

# LAZ 10. HID USB Device AT90USB162

## Prolog

Für die Durchführung dieses Projektes benötigt man das [USB Addon von MCS](#).

Die Veröffentlichung von Teilen dieses Projektes ist nicht erlaubt!

Die eigentliche BASCOM Projektdatei ist dem Download beigelegt.

Zum Compilieren die Datei "usbinc.bas" des USB Addon von MCS in den Ordner der Datei "hid\_generic-162.bas" kopieren.

Die nötigen Dateien für das Lazarus Projekt, sind in der Downloaddatei enthalten.

Das Lazarus Projekt ist unter Windows und Linux compilierbar.

Die Zielhardware für die Beispielsoftware ist das MCS USB Motherboard mit aufgestecktem AT90USB162 Board.

-Display 16x2

-4 x LED

-3 x Taster

[MCS\\_USB\\_Motherboard.png](#)

Die genaue Anschaltung von Display, LED und Tastern kann dem Bascom Quellcode entnommen werden.

## Atmel AT90USB(162)

Die AT90USB Chips von Atmel sind bei Auslieferung mit einem Bootloader versehen.

Nach Anschaltung dieser Bausteine an den USB-Bus, melden sich diese im Gerätemanager in der Rubrik "Atmel" als "AT90USB162" Device an.

Dies ist kein "richtiger" Gerätemodus, sondern der Atmel "DFU" Modus. (**D**evice **F**irmware **U**ppgrade)

Dieser Modus wird von Atmel als "PROM" Modus beschrieben.

Nur, wenn ein intakter Bootloader aufgespielt ist, kann der Chip sich in diesem Modus anmelden.

Wird durch programmieren des Chip mit einem Programmer der Bootloader überschrieben (z.B. einspielen eigener Software), funktioniert dieser DFU Modus nicht mehr.

Der Bootloader muss erneut eingespielt werden. Hierzu für den AT90USB162 in den Fuses als Bootloader Adresse \$1800 und Bootvektor = Bootloader einstellen.

Danach das HEX File des Bootloader laden und programmieren.

[Download Atmel Bootloader Files](#)

Bei intaktem Bootloader kann man eigene Software mit Hilfe von Atmel FLIP über die BASCOM IDE einspielen.

Dazu unter [Optionen] [Programmer] "FLIP" auswählen.

Zum Übertragen der Software mit Flip, muss der AT90USB162 in den DFU Modus gebracht werden.

Dies erreicht man, indem man den PORTD.7 auf GND legt, danach ein Reset Signal auslöst und den PORTD.7 wieder auf VCC legt.

Danach erscheint der AT90USB162 wieder als "Atmel Device" im Gerätemanager.

Jetzt kann man mit der BASCOM-IDE mit "FLIP" die eigene Software übertragen.

[Download Atmel FLIP](#)

## Bascom

Der Bascom Quellcode basiert auf dem "generic HID input and output sample" Code vom MCS.

Um den Datenverkehr auf das Nötige zu reduzieren, habe ich drei Flags eingeführt.

Der Report ist auf die Übertragung von 8 Byte eingestellt, kann aber leicht angepasst werden.

Der Feature Report überträgt ebenfalls 8 Byte. Lediglich das erste Byte, Bit 0 - 4, wird zur Ansteuerung von 4 LED verwendet.

## Lazarus

Das Lazarus Projekt ist einfach gehalten und bedient den Feature Report, den normalen Report IN und OUT mit jeweils 8 Byte.

Zum Betrieb darauf achten, dass im Bascom Quellcode und den Felder im Lazarus Programm die gleichen PID/VID stehen!

Danach "Aktive" anhaken. Der Rest geschieht automatisch.

Wird das HID Device vom USB Bus getrennt bzw verbunden, wird dies von der Software automatisch erkannt.

Es sind keine weiteren Treiber unter Windows nötig!

Unter Linux müssen die Rechte am Gerät erweitert werden.

Hierzu eine Datei unter /etc/udev/rules.d/ anlegen.

Der Name könnte z.B. sein: 10-HID.rules

Inhalt der Datei:

```
# HIDAPI/libusb
```

```
SUBSYSTEM=="usbmisc", ATTRS{idVendor}=="16d0", ATTRS{idProduct}=="201d", MODE="0666"
```

```
# HIDAPI/hidraw
```

```
KERNEL=="hidraw*", ATTRS{busnum}=="1", ATTRS{idVendor}=="16d0", ATTRS{idProduct}=="201d", MODE="0666"
```

Hier die VID/PID anpassen! Buchstaben in Kleinschrift!

Zur Durchführung werden Administratorenrechte benötigt!

[HID\\_Sample.png](#)

Viel Spaß, Michael

[Laz HID.zip](#)