

1 WLib-SPI-Abstraction

1.1 Motivation

Angenommen man bekommt die Aufgabe einen Treiber für den über SPI angebundenen Chip-XY zu schreiben. Damit die Wiederverwendbarkeit des Treibers sichergestellt werden kann sollte er möglichst plattformunabhängig sein. Jedoch ist der Featureset und die Bedienung der SPI-Hardware eine stark plattformabhängige Angelegenheit. Es empfiehlt sich deshalb die SPI-Kommunikation als ganzes zu Abstrahieren. Die Bedienung der SPI-Hardware wird an einen spezialisierten SPI-Hardware-Treiber delegiert.

WLib-SPI-Abstraction stellt ein solches Interface zur Verfügung. Der Treiber für den Chip-XY benutzt es. Die spezialisierten SPI-HW-Treiber der unterschiedlichen Plattformen stellen es zur Verfügung.

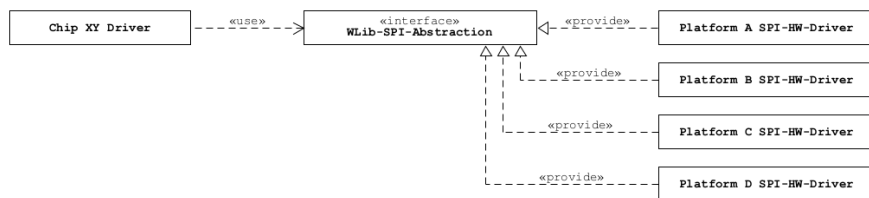


Abbildung 1: Dependency graph

1.2 Abstraktes Kommunikationsmodell

Zuerst muss die SPI-Hardware aktiviert bzw. konfiguriert werden. Danach wird der/die Chips ausgewählt. Nun können die Daten übertragen werden. Nach der Datenübertragung wird der Chip wieder abgewählt. Zum Abschluss wird die SPI-Hardware deaktiviert.