

USB-DMX Debugging

Digital Enlightenment

<http://www.digital-enlightenment.de>

Das Zusammenspiel der einzelnen Komponenten des Interface ist zum Teil recht komplex, weshalb man nur schwer eine eindeutige Lösung zu einem bestehenden Problem geben kann.

Ich werde daher im Folgenden versuchen den Start-Up des Interface Schritt für Schritt zu beschreiben und für jeden Zeitpunkt mögliche Fehler und deren Lösungen aufzeigen. Je nach Fehlerbild lassen sich dann mögliche Ursachen ermitteln. Der dargestellte Ablauf gilt für den Betrieb an einem USB Port. Die Initialisierung für den Stand Alone Repeater Mode läuft etwas anders ab, sollte aber keine Probleme machen, wenn das Interface am USB Port funktioniert.

1. Check im ausgeschalteten Zustand

Wenn es Probleme gibt, ist eine Sichtprüfung der Platine der erste Schritt der Fehlersuche:

- Sind die Pins aller ICs angelötet?
- Sind Kurzschlüsse von Leiterbahnen durch zu viel Lötzinn zu sehen?
- Sind kalte Lötstellen zu entdecken?
- Wurden (bei einer selbstgemachten Platine) Durchkontaktierungen oder Brücken vergessen?
- Sind (bei einer selbstgemachten Platine) Kurzschlüsse oder Unterbrechungen von Leiterbahnen zu sehen?

Als nächster Schritt sollten alle Verbindungen mit einem Durchgangsprüfer getestet werden. Der Durchgangsprüfer sollte eine Testspannung $< 0.5V$ verwenden (machen Multimeter in der Regel, wenn sie nicht im Diodentestmodus sind), damit die Eingangsschutzdioden der ICs die Messung nicht verfälschen. Außerdem sollte man darauf achten direkt an den Pins der ICs selbst zu messen und nicht etwa am Lötpad, denn nur so fallen z.B. auch kalte Lötstellen auf. Weiterhin ist es sinnvoll benachbarte Pins an ICs auf Kurzschlüsse zu untersuchen.

Welche Pins verbunden sein sollten und welche nicht kann man gut auf dem Bild "Layout.jpg" erkennen.

2. Power-Up

Ist der erste Schritt erfolgreich abgeschlossen, kann das Interface an einen USB Port angeschlossen werden.

Nach dem Einstecken liegt aufgrund der Schottky Diode D1 an Vcc der ICs 1 bis 4 nicht die volle USB Versorgungsspannung an, sondern ca. 0.4V weniger. Hier kann es ein Problem geben, wenn der USB Port schon etwas weniger als 5V liefert. Das testweise Anschließen der externen Versorgung stellt sicher, dass dem Interface genügend Spannung zur Verfügung steht. Für den Masseanschluss bei Messungen eignet sich übrigens die Kühlfahne des 7805 Spannungsreglers gut.

Ist an den ICs keine Spannung oder wesentlich weniger als 4.6V messbar, gibt es irgendwo eine Unterbrechung oder einen Kurzschluss. Hier empfiehlt es sich, den Weg der Versorgungsspannung nach und nach von der Sicherung (F1) am USB Port bzw. von der externen Versorgung zu den ICs zu verfolgen. Sollten ihr feststellen, dass die Sicherung defekt ist, diese bitte nicht einfach überbrücken, denn sie hat sicherlich einen Grund für dieses Verhalten. Ein kurzgeschlossener USB Port kann im schlimmsten Fall zu einem Defekt der Southbridge in eurem Rechner führen, was neben USB durchaus auch andere Komponenten wie z.B. Ethernet oder IDE schädigen kann.

Bevor ihr eine kaputte Sicherung austauscht, solltet ihr natürlich den Grund für ihr Auslösen (vermutlich ein Kurzschluss) suchen.

3. Clocksystem Start-Up

Liegt eine stabile Spannung an der Schaltung, so lädt sich C5 langsam über R16 auf und AVR sowie USBN verlassen den Hardware-Reset Zustand nach einigen Millisekunden. Der USBN startet nun den Quarz Oszillator, welcher mit 24 MHz arbeitet. Er benötigt dafür einen sog. Grundton Quarz. 24 MHz Quarze, bei denen dies nicht explizit dabei steht, sind in der Regel sog. Obertonquarze. Der Unterschied ist (wie die beiden Bezeichnungen auch schon vermuten lassen), dass der Grundtonquarz tatsächlich mit seiner Grundfrequenz auf 24 MHz schwingt, während es sich bei Obertonquarzen eigentlich um 8 MHz Quarze handelt, welche durch eine geeignete Oszillatorschaltung auf ihrer dritten Oberwelle (= 24 MHz) betrieben werden. Dazu ist der Oszillator im USBN aber nicht geeignet, weshalb ein Grundtonquarz benötigt wird.

Schwingt der Oszillator korrekt mit 24 MHz (bei Messungen einen 1:10 Tastkopf benutzen und an XOUT messen, damit die Schwingung nicht abreißt), so werden diese intern auf 48 MHz verdoppelt. Diese Frequenz wird für die 12 MBit USB Kommunikation benötigt und findet sich bei vielen Full Speed USB Geräten.

Außerdem werden mit Hilfe eines Taktteilers defaultmäßig 4 MHz am CLKOUT Pin (28) des USBN ausgegeben, welche dem AVR als Taktquelle dienen.

Ist der AVR korrekt geflasht, beginnt er mit der Ausführung der Reset-Routine. Ist das nicht der Fall (z.B. meldet Pony Prog wohl keinen Fehler wenn das erzeugte Hex File ein falsches Format hat), bleibt der Start-Up des Interface an dieser Stelle hängen.

Während der Reset Routine des AVR wird der DMX Transceiver für eine kurze Zeit (250ns) eingeschaltet. Vereinzelt USBN Bausteine reagieren scheinbar empfindlich auf den Impuls auf der Versorgungsspannung, welchen der Transceiver aufgrund dieses kurzen Einschaltimpulses generiert. Abhilfe schafft alles, was die Spannung am USBN glättet, also z.B. ein 100nF am USBN zwischen den Pins 22 und 24 (bzw. an der Stelle eben zwischen GND und Vcc), oder auch z.B. eine Spule anstelle der Brücke für Vcc unter IC3. Zum kurzen Test ob hier das Problem liegt, kann auch

einfach der Transistor überbrückt werden (E und C verbinden), so dass der Transceiver permanent eingeschaltet ist und somit keine Störspitze erzeugt. Stürzt der USBN aufgrund eines solchen Impulses ab, dann wartet der AVR vergeblich auf das Ende des Soft-Reset Kommandos des USBN, welches zuvor während der Reset Routine gesendet wurde.

Das gleiche Fehlerbild ergibt sich, wenn die Buskommunikation zwischen AVR und USBN aufgrund von Kurzschlüssen oder Unterbrechungen gestört ist, da auch in diesem Fall der AVR niemals das Ende des Soft-Resets auslesen kann. Je nach Position des Fehlers zwischen AVR und USBN (d.h. je nachdem welche Datenleitung betroffen ist) wäre es theoretisch möglich, dass das Ende des Soft-Resets trotzdem korrekt ausgelesen werden kann. Der Fehler wird sich aber spätestens ein paar Übertragungen später bemerkbar machen.

Alle diese Fehler (falsch geflashter AVR, USBN Absturz und Kommunikationsstörung) lassen sich vor allem daran erkennen, dass das Clocksignal am CLKOUT vom USBN und damit XTAL1 des AVR eine Frequenz von 4 MHz hat. Bei einer fehlerhaften Verbindung zwischen AVR und USBN wäre es auch möglich, dass eine andere Frequenz als 4 oder 12 MHz am CLKOUT anliegt, weil vereinzelte Bits des Configurationswertes falsch übertragen werden. Sofern der Quarz am USBN auch tatsächlich mit 24 MHz schwingt, würde das dann stark auf einen Kommunikationsfehler hindeuten.

Ist bis zu dieser Stelle alles glatt gelaufen, dann schreibt der AVR einen neuen Wert in das CLKOUT Config Register des USBN, so dass dort ab sofort ein 12 MHz Signal ausgegeben wird und der AVR mit einer dreimal höheren Geschwindigkeit arbeitet. Das ist notwendig, um genügend Rechenleistung für Mode 7 des Interfaces bereit stellen zu können.

Ein erfolgreicher Start-Up des Clocksystems ist also daran zu erkennen, dass an Pin 19 (XTAL1) des AVR 12 MHz zu messen sind.

4. Anmeldung am PC

Zu diesem Zeitpunkt sind die meisten Hürden bereits genommen. Der AVR initialisiert jetzt den USBN und seine internen Peripheriekomponenten. Dabei wird auch der 3.3V Regler des USBN aktiviert, was man an Pin 18 überprüfen kann. Anschließend werden die beiden USB-Datenpins (D+/D-) freigegeben (sie wurden bis jetzt aktiv auf GND Pegel gehalten), so dass der PC ein neues USB Gerät erkennt. Dieser sendet nun eine Reihe sog. USB Requests an das Interface, welche mit Hilfe des USBN vom AVR empfangen und beantwortet werden. Diesen Vorgang nennt man Enumeration. Das Betriebssystem des PC fragt dabei alle möglichen Daten wie z.B. die Seriennummer ab und erkennt daraus, dass es sich um ein HID (Human Interface Device) handelt. Daraufhin wird vom Betriebssystem automatisch der HID Treiber für dieses Gerät geladen. Probleme gibt es an dieser Stelle eher selten. Möglich ist eine fehlerhafte Verbindung der USB Datenleitungen, so dass der 1.5K Pullup Widerstand an D+ (welcher dem USB Hub ein Full Speed Device signalisiert) zwar erkannt wird, eine Datenkommunikation mit 12 MBit aber nicht möglich ist. Das Betriebssystem erkennt dann zwar ein Gerät am USB Port, kann aber nicht mit ihm kommunizieren. Windows z.B. meldet dann, dass es ein neues Gerät gefunden hat, dieses aber nicht richtig funktioniert.

Sobald die Enumeration erfolgreich abgeschlossen ist, ist das Interface einsatzbereit und taucht bei Windows in der Systemsteuerung unter "Human Interface Devices" auf.

5. Betrieb

Mit Hilfe der usbdmx.dll kann nun eine beliebige Anwendung mit dem Windows HID Treiber und so mit dem Interface kommunizieren. Einige Fehler fallen jedoch erst während des Betriebs des Interface auf:

Wenn der Mode des Interface von Standby auf einen Mode ungleich Standby gesetzt wird, aktiviert das Interface den Transceiver. Das führt wieder zum selben Spike auf der Versorgungsspannung wie bereits unter Punkt 3 beschrieben. Hier soll nur kurz erwähnt werden, dass es auch beim Setzen eines aktiven Modes zu diesem Problem kommen kann. Abhilfen sind natürlich die selben wie unter 3 beschrieben.

Ein weiteres mögliches Problem ist eine fehlerhafte Programmierung der Clock Fuses des AVR, so dass dieser mit der internen Clock von 1 MHz arbeitet anstatt die ext. 12 MHz zu verwenden. Die USB Enumeration am PC funktioniert auch mit diesen 1 MHz, da der USBN die benötigten 48 MHz zu Verfügung hat und die Busverbindung zwischen AVR und USBN asynchron arbeitet. Allerdings kann so natürlich kein 250 KBit DMX Signal erzeugt oder empfangen werden. Dazu sind 12 MHz notwendig. Diesen Fehler erkennt man daran, dass ein Loop-Back Test (d.h. direkte Verbindung des DMX Ausgangs mit dem DMX Eingang und dann überprüfen, ob gesendete DMX Daten korrekt wieder empfangen werden) zwar funktioniert (da sowohl DMX Sender als auch Empfänger mit der selben, jedoch falschen Bitrate arbeiten), ein externes DMX Gerät aber nichts empfängt.

Ein weiteres Problem tritt auf, wenn die Verbindung zwischen AVR und seinem externen RAM gestört ist. Z.B. durch Kurzschlüsse oder Leitungsunterbrechungen zwischen den ICs 1, 2 und 3. Das Interface zeigt dann korrektes Verhalten am PC und auch das Format der generierten DMX Daten (Bitrate usw.) ist in Ordnung. Da der RAM aber als Zwischenspeicher für zu sendende und empfangene DMX Daten dient, sind diese korrupt. Je nach Ort des Verbindungsfehlers (Datenleitung, Adressleitung oder Controlleitung) können sich viele verschiedene Fehlerbilder ergeben. Es ist z.B. denkbar (wenn auch sehr unwahrscheinlich), dass der DMX Empfang einwandfrei funktioniert, das Senden aber nicht. Auch wenn natürlich ein Defekt an den Bauelementen nicht ausgeschlossen werden kann, hatten bisher sämtliche Probleme dieser Art ihre Ursache in einer fehlerhaften Verbindung.