

# Briefing Thesis

Raphael Frey, Noah Hüsler

5. April 2017

## 1 Aktueller Stand

### 1.1 Evaluation Vorgängerprojekt & Red Pitaya Ecosystem

- Softwareseite des Red Pitaya-Projekts ist auf der Herstellerseite stark um Umbruch (gesamte Codebasis wird momentan neu geschrieben) und daher auch ziemlich chaotisch.
- Der alte Codebranch wird nicht mehr maintained und ist schlecht dokumentiert. Er stellt in unserer Ansicht keine zukunftsfähige Lösung dar.
- Zitat Entwickler zur neuen Codebase: “The code is still under heavy development and not stable.” [1]
- Neben der Codebase selbst ist auch die Toolchain der FPGA-Seite nicht gut dokumentiert.
- Die Dokumentation des Linux-Teils des Pitaya-Projekts scheint aber passabel.
- Wir sind uns nicht sicher, ob unsere Vorgänger wirklich jemals erfolgreich einen Bitstream auf das FGPA aufgespielt und betrieben haben.

### 1.2 Entwickeln einer sauberen Toolchain

- Die Community um den Red Pitaya ist ebenfalls nicht so zufrieden mit der Dokumentation des Projekts. Wir haben eine Person gefunden [2], die sich daran gemacht hat, hier Abhilfe zu schaffen und eine eigene Toolchain am Entwickeln ist.
- Diese Toolchain ist ein laufendes Projekt dieser Person, ist modular gehalten, einigermassen gut dokumentiert (bzw. die Struktur transparent und so aufgebaut, dass sie grösstenteils selbsterklärend ist).
- Daher sind wir nicht auf die (schlechte) Dokumentation des Pitaya-Projekts angewiesen.
- Ebenfalls sind wir nicht auf das Fortbestehen des Pitaya-Projekts angewiesen (es scheint etwas unsicher, ob das Projekt langfristig überlebensfähig).

**Folgerung:** Aufgrund des ungeordneten Zustandes des Pitaya-Projekts (insbesondere der FPGA-Codebase) scheint uns eine Grundlage für unser Projekt wichtig, welche nicht mit dem Erfolg oder Misserfolg des Pitaya-Projekts selbst steht oder fällt.

- Wir haben uns daher entschieden, aufbauend auf [2] unsere eigene Toolchain zu entwickeln. Dies sollte es uns erlauben:
  - Nicht vom Wohlwollen der Pitaya-Entwickler abhängig zu sein.
  - Unsere Toolchain gründlich zu dokumentieren und unseren Nachfolgern eine saubere Grundlage sowohl in technischer wie auch in dokumentarischer Hinsicht liefern zu können.
- Wir haben uns mit der Entwicklungsumgebung für den Zynq vertraut gemacht (Vivado, TCL, ARM Linux).
- Wir haben TCL-Scripts von [2] an unsere Bedürfnisse angepasst und ein Blockdesign soweit fertig, um Daten vom ADC auslesen zu können.
- Um dieses Blockdesign auf dem FPGA zum Laufen zu bringen, müssen wir ein zugehöriges Kernelmodul auf dem Linux des Red Pitaya installieren. Dies erfordert einen Rebuild des ARM-Linux des Pitaya. Dies ist unser Momentanes Arbeitspaket (“Clean Linux Build for Pitaya” unten im Zeitplan).

## 2 Weiteres (geplantes) Vorgehen

Siehe Zeitplan.

## Literatur

[1] <https://github.com/RedPitaya/RedPitaya/issues/107>

[2] <https://pavel-demin.github.io/red-pitaya-notes/>

