

Aufbau einer Softwarebibliothek für das IO-Link-Master-Shield/Hat

Studiengang: BSc in Elektrotechnik und Informationstechnologie | Vertiefung: Embedded Systems
Betreuer: Prof. Dr. Torsten Mähne, Ivo Adrian Oesch
Experte: Daniel Kühni (Inetronic AG)
Industriepartner: Balluff GmbH, Neuhausen a.d.F.

Die Kommunikationsschnittstelle IO-Link verbreitet sich immer mehr in der Automatisierungsindustrie. Zur einfacheren Nutzung von IO-Link-Geräten in der Ausbildung und Kleinprojekten wurde an der BFH ein IO-Link-Master entwickelt, der als Aufsatz auf einem Arduino Due oder einem Raspberry Pi nutzbar ist. In dieser Arbeit wurde eine Software-Bibliothek für diesen Master entwickelt, die beide Plattformen unterstützt und einfach an neue Hardware anpassbar ist.

Ausgangslage und Ziel

IO-Link ist ein Master-Slave-Kommunikationsprotokoll. Jeder Sensor oder Aktor (Device) wird direkt an einen IO-Link-Master angeschlossen, der als Schnittstelle zur Anlagen-Steuerung dient. IO-Link-Geräte sollen auch im Unterricht und für Kleinprojekte einfach nutzbar sein. Dazu entwickelte Pascal Frei in seiner Bachelor-Thesis das IO-Link-Master-Shield/Hat, das auf einen Arduino oder Raspberry Pi aufgesteckt werden kann. Die von ihm erstellte rudimentäre Software-Bibliothek zur Demonstration auf einem Arduino wurde später auf Raspberry Pi portiert und funktionsell erweitert.

Ziel dieser Arbeit war es, die Treiberbibliothek für das Shield/Hat neu aufzubauen, so dass sie zukünftig einfach an neue Plattformen und IO-Link-Hardware angepasst werden kann.

Konzept

Das IO-Link-Protokoll ist in der „IO-Link Interface and System Specification“ standardisiert. Weil diese so umfangreich ist, kann es nicht vollständig im Rahmen dieser Bachelor-Arbeit implementiert werden. Trotzdem sollte sich die Struktur der Bibliothek stark an der Spezifikation orientieren (Abb. 1). Die Namen der Klassen und Methoden entsprechen genau den Elementen in der Spezifikation, die sie abbilden. Dies erleichtert die Orientierung in der Software.

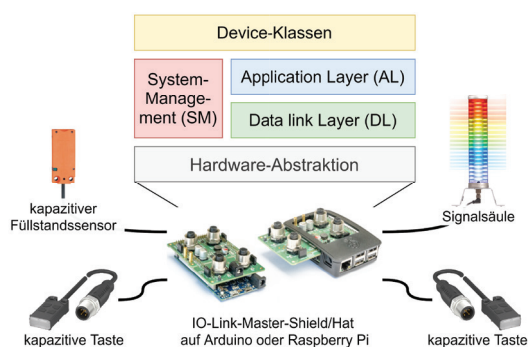


Abb. 1: Beispiel für den Einsatz der Software-Bibliothek

Ergebnisse und Ausblick

Es wurde ein detailliertes Softwaredesign erarbeitet, das von der Hardware-Abstraktion bis hinauf zu den Device-Klassen reicht. Die Methoden für alle „Services“ zwischen den Schichten sind vorhanden, aber noch nicht implementiert. Die Hardware-Abstraktion (Abb. 2) ist teilweise umgesetzt und getestet. Die Pin-Klassen (rosa und grün) abstrahieren den Zugriff auf die GPIOs. Die Klassen Max14819 (gelb) und Max14819_Port (orange), welche den verwendeten IO-Link-Transceiver-Chip mit seinen Ports abstrahieren, nutzen die Pin-Klassen zum plattformunabhängigen Zugriff auf die Hardware. Der Benutzer interagiert einzig mit der IOLMaster-Klasse (blau), die für den gesamten IO-Link-Master steht. Diese besitzt Objekte für das verwendete PCB (Shield_Hat in blau) mit den Chips und die IO-Link-Ports. IOLMaster erlaubt derzeit den Aufbau der IO-Link-Verbindung. Zukünftig wird es auch möglich sein pro Port Informationen über das angeschlossene Device abzufragen und ein passendes Device-Objekt zu registrieren.

Die Bibliothek ist unter dem Namen „openioliink“ auf GitHub veröffentlicht und steht unter der Apache-2.0-Lizenz, damit andere Personen einfach zu ihr beitragen und diese in ihre eigenen Projekte integrieren können.



Tobias Gammeter

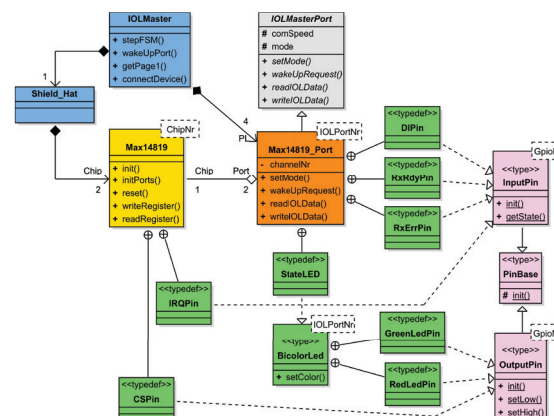


Abb. 2: Vereinfachtes Klassendiagramm der Hardware-Abstraktion