TECHNISCHE
UNIVERSITÄT
DARMSTADT

ANSIAN - ANDROID SIGNAL ANALYZER

MATTHIAS KANNWISCHER

Secure Mobile Networking Project Documentation

March 9, 2017

Secure Mobile Networking Lab
Department of Computer Science

AnSiAn - Android Signal Analyzer
Secure Mobile Networking Project Documentation

Submitted by Matthias Kannwischer
Date of submission: March 9, 2017

Advisor: Prof. Dr.-Ing. Matthias Hollick
Supervisor: Jiska Classen

Technische Universität Darmstadt
Department of Computer Science
Secure Mobile Networking Lab

ABSTRACT

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

AnSiAn	Android Signal Analyzer
BPSK	Binary Phase Shift Keying
FM	Frequency Modulation
QAM	Quadrature amplitude modulation
RDS	Radio Data System

1 INTRODUCTION

Android Signal Analyzer (AnSiAn) is an Android application that has been initially developed by Steffen Kreis and Markus Grau at the Secure Mobile Network Lab (SEEMOO) in 2015. It has been extended and improved by Dennis Mantz and Max Engelhardt in summer 2016. The application is designed to allow the user the explore, capture, demodulate and decode radio frequency signals on a broad range of frequencies easily. The application supports low cost receive-only RTL-SDR dongles, but also more advanced devices like HackRF One, rad1o and PlaySDR which are capable of receiving and transmitting signals.

1.1 MOTIVATION

This project further extends and improves the current implementation of AnSiAn. Since some of the supported devices are capable of transmitting signals, the focus of this project is to properly implement the transmission of signals. The most recent version of AnSiAn does implement some prototyped transmission of Morse Signals, but there are still some issues that need to be addressed and the implementation of the transmission chain is currently very limited and needs to be generalized for the transmission of more complex signals. Additionally this projects tries to further improve the performance of AnSiAn and to fix existing issues.

1.2 FEATURE PLANNING

The project is implemented by only one developer and therefore the feature scope needs to be defined accordingly. Since signal processing on Android often causes unforeseen issues, it has been decided to separate required features and optional (stretch) features to be able to plan enough time buffer for unplanned activities without failing to implement crucial functionality.

The following features are currently planned to be implemented during this project:

1. **Implement Transmission Chain**: Currently AnSiAn only implements a DummyTransmissionChain with very limited functionality. The conclusion of the previous project documentation stated that it is required to entirely reimplement the transmis-

sion chain to enable various use cases and modulations to use it [2].

2. **Radio Data System (RDS) Transmission**: RDS is a specification on how to transmit additional information (like the stations name) alongside conventional FM radio broadcasts. The current version of AnSiAn already supports the demodulation and decoding of RDS signals. This functionality should be extended to being able to transmit own RDS data packets together with an FM modulated audio signal.

3. **Binary Phase Shift Keying (BPSK) Demodulation Improvements**: AnSiAn is capable of demodulating BPSK signals, which is already used for RDS and PSK31. However the developers faced some performance issues with the current implementation and suggested that this should be fixed in future releases. The improvement alternatives should be evaluated and implemented as a part of this project.

4. **Walkie-Talkie-Mode**: The AnSiAn should be extended for an Walkie-Talkie functionality. It should be possible to use two smartphones with AnSiAn and a HackRF each as Walkie-Talkies. Therefore AnSiAn should constantly receive and demodulate on a specified frequency and the user should then be able to quickly switch into transmission mode to send an own audio signal recorded from the included microphone. The time to switch to transmission mode and then back to reception mode should be as short as possible.

5. **Optional: Quadrature amplitude modulation (QAM) Demodulation**: AnSiAn already implements BPSK, which is a special case of QAM. If there is time left at the end of this project this implementation could be extended to support 4-QAM, 16-QAM, 64-QAM and 256-QAM. This could for example be used to implement 802.11 in the future.

1.3 SPRINT PLANNING

The project will be implemented by a single developer. Therefore the time available is 270 hours distributed over the entire winter term. We tried to estimate the required time for all features by splitting each feature up into consecutive tasks:

- **Project Initiation Phase [20h]**
- **Meetings with supervisor, final presentation [10h]**
- **Feature 1: Implement Transmission Chain [50h]**
  - Task 1: Investigation of Existing Code [5h]

  - Task 2: Rewrite Modulator [10h]
  - Task 3: Implementation Interpolator [10h]
  - Task 4: Refactor Existing Code and Finalize [5h]
  - Task 5: Integration with already implemented transmission modes [10h]
  - Task 6: Regression Testing [10h]
- **Feature 2: RDS Transmission [35h]**
  - Task 7: Investigation of Existing Code and Specification [5h]
  - Task 8: Implementation in MATLAB [5h]
  - Task 9: Portation to Java [15h]
  - Task 10: Testing and Bug Fixing [10h]
- **Feature 3: BPSK Demodulation Improvements [55h]**
  - Task 11: Investigate Existing Code [5h]
  - Task 12: Research Improvement Alternatives [5h]
  - Task 13: Implement Improvement [15h]
  - Task 14: Analyze Performance Increase [15h]
  - Task 15: Testing and Bug Fixing [10h]
  - Task 16: Regression Testing [5h]
- **Feature 4: Walkie-Talkie-Mode [60h]**
  - Task 17: Design and implement UI [15h]
  - Task 18: Implement AM [15h]
  - Task 19: Implement FM [10h]
  - Task 20: Implement SSB [10h]
  - Task 21: Testing and Bug Fixing [10h]
- **Optional: Feature 5: QAM Demodulation [ca. 50h]**
- **Documentation [25h]**
- **Presentation Preparation [15h]**

Excluding the optional feature this sums up to 270 hours. To distribute the work equally over the entire winter term, we assigned each feature to one of the predefined submission dates:

- **Alpha Release - 22.12.2016**
  - Feature 1
  - Feature 2
- **Beta Release - 02.02.2017**

  - Feature 3
  - Feature 4

- **Final Release - 09.03.2017**
  - Documentation
  - Extensive Testing
  - Bug Fixes
  - Optional: Feature 5

1.4 DETAILED PROJECT PLAN

In this project we try to apply agile software development methods to reduce organizational and planning overhead. Therefore we decompose each submission into 3 sprints of 2 to 3 weeks. After each sprint the achieved progress should be reviewed and the the plans for the next sprints should be adjusted.

We tried to assign each task to one of the sprints. However this is not a static assignment and it must be reviewed and adjusted regularly. The preliminary planning currently is:
Sprint 1 04.11.2016-20.11.2016 [2,5 weeks] - Task 1,2,3,4 [30h]
Sprint 2 21.11.2016-04.12.2016 [2 weeks] - Task 5,6,7 [25h]
Sprint 3 05.12.2016-25.12.2016 [3 weeks] - Task 8,9,10 [30h]
////////// ALPHA RELEASE
Sprint 4 26.12.2016-08.01.2017 [2 weeks] - Task 11,12,13,14,15 [50h]
Sprint 5 09.01.2017-22.01.2017 [2 weeks] - Task 16,17,18 [35h]
Sprint 6 23.01.2017-05.02.2017 [2 weeks] - Task 19,20,21 [30h]
////////// BETA RELEASE
Sprint 7 06.02.2017-19.02.2017 [2 weeks] - Doc and Bug Fixes
Sprint 8 20.02.2017-05.03.2017 [2 weeks] - Doc and Bug Fixes
Sprint 9 06.03.2017-19.03.2017 [2 weeks] - Presentation
////////// FINAL RELEASE

To track the time spend on each task we use an online tool called Agilefant, which can be used to generate charts and to compare the expected and the actual time required to implement a feature.

1.5 SOURCECODE

For development we use a git repository on [1] which has been forked from the last version of Dennis Mantz and Max Engelhardt. All changes (including the documentation) will be available there.

1.6 TESTING

Testing is done on a Samsung Galaxy S6 and a Samsung Galaxy S6 Edge with Android 6.0.1. For transmission and reception we use two

HackRF Ones connected to either one of the smartphones or a linux computer with gqrx. For regression tests we will also use a RTL-SDR (RTL2832U). For the final tests we will also use some other Android smartphones.

2 DESIGN

The following chapter describes the design of the components required to implement the specified features from an architectural view. Each section represents one feature and illustrates which components will be developed.

- **Feature 1: Implement Transmission Chain**: Figure 1 illustrates the current implementation of the transmission chain in AnSiAn. This needs to be refactored and extended to a full transmission chain as illustrated in Figure 2.
- **Feature 2: RDS Transmission**: Figure 3 shows the required components for the RDS transmission feature. We need to be able to feed RDS data and audio into the modulator. Then we need to implement the construction of an RDS signal, which uses Frequency Modulation (FM) for the audio and BPSK for the additional information. As BPSK modulation is already implemented, this can be reused. Additionally we need to extend the UI to enable the user to define which RDS data should be transmitted.
- **Feature 3: BPSK Demodulation improvements**: Figure 4 shows the context of the BPSK Demodulation class which is used by RDS and PSK31. This class should be refactored and improved.
- **Feature 4: Walkie-Talkie-Mode**: This feature consists of UI changes and transmission changes. The UI should provide the functionality to select the Walkie-Talkie-Mode, listen to a specific frequency and enable/disable the transmission at any time. The transmission should then use the audio stream from the microphone, modulate it with a user-defined modulation scheme and transmit it using the HackRF. Figure 5 depicts the components required for the transmission part. The UI changes are straightforward.

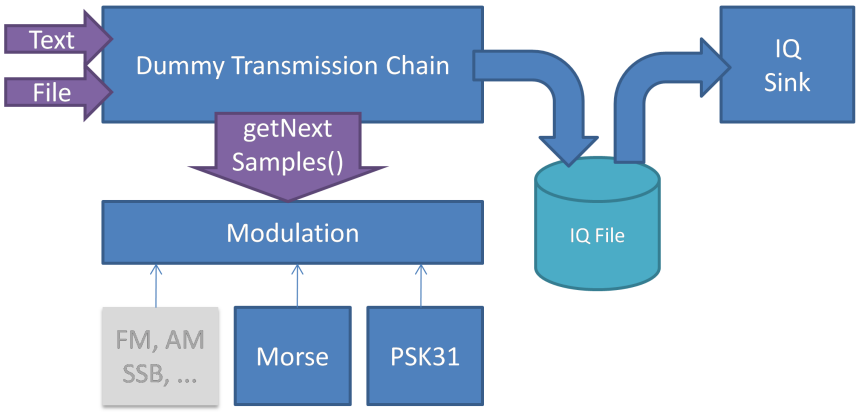

Figure 1: Transmission in current version of AnSiAn [2]

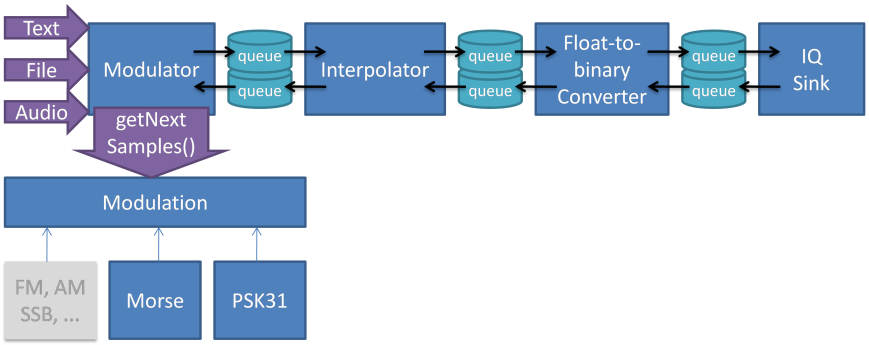

Figure 2: Final transmission chain as suggested by [2]

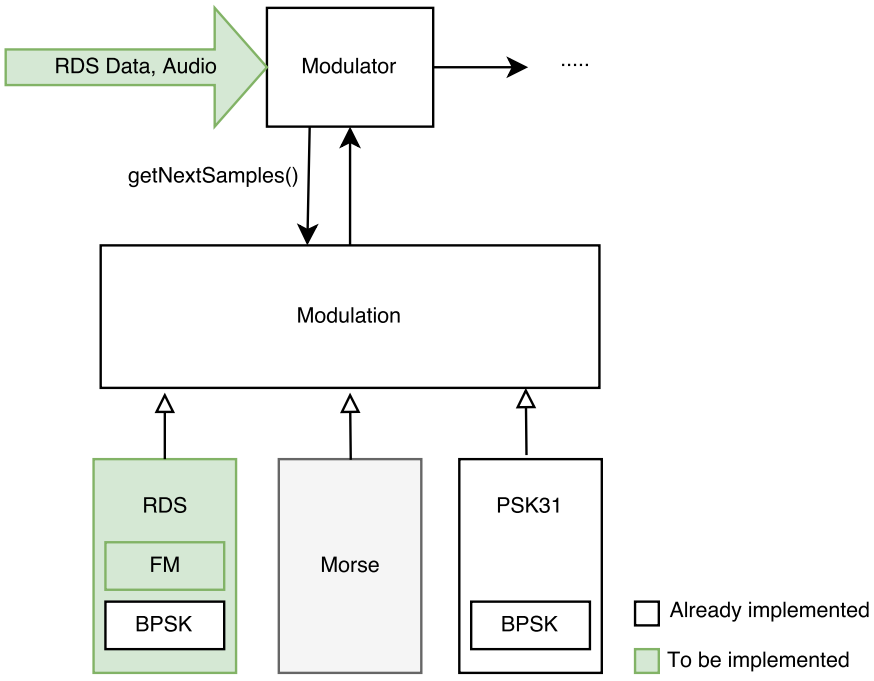

Figure 3: Required Components for RDS Transmission

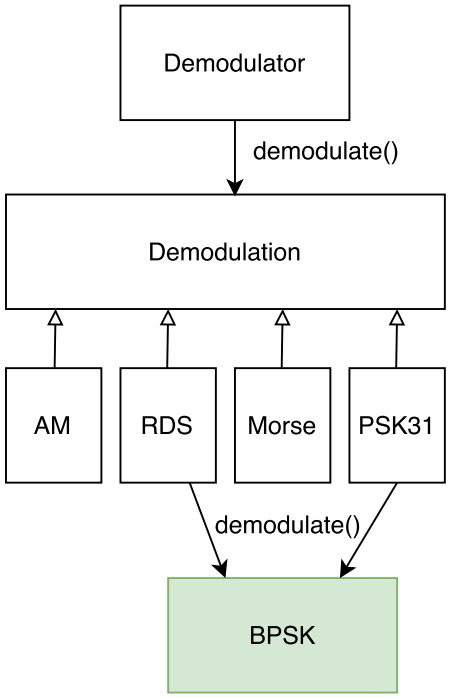

Figure 4: Required Components for BPSK Demodulation Improvements

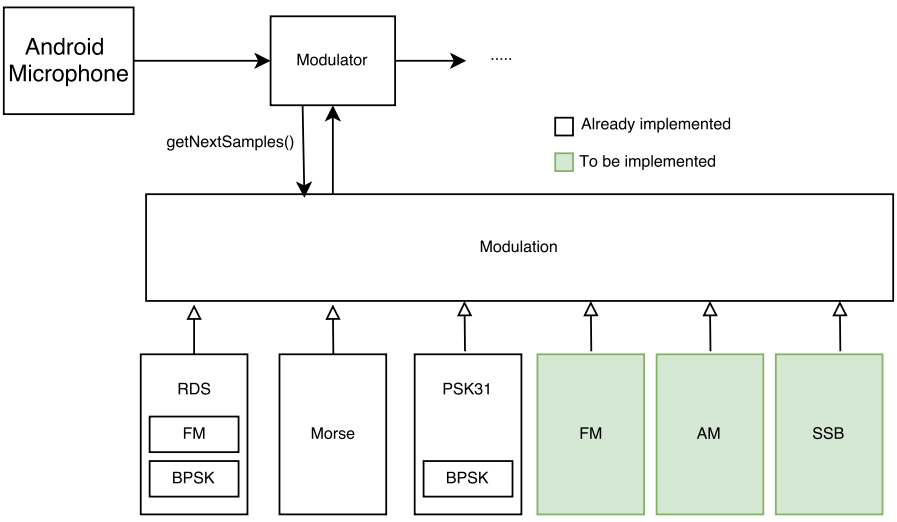

Figure 5: Required Components for Walkie-Talkie-mode

BIBLIOGRAPHY

[1] *Android Signal Analyzer Sourcecode*. https://github.com/matze765/AnSiAn.

[2] D. Mantz and M. Engelhardt. *AnSiAn - Android Signal Analyzer Documentation*. Secure Mobile Networking Lab, TU Darmstadt, 2016.

ERKLÄRUNG

gemäß § 22 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich die vorliegende Secure Mobile Networking Project Documentation ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. In der abgegebenen Arbeit stimmen die schriftliche und elektronische Fassung überein.

Darmstadt, 9. März 2017

Matthias Kannwischer