

Erledigt

"Du hast ja Alles" - hmmm vielleicht, wenn ich einen Laptop habe.

Beitrag von „Brumbaer“ vom 7. Mai 2018, 00:06

Windows-Phone

Irgendwie schon, aber Windows-Modem trifft es besser.

Verbindung mit dem Netz hergestellt, den Schnüffler an der Nase gezogen, damit er anfängt seinem Namen Ehre zu machen, und schon "geht dat los".

Wer zu spät kommt, den bestraft das Leben. In diesem Falle ist es zu spät für Screenshots, denn die Lizenz für die Testversion des Sniffers ist abgelaufen. Deshalb gleich weiter zu den Ergebnissen.

- Eintracht - Hamburg 3:0
- Hannover - Hertha 3:1
- USB - Modem Encapsulated commands und responses.

Das Ergebnis ist überraschend, man hätte meinen sollen, dass Hertha gewinnt und dass das mit dem *Encapsulated commands and responses* nicht funktioniert.

Es gibt aber einen Grund warum die *EC&R* hier funktionieren.

MBIM

steht nicht für eine mobile Glocke, sondern für *Mobile Broadband Interface Model*. Der eine oder der andere (mehr als zwei werden es kaum sein) wird sich vielleicht erinnern, dass es unterschiedliche USB-Interface-Deskriptoren in Abhängigkeit vom *Alternate Setting* gab. Eine Variante trug *NCM* im Namen, die andere *MBIM*.

MBIM ist eine Spezifikation zur Kommunikation mit USB-Breitband-Modems. Sie unterscheidet sich in für uns zwei wichtigen Dingen von NCM.

- Das Modem muss über Modem Encapsulated commands und responses gesteuert werden können.
- Die Daten entsprechen nicht Ethernet-Paketen, sondern Internet-Paketen. Es gibt Internet-Pakete mit 4 Byte langen IPs (IPv4) und welche mit 16 Byte langen IPs (IPv6), das soll uns an der Stelle aber nicht belasten.

À la carte

Was hätten wir den gerne ?

Da schauen wir erst mal in die Karte, was mich dazu bringt eine Prepaid Karte für den Miix zu kaufen, denn die Bastelei wird wohl noch etwas dauern und mein iPad braucht seine Karte.

Prepaid ist ok und schon habe ich so eine neue, schlanke und ranke Karte. Wenn das Miix in Betrieb geht, kann ich sie durch die andere Karte ersetzen.

Ziel ist es das L831-EAU auf MBIM umzuschalten und dann erst die Kommunikation zu starten.

In Memoriam



Das ist die "Treiber" Struktur. Als erstes knöpfen wir uns *AppleUSBNCMControl* vor. Einen Ersatz-Treiber hatte ich ja schon geschrieben, nur hat er nicht funktioniert.

Eigentlich sollte er funktionieren, also nur die Umschaltung auf MBIM hinzugefügt. Neue Sim Karte rein, Neustart und das Modem antwortet.

Wieder einmal eine Spec runtergeladen und nachgeschaut, welche Befehle das Modem laut Spec unterstützen muss. Hatte ich erwähnt, dass Vorschau keine gute Wahl zum Anschauen von großen PDFs ist. Falls nicht: **Vorschau ist eine miese Wahl zum Anschauen großer PDFs.**

Die Liste mit den unter Windows übertragenen Befehlen abgeglichen und eine Initialisierungssequenz abgeschickt.

Sieht ganz gut aus, bis mit der Sim Karte kommuniziert werden soll, dann überträgt das Modem nur eine Fehlermeldung.

Die Fahrscheine bitte

Das Modem findet keine Sim Karte. Ich finde sie hingegen sofort, sie steckt im Kartenslot. Also Windows gebootet und auch dort wird die Karte nicht gefunden. Ich wollte schon auf Windows schimpfen, bis mir einfiel, dass es mit meinem Treiber auch nicht geht. Karte rein, Karte raus, kein Unterschied. Lustig, die Karte rastet ein und um sie auszuwerfen drückt man sie erst tiefer rein. Das war bei der anderen Karte nicht so. Da sieht man mal wieder, dass diese halbe Hemden nicht die Fusstapfen von uns King Size Katzen füllen können. Die andere Karte hat wohl die Kontakt Pins ein wenig zurückgebogen und nun kommen die Pins nicht mehr an die Kontakte der schlanken Karte. Ein Fetzen Papier behebt das Problem, bis ich dazu komme, den Miix zu öffnen und die Pins zurück zu biegen.

Contact

Lange darauf hingearbeitet und trotzdem kommt der Erstkontakt mit dem Handy-Netz

irgendwie überraschend.

Nun besteht die Hoffnung die bestehende Verbindung in den NCM Modus rüberzuretten, damit ich nicht auch noch den Daten-Treiber neu schreiben muss.

Geht nicht. Hätte ich die Spec aufmerksamer gelesen hätte ich es erst gar nicht probiert, das steht nämlich drin, dass es nicht geht. Wenn man zwischen den Modi umschaltet, wird alles zurückgesetzt und Control Interface auf MBIM und Daten Interface auf NCM geht schon gar nicht.

IOEthernetController

Macos ist gutgläubig. Wenn es aussieht wie ein Ethernet-Controller und redet wie ein Ethernet-Controller, dann denkt macos es ist ein Ethernet-Controller und lädt die zugehörigen Treiber für die Einbindung in das System.

Die Klasse *IOEthernetController* ist leidlich dokumentiert und es gibt ein paar Beispieldateien. Das lässt einen den Treiber schneller schreiben, fördert aber nicht unbedingt das Verständnis.

IOEthernetController dient letztendlich dazu Ethernet-Daten-Pakete zu empfangen und zu verschicken. Wie vorher erwähnt, hat bzw. will MBIM, aber Internet-Daten-Pakete. Es mag sein, dass es möglich ist *IOEthernetController* zu umgehen, aber ich habe leider keine Möglichkeit gefunden.

MBIM und *IOEthernetController* sind eher eine von den Ehen bei denen man von vorne herein vermutet, dass es sich um eine Zweckehe handelt.

Was will uns der Künstler damit sagen ?

Laut OSI-Schichten-Modell ist die Internet-Datenpaket-Übertragung Teil des Layers 3, des *Network Layers*. Die Ethernet-Datenpaket-Übertragung Teil des Layers 2, des *Data Link Layers*. Das macht es nicht klarer, oder ?

Im Layer 3 wird definiert, was von wo nach wo soll. In unserem Falle sollen von der IP 123.456.789.6 zur IP 123.456.789.77 übertragen werden. Wie die Daten, Adressen und Hilfsinformationen formatiert werden sollen, gibt die jeweilige Spezifikation vor. In unserem Fall als IPv4- oder IPv6-Paket.

Der Layer 2 präzisiert, wie das geschehen soll. In unserem Fall laut Ethernet-Protokoll. Dummerweise überträgt Ethernet aber keine Daten zwischen Geräten mit IPs. Das Ethernetprotokoll verwendet die *Ethernet Hardware Address (EHA)* statt der IP um die Geräte zu identifizieren. Ein anderer Name für *Ethernet Hardware Address* ist *MAC Adresse*. Das *MAC* kommt von einem Sublayer des *Data Link Layers* indem die Adresse Verwendung findet.

Soll ein Internet-Paket über Ethernet übertragen werden, muss man zuerst die *EHA* finden, die zur jeweiligen *IP* gehört und dann das Internet-Paket als Payload (Daten) in ein Ethernet-Paket packen. Dazu wird dem Internet-Paket ein Header vorangestellt und eine Prüfsumme hintenangestellt.

Früher war Alles besser

Früher bei NCM war das alles super. Denn NCM arbeitet mit Ethernet-Paketen. Bekommt unser Pseudo Ethernet-Controller ein Ethernet-Paket, so muss er es nur an das Modem weiterschicken.

Aber MBIM hat mit Ethernet und dessen Paketen nichts am Hut. Versucht ein Treiber die EHA zu einer IP zu finden, so antwortet normalerweise der Router. Da der Router aber nichts von unserem USB Gerät weiß, hält er sich raus. Also habe ich dem Treiber eine Funktion spendiert, die Anfragen nach der EHA zur IP des Modems beantwortet. Der Interessierte schickt eine Anfrage über das Ethernet. Diese erfolgt laut *ARP (Address Resolution Protocol)* Protokoll. Und der Treiber schickt eine Antwort laut des selben Protokolls. Jetzt kann jedes Ethernet Gerät das Modem anhand dessen IP finden, schon mal sehr hilfreich.

Zahlenspiele

Die IP des Modems wird vom Netzwerkprovider zugewiesen, aber im Netzwerk-Kontrollfeld erscheint die IP nicht. Es gibt laut ARP Protokoll eine Möglichkeit, seine IP in die Welt herauszuschreien, damit alle Geräte die IP zu einer EHA oder umgekehrt kennen. Leider lässt sich das Netzwerk-Kontrollfeld von dem Geschrei nicht einschüchtern und ignoriert diese ARP-Pakete.

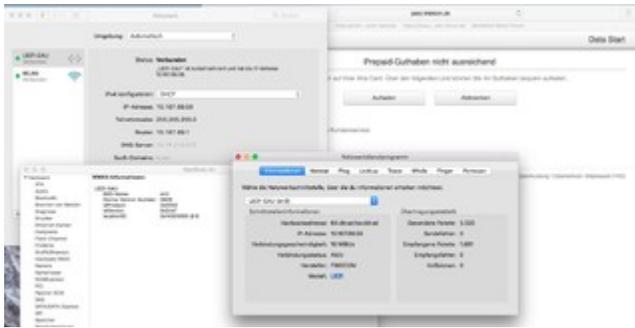
Das Netzwerk-Kontrollfeld sieht allerdings die Option vor die IP Adresse über DHCP zu beziehen. Also hat der Treiber eine weitere Funktion bekommen. Er mimt einen DHCP Server und gibt die IP Adresse, Gateway Adresse und DNS Server Adresse des Modems an das Netzwerk-Kontrollfeld weiter. Meine Güte bei den vielen Rollen, die der Treiber übernimmt, bekommt er noch eine Identitätskrise.

Jetzt wo ich das schreibe frage ich mich, ob es nicht einfacher gewesen wäre, im Netzwerk-Kontrollfeld eine beliebige IP zu setzen und den Treiber eine Adressumwandlung vornehmen zu lassen. Die guten Ideen hat man immer zu spät.

Geht's ?

Na ja, kommt ein Paket von macos muss man das Internet-Paket aus den Ethernet-Paketen herauslösen und an das Modem schicken. Im umgekehrten Fall muss man ein Ethernet-Paket aus den Internet-Paketen machen und an macos weiterleiten.

Es lässt viel Spielraum für Fehler, aber mit Geduld, Spucke, den Specs und Wireshark, geht's dann doch.



Und jetzt ?

Gute Frage. Das Modem funktioniert.

Es kommt als Ethernetkarte daher, weshalb es immer vor dem WLAN bevorzugt wird. Eine automatische Umschaltung erfolgt nicht. Will man es nicht muss man es ausschalten. Da gilt es herauszufinden, wie man es als WWAN kennzeichnet so dass es auch das Netzwerk-Kontrollfeld als solches erkennt.

Aber das ist eher nervig als spannend.

Punkt ist, dass Modem und Touch funktionieren, aber noch Raum für Verbesserungen lassen. Aber wie sooft, ist der interessante Teil, das Zum-Laufen-Bringen, das Falten-Rausbügeln ist dann mühselig.

Ich könnte mir natürlich auch erst anschauen ob man die Kameras zum Laufen bringt.

Ich denke ich werde eine Pause einlegen und dann Touch und Modem überarbeiten, und dann erst die Kameras anschauen, falls mir bis dahin nichts besseres einfällt oder mir ein interessanterer 2 in 1 unterkommt.

Denn irgendwie ist der Funke zum Miix nicht wirklich übergesprungen. Vielleicht kennen wir uns einfach zu gut.