Raspberry Pi-Treiberpaket für WinFACT

(Stand: 23.01.2014)

Leistungsumfang

Dieses Treiberpaket enthält Treiber für das Blockorientierte Simulationssystem BORIS aus dem Programmsystem WinFACT für verschiedene Raspberry Pi-Konfigurationen. Jeder Treiber besteht aus einer entsprechenden User-DLL für BORIS sowie dem zugehörigen Python-Skript für den Raspberry Pi. Ebenfalls enthalten sind die DELPHI 6-Quelltexte der User-DLLs. In der aktuellen Version werden folgende Module bzw. Schnittstellen unterstützt:

Modul bzw. Schnittstelle	Zugehörige User-DLL	Zugehöriges Python-Skript
Onboard-GPIO	`RaspberryGPIO.DLL`	`BORIS_RaspberryGPIO.py`
PiFace Digital	`RaspberryPiFace.dll`	`BORIS_RaspberryPiFace.py`
GERTBoard	`RaspberryGert.dll`	`BORIS_RaspberryGert.py`

Kopieren der Dateien und Einrichtung der Treiber

Nachfolgende Tabelle listet alle im Treiberpaket enthaltenen Dateien (größtenteils mit Wildcards) zusammen mit den entsprechenden Zielverzeichnissen auf. *User-DLL-Unterverzeichnis* ist dabei das Unterverzeichnis *UserDLLs* Ihrer WinFACT-Installation.

Dateiname	Zielverzeichnis	Kommentar
`*.DLL`	User-DLL-Unterverzeichnis	BORIS-User-DLL
`*.BMP`	User-DLL-Unterverzeichnis	Block- und Toolbar-Bitmaps
`*.BSY`	beliebig	BORIS-Beispieldatei
`*.DFM`	beliebig	DELPHI-Formulardatei
`*.DPR`	beliebig	DELPHI-Projektdatei
`*.PAS`	beliebig	DELPHI-Quelltextdatei
`*.PY`	Raspberry Pi	Python-Skript zum Treiber
`ReadMe.PDF`	beliebig	diese Datei

Überprüfen Sie nach dem Kopieren der Dateien, ob das User-DLL-Unterverzeichnis Ihrer WinFACT-Installation in BORIS als Suchverzeichnis eingerichtet ist (Menüoption *Optionen* | *Anpassen...*). Ist dies nicht der Fall, holen Sie dies zunächst nach (Bild 1).

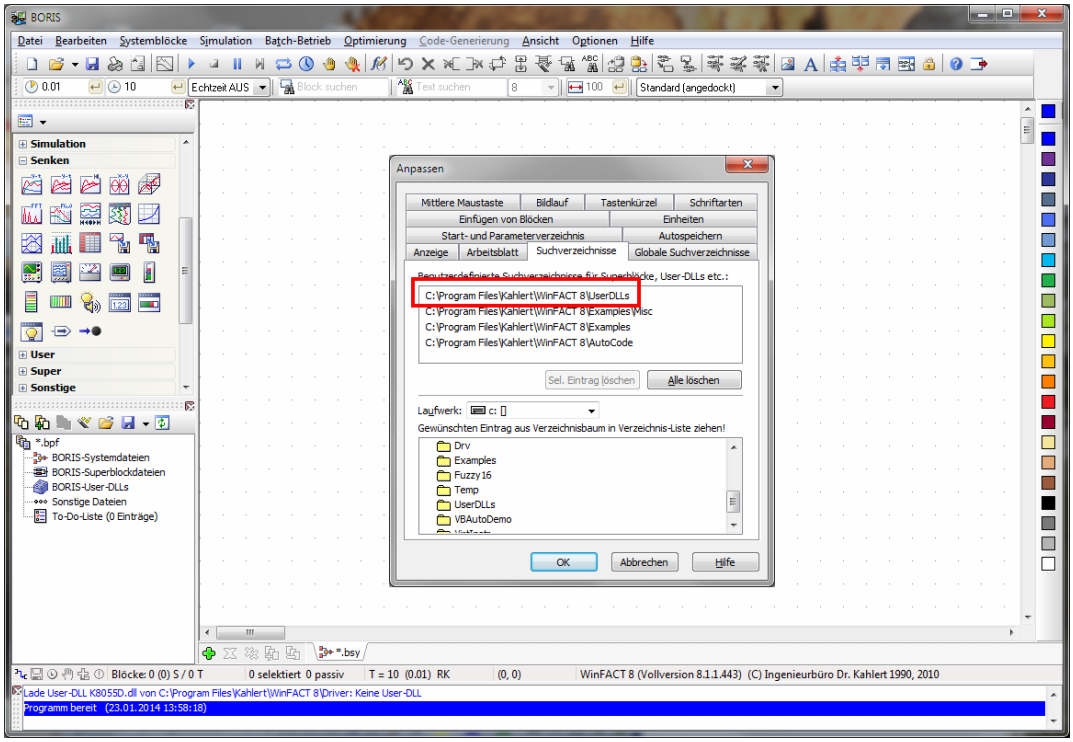

Bild 1. Einrichten des User-DLL-Unterverzeichnisses als Suchverzeichnis in BORIS

Schließen Sie BORIS anschließend und starten Sie es erneut. Innerhalb der *User*-Palette der Systemblockbibliothek sollten Sie nun die Schaltflächen für die verschiedenen Raspberry Pi-Treiberblöcke finden (Bild 2).

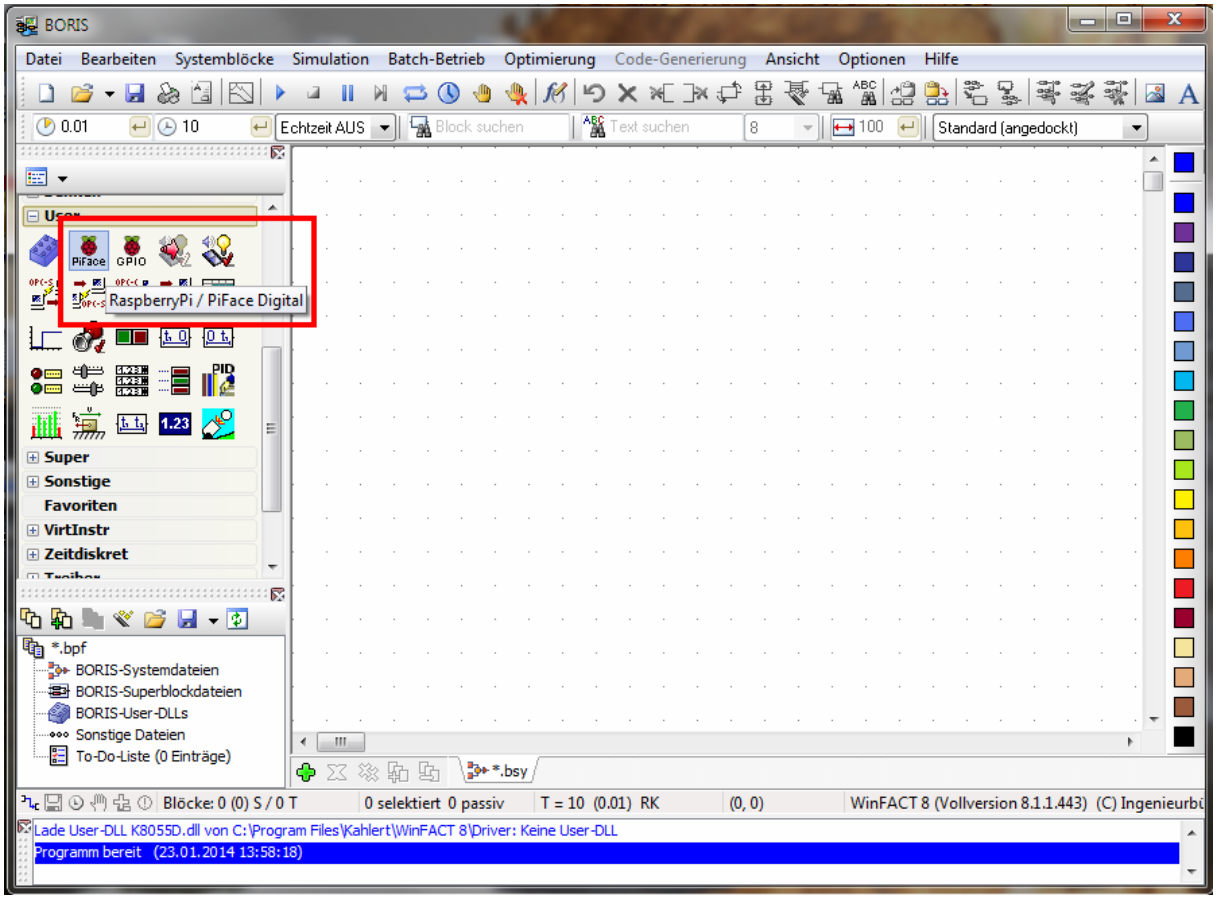

Bild 2. Schaltflächen für Raspberry Pi-Treiber in der BORIS-Systemblockbibliothek

Nutzung der Treiber

Jeder Treiber besteht aus zwei Komponenten: einer User-DLL für BORIS sowie dem zugehörigen Python-Skript für den Raspberry Pi. BORIS kommuniziert dabei mit dem Raspberry Pi über eine LAN- bzw. WLAN-Verbindung via UDP (*User Datagram Protocol*).

Innerhalb von BORIS fügen Sie einen Treiber-Block über die entsprechende Schaltfläche der Systemblock-Bibliothek in eine Simulationsstruktur ein. Durch Doppelklick auf den Block und anschließende Betätigung der Schaltfläche *Dialog...* innerhalb des Standard-User-DLL-Parameterdialogs gelangen Sie in den Dialog zur Konfigurierung des Treiber-Blocks. Hier ist die IP-Adresse des Raspberry Pi anzugeben, der angesprochen werden soll, sowie die Portnummer, über die die Kommunikation stattfinden soll. Außerdem können hier je nach Treibertyp verschiedene I/O-Konfigurationen vorgenommen werden. Bild 3 zeigt dies am Beispiel des Treiber-Blocks für die Onboard-GPIO des Raspberry Pi.

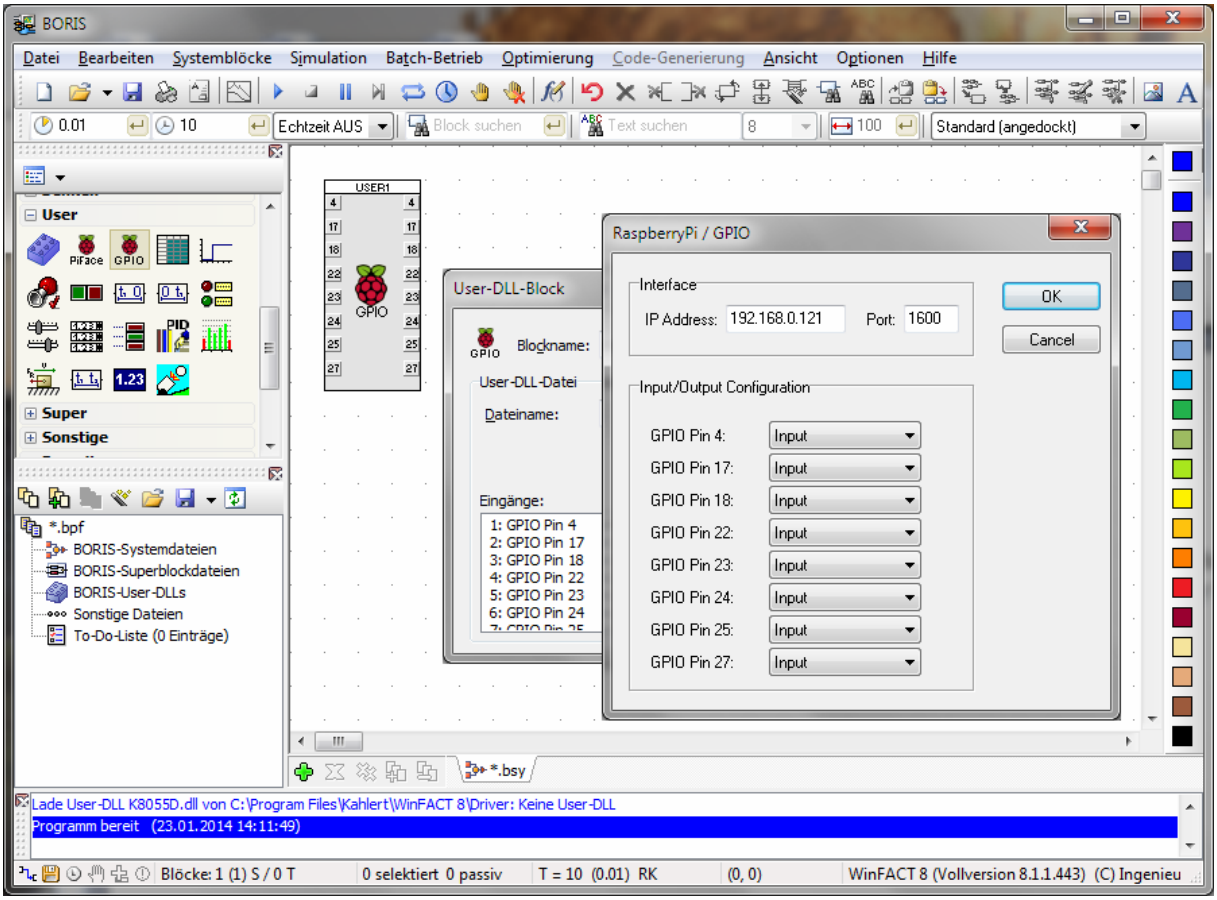

Bild 3. Konfigurierung des Treiber-Blocks (hier für Onboard-GPIO)

Auf dem Raspberry Pi muss parallel das zugehörige Python-Skript (nach einer entsprechenden Anpassung von IP-Adresse und Portnummer) gestartet werden. Bild 4 zeigt beispielhaft das Skript für die Onboard-GPIO. Bitte beachten Sie, dass je nach verwendetem Skript ggf. diverse Python-Pakete (z. B. *Rpi.GPIO*) auf dem Raspberry Pi installiert werden müssen, damit das Skript lauffähig ist. Hinweise dazu entnehmen Sie bitte den entsprechenden Raspberry-Quellen im Internet oder der einschlägigen Literatur.

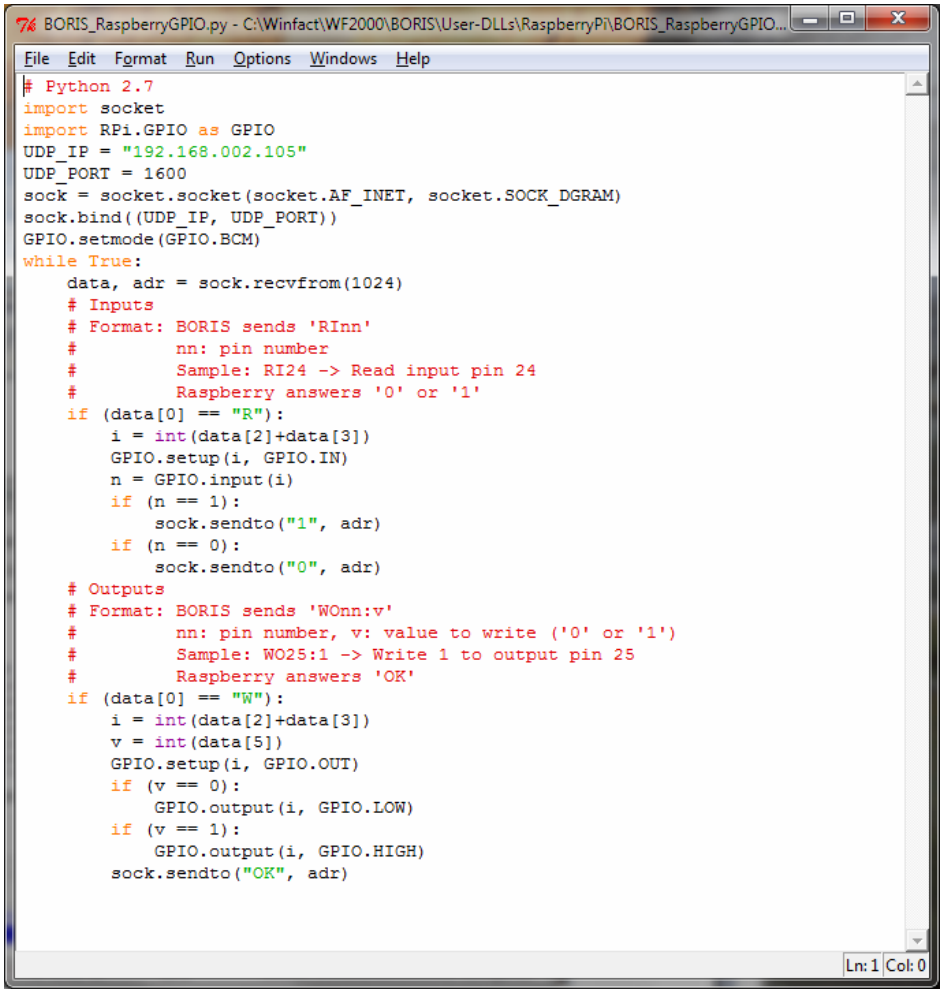

Bild 4. Python-Skript für Raspberry Pi (hier für Onboard-GPIO, angezeigt unter Windows!)

DELPHI-Projekte der User-DLLs

Alle Treiber-User-DLLs wurden unter DELPHI 6, einer bereits recht alten DELPHI-Version, entwickelt. Die entsprechenden Quelldateien sind im Treiberpaket enthalten, sodass sich die Treiber bei Bedarf anpassen oder erweitern lassen. Wird dabei eine DELPHI-Version ab DELPHI 2009 benutzt, sind aufgrund der Unicode-Umstellung von DELPHI Folgendes zu beachten:

> Bei Verwendung von DELPHI 2009 oder einer neueren Version ist der Datentyp `Char` jeweils durch den Datentyp `AnsiChar` und der Datentyp `PChar` jeweils durch den Datentyp `PAnsiChar` zu ersetzen!

Weitere Hinweise

Bei der Kommunikation mit einem Raspberry Pi sollte BORIS grundsätzlich im *Echtzeitbetrieb* betrieben werden. Die Simulationsschrittweite sollte nicht unter 0.1 s (d. h. 100 ms) liegen.