

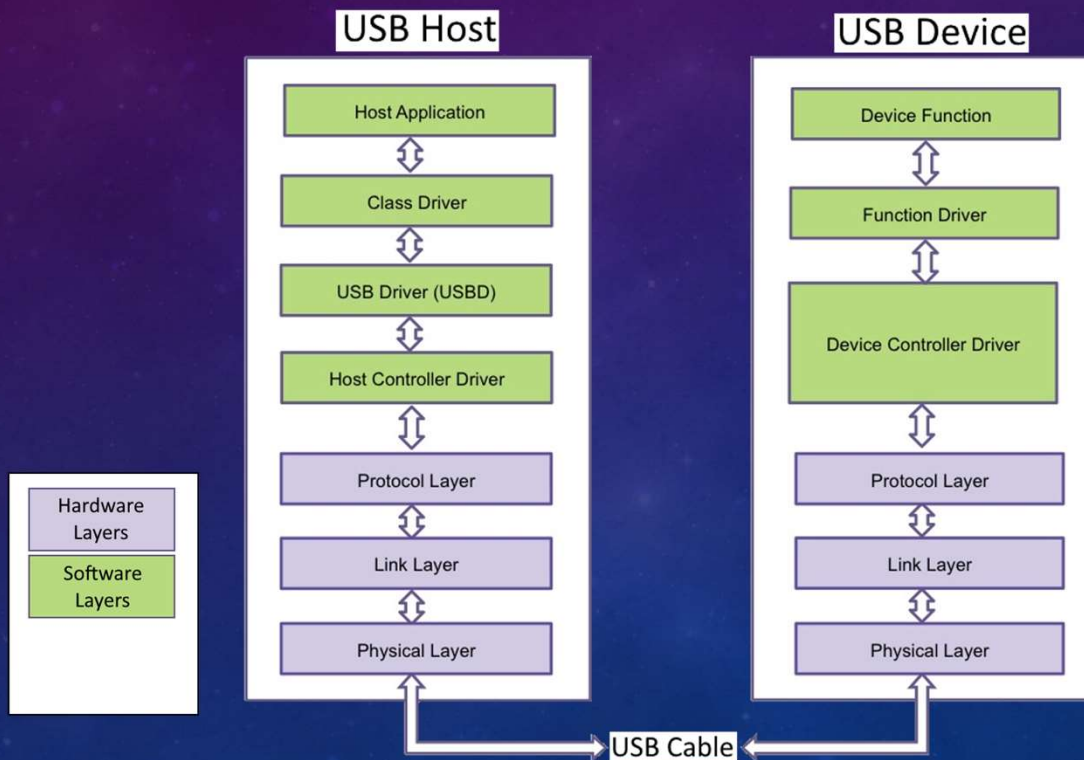
The background is a dark blue gradient with a subtle pattern of white dots. On the left side, there are several concentric circles and a large arc with a scale. The scale has numbers ranging from 140 to 260 in increments of 10. There are also some dashed lines and arrows pointing in different directions, creating a technical or scientific feel.

# USB SCHICHTEN

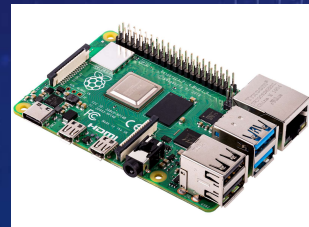
# STRUKTUR

- USB Schichten Überblick
- USB Host
  - Host Applikation
  - Klassen Treiber
  - USB Treiber
  - Host Kontroller Treiber
- USB Geräte
  - Geräte Funktion
  - Funktion Treiber
  - Geräte Kontroller Treiber
- Hardware
  - Protokoll Schicht
  - Link Schicht
  - Physikalische Schicht

# USB ÜBERBLICK DER SCHICHTEN



# USB HOSTS

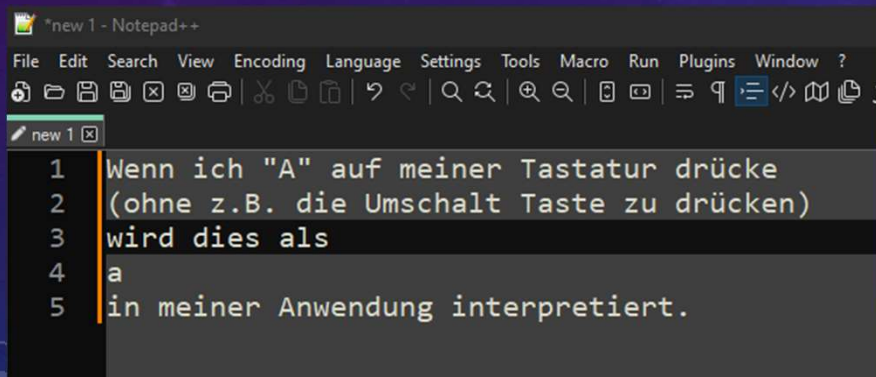




# HOST APPLICATION

Die Host Anwendung ist wo die Daten am Ende ankommen oder von abgefragt werden.

Dies könnte z.B. eine Notier Software, ein Spiel oder eine Audio Aufzeichnung Software sein.



```
*new 1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
1 Wenn ich "A" auf meiner Tastatur drücke
2 (ohne z.B. die Umschalt Taste zu drücken)
3 wird dies als
4 a
5 in meiner Anwendung interpretiert.
```

# CLASS DRIVER

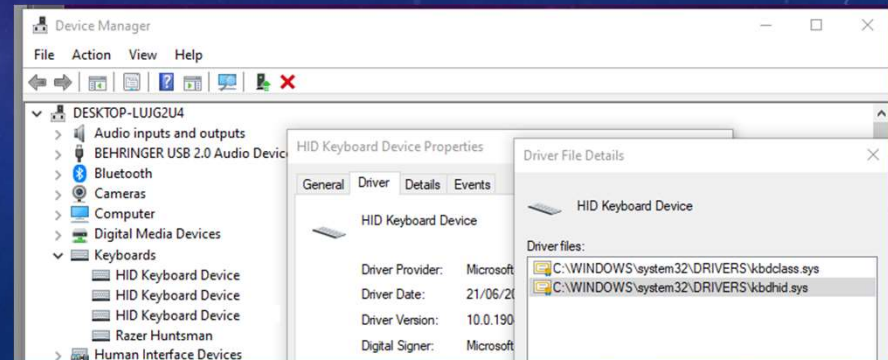
Class Drivers sind definiert unter anderem hier:

<https://www.usb.org/defined-class-codes>

Als Beispiel gibt es die

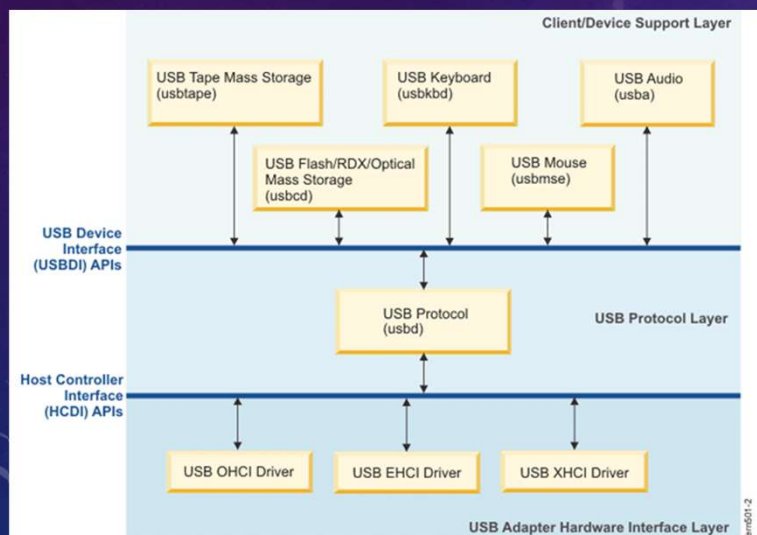
„**HID – Human Interface Device**„ Klasse.

Ein Klassen Treiber für z.B. Tastaturen, erhält alle Daten von allen verbundenen Tastaturen, verarbeitet die Daten und benachrichtigt das Betriebssystem.



# USB DRIVERS (USBD)

USB Treiber ist Software, welche Sachen behandelt wie z.B. was passieren soll wenn ein neues Gerät verbunden wird, oder auch das Verwalten des Stroms an den USB Ports.



Pk3

# HOST CONTROLLER DRIVER

Wie der Name schon beschreibt, sind Host Controller Treiber, Software welche die Hardware der USB Port Controller Chips mit dem Betriebssystem verbindet.



Pk4

## Slide 6

---

**Pk3**      <https://www.ibm.com/docs/en/aix/7.1?topic=concepts-universal-serial-bus-usb-subsystem>  
Pauk k, 30/11/2022

**Pk4**      <https://www.ebay.co.uk/p/1375911908?iid=291601349697>  
Pauk k, 30/11/2022

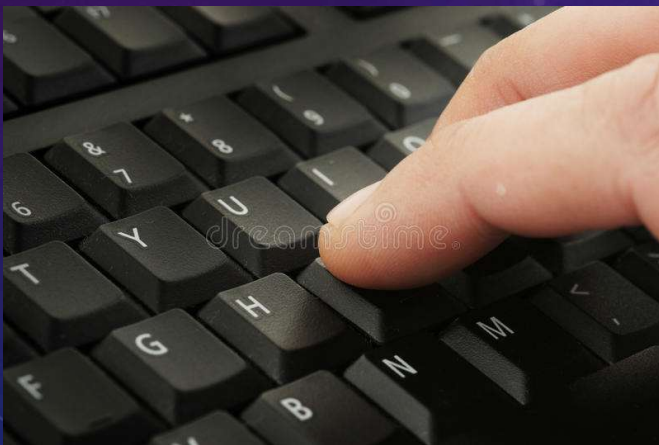
# USB DEVICES / USB GERÄTE





# DEVICE FUNCTION / ENDPOINT

Die Funktion des Gerätes, z.B. das Klicken einer Taste auf einer Tastatur.



Pk2

# FUNCTION DRIVER

Der Funktion Treiber bezweckt das erstellen von Paketen für die spezifische Klasse des Gerätes.

Sozusagen das übersetzen von Informationen die das Gerät erkennen kann zu Informationen die USB erkennen kann, oder anders herum.

```
/** HID class driver callback function for the creation of HID reports to the host.
 *
 * @param[in]  HIDInterfaceInfo  Pointer to the HID class interface configuration structure being referenced
 * @param[in,out] ReportID       Report ID requested by the host if non-zero, otherwise callback should set to the generated report ID
 * @param[in]   ReportType       Type of the report to create, either HID_REPORT_ITEM_In or HID_REPORT_ITEM_Feature
 * @param[in]   ReportData       Pointer to a buffer where the created report should be stored
 * @param[in]   ReportSize       Number of bytes written in the report (or zero if no report is to be sent)
 *
 * @return Boolean is true to force the sending of the report, is false to let the library determine if it needs to be sent
 */
bool CALLBACK_HID_Device_CreateHIDReport(USB_ClassInfo_HID_Device_t* const HIDInterfaceInfo,
                                          uint8_t* const ReportID,
                                          uint8_t ReportType,
                                          void* ReportData,
                                          uint16_t const ReportSize)
{
    USB_MouseReport_Data_t MouseReport = (USB_MouseReport_Data_t){ReportData};

    float adc = deConversion((uint16_t)0);
    int8_t calc = (int8_t)((adc/1023*20)-10);

    MouseReport->y = calc;

    adc = deConversion((uint16_t)1);
    calc = (int8_t)((adc/1023*20)-10);

    MouseReport->x = calc;

    MouseReport->button |= ((1 << Pin_PIO_PIO_1) | (1 << Pin_PIO_PIO_0)) << 0;

    MouseReport->button |= (1 << Pin_PIO_PIO_4) << 1;

    *ReportSize = sizeof(USB_MouseReport_Data_t);
    return true;
}
```

## Slide 8

---

**Pk2**

<https://www.dreamstime.com/stock-photo-finger-pressing-keyboard-image14255180>

Pauk k, 30/11/2022

# DEVICE CONTROLLER DRIVER

Der Geräte Kontroller Treiber übersetzt die Software zur oder von der Hardware.



Pk1

## Slide 9

---

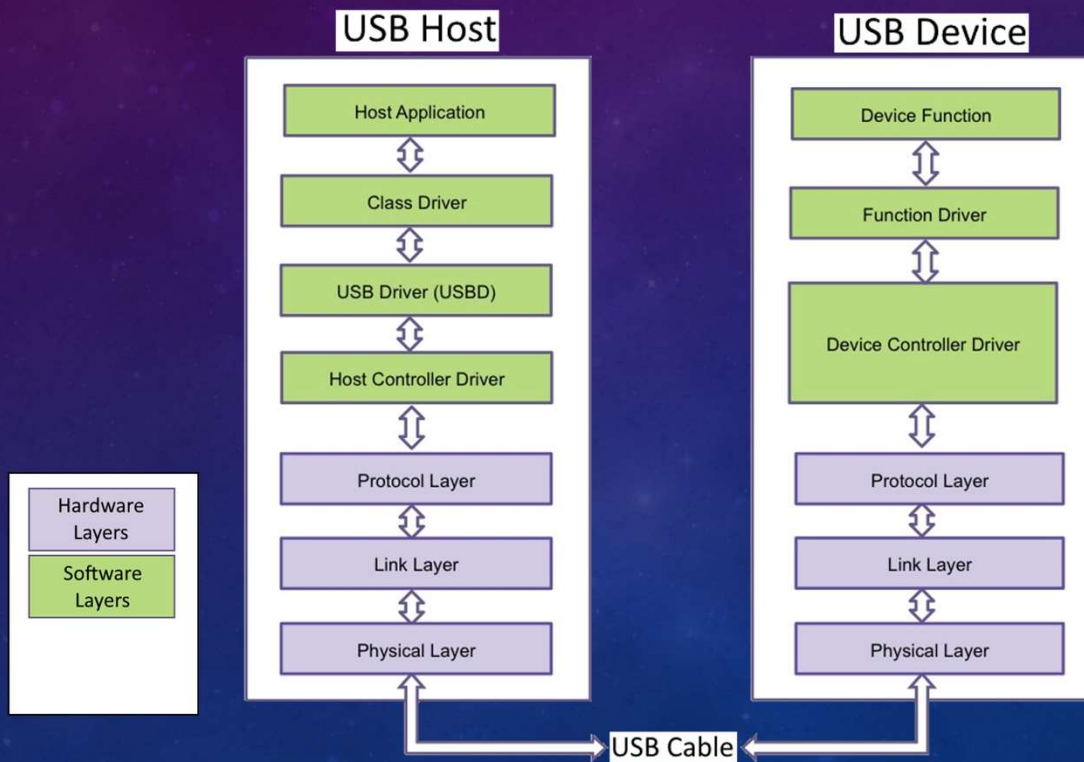
**Pk1**

<https://www.hood.de/i/usb-flash-drive-circuit-board-pcb-board-pcba-usb-stick-2-0-3-0-ohne-gehaeuse--84629>

Pauck, 30/11/2022



# HARDWARE



# PROTOKOLL SCHICHT

Die Protokoll Schicht hat 4 Typen von Transaktionen:

- Control Transfers
  - Für die send von Befehlen, konfigurieren des Gerätes, anfragen machen, etc..
- Interrupt Transfers
  - Senden von kleinen Daten vom Gerät zum Host, statt durch einer Anfrage vom Host.
- Bulk Transfers
  - Senden von großen Daten.
- Isochronous Transfers
  - Senden von Daten in einer sehr bestimmten Rate.
  - Ohne Fehlerkorrektur.

# LINK SCHICHT

- Behandeln der Strom Status
- Klein Frequenz Kommunikation zwischen zwei Ports
- Fehlerbehandlung

# PHYSIKALISCHE SCHICHT

- Vermischen der Daten um EMI zu verringern von 8Bit zu 10Bit
- (De)Serialisierung der Daten
- Datenwiederherstellung
- Clock Erstellung
- Daten Buffer



## Slide 12

---

**Pk7**

<https://blog.leviton.com/charging-technology-pd>

Pauk k, 30/11/2022



# TEXT QUELLEN

- <https://www.ibm.com/docs/en/aix/7.2?topic=concepts-universal-serial-bus-usb-subsystem>
- <https://www.linkedin.com/pulse/protocol-depth-usb-luigi-c-filho/>
- <https://www.techdesignforums.com/practice/technique/usb-3-0-system-overview/>

Bild Quellen sind  
gegebenenfalls als  
Kommentar vorhanden.