

A. Zugriff unter Linux auf die Schnittstelle, an der das Anytone angeschlossen ist.

Windows und Linux unterscheiden sich deutlich in der Behandlung von Schnittstellen. Unter Linux wird alles als File abgebildet und in das Filesystem eingebunden. Die Schnittstellen unter Linux finden sich im Filesystem unter **/dev/** wieder. Folglich haben die Hardware Schnittstellen auch alle Eigenschaften von Files: Es gibt einen Besitzer und die Zugriffsrechte, die bei der Benutzung von Windows Programmen unter Wine zu beachten sind. Zwar werden die seriellen Schnittstellen unter Wine für Windows Programme per Link „durchgereicht“, doch nehmen sie dabei auch alle Attribute (so auch die Zugriffsrechte), die sie bei der Erstellung (z.B. durch Einstecken des USB-Kabels) bekommen haben mit. Aufgrund der Zugriffsrechte entstehen bei der Nutzung unter Windows häufig Probleme.

Schließt man nun das Anytone über das USB-Kabel an und schaltet das Anytone ein, so wird dazu unter **/dev/** das passende Device-File angelegt. Bei mir (Lubuntu 24.04.1 LTS) wird das File **/dev/ttyACM3** erzeugt. Der Name nach tty hängt von dem Schnittstellenbaustein ab und davon, ob bereits andere Schnittstellen dieser Art installiert sind. Bei mir hängt bereits ein DXP von microHAM per USB am Rechner, wodurch bereits die Schnittstellen **/dev/ttyACM0** bis **2** erzeugt wurden. Das Anytone bekommt folglich die nächste Nummer, nämlich **tty/dev/ACM3**.

Unter den meisten Linux-Systemen gehören die Files für serielle Schnittstellen dem User **root**. Außerdem gehören sie in die Usergruppe **dialout**. Sowohl für den Besitzer (root) als auch für User, die Mitglieder der Gruppe dialout sind, sind die Zugriffsrechte auf **rw** gesetzt. Root und dialout-Mitglieder dürfen also (rw = read write) Zeichen aus der Schnittstelle lesen, als auch Zeichen in die Schnittstelle schreiben. Anders sieht es bei allen anderen Usern (other) aus: Hier stehen die Zugriffsrechte nur auf **-** (weder r noch w), also kein Zugriff auf die Schnittstelle. Da Windows Programme unter Wine aber weder unter dem User root laufen, noch der Wine-User Mitglied der Gruppe dialout ist, quittiert das Windows Programm den Versuch des Öffnens der Schnittstelle bereits als Fehler.

Weil unter Wine noch viele andere Programme laufen (können), halte ich es nicht für ratsam, die vorgegebenen User von Wine zu ändern. Alternativ kann man eben auch die Zugriffsrechte des Schnittstellenfiles ändern, so dass sie für alle User zugreifbar ist. Der Befehl hierzu lautet:

sudo chmod o+rw /dev/ttyACM3 (hier euer File angeben).

Allerdings muss dieser Befehl jedes Mal neu gegeben werden, wenn das Anytone mit dem Rechner verbunden und eingeschaltet wird.

Durch eine Änderung der Behandlung des Betriebssystems beim Einstecken des USB-Kabels kann man jedoch die Zugriffsrechte sofort beim Einsteckvorgang automatisch ändern lassen. Hierzu macht man sich zu nutze, dass der udev-daemon, der für das Anlegen der Device-Files zuständig ist, sich seine Anweisungen, wie er mit bestimmter Hardware umgehen soll, aus Konfigurationsfiles holt. Wie bei Linux üblich, sind die Konfigurationsfiles Textfiles, also mit dem Standard-Texteditor des Betriebssystems veränderbar.

Da bekanntlich viele Wege nach Rom führen, gibt es auch hier mehrere Möglichkeiten. Ich zeige hier zwei Möglichkeiten auf:

1. eine einfach zu realisierende, die lediglich die Zugriffsrechte auf alle per USB eingesteckten Geräte ändert und
2. eine etwas kompliziertere, die genau nur die Anytone-Schnittstelle ändert und dabei weitere Möglichkeiten wie eindeutiger Name der Schnittstelle und anderer User- und / oder Gruppenzugehörigkeit der Schnittstelle.

Die Konfigurationsfiles für den **udev-daemon** liegen im Verzeichnis `/etc/udev/rules.d`. Um nicht am System etwas zu verdrehen, was man später nicht mehr richtig fixen kann, legt man einfach ein neues File an:

```
sudo nano /etc/udev/rules.d/80-myusb.rules
```

Der Name des Files ist egal, wichtig ist nur, dass die Endung des Files `.rules` ist (der `udev-daemon` liest nur Files mit der Endung `.rules`).

1. Einfache Lösung

In dieses File schreibt kommt nun die Zeile:

```
KERNEL=="ttyACM[0-9]*",MODE="0666"
```

Nach dem Speichern und Aus- und wieder Einschalten der Geräte wird der `udev-daemon` nun alle `/dev/ttyACM` Devices für alle User auch beschreibbar erzeugen; die Zugriffsrechte sind nun für alle User auf `rw` gesetzt.

2. Aufwändigere Lösung

Im Gegensatz zu Lösung 1, bei der die Zugriffsrechte für alle seriellen Geräte, die über USB verbunden werden, greift Lösung 2 für genau das eine Gerät (Anytone), das wir verbinden; bei allen anderen USB-Geräten ergibt sich keine Änderung. Zwingend notwendig dazu ist aber, dass der `udev-daemon` das eingesteckte Gerät eindeutig von allen anderen unterscheiden. Normalerweise bekommt ein neu eingestecktes Gerät den nächsten freien Filenamen für die betreffende Gerätekategorie. In meinem Fall `/dev/ttyACM0` oder `/dev/ttyACM3`, je nach dem, welches Gerät zuerst eingesteckt und eingeschaltet wird. Hier unterscheidet sich Linux von Windows: Windows weis jedem einmal erkannten Gerät immer die selbe `com`-Schnittstelle zu, unter Linux kann das selbe Gerät je nach Zeitpunkt des Einsteckens eine unterschiedlichen Device-Filenamen bekommen.

Um den `udev-daemon` dazu zu bringen, beim Einstecken eines Gerätes nur ganz spezielle für dieses eine Gerät vorgesehene Aktionen durchzuführen, kann man sich den Prozess zu Nutze machen, der immer dann abläuft, sobald ein Gerät eingesteckt wird (Plug and Play unter Linux). Nach dem Einsteckvorgang übermittelt das eingesteckte Gerät zahlreiche Eigenschaften wie Hersteller, Produktnummer, Art des Gerätes, etc. Der `udev-daemon` matcht nun diese Informationen zeilenweise mit den Anweisungen in den `.rules` Files. Jede Zeile in einem `.rules` File beinhaltet Bedingungen (eine oder mehrere) und Anweisungen (eine oder mehrere). Wenn alle Bedingungen einer Zeile erfüllt sind, wird die Anweisung aufgeführt.

Nun muss man „nur“ noch die Informationen aus dem Datenstrom, der beim Einsteckvorgang übermittelt wird zu finden, die für das individuelle Gerät einmalig sind. Hat man sich Informationen heraus gesucht, die auf mehrere Geräte zutreffen (z.B. den Hersteller), so werden alle Geräte mit dieser Herstellerkennung entsprechend der Anweisung behandelt, was meist nicht erwünscht ist. In meinen `.rules` Files verwende ich die Hersteller ID, die Produkt ID und die Seriennummer.

Anmerkung: Für Geräte, die mehrere serielle Schnittstellen haben wie viele Transceiver reicht das nicht aus; hier muss zusätzlich über die Interface Nummer selektiert werden. Für unseren Fall des Anytones mit nur einer Schnittstelle trifft dies aber nicht zu.

Zuerst müssen wir wissen, welches Schnittstellenfile dem Gerät beim Einsteckvorgang gerade zugeordnet worden ist. Auch hier wieder zwei Möglichkeiten:

Entweder

bei noch nicht eingeschaltetem Gerät alle `tty` Devices anzeigen lassen:

```
ls -al /dev/tty*
```

Danach Gerät einschalten, warten bis das Anytone komplett gebootet hat und ein zweites Mal nach den tty Devices schauen (ls -al /dev/tty*). Beide Listings unterscheiden sich durch einen Filenamen, bei mir ist das /dev/ttyACM3.

Oder

bei noch ausgeschaltetem Gerät folgenden Befehl geben:

sudo dmesg -W

Die Bildschirmausgabe wartet und kommt nicht mit einem neuen Prompt zurück. Nun Anytone einschalten und den Bootvorgang komplett abwarten. Auf dem Bildschirm erscheinen mehrere Zeilen. In einer der letzten Zeile steht der Name des Device-Files, beginnend immer mit tty. Bei mir also ttyACM3. Den Bildschirm durch Eingabe von Ctrl C wieder freigeben.

In beiden Fällen Gerät eingeschaltet lassen. Mit dem Wissen, wie der richtige Filename des Gerätes ist, geht es nun weiter.

sudo udevadm info --attribute-walk /dev/ttyACM3 > anytone.txt

Der Filename, in den die Ausgabe umgelenkt wird (anytone.txt) ist egal, nur bitte merken. Statt ttyACM3 den Filenamen angeben, der in eurem Fall für das Anytone vergeben wurde (s. Oben).

In diesem File sind nun alle Informationen gespeichert, die auch der udev-daemon bekommt. Wir schauen nun nach, welche Produkt ID etc. übermittelt wurden:

grep idProduct anytone.txt (Groß- Kleinschreibung beachten)

Wir bekommen mehrere Zeile ausgegeben; interessant ist die erste, in meinem Fall

ATTRS{idProduct}=="018a"

Weiter geht es analog mit idVendor (Hersteller) und serial (Seriennummer). Wir nehmen immer die erste Zeile, die grep uns auswirft.

Nun können wir unser .rules File schon einmal mit der Bedingung füllen. Also neues File anlegen und in das File die durch grep erhaltenen Informationen eintragen:

sudo nano /etc/udev/rules.d/80-anytone.rules

Für mein Anytone sind die folgenden Bedingungen in einer einzigen Zeile einzutragen:

**#SUBSYSTEM=="tty", ATTRS{idProduct}=="018a", ATTRS{idVendor}=="28e9",
ATTRS{serial}=="00000010000"**

Alles in einer Zeile, hier nur wegen der besseren Lesbarkeit umgebrochen. Wir beginnen die Zeile auch zunächst mit einem Hashtag, um die bisher noch unvollständige Regel noch nicht „scharf“ zu schalten. Außerdem muss im Anfang **SUBSYSTEM=="tty"** stehen, bitte nicht das Keyword SUBSYSTEMS (mit S hinten) verwenden, das ebenfalls in dem mitgeschriebenen File anytone.txt auftaucht.. Die einzelnen Bedingungen sind durch Kommata und Leerzeichen voneinander getrennt. Statt meiner Werte (die in den Anführungszeichen) tragt ihr eure Werte ein, und zwar exakt so, wie sie die vorherige grep Anfrage geliefert hat, z.B. auch die vielen führenden Nullen in der Seriennummer.

Nun müssen wir noch die Anweisung dahinter schreiben, die der udev-daemon ausführen soll, wenn alle Bedingungen zutreffen. Mehrere Anweisungen werden wieder durch Kommata und Leerzeichen voneinander getrennt.

Interessant für unseren Fall sind nun folgende Anweisungen; sie müssen nicht alle aufgenommen werden. Ihr überlegt euch einfach welche Anweisung(en) ihr wollt.

MODE="0666" setzt die Zugriffsrechte auf rw für alle user
OWNER="egon" ordnet die Schnittstelle dem User egon zu
GROUP="hunde" ordnet die Schnittstelle der Gruppe hunde zu

Anmerkung: Hier muss der User und die Gruppe eingetragen werden, unter denen Wine arbeitet. Das ist normalerweise auch der User und die Gruppe, unter der sich der Benutzer angemeldet hat.

SYMLINK+="ttyAnytone" erzeugt durch Link ein Device-File mit nun
eindeutigem Namen

ENV{ID_MM_DEVICE_IGNORE}=1 schaltet die Modeminitialisierung per AT-Befehle ab.
War bei mir nicht nötig, da das Anytone nicht als Modem eingestuft wurde. Bei meinem DXO von microHAM dagegen schon.

Die gewünschten Anweisungen in die selbe Zeile hinter die Bedingungen einfügen, das Hashtag Zeichen am Zeilenbeginn nach Fertigstellung der Zeile löschen, File speichern und Anytone aus- und wieder einschalten. Von jetzt an sollte das Anytone so mit Linux verbunden sein, wie es gewünscht wird. Bitte inmal mit `ls -al /dev/tty*` kontrollieren.

Meine Zeile für das Anytone sieht wie folgt aus (Zeilenumbruch hier nur wegen der besseren Lesbarkeit):

```
SUBSYSTEM=="tty", ATTRS{idProduct}=="018a", ATTRS{idVendor}=="28e9",  
ATTRS{serial}=="000000010000", OWNER="xxx", GROUP="xxx",  
SYMLINK+="ttyAnytone"
```

B. Durchreichen der Linux-Schnittstelle nach Wine

Wine mappt die Linux-Schnittstellen per Link als com Device unter dem Home-Directory des angemeldeten Users in dem Verzeichnis `dosdevices`, Unterverzeichnis von `.wine`; in der Linux-Notation also unter `~/wine/dosdevices` Um zu wissen, welcher COM-Port mit dem Windows-Programm unter Wine angesprochen werden muss, lässt man sich einfach das Verzeichnis der Ports anzeigen:

```
ls -al ~/wine/dosdevices/com*
```

Es werden alle com Files angezeigt mit ihrem Link auf die eigentlichen Linux-Devices. Da wir vorher heraus gefunden haben, an welcher Schnittstelle das Anytone angeschlossen ist (bei mir mit `/dev/ttyACM3` bzw. `/dev/ttyAnytone`), brauchen wir nur zu schauen, mit welchem com File die richtige Schnittstelle verlinkt ist. Bei mir ist das **com43**. Falls Wine noch nicht den Link gesetzt hat (beim ersten Mal ist ein Bootvorgang von Wine nötig), Wine nach Beenden aller Windows Programme neu booten:

```
wineboot
```

Nun ist das Anytone unter Wine immer unter der selben Schnittstelle anzusprechen.

Will man, dass das Gerät in Wine unter einem bestimmten COM-Port ansprechbar ist, z.B. weil das Windows Programm nur die Schnittstellen von COM1 bis COM 20 anbietet, so kann man dieses mit einem Eintrag in die Registry realisieren:

Unter Linux bei eingeschaltetem Anytone **winetricks** aufrufen. Es geht ein Fenster auf, in diesem **Standard wineprefix auswählen** aktivieren, auf **OK** drücken. Es geht ein neues Fenster auf, in dem **regedit starten** auswählen und wieder **OK** drücken. Es starten nun der Registry Editor. Dort gehen wir zu

HKEY_LOCAL_MACHINE\Software\Wine\Ports

Dort können wir mittels Rechtsklick, **neu** und dann **Zeichenfolge** einen neuen Eintrag für einen Port anlegen. Wir legen also einen neuen Eintrag mit dem Namen com1 an und tragen als Daten unseren Linux Filename für das Anytone Device-File ein, also /dev/ttyACM3 oder /dev/ttyAnytone .Tragen wir keine Daten ein, so wird dieses com-Device bei der Verlinkung unter Wine ausgelassen. Ich habe zunächst vier leere Einträge von com1 bis com4 erzeugt, später dann com1 mit dem Link zu /dev/Anytone belegt, com2 bis com 4 sind noch für eventuelle zukünftige Anwendungen, die eine spezielle com-Schnittstelle brauchen, reserviert.

Zum Schluss Fenster des Registry-Editors schließen und die beiden Fenster von winetricks durch **Abbrechen** ebenfalls schließen. Nach dem booten von Wine (bootwine) ist nun die Schnittstelle für das Anytone unter Wine richtig zugeordnet.

Mit diesen Einstellungen läuft bei mir das Linux-Wine CPS (Wine D878UV_WineHQ_3.05) problemlos.