



Armin Kessler

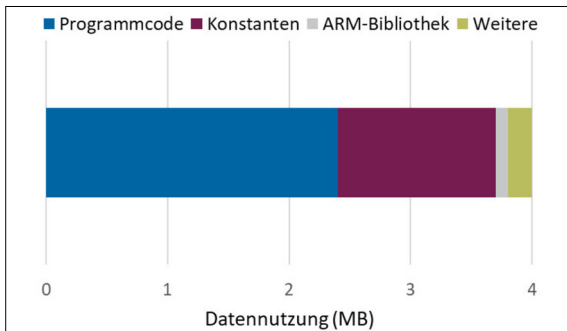


Ambroise Suter

Studenten	Armin Kessler, Ambroise Suter
Examinator	Prof. Reto Boderer
Themengebiet	Embedded Software Engineering

## Bare Metal Qt

### Ressourcenoptimierte Qt-Portierung auf einen Embedded Controller



Speicherbedarf eines minimalen Qt Programms.

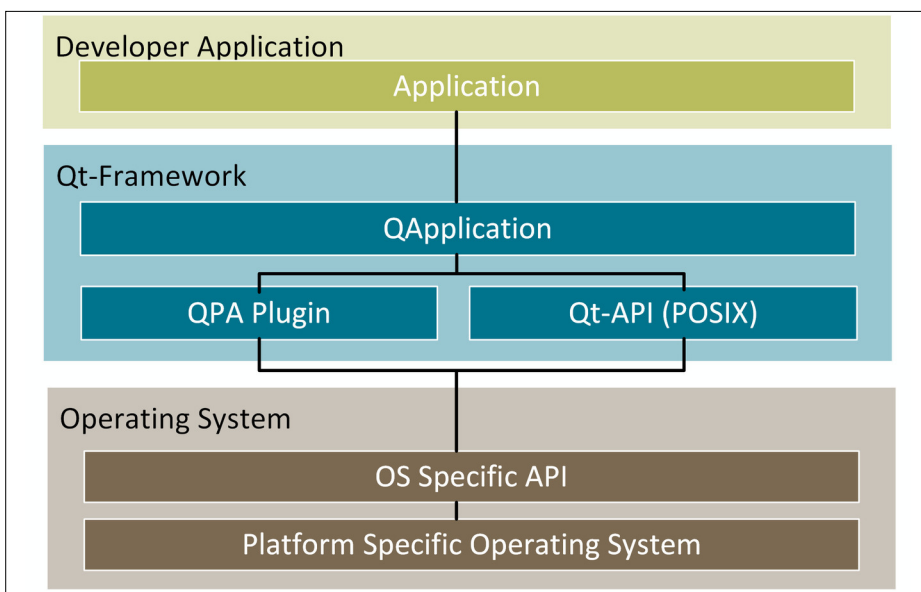
**Aufgabenstellung:** Qt ist ein sehr verbreitetes und Plattform übergreifendes Framework, das vor allem für grafische Benutzeroberflächen eingesetzt wird. Nach eigenen Angaben benötigt Qt ein Betriebssystem, um ordnungsgemäss ausgeführt werden zu können. Dies stellt jedoch bereits eine erhöhte Leistungsanforderung an ein Embedded System dar. Im Rahmen dieser Studienarbeit wird untersucht, ob und wie Qt ressourcensparend auf Bare Metal Systemen, insbesondere auf einem Cortex-M4, eingesetzt werden kann. Das Qt-Framework soll genauestens untersucht und auf ressourcensparende Möglichkeiten eingegangen werden. Aufgrund der erlangten Erkenntnisse soll ein geeignetes Betriebssystem evaluiert und ein Qt-Demonstrator erstellt werden.

**Vorgehen / Technologien:** Im ersten Teil der Arbeit wurde die Struktur von Qt untersucht. Das Ziel war, herauszufinden wie Qt seine Plattformunabhängigkeit umsetzt und ob gewisse Anforderungen an ein Betriebssystem eliminiert werden könnten. Aufgrund dieser Erkenntnisse wurde anschliessend versucht, Qt auf das nicht offiziell unterstützte Betriebssystem eCos zu portieren. Zuletzt wurden alle neuen Erkenntnisse genau dokumentiert und eine Empfehlung für das Entwickeln mit Qt auf Embedded Systems erstellt.

**Ergebnis:** Qt kann nicht ohne tieferes Eingreifen in das Qt Framework Bare Metal betrieben werden. Es wird mindestens erwartet, dass die Plattform multithreadingfähig ist, eine POSIX-konforme Schnittstelle und eine Standard C Library enthält. Weiter benötigt ein minimales GUI Programm bereits einiges an Speicher, mehr als ein Cortex-M4 bieten kann. Möchte man eine grafische Benutzeroberfläche auf einem Bare Metal Device umsetzen, lohnt es sich zuerst, nach geeigneteren Frameworks Ausschau zu halten. Möchte man dennoch Qt benutzen, sollte man auf bereits unterstützte Betriebssysteme zurückgreifen. Qt kann in der Theorie zwar auf jede Plattform mit GCC Compiler portiert werden, jedoch erschweren fehlende und teils veraltete Dokumentationen diesen Vorgang.



Analoguhr mit Qt auf dem Raspberry Pi implementiert



Aufbau des Qt-Frameworks