

Erhöhung der Ausfallsicherheit einer Mikropumpensteuerung mit Hilfe einer hierarchisch organisierten, heterogenen Controllerplattform

Verteidigung des großen Belegs

Najdet Charaf – najdet.charaf@tu-dresden.de

Dresden, 09.06.16

Gliederung

1. Motivation

2. Controllerplattform Beaglebone Black

3. ARM PRU Kommunikation

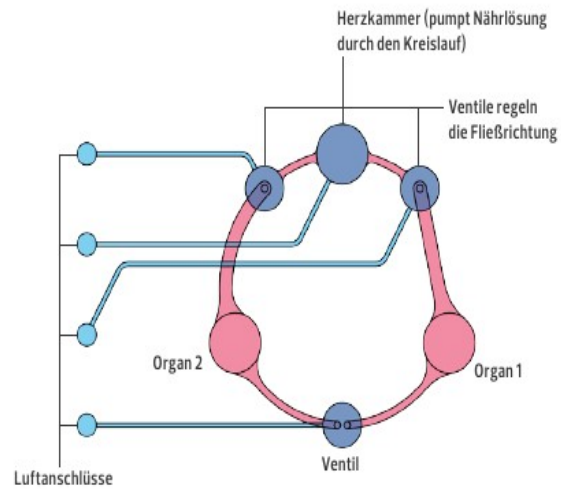
4. Mikropumpensteuerung

5. Implementierung und Evaluation

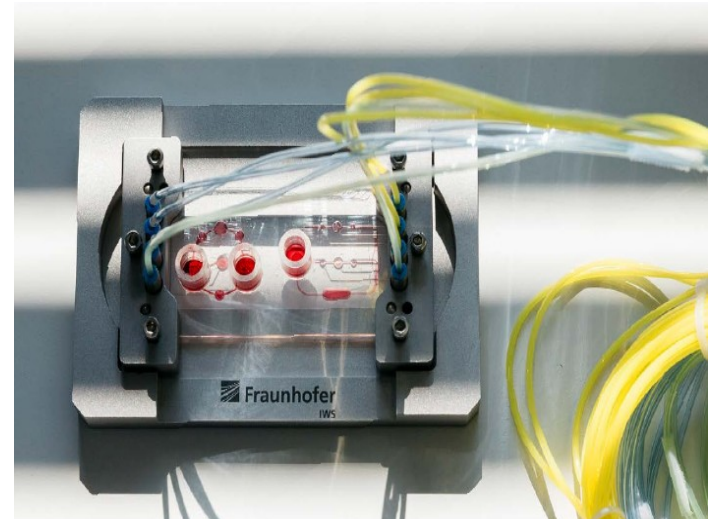
6. Zusammenfassung und Ausblick

01. Motivation

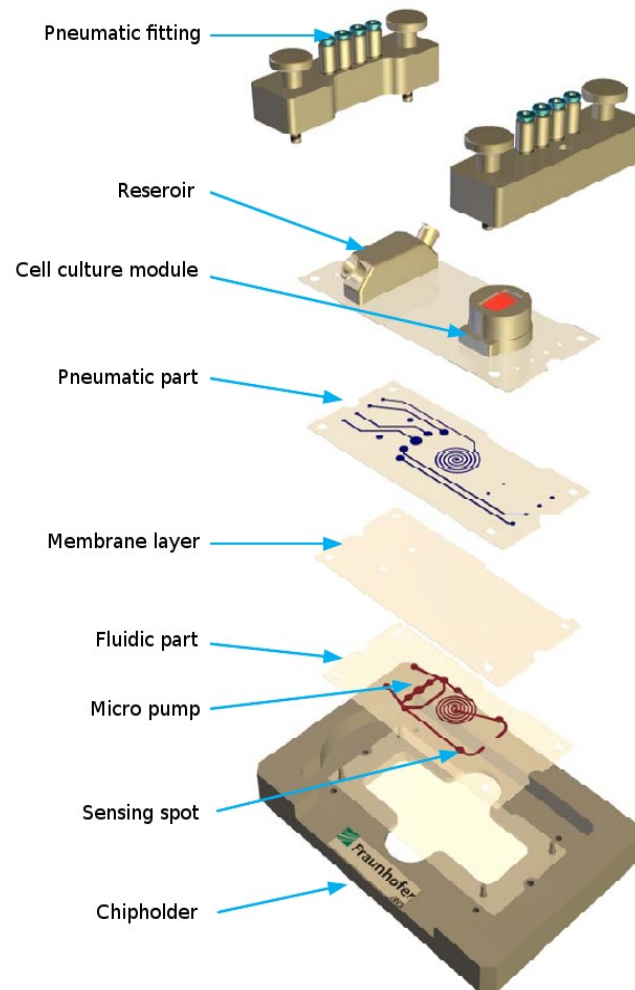
- **Lab-on-a-Chip**
 - Nachbildung biologischer Abläufe
 - Unterbringung gesamter Funktionalität eines makroskopischen Labors auf einer nur kartengroßen Plattform
 - benötigt minimale Mengen an chemische Substanzen
- **Multi-Organ-Chip (MOC)**
 - Nachbildung der im lebendigen Organismus ablaufenden Prozesse
 - Kultivierung komplexer Gewebekulturen, wie Niere, Leber und Haut, in einem Chip
 - besteht aus:
 - mehreren verschiedenen Schichten
 - feine Leitbahnen
 - Reservoiren und Mikropumpen



Schematischer Aufbau eines Organchips [Röm15]



Organchip mit zwei Blutkreisläufe [Röm15]



Eine
Einzelteildarstellung
eines MOC [SSS+16]

Anwendungsbeispiele und Ziel:

- Testen von neuen Medikamenten und kosmetischen Produkten
- MOCs sollen die Tierversuche ersetzen
- reibungslos, robust, sicher und mindestens 28 Tage lang ausfallfrei funktionieren
- Ausfallsicherheit der Mikropumpensteuerung erhöhen
- Umsetzung der Implementierung in Echtzeit

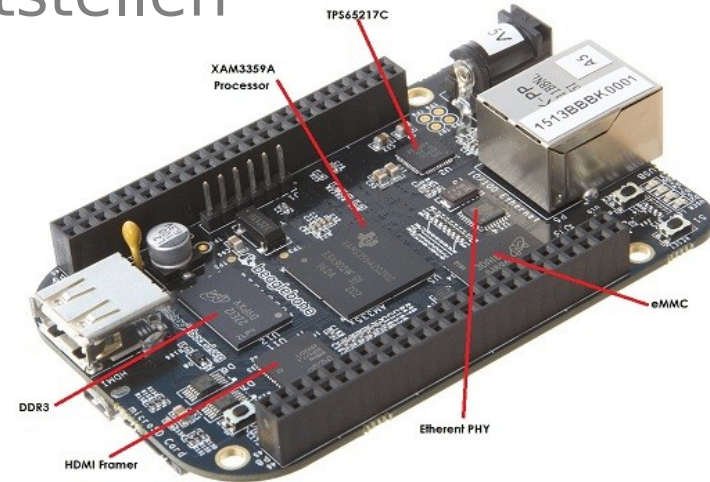
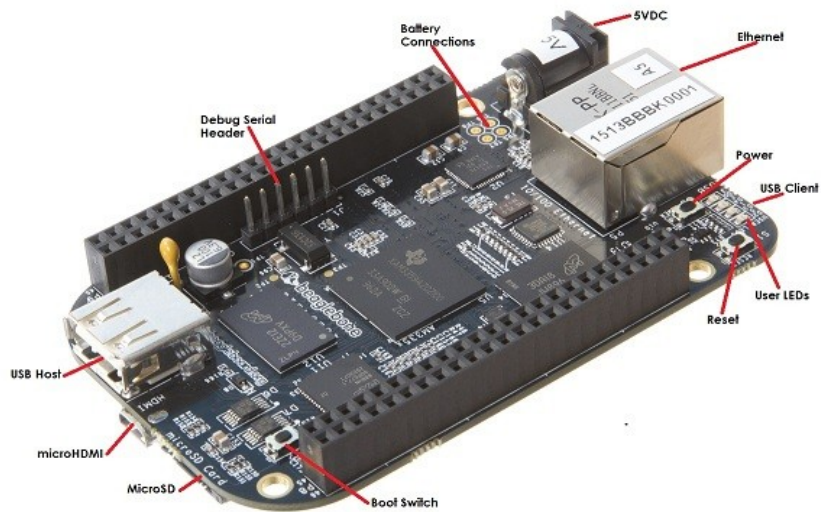
02 Controllerplattform Beaglebone Black

- Neustes Board der Beagleboard Familie von Texas Instrument
- Einplatine-Computer und Kostengünstige Entwicklungsplattform
- unterstützt primär Linux-Distributionen

Komponenten	Eigenschaften
Prozessor	Sitara Cortex A8 1 GHz
SDRAM	512 MB DDR3L 800 MHz
OnBoard Flash	4 GB 8 Bit Embedded MMC
HDMI Framer	Max. Auflösung 1280x1024
3 Tasten	Einschalter, eine Reset Taste und eine Boot-Wahl Taste

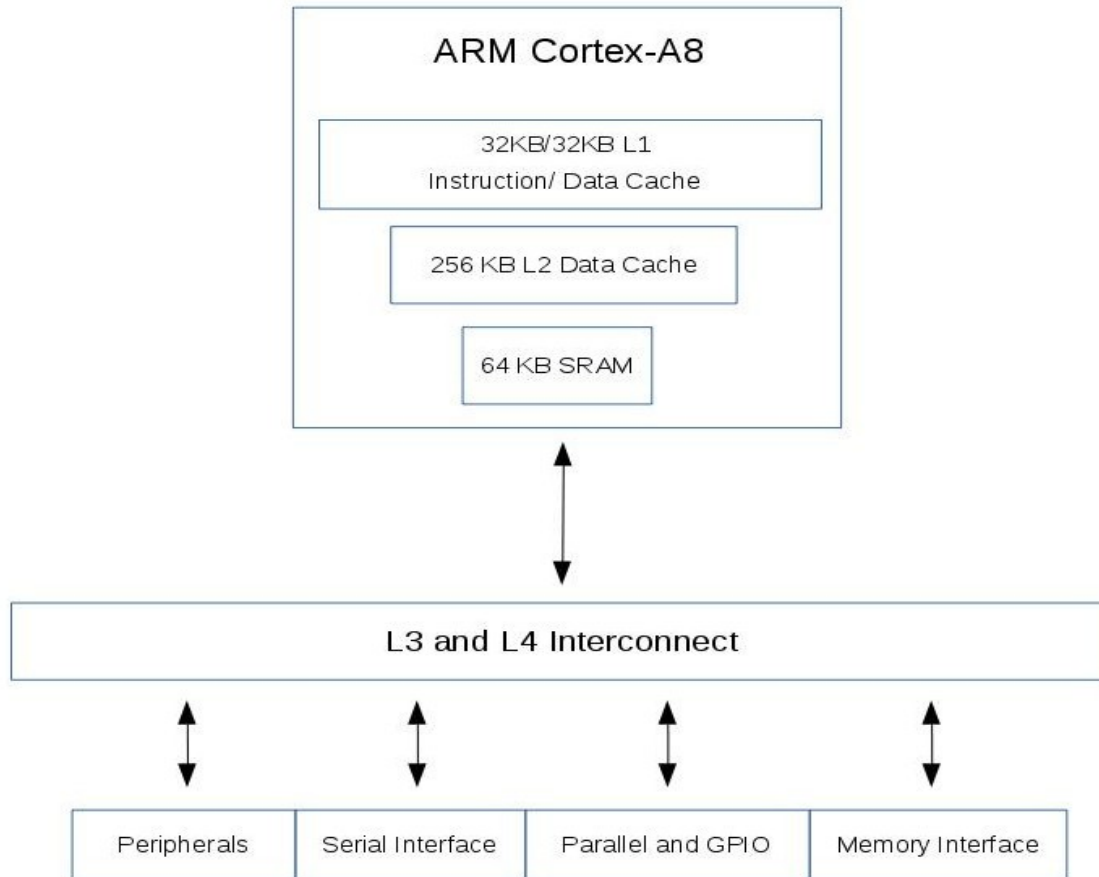
Komponenten und Schnittstellen

[A5A][RA5A]

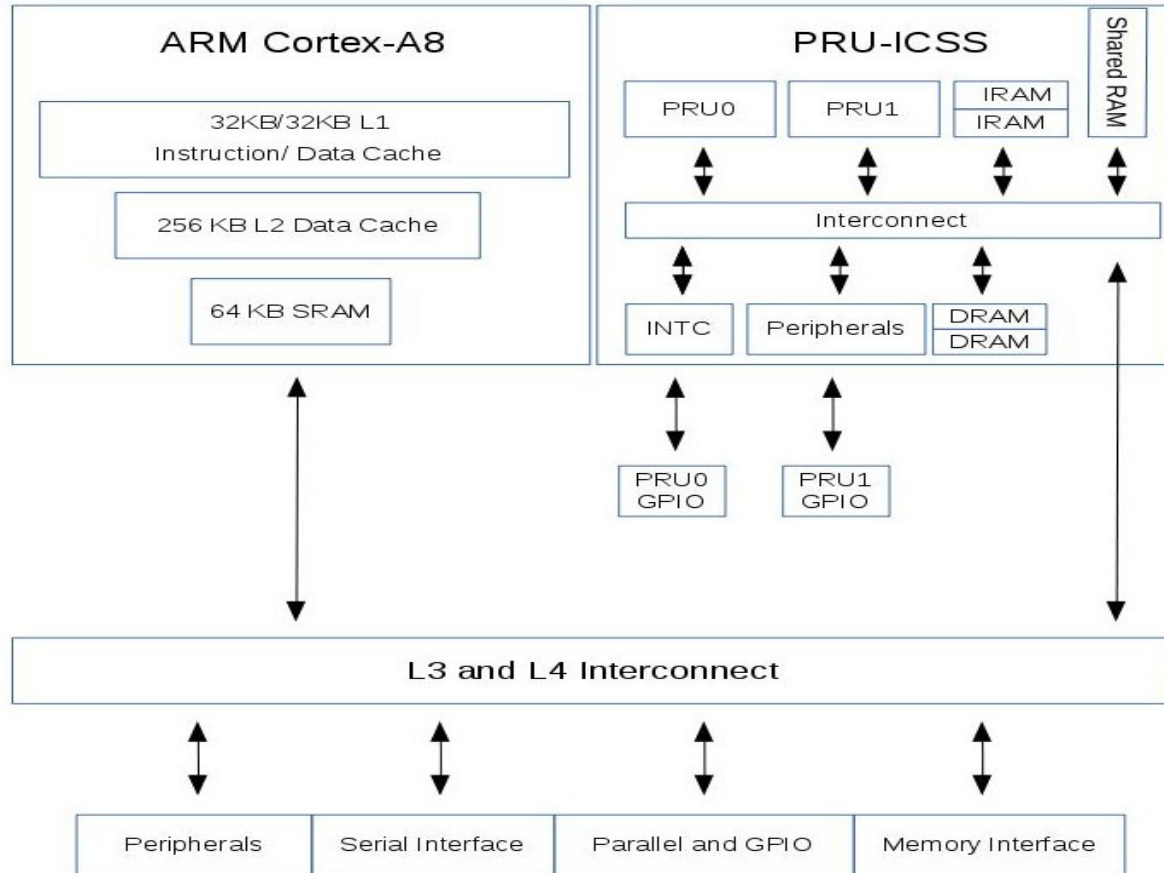


- **Besonderheiten:**
 - 2x Programmable real-time unit (PRU)
 - bietet eine große Anzahl an Pins an
 - unterstützt zahlreiche Standardschnittstellen
 - Booten von verschiedenen Medien
- **PRU-ICSS:**
 - Subsystem mit sehr geringer Latenz
 - 2x 32-bit RISC PRUs
 - 2x 200 MHz Taktfrequenz
 - 2x 8 KB Daten- und Befehlsspeicher
 - 12 KB Shared Data Speicher
 - Kein Pipelining – 1 Zyklus pro Befehl
 - Direkte Verbindung mit den GPIOs
 - Zugriff auf alle Interfaces, Speichern und Modulen auf dem ganzen SoC
 - Interrupt-Controller
 - Multiply and Accumulate Unit

ARM SoC Architektur [TI]



ARM + PRU-ICSS SoC Architektur [TI]

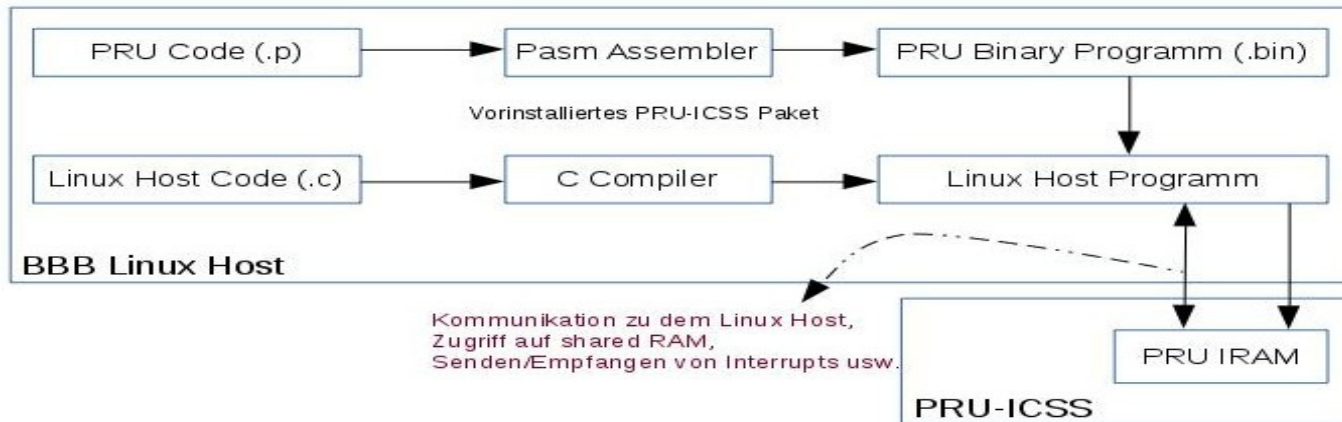


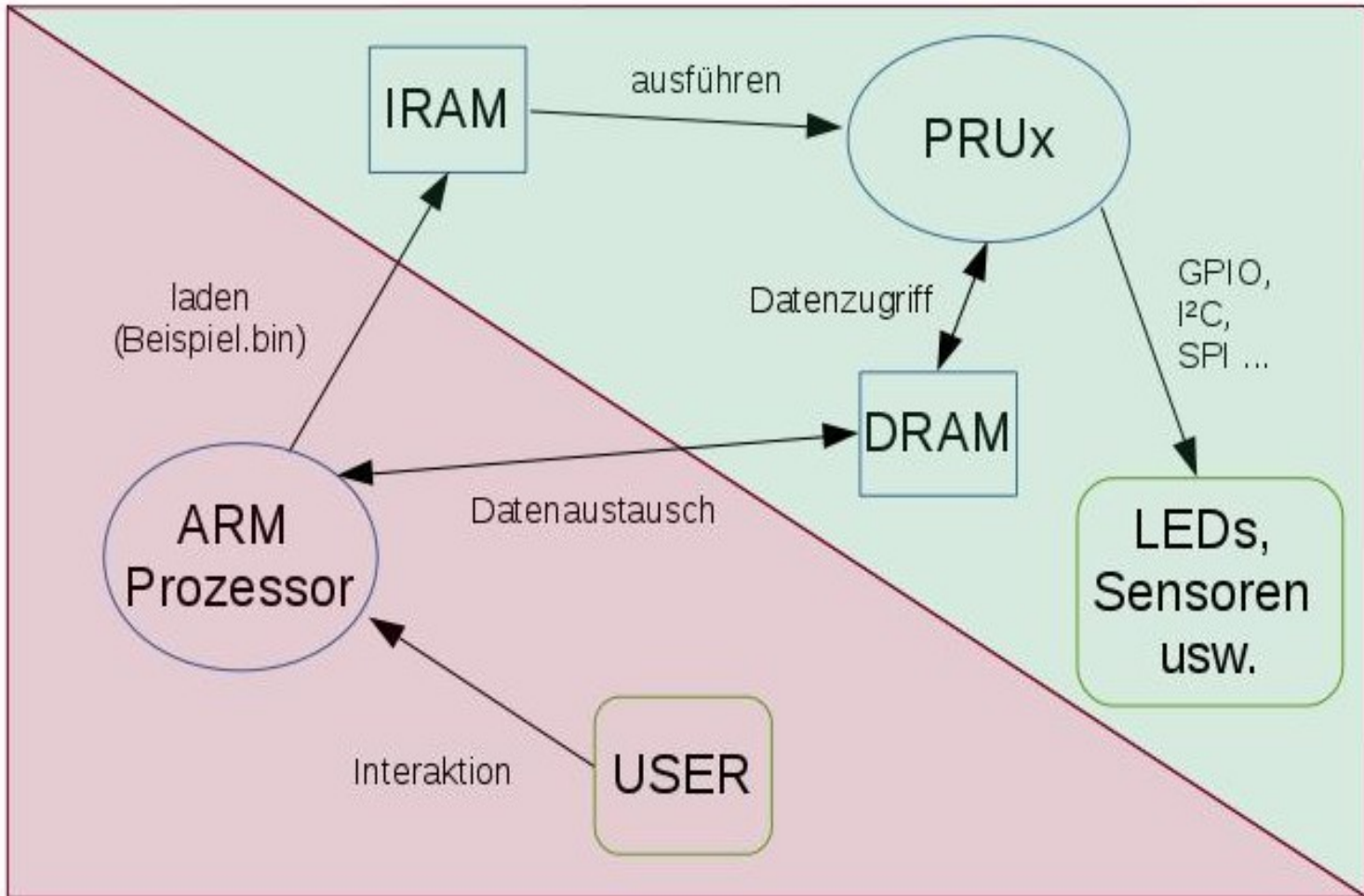
Warum Beaglebone Black ?

- Zwei unabhängige Programmierbare Echtzeit Einheiten
- 4 GB OnBoard Flash
- unterstützt zahlreiche Standardschnittstellen (I²C, SPI, CAN usw.)
- unterstützt Webbasierte Entwicklungsumgebung
- Bootable von verschiedenen Medien
- bietet eine separate Haupt- und Substeuerung
- bietet eine Hardwarelösung für die nicht-preemptive Eigenschaft des Linux Kernels

03 ARM PRU Kommunikation

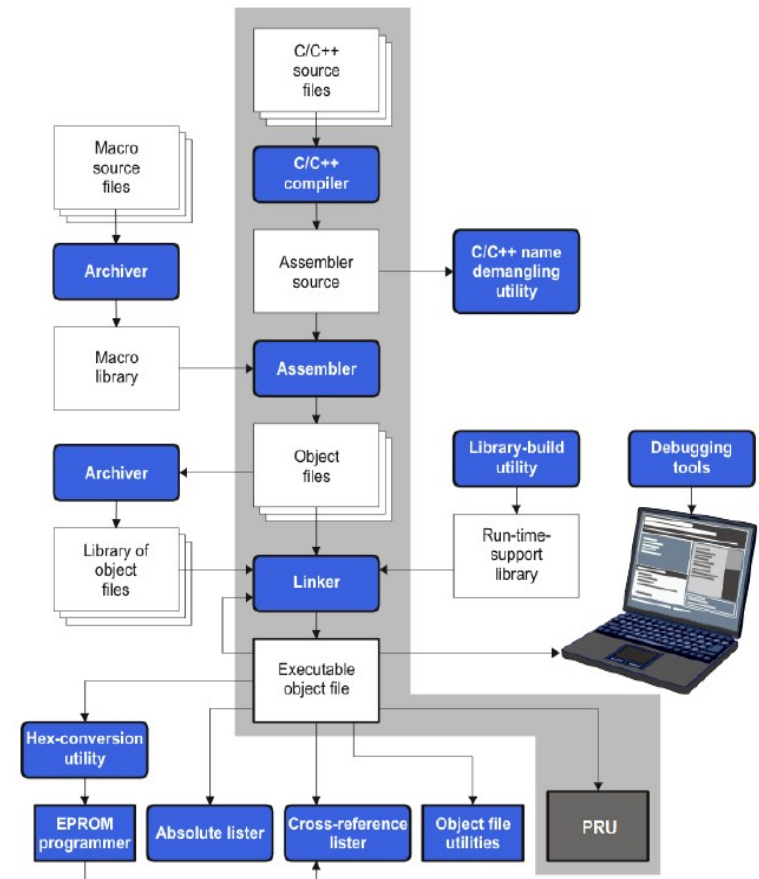
- PRU-Modul aktivieren
- Device Tree Overlay in den Kernel laden
- C/C++-Programm für den Linux Host
- Assembler-Programm für die PRU
- C-Code compilieren und Assembler-Code assemblieren





Alternative zum normalen Weg

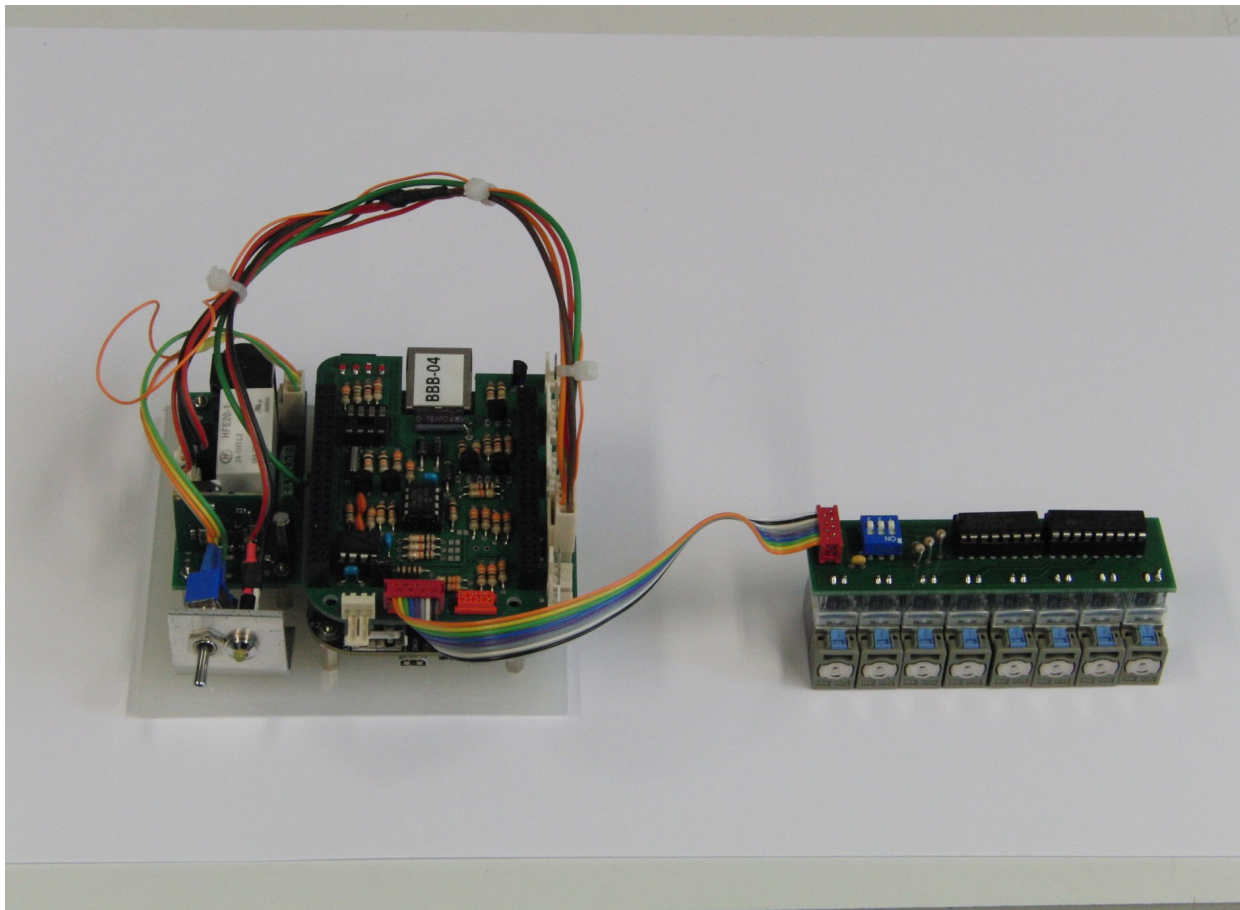
- Texas Instrument C Compiler
 - Compiliert, assembliert, optimiert und linkt in einer Anweisung
 - Syntax:
 - `cpru [options] [filenames] [--run_linker [link_options] object files]`
 - durch options das Compiler-Verhalten ändern
- Cloud9 IDE Integrierte online Entwicklungsumgebung, Node.js und BoneScript



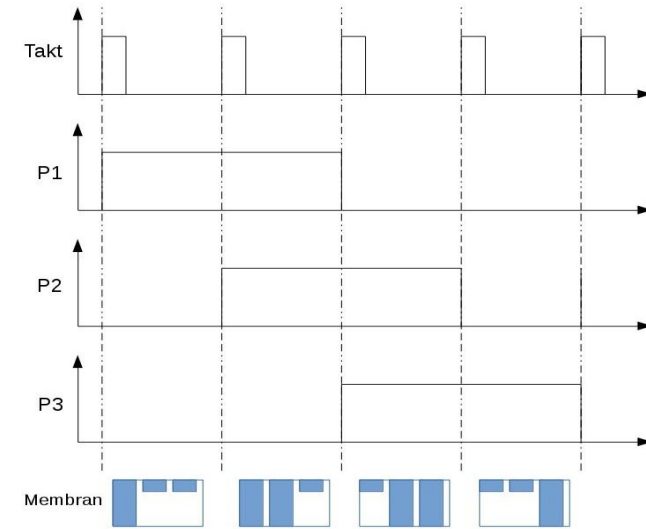
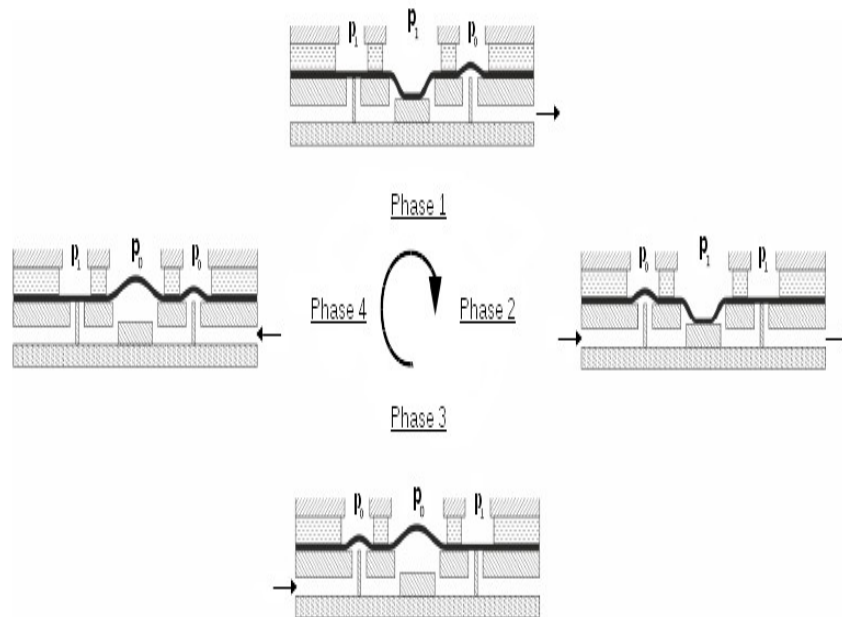
PRU Software Development Flow [TIPRU]

04 Mikropumpensteuerung

- besteht aus:
 - Controllerplattform BBB
 - von Fraunhofer angefertigtes Cape
 - Stromversorgungsinsel
 - Ventilinsel, besteht aus:
 - Expander PCF8574P
 - Darlington Transistor Arrays ULN2803A
- erzeugt einen Unterdruck oder Überdruck an der elastischen Membran
→ ist es möglich, die Flüssigkeit innerhalb des mikrofluidischen Systems zu pumpen



Pumpvorgang



Vier Phasenvorgänge [SSS+16]

05 Implementierung und Evaluation

- Device Tree Overlay in den Kernel laden
- benötigten Pins konfigurieren
- PRU-Programm, um die Mikropumpen anzusteuern
 - Zugriff auf das I²C-Interface durch den OCP Port
 - OCP Port aktivieren
 - I²C-Modul konfigurieren und Clocks synchronisieren
 - Slave-Adresse und Register Offsets des I²C-Moduls festlegen
 - I²C-Modul zurücksetzen und für die PRU bereitstellen
 - Druckwerte an den Expander schicken

05 Implementierung

- 3 Linux Host Programme,
 - 1) um die PRU aktivieren, das Memory Mapping und PRU-DRAM initialisieren und den PRU-Code in den gewünschten PRU-IRAM laden zu können
 - 2) um die Frequenz der Pumpen ändern zu können
 - 3) um die PRU und das Memory Mapping beenden und deaktivieren zu können

05 Evaluation

- Die PRU führt den Code unabhängig aus
- Der User kann die Daten in dem PRU-DRAM ändern
- Der User kann die PRU jeder Zeit beenden und deaktivieren
- Die PRU führt das Programm, trotz eines unerwarteten Absturzes des Systems, weiter aus

06 Zusammenfassung und Ausblick

- Konzipierung einer Strategie zur Implementierung
- Verwendung Linux-Distribution Debian 8.4
- Aktivierung und Modifizierung der gewünschten GIPOs
- laden das hierfür geeignete Overlay
- Die Steuerungsabläufe partitionieren in Haupt- (ARM) und Elementarsteuerung (PRU)
- Ausführung eines Forkbomb-Programms
 - ➔ Die Unabhängigkeit der Abarbeitung und die Weiterführung des PRU-Programms

06 Ausblick

- Die Ausfallsicherheit verbessern
 - ➔ bei einem Restart
- Der von Texas Instruments C Compiler verwenden
 - ➔ für komplexere Implementierungen
- Eine Standard-Library
 - ➔ reduziert Programmanzahl
 - ➔ in C- oder C++-Sprache
 - ➔ vereinfacht die Ausführung
- Assembler-Code und Linux Kernel weiter optimieren

Quellen

- [A5A] http://elinux.org/File:COMP_A5A.jpg
- [BIO] <https://www.biotechnologie.de/BIO/Navigation/DE/root,did=179124.html>
- [BBB] <http://elinux.org/Beagleboard:BeagleBoneBlack>
- [MODE] D. Molloy, Exploring Beaglbone: Tools and Techniques for building with embedded Linux. Wiley, 2015
- [RA5A] http://elinux.org/File:CONN_REVA5A.jpg
- [TI] http://processors.wiki.ti.com/images/3/34/Sitara_boot_camp_pr-module1-hw-overview.pdf
- [TIPRU] <http://www.ti.com/lit/ug/spruhv6a/spruhv6a.pdf>

Quellen

- [Röm15] G. Römer. „Der ganze Mensch auf einem Chip“. In: P.M. Magazin (Juli 2015), S. 74-79
- [SSS+16] F. Sonntag u.a., Hrsg. Universal lab-on-a-chip platform for complex, perfused 3D cell cultures. Conference „Microfluidic, BioMEMS, and Medical Microsystems“. 2016



»Wissen schafft Brücken.«