

Ruby Defined Radio

Draft

Abstract

This document contains a draft for the presentation "Ruby Defined Radio" hold to be at the annual RubyWorld Conference on 8. November 2012 in Matsue, Japan.

Topic

In the last 7 years we deployed an unique radio communication system in more than 15 mass transit lines world-wide. During our ongoing process of building a next-generation radio for signalling systems, we evaluated the concept of a Software-Defined Radio. This presentation will cover the idea of a Software Defined Radio, why we are interested in it and where Ruby has its place in all of it.

Slides

The presentation will consist of 19 slides which will be used to visualize the topic described by the presenter. The presentation slides won't contain all information described by the presenter. Only the main idea and necessary graphics for visualization will be shown on the slides. The presentation will be split up into the following five main parts:

1. Introduction

Slides: 1 to 3

Approx. time: 2 minutes

Content: Introduction of the presenter and the topic.

2. Past

Slides: 4 to 9

Approx. time: 7 minutes

Content: A past project will be described in which the high frequency radio part of a standard hardware platform was modified by software to improve certain features.

3. Present

Slides: 10 to 13

Approx. time: 7 minutes

Content: The concept of software defined radio will be explained. Current state of the art hardware and software will be presented.

4. Future

Slides: 14 to 17

Approx. time: 7 minutes

Content: Explanation of current work to improve the current usage of software defined radios will be presented. I.e. using Ruby to program hardware (FPGAs).

5. Summary (including Q&A)

Slides: 18 to 19

Approx. time: 7 minutes

Content: The past (few SDR systems exist), present (increase of SDR) and future (SDR will be ubiquitous) will be summarized. The use of Ruby is emphasised.

The following headlines are describing each slide of the presentation with its purpose and the content.

Slide 1 (Headline)

Purpose:

Rough information to the audience about the topic.

Content:

- Title of presentation: Ruby Defined Radio
- Name of presenter: Daniel Bovensiepen

Slide 2 (Presenter Introduction)

Purpose:

Give a short overview of the presenter.

Content:

- Name of presenter: Daniel Bovensiepen
- Occupation: Lead Engineer at Siemens Limited China
- Experience: Using Ruby for 12 years. Working as a software engineer for 11 years and designing high frequency radio systems for 5 years.

Slide 3 (Topic Introduction)

Purpose:

Describes the field in which the presenter is working.

Content:

- Building Signaling Systems for Metros
- These Metro Trains need to communicate
- The presenter is building the radio communication channel between every train to ensure the safety of the complete Metro Line

Slide 4 (Roaming Problem)

Purpose:

Describes the problem of roaming.

Content:

- Moving objects have to roam between different systems
- Roaming introduces unpredictable effects
- We would like to avoid roaming

Slide 5 (Remove Roaming)

Purpose:

An idea to avoid roaming by modifying standard radio hardware with software.

Content:

- By modifying the 802.11 standard it is possible to remove roaming
- This can be done by using standard 802.11 hardware

Slide 6 (Proof Of Concept)

Purpose:

A proof of concept should present the feasibility of this idea.

Content:

- Creating a proof of concept
- Lots of expensive hardware would normally be necessary

Slide 7 (Reverse Engineering of Hardware)

Purpose:

Searching for an inexpensive proof of concept by reverse engineering existing hardware.

Content:

- Findings of existing inexpensive hardware
- Hardware specification and software support

Slide 8 (Implement A Framework)

Purpose:

Implement a software framework to test the proof of concept on an inexpensive piece of hardware.

Content:

- Analyze hardware and replace existing software platform
- Cross Compile Linux System
- Usage of alpha version of mruby on MIPS platform
- Implementing a binding to libpcap

Slide 9 (Summary of Proof Of Concept)

Purpose:

What was the outcome of using inexpensive hardware for the proof of concept?

Content:

- By using inexpensive hardware - a proof of concept could be created
- Implementing radio logic partly in software made this possible

Slide 10 (Interest in Software Defined Radio)

Purpose:

Emphasise the possibilities of Software Defined Radios

Content:

- Manipulating radio behavior in software provides an inexpensive way to experiment with radio algorithm
- By skipping the need of new hardware revisions for each new real-life test, an iterative software development approach can be applied to developing radio systems

Slide 11 (Standard Radio System Architecture)

Purpose:

How does a normal radio system looks like?

Content:

- Signaling processing is done completely in hardware
- Very small radio parts are implement in software (i.e. high protocol layer as firmware)

Slide 12 (Software Defined Radio Architecture)

Purpose:

How does a software defined radio system looks like?

Content:

- Primitive Signaling Processing done in Hardware (i.e. FPGAs)
- Complex Signaling Processing done in hardware near language (Assembler or C)
- Protocol definition in high languages like Python or Ruby.

Slide 13 (Current Use Cases)

Purpose:

Where are software defined radios used today?

Content:

- military usage
- researching of new receiving and transmission algorithmen
- LTE

Slide 14 (Performance)

Purpose:

Currently performance need of Software Defined Radio systems are too high for software-only solution.

Content:

- The limit of software programs to solve high frequency radio issues
- Too much data
- Too high real time requirements
- Expensive hardware necessary

Slide 15 (Protocol Layer)

Purpose:

Describes the highest layer of a software defined radio which can already be handled by Ruby.

Content:

- Combining Signal Processing routines to process data
- Implement a high level communication protocol (i.e. for passing ethernet frames)

Slide 16 (Complex Signal Processing)

Purpose:

More complex signal processing which is currently be handled by hardware

Content:

- Real Time requirements
- Problem of garbage collection
- Using generated C code of mruby and include into C programs

Slide 17 (Primitive Signal Processing)

Purpose:

Primitive functions for signal processing which have to be applied to a big amount of data.

Content:

- Done in FPGAs
- Not many developer can program VHDL or similar languages
- Work in progress to generate VHDL by a Ruby DSL which can be executed in Software and Hardware

Slide 18 (Past Present Future)

Purpose:

Summarize the content of the presentation.

Content:

- Past (why are we interested in SDR?)
- Present (what is currently possible with SDR?)
- Future (what do we want to achieve with SDR?)

Slide 19 (Final)

Purpose:

Closing with the perspective that Ruby could glue future technologies together. Ask the audience for questions and feedback.

Content:

- Ruby Is The Glue For The Future