMs entfernen, bzw. die Brücke zum Ausschalten der internen ROMs in Position belassen.

Wie immer inspizieren wir zuerst den Prozessor und den ‚selection circuit‘. Dadurch dass kein Programm vorhanden ist (keine PROMs bzw. PROMs ausgeschaltet) zwingen wir den Prozessor sehr schnell alle seine Adressen anzusprechen. Schauen wir uns die essentiellen Signale an, die der Prozessor braucht.

Da wäre erst einmal die Spannungsversorgung 5V, zu messen zwischen PIN1 (Ground) und PIN8 (5Volt). Die nächsten Spannungen sind: : Pin 40 = 4 Volt; Pin 4 = 5Volt ; Pin 2 = 5 Volt ; Pin 37 = 2 Volt : Pin 34 = 2Volt. So etwas fehlt, nimmt man den Schaltplan zur Hand und verfolgt das fehlende Signal zurück zu seinem Ursprung. Für die Signale an Pin 2 und Pin 4 ist nur der entsprechend ‚Pull-UP‘ Widerstand zu prüfen. RES an Pin 40 und Clock an Pin 37 sind schon durch GORAT80 geprüft. Pin 34 ist das R/W Signal, fehlt dieses ist entweder die CPU defekt oder das Signal hat einen Kurzschluss. Zum Test kann man (so die CPU gesockelt ist) das Signal einmal freilegen und noch einmal messen. Ist das Signale dann immer noch nicht da, ist die CPU defekt, andernfalls hat das Signal einen Kurzschluss.

Sind diese Signale vorhanden ist es Zeit nach den Signalen am Address- und Data-Bus zu schauen. Mit einem Standard Multimeter gemessen sollten alle zwischen 0,5 und 3 Volt liegen. Ist das Signal nicht vorhanden, wie üblich durch freilegen checken ob ein Kurzschluss oder eine defekte CPU vorliegt.

Programm Selektion

GORAT80 nutzt BAB11 vom TC1-Adapter zur Adressselektion. Im Gegensatz zu allen anderen Adresssignalen kommt dieses Signal nicht direkt von der CPU sondern läuft über das erste (Pin 1&2) und das zweite (Pin 3 & 4) Gatter von Z10. Damit GORAT80 funktionieren kann, müssen dieses IC und die angeschlossenen ‚Pull-up‘ Widerstände in Ordnung sein.

4.3.Nur ein Flash

Das bedeutet dass die CPU in Ordnung ist, höchstwahrscheinlich ist der RAM Chip (5101) defekt, allerdings koennte auch ein defekter RIOTs dies nur vortäuschen. Um die Tests trotzdem weiter laufen zu lassen kann man Pin 6 von U1 (CPU 6502) kurz auf Ground ziehen. Dadurch wird ein Interrupt ausgelöst, welcher das Testprogramm direkt zum RIOT Test springen lässt. Dies wird, so der RIOT OK ist, direkt durch einen

Flash angezeigt wonach das Testprogramm normal mit dem naechsten RIOT fortgesetzt wird.

In den meisten Fällen wird der RAM Chip selbst defekt sein, um 100% sicher zu sein sollten die ‚Selection Signals‘ für diesen Chip überprüft werden. Dazu wieder neu starten, wonach der Test wieder nach dem ersten Flash stoppen sollte, bzw. weiterhin das RAM testet und damit selektiert. Die anliegenden Signale sollten sein: pin 20 = 3.5volt, pin 19 = 3.5volt, pin 17 = 0.2volt, pin 18 = 0.2volt. Danach sind die ‚data signals‘ Pin 9 bis 16 zu prüfen (0.2 Volt). Sind alle Signale vorhanden ist der RAM Baustein defekt und auszutauschen.

4.4.Zwei Flashes = U4 , drei Flashes = U5, und vier Flashes = U6

‚Selection Siganalns‘ für die Riot's U4, U5 und U6

In einem von diesen Fällen schaut man auf die ‚selection signals‘ des betreffenden RIOTs . Das sind SEL2 pin 37 = 3,5 volt (Z7 und Z9) oder manchmal auch Z8 bei einem fehlenden Pin 38 Signal, hier sind wieder die Schaltpläne zu konsultieren. Um ganz sicher zu gehen pin 34 (RES) 5 Volt und Pin 39; 4 Volt überprüfen. Wenn alle diese Signale vorhanden sind, ist der betreffende RIOT auszutauschen.

Dauerhaftes Blinken

Dies bedeutet, dass lles in Ordnung ist, undman jetzt die Ausgänge aller RIOTs zu kontrollieren hat.

Mit einem Multimeter kontrolliert man dies, sie müssen alle im mittleren Bereich zwischen 0 und 5 Volt liegen (pulsieren). Die betreffenden Pins sind: U4, U5 en U6 Pins 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23 und 24 . Einige Signale werden nicht pulsieren, das sind diejenigen die als Input konfiguriert sind. Trotzdem kann man diese kontrollieren. Pin 15 von U5 wird seinen Wert ändern wenn Pin 10 von U5 auf Ground gelegt wird. Um die Pins 8 bis 15 von U4 zu kontrollieren mus man zuerst die dip-switches 1 bis 8 auf ‚On‘ stellen. Wird jetzt an Z15 eine Brücke zwischen Pin3 und 7 gelegt fangen diese Pins an zu pulsieren (Werte zwischen 2 und 4 Volt mit dem Multimeter)

Nun sind alle Ausgänge kontrolliert. Jeder Ausgang an dem kein Signal festzustellen ist, kann er testweise mit einem seiner ‚Nachbarn‘ verbunden werden. Zwei Möglichkeiten: Wenn nun beide pulsieren ist der RIOT defekt, pulsiert keiner von beiden mehr, dann liegt ein Kurzschluss an dem ersten Pin vor. Auch hier, um 100% sicher zu sein, ist der verdächtige Pin frei zu legen. Pulsiert er dann wieder, liegt ein Kurzschluss vor und die Leiterbahn ist anhand des Schaltplans zu überprüfen. Im anderen Fall, ist der RIOT defekt und muss ausgetauscht werden.

4.5.Über das Testprogramm

Das Programm ist so organisiert, dass es in den Sektionen des Adressbereichs läuft in denen Adresse 6 nicht involviert ist (Low); mit Ausnahme der Stellen wo die LED ‚an‘ ist, in diesen Bereichen ist Adresse 6 involviert (High) und die LED damit an.

HINWEIS: Dies konnte nur erreicht werden indem Leon keine Hochsprache, noch nicht einmal Assembler, für sein Programm verwendet hat, sondern die Instruktionen direkt mit ihrem hexadezimalen Werten an den Stellen platziert hat, wo er sie brauchte, verbunden durch ‚Jump‘ Anweisungen. Die ist auch der Grund warum kein Source Programm vorhanden ist, einfach weil es nicht existiert!