



IR-Schreib-Lesekopf, USB-Interface

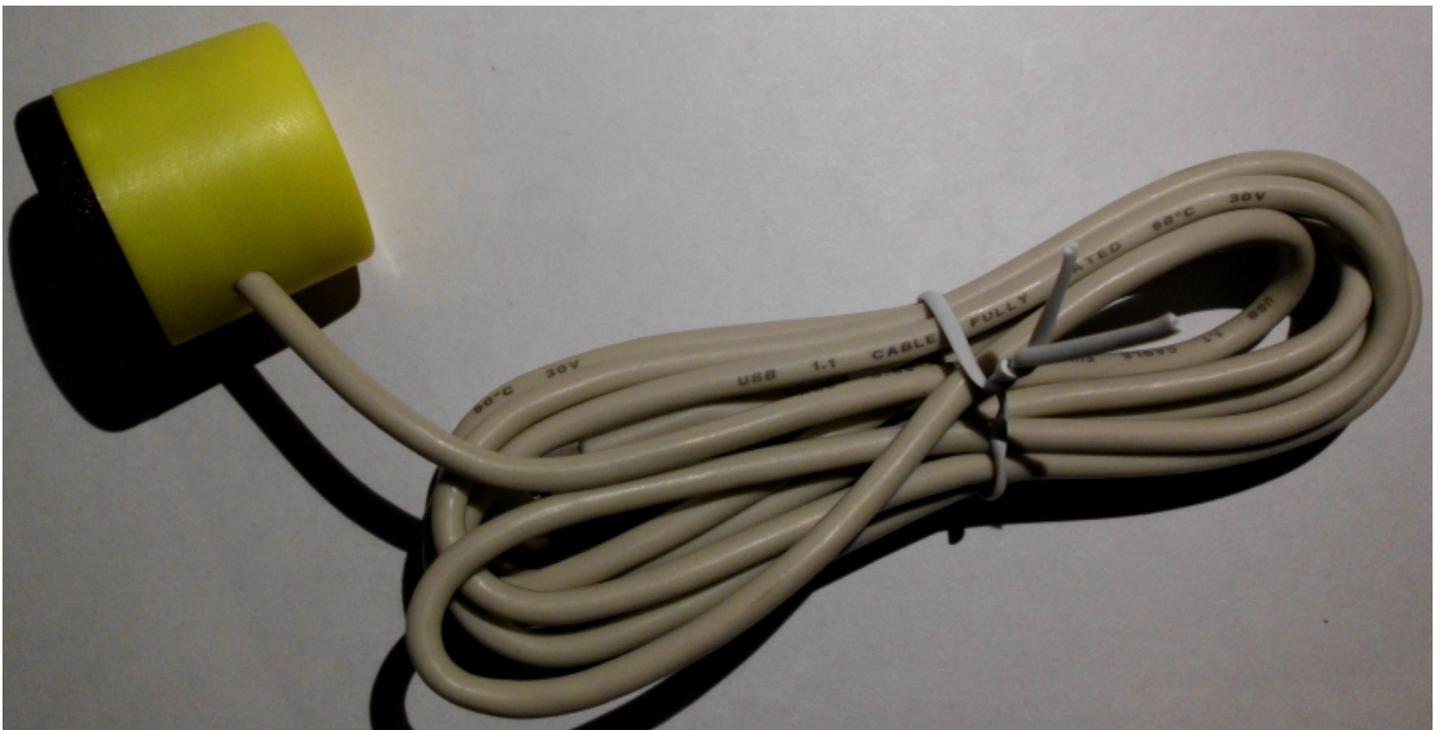
Hier der IR-Schreib-Lesekopf mit USB-Ausgang. Dieser ist für alle eHZ mit IR-Schnittstelle geeignet.

Features

- Optischer Sensor mit USB-Schnittstelle
- Schreib- und Lesefunktion
- Einfache Montage am Zähler durch eingebauten Ringmagnet
- Normgerechte Abmessungen
- 2,50 Meter Anschlussleitung
- Treiber für übliche Betriebssysteme

Hinweise zum IR-Kopf gibt es bei nilsvolksz@gmail.com

Bilder



Die einen Dauermagnet umgebenden Magnetfelder können empfindliche elektronische und mechanische Messgeräte beeinflussen oder sogar zerstören. Achten Sie auf ausreichenden Abstand (z.B. größer 2 m) insbesondere von Kredit- und EC-Karten, Datenträgern, Videokassetten, Monitoren, mechanische Uhren, Hörgeräten und Herzschrittmachern usw.).

Linux-Support

Wird der Lesekopf via USB an einen Linux-PC angeschlossen, bekommt er einen Gerätenamen. z.B. `/dev/ttyUSB0`. Werden noch andere USB-Geräte benutzt, die ebenfalls eine serielle Schnittstelle bereitstellen (z.B ein 2. Lesekopf) ist der Name der Schnittstelle nicht eindeutig.

Bei den meisten Distributionen sollte schon ein udev-script vorhanden sein, dass in

```
$ ls -l /dev/serial/{by-path,by-id}/*
```

symlinks anlegt, die die geräte einmal ueber die position am bus (usb-port) und einmal ueber ihre seriennummer identifizieren.

Falls diese nicht vorhanden sind, oder man es noch eleganter haben moechte, kann man auch eine eigene udev-regel anlegen, die das Geraet (zB ueber die beim Silab-Chip vorhandene Seriennummer) identifiziert, und mit einem sprechenden, eindeutigen Namen versieht:

Folgendes Kommando ausfuehren: (ttyUSB0 ist evtl. anzupassen...)

```
/sbin/udevadm info --query=all --name=/dev/ttyUSB0
```

In der Ausgabe findet sich eine Zeile „E: ID_SERIAL_SHORT=ABC1234“. ABC1234 ist die Seriennummer des USB-Chips.

Nun kann man eine Datei /etc/udev/rules.d/99-lesekopf.conf mit folgendem Inhalt anlegen:

```
SUBSYSTEM=="tty", ATTRS{product}=="CP2104 USB to UART Bridge Controller", ATTRS{serial}=="ABC1234", NAME="lesekopf0"
```

Ergaenzend ist zu erwaehnen, dass je nach Linux Distribution die udev-„Rules“-Datei, also z.B. /etc/udev/rules.d/99-lesekopf.conf nicht mit .conf, sondern mit .rules enden muss, damit der udev-Dienst diese Datei auch beruecksichtigt. Dies betrifft unter anderem Ubuntu und Debian. Siehe [debian-wiki \(en\) \[http://wiki.debian.org/udev\]](http://wiki.debian.org/udev).

Nach dem die Datei angelegt wurde noch kurz den udev-Dienst neu starten/laden (z.B. /etc/init.d/udev reload) und man kann ueber /dev/lesekopf0 auf den Lesekopf zugreifen. Egal, welche anderen ttyUSB Gerate es noch gibt 😊

Beim VZ-Image (Debian) oder Ubuntu 12.10 folgendes:

```
SUBSYSTEM=="tty", ENV{ID_SERIAL_SHORT}=="ABC1234", SYMLINK+="lesekopf0"
```

legt einen sym. Link „/dev/lesekopf0“ auf das eigentliche Device an.

den udev-Dienst dann mit

```
sudo udevadm trigger
```

die neue Regel anwenden lassen.

Testmoeglichkeiten unter Linux

Will man schnell pruefen ob der IR-Kopf in Verbindung mit einem Zaeher funktioniert und wie die Daten aussehen die empfangen werden, so gibt es verschiedene Moeglichkeiten, wobei minicom die naeheliegenste ist. Wenn Binärdaten uebermittelt werden wird es ein wenig spannender, denn diese sind unter minicom nicht von etwa gestoerten Daten zu unterscheiden.

Serielle Schnittstelle einstellen

Unter vielen anderen Moeglichkeiten bietet sich fuer die Konfiguration der seriellen Schnittstelle das Utility stty an und sollte im Zweifelsfall nachinstalliert werden. Um sich die vielen Optionen und Parameter einzusparen, bietet sich folgende Vorgehensweise an (als Schnittstelle soll /dev/ttyUSB0 dienen): man stellt die richtige Konfiguration unter minicom ein und verlaesst dieses ohne Reset (Ctrl-A, Q). minicom mit sudo aufrufen.

```
Welcome to minicom 2.7

OPTI+-----+
Comp| A -   Serial Device       : /dev/ttyUSB0  |
Port| B - Lockfile Location     : /var/lock    |
    | C - Callin Program       :              |
Pres| D - Callout Program      :              |
    | E -   Bps/Par/Bits       : 9600 7E1     |
    | F - Hardware Flow Control : No          |
    | G - Software Flow Control : No          |
    |                               |
    |   Change which setting? █             |
+-----+-----+
    | Screen and keyboard  |
    | Save setup as dfl   |
    | Save setup as..    |
    | Exit                 |
+-----+-----+
```

Wobei die Punkte A und E natuerlich an die Gegebenheiten angepasst werden koennen.

Wenn man dann aber schon in minicom ist, kann man auch gleich den IR-Kopf testen, indem man ueber die Tastatur Zeichen ausgibt, die dann, wenn man den IR-Kopf 1-2 cm ueber eine weisse Flaechе (Papier) haelt, sofort wieder empfangen werden und in minicom angezeigt werden.

In vzlogger ist fuer das Protokoll „sml“ 8N1 und fuer „d0“ 7E1 der Defaultwert und braucht nicht extra eingestellt werden.

Serielle Schnittstelle auslesen

Hat man die richtige Einstellung gefunden, dann reicht ein

```
cat /dev/ttyUSB0
```

oder für binäre Daten

```
cat /dev/ttyUSB0 | od -tx1
```

um zu sehen was über die serielle Schnittstelle empfangen wird.

Die Ausgabe sollte dann für binäre Daten so aussehen:

```
0000000 1b 1b 1b 1b 01 01 01 01 76 07 00 0a 00 6f 31 74
0000020 62 00 62 00 72 63 01 01 76 01 01 07 00 0a 00 20
0000040 10 7c 0b 06 45 4d 48 01 04 c5 6b 7d ce 01 01 63
0000060 86 9d 00 76 07 00 0a 00 6f 31 75 62 00 62 00 72
0000100 63 07 01 77 01 0b 06 45 4d 48 01 04 c5 6b 7d ce
0000120 07 01 00 62 0a ff ff 72 62 01 65 00 20 79 be 7a
...
```

Der Treiber fuer den verwendeten Silabs cp2104 USB-Seriell Wandler ist in Linux Kernel Versionen bis 3.2 DEFEKT, Das Geraet wird zwar erkannt, aber das Einstellen der Baudrate funktioniert nicht. Das Problem ist erst in Linux ab Kernel-Version v3.3-rc2 behoben

Details: https://bugzilla.kernel.org/show_bug.cgi?id=42586 [https://bugzilla.kernel.org/show_bug.cgi?id=42586]

Windows und das Linux vom Raspberry Pi sind davon nicht betroffen.

Silabs stellt Treiber für alle gängigen Betriebssysteme zur Verfügung:

<https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

[<https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>]

hardware/controllers/ir-schreib-lesekopf-usb-ausgang.txt · Zuletzt geändert: 2018/03/25 14:53 von udo1