

Entwicklung eines USB-Video-Grabber-Treibers für RIOT

Philipp Rosenkranz, Maximilian Ferdinand Müller

I. EINLEITUNG

DIESES Softwareprojekt hatte zum Ziel, einen Gerätetreiber für einen USB-Video-Grabber für RIOT zu entwickeln.

II. VERWENDETE TECHNOLOGIEN

A. RIOT

RIOT ist ein neues Betriebssystem für das Internet of Things (IoT). Es ist ein teilweise POSIX-konformes System, das auf mehreren Hardwareplattformen lauffähig ist. Die Programmierung gestaltet sich sehr bequem, da einfach in C oder C++ programmiert werden kann, unter Verwendung von Standardwerkzeugen wie gcc und gdb. Es zeichnet sich durch eine minimale Hardwareabhängigkeit aus ist dementsprechend einfach zu portieren.

RIOT ist außerordentlich ressourcenschonend und ermöglicht dabei eine maximale Energieeffizienz, was vor allen Dingen für eingebettete Systeme entscheidend ist. Zusätzlich ist das System Multithreading-fähig und unterstützt viele für das IoT relevante Netzwerkprotokolle. Dazu kommt eine Echtzeitfähigkeit.

1) *native port*: Der native port ist eine Version von RIOT, welche als regulärer Prozess innerhalb eines Linux-Systemes ausgeführt werden kann. Vergleichbar ist dies mit Lösungen wie User Mode Linux. Dies ermöglicht das Testen und Implementieren von hardwareunabhängigen Funktionen innerhalb von RIOT ohne auf einer eingebetteten Plattform jedes mal flashen zu müssen. Ferner können alle Funktionen des Wirtsystemes verwendet werden. Insbesondere bedeutet dies auch, dass der Arbeitsspeicher des Wirtsystemes im vollen Maße verwendet werden kann.

2) *Treiberarchitektur*: Um die hohe Portabilität von RIOT zu gewährleisten, sind alle Treiber prinzipiell in einen hardwareabhängigen und einen hardwareunabhängigen Teil getrennt. Dies hat zur Folge, dass bei der Portierung von RIOT auf eine neue

Plattform der hardwareabhängige Teil angepasst werden muss.

B. USB

Der Universal Serial Bus ist ein serieller Bus, der die Kommunikation von Peripheriegeräten mit dem Computer ermöglicht. Die Kommunikation läuft immer zwischen zwei Endpunkten ab. Ein Endpunkt ist eine Art Unteradresse in einem USB-Gerät. Viele Geräte besitzen Unterfunktionen, denen dann jeweils ein Endpunkt zugeordnet ist. USB kennt verschiedene Übertragungsmodi:

a) *Synchroner Transfer*: Beim synchronen Transfer (auch Kontrolltransfer genannt) werden durch eine bidirektionale Pipe kurze Datenpakete zwischen zwei Endpunkten hin und her geschickt. Diese Art des Datentransfers ist speziell für die Konfiguration von Geräten entworfen worden.

b) *Asynchroner Transfer*: Der asynchronen Transfer kann dadurch charakterisiert werden, dass die Antwortpakete vom Gerät zu beliebigen Zeitpunkt beim Sender eintreffen. Die Verarbeitung der übertragenen Daten muss folglicherweise auch asynchron erfolgen. Die Übertragung asynchroner Pakete ist generell mit einer CRC16-Prüfsumme gesichert. Dies bedeutet jedoch nicht, dass bei einem fehlerhaften Transfer automatisch eine Übertragungswiederholung durchgeführt wird, sondern dass erkannt werden kann, ob ein Paket fehlerhaft ist. Beim asynchronen Transfer unterscheidet man wiederum in folgende drei Submodi:

Bulk-Transfer: Der Bulk-Transfer ist vor allen Dingen für die Übertragung von großen Datenmengen gedacht. Da diese oftmals nicht zeitkritisch sind, wird diese Form des Transfers niedrig priorisiert und erst durchgeführt, wenn isochrone Transfers und Interrupt-Transfers abgeschlossen sind.

Interrupt-Transfer: Interrupt-Transfers sind dafür da, um kleine Datenmengen schnell und unter Umständen häufig zu übertragen, daher wird diese

Form des Transfers hauptsächlich für Eingabegeräte wie Maus und Tastatur verwendet.

Isochroner Transfer: Die Form des Transfers garantiert eine bestimmte Datenrate zwischen zwei Endpunkten. Bei der Initialisierung einer solchen Verbindung muss die gewünschte Datenrate explizit auf dem USB-Hostcontroller reserviert werden. (vgl. *Alternate Setting*) Zur Steigerung der Datenrate ist es möglich, mehrere isochrone Transfers parallel zwischen mehreren Endpunkten ablaufen zu lassen. Isochroner Transfer läuft im Grunde genommen so ab, dass vorbereitete, leere Datenpakete verschickt werden, die dann auf der Geräteseite mit den zu übertragenden Daten gefüllt werden und als Antwortpakete zurückgesendet werden.

C. USB-Video-Grabber