

LAZ 5. Lazarus für serielle Kommunikation aufrüsten

1 Der USB/Seriell Wandler

Die serielle Kommunikation war seit Anfang an bei PC's Bestandteil der Hardware. In früheren Zeiten war jedes Motherboard mit so genannten "COM- Schnittstellen" ausgerüstet.

Diese **COM**munication Schnittstellen waren SUB-D 9 Buchsen auf dem Motherboard.

Neue, universelle Bus-Systeme in moderneren Generationen der Hardware, machten es überflüssig spezielle Protokolle in Hardware auszuführen.

So haben alle moderne Rechner USB Schnittstellen, über welche man serielle Kommunikation abwickeln kann.

Damit man wieder eine SUB-D 9 Buchse zur Verfügung hat, benötigt man ein "USB/Seriell Kabel"

Ein Beispiel eines auf dem Markt verfügbaren, typischen Adapters

Image not found or type unknown



Dieser Adapter verwendet intern einen FTDI Chip (rs232rl), welcher in jedem Fall eine sichere und gute Wahl ist.

FTDI Chips werden von Windows mit signierten Treibern unterstützt und auch unter Linux arbeiten diese ohne Probleme.

2 Lazarus und Zugriff auf serielle Kommunikation

2.1 Die Komponente LazSerial, eine COM Schnittstelle

bascomforum.de/index.php?attac...8883d0fa11c524281390d339d

Sicherlich hast du schon mal die einzelnen Seiten der Komponentenpalette in der Lazarus "IDE" (Integrierte Entwicklungsumgebung) durchstöbert.

Dort finden sich allerlei Komponenten, jedoch keine serielle Schnittstelle.

Was machen? Lazarus weitwerfen?

Nein, die Lazarus Komponenten Palette lässt sich relativ einfach erweitern! Da du sicherlich noch keine Idee hast, wie man solch eine Komponente entwickeln könnte, nehmen wir etwas fertiges "von der Stange". Es kostet nichts und funktioniert einwandfrei: **LazSerial**

Die Komponente "LazSerial" wurde vom Benutzer "Jurassic Pork" , welcher auf der Seite forum.lazarus.freepascal.org/ vertreten ist, programmiert.

bascomforum.de/index.php?attac...8883d0fa11c524281390d339d

Am besten entpackt ihr das enthaltene Verzeichnis "LazSerial" in das Installationsverzeichnis von Lazarus unter "Komponenten" in einen Unterordner "_Eigene Komponenten"

also z.B.: C:\lazarus_1_6\components_Eigene Komponenten\LazSerial

Danach starte Lazarus. Wähle [Package] --> [Package Datei (*.lpk) öffnen] aus.

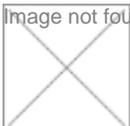
Image not found or type unknown



Wähle nun das Verzeichnis "LazSerial" aus, welches wir vorhin eingespielt haben. Dort erscheint die Datei "**LazSerialPort.lpk**". Diese Selektieren und "OK" klicken. (...oder Doppelklick)

Es erscheint nun folgendes Fenster:

Image not found or type unknown



Hier "Installieren" anklicken und die folgende Frage, ob Lazarus neu kompiliert werden soll mit "OK" bestätigen.

Lazarus wird nun neu kompiliert und die Komponente LazSerial wird eingefügt. Nach dem automatischen Neustart von Lazarus siehst du den Tabreiter "LazSerial" in der Komponentenpalette.

Wenn bis hierher alles geklappt hat, dann hast du die Komponente erfolgreich installiert und wir können diese benutzen. Falls es Probleme gab, helfe ich gerne weiter und werde die Anleitung ggf. ergänzen.

2.2 Die Komponente "FTDI"

bascomforum.de/index.php?attac...8883d0fa11c524281390d339d

Die FTDI Komponente habe ich entwickelt. Sie greift über einen speziellen Treiber direkt auf den FTDI Baustein zu. Dadurch ist es nicht nur möglich, die serielle Schnittstelle zu verwenden, sondern auch noch Ein- und Ausgänge des FTDI Bausteins. Praktisch könnte man so einen Ausgang verwenden, um einen AVR Baustein zu testen. Um eine Verbindung zu einem FTDI Baustein aufzubauen, muss hierfür auch kein COM-Port ausgewählt werden, die Komponente kann das Nötige selbstständig, um Kontakt aufzunehmen!

[Download FTDI Komponente](#)

Verfahre mit der Installation der Komponente wie unter 2.1

Nach der Installation bitte folgendes beachten:

Unter Windows wird die DLL ftd2xx.dll benötigt. Diese entweder einem Projekt zufügen, indem diese DLL im gleichen Verzeichnis wie die EXE Datei liegt, oder allgemein verfügbar unter c:\windows\system32 kopieren.

Unter Linux die Datei libftd2xx.so.1.3.6 nach /usr/lib kopieren

zum nächsten Teil: ["Houston, we've got Contact..." oder TXD und RXD](#)

Michael Köcher / six1