



Christoph Hälg

| | |
|--------------|-------------------------------|
| Student | Christoph Hälg |
| Examinator | Prof. Reto Boderer |
| Themengebiet | Embedded Software Engineering |

Schlanke Serialisierungsbibliothek

Schlanke Serialisierungsbibliothek für Embedded Systems

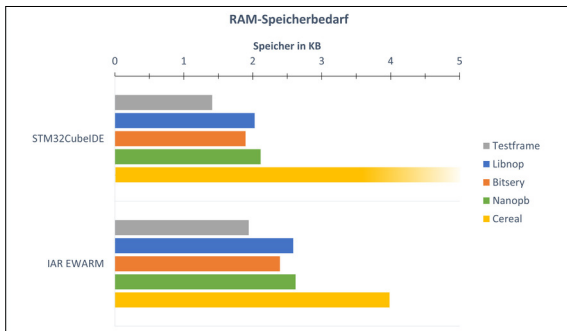


Abb. 1: Read-write Speicherbedarf der Bibliotheken inklusive Testprogramm
Eigene Darstellung

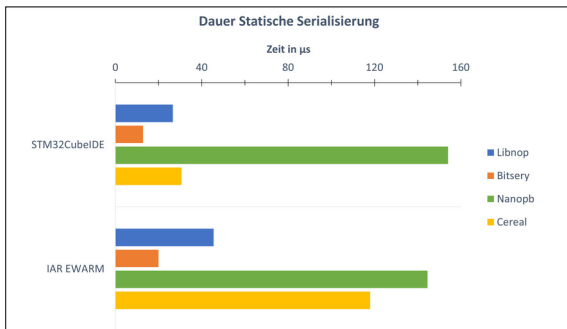


Abb. 2: Benötigte Laufzeit zur statischen Serialisierung des Testobjekts
Eigene Darstellung

Einleitung: Damit Datenstrukturen und Objekte aus einem lokalen Programm in einer Datei abgespeichert oder in einem Netzwerk übertragen werden können, müssen diese als Bytestream serialisiert, d. h. umgewandelt werden. Die Rückwandlung des Bytestreams und Rekonstruktion der Struktur wird als Deserialisierung bezeichnet. Die meisten Serialisierungsbibliotheken sind aber nicht für die Anwendung in einem Embedded-Umfeld geeignet. Gründe dafür sind etwa deren Spezialisierung auf spezifische Betriebssysteme, Abhängigkeiten von anderen, speicherintensiven Tools oder die Verwendung von für Menschen leserlichen, aber ineffizienten Serialisierungsformaten.

Ziel der Arbeit: Es sollen mehrere C++ Serialisierungsbibliotheken auf ihre Eignung zur Anwendung in einem Embedded-Umfeld überprüft werden. Da die Bibliotheken mit wenig Speicher auskommen müssen, sollten sie möglichst keine Abhängigkeiten aufweisen und Daten in kleiner Form serialisieren. Ebenfalls sollen die Einfachheit der Implementierung sowie die Laufzeit von Serialisierung und Deserialisierung bewertet werden. Das Resultat der Untersuchungen sowie dessen Beurteilung soll Embedded SW Entwicklern dabei helfen, einen begründeten Entscheid für eine der Bibliotheken fällen zu können.

Ergebnis: Von den vier vorgeschlagenen Bibliotheken erwiesen sich zwei als nicht umsetzbar mit den beiden gewählten Entwicklungsumgebungen STM32CubeIDE und IAR Embedded Workbench bzw. deren Toolchains. Diese Bibliotheken mussten durch andere ersetzt werden. In der Mehrzahl der Fälle benötigen die Bibliotheken länger zur Deserialisierung als zur Serialisierung, die Dauer hängt aber nicht merklich davon ab, ob das betreffende Objekt bereits zur Compile-Zeit bekannt ist oder nicht. Betreffend Laufzeit und Speicherbedarf konnten zwischen den Bibliotheken erhebliche Unterschiede festgestellt werden. Des Weiteren wurden zumeist bessere Resultate erreicht durch Verwendung der vom Hardwarehersteller kostenlos zur Verfügung gestellten CubeIDE. Die kostenpflichtige Software von IAR erzielte nur vereinzelt bessere Werte. Die Grösse des serialisierten Objekts hingegen ist unabhängig von der Toolchain. Auch die Einfachheit der Implementation und Verfügbarkeit von Dokumentation und Codebeispielen variiert stark von Bibliothek zu Bibliothek.

Als beste Bibliothek insgesamt schneidet Bitsery ab, gefolgt von Libnop. Nanopb benötigt am meisten Laufzeit und Cereal erfordert viel Speicher.