1541-rebuild

von Thorsten Kattanek
Berlin, 01.05.2020

Inhaltsverzeichnis

1.0 Was ist 1541-rebuild

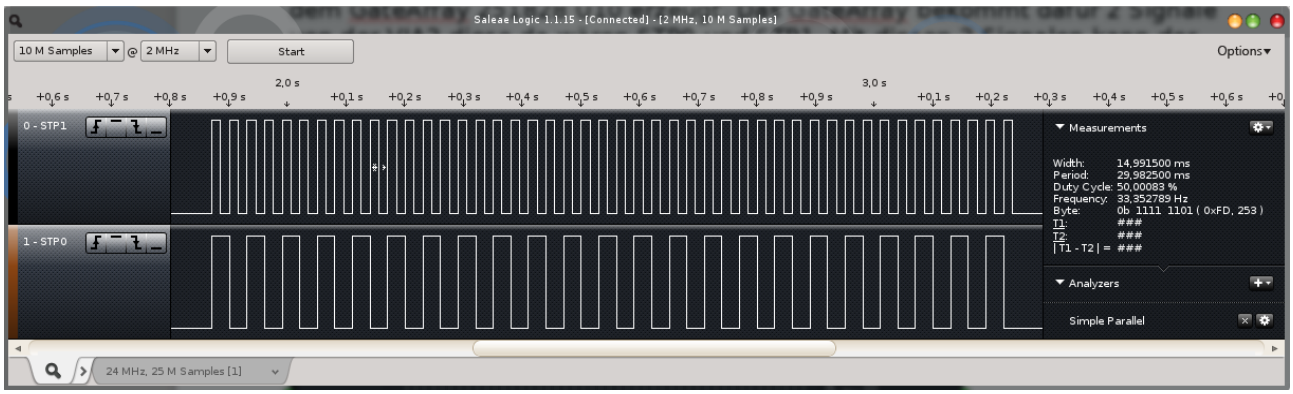
1541-rebuild ersetzt den mechanisch-analogen Teil der Commodore Floppy 1541 und 1541 II durch eine Elektronik. Die Floppy Disk wird in einem Mikrocontroller emuliert. Die Disk Images werden über eine SD Karte geladen. Die Bedienung erfolgt über 3 Taster und ein LCD Display. Als Mikrocontroller kommt ein Atmega1284P zum Einsatz, der mit 24 MHz getaktet wird.

2.0 Beschreibung der Internen Prozesse

2.1 Emulation des Schrittmotors

Der Schrittmotor ist der Antrieb für den Schreib- Lesekopf der Floppy. Dieser wird auf die zu zugreifende Spur gefahren. Der Schrittmotor hat 4 Anschlüsse welche an die 4 Spulen des Schrittmotors gehen. Durch geeignete Ansteuerung der Spulen kann der Schrittmotor um ein ganz kleinen Winkel, nach links oder rechts gedreht werden. Die vier Steuersignale für den Schrittmotor werden von dem GateArray 251828 U10 erzeugt. Das GateArray bekommt dafür 2 Signale von der VIA2 diese da wären STP0 und STP1. Mit diesen 2 Signalen kann der Lesekopf um genau eine Halbspur vor oder zurück bewegt werden. Diese beiden Signale muss in unserem Fall der µC auswerten um intern die Virtuelle Spur darstellen zu können.

Hier mal die Signale mit einem Logic Analyzer aufgenommen, wenn die Floppy Disk formatiert wird. Der Lesekopf wird dabei an den Anschlag (BUMP) nach außen gefahren.



Die beiden STP Signale sind „verdreht“ am VIA2 angeschlossen. STP1 ist am PB0 und STP0 ist am PB1 angeschlossen. Wenn wir die Signale mal in einer Tabelle aufnehmen sieht man die Wirkungsweise der Signale.

Step	PB1 (STP0)	PB0 (STP1)	Dezimal
0	1	1	3
1	1	0	2
2	0	1	1
3	0	0	0
4	1	1	3
5	1	0	2

Man sieht hier deutlich das der VIA2 nur die beiden PINS „runter zählen“ muss um den Lesekopf nach außen zu bewegen und zwar immer um eine Halbspur. Zählt die VIA2 hoch fährt der Lesekopf entsprechend nach innen. Steht der Lesekopf nun auf eine „voll“ Spur (Bit 0 der aktuellen Halbspur gelöscht) so müssten sofort die neuen Daten von der aktuellen Spur in das RAM geladen werden. Um das aber zu verhindern und dabei Zeit zu sparen, werden erst neue Daten geladen wenn es 20ms keine Signaländerungen an STP0 und STP1 gegeben hat. Je nach Postion der Spur muss die Schreib-/ Lesegeschwindigkeit der GCR Daten neu gesetzt werden. Dazu mehr weiter unten im Kapitel 2.3.

Das ist eigentlich auch schon alles was man wissen muss um die Signale auszuwerten zu können und den internen Spurzeiger zu setzen.

2.2 Emulation des Laufwerkmotors

Der Laufwerksmotor wird gesteuert durch die VIA2 PB2. Ist es High läuft der Motor ist er Low ist er aus. Dieses Signal wird benutzt um das senden der GCR Bytes zu steuern. Ist das Signal High werden aus dem Ringpuffer fleißig die GCR Bytes der aktuellen Spur gesendet. Ist das Signal Low werden nur noch 0 Bytes gesendet. (Keine Flusswechsel da keine Drehung !)

2.3 Schreib-/ Lesekopf (GCR Daten)

Gleich erstmal vorweg. Im momentanen Entwicklungszustand unterstützt die

1541-rebuild nur das lesen von einem Disetten Image (d64, g64). Das schreiben und insbesondere das zurückschreiben in einem Disketten Image auf die SD Karte ist noch nicht vollständig implementiert. Es wurde schon auf Machbarkeit geprüft und G64 Dateien konnten beschrieben werden, jedoch ist das ganze in der aktuellen Firmware (1.2.3) sehr unzuverlässig.

Kommen wir nun zum lesen der Daten von einem GCR

3.0 Spannungsversorgung

Die Spannungsversorgung erfolgt über ein Steckernetzteil mit 5VDC und 1A. Leider ist das Originale Floppy Netzteil etwas schwachbrüstig. Die Spannung brach mit der Zusatzschaltung auf 4.7V ein. Wenn ich den Datenport komplett auf High gesetzt habe, brach sie sogar auf 4.2V ein. Somit konnte der 251828 intern die Transistoren nicht mehr komplett durchschalten und der Effekt war eine glühende GateArray.

Momentan läuft bei mir die Schaltung auch mit dem Originalen Floppy Netzteil!

4.0 Hardware – Aufbauanleitung (Rev. 1.4.0)

5.0 Flashen der Firmware (unter Linux)

6.0 1541-rebuild Bedienungsanleitung

7.0 Quellen

- Die Floppy 1541 von Karsten Schramm