GSoC'14:GSM Analysis Tool

Zhenhua HAN
hzhua201@gmail.com

Mentor: Sylvain Munaut

March 15, 2014

Contents

1 Introduction

GSM (Global System for Mobile Communications) is the most used standard in mobile communication. GNU Radio is a free software development toolkit that provides the signal processing runtime and processing blocks to implement software radios using readily-available, low-cost external RF hardware and commodity processors.[1] This proposal addresses ideas to create an out of tree module for GNU Radio to decode GSM signals as suggested by Sylvain Munaut. Section 2 is the general description of the GSM physical layer as this project mainly focuses on decoding GSM signal. Section 3 is the deliverables. A rough project schedule is given in Section 4. Section 5 are the expanding tasks I plan to do after GSoC'14. My personal statements are in Section 6.

2 Background of GSM Signal

GSM (Global System for Mobile Communications) is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe protocols for the second generation (2G) digital cellular networks used by mobile phones. According to the GSM Association, it is approximated that 80 percent of the world use GSM when placing wireless calls.

2.1 Multiple access and time slot structure

The access scheme of GSM is Time Division Multiple Access (TDMA) with 8 basic physical channels per carrier. The carriers are separated by 200kHz.

Figure 1 shows the time slot structure and bursts. The longest recurrent time period is called hyperframe which has a duration of 3h 28mn 53s 760ms. One hyperframe contains 2048 superframes which have a duration of 6.12 seconds. A superframe contains 26-frame multiframe (51 per superframe) or 51-frame multiframe(26 per superframe). One (26-frame) multiframe contains 26 TDMA frames and one (51-frame) multiframe contains 51 TDMA frames. A TDMA frame contains 8 time slots. A time slot is the basic radio resource of GSM which lasts about 576.9 μs (15/26 ms) and 156.25 bit durations, and its physical content is called a burst. There are four types of bursts in GSM.

- normal burst (NB): This burst is used to carry information on traffic and control channels.
- frequency correction burst (FB): This burst is used for frequency synchronization of the mobile.
- synchronization burst (SB): This burst is used to time synchronization of the mobile.
- access burst (AB): This burst is used for random access from a mobile which does not know the timing advance at the first access (or after the handover).

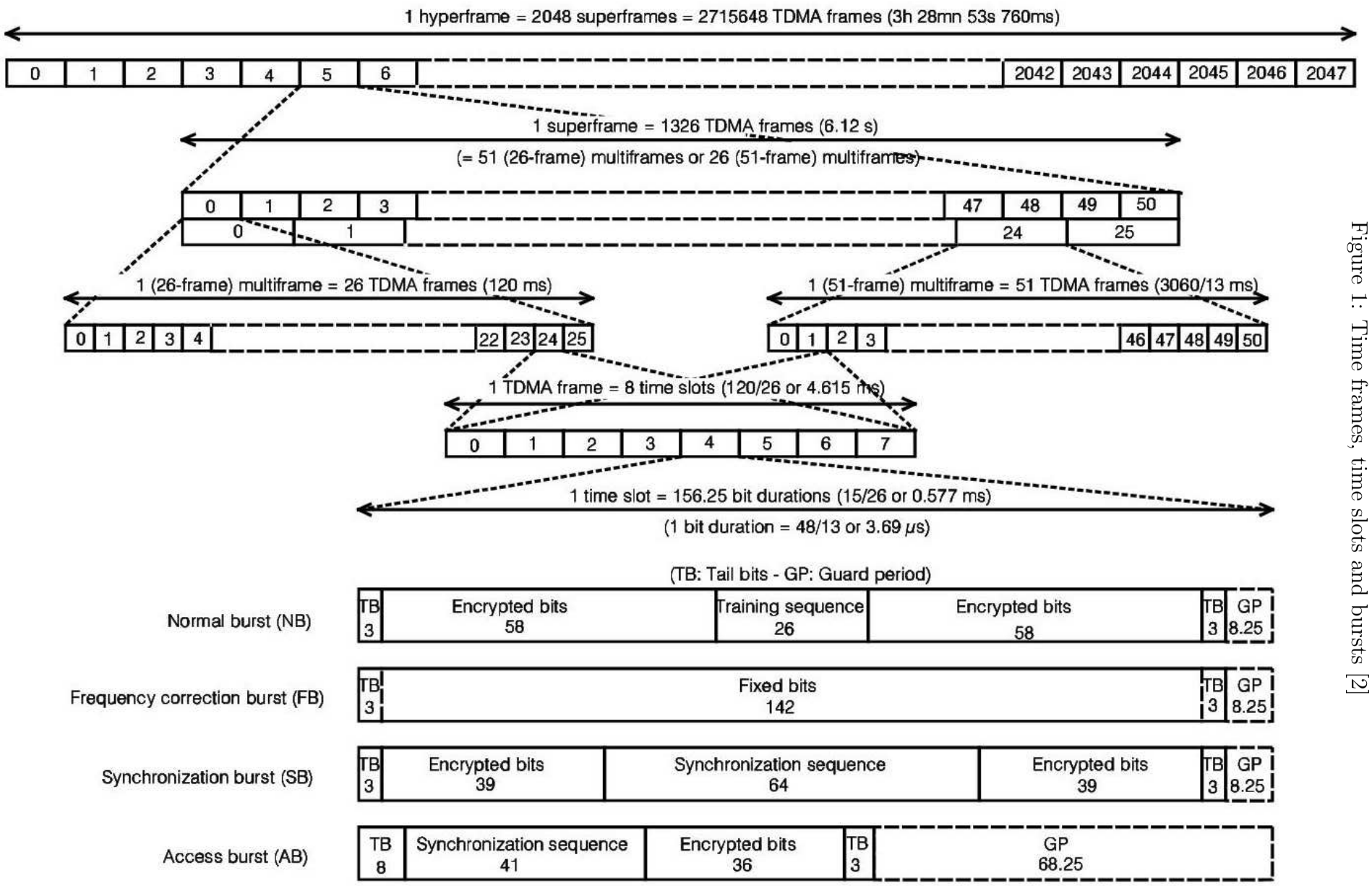

Figure 1: Time frames, time slots and bursts [2]

2.2 Channel organization

In the GSM, there are two types of logical channels: Common Channels (CCH) and Dedicated Channels (DCH). As figure 2 shows, the CCH consists of Broadcasting Channels (BCH) and Common Control Channels (CCCH) and the DCH consists of Dedicated Control Channels (DCCH) and Traffic Channels (TCH). Figure 3 and 4 shows how the channel organized in the 26-frame multiframe and 51-frame multiframe respectively.

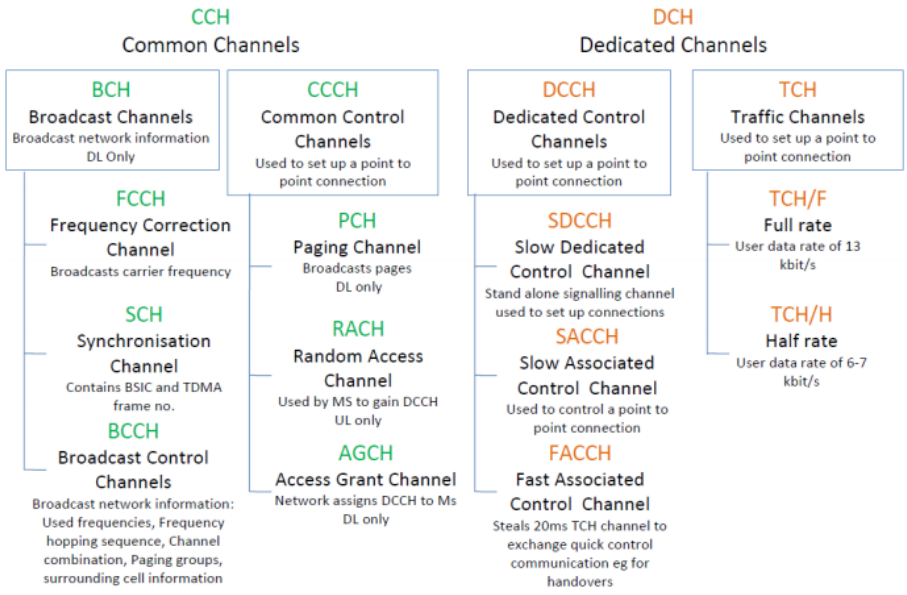

Figure 2: Logic Channels of GSM [3]

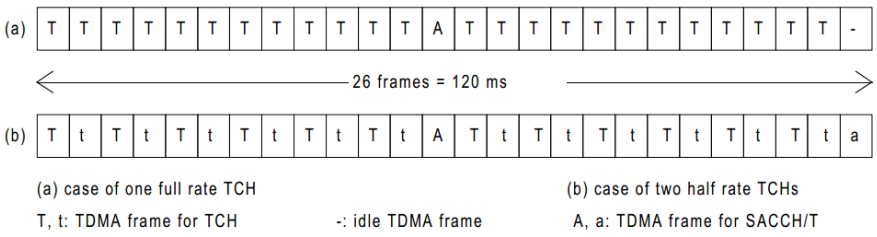

Figure 3: 26-frame organization [2]

Figure 4: 51-frame organization [2]

2.3 Coding schemes

The table in figure 5 shows the summarised description of the coding schemes of GSM. For all types of channels, the operations are made by the order of external coding(block coding), internal coding(convolutional coding) and then interleaving.

Type of channel	bits/block data+parity+tail[1]	convolutional code rate	coded bits per block	interleaving depth
TCH/FS			456	8
class I[2]	182 + 3 + 4	1/2	378	
class II	78 + 0 + 0	-	78	
TCH/HS			228	4
class I[3]	95+3+6	104/211	211	
class II	17+0+0		17	
TCH/F9.6	4*60 + 0 + 4	244/456	456	19
TCH/F4.8	60 + 0 + 16	1/3	228	19
TCH/H4.8	4*60 + 0 + 4	244/456	456	19
TCH/F2.4	72 + 0 + 4	1/6	456	8
TCH/H2.4	72 + 0 + 4	1/3	228	19
FACCH/F	184 + 40 + 4	1/2	456	8
FACCH/H	184 + 40 + 4	1/2	456	6
SDCCHs SACCHs BCCH NCH AGCH PCH				
CBCH	184 + 40 + 4	1/2	456	4
RACH	8 + 6 + 4	1/2	36	1
SCH	25 + 10 + 4	1/2	78	1
NOTE 1: The tail bits mentioned here are the tail bits of the convolutional code.				
NOTE 2: The 3 parity bits for TCH/FS detect an error on 50 bits of class I.				
NOTE 3: The 3 parity bits for TCH/HS detect an error on 22 bits of class I.				

Figure 5: Channel block structure [2]

3 Deliverables

Airprobe is a tool that was used to analyse GSM signals a while back. It no longer works with current GNU Radio versions, and doesn't make use of any of the new GNU Radio features. A modular rewrite is the main part of this project. Figure 6 shows the blocks of GSM Analysis Tool which will be implemented in this project.

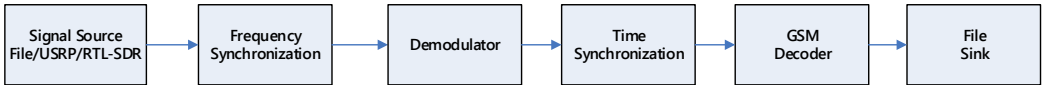

Figure 6: Blocks of GSM Analysis Tool

3.1 FCCH Burst Detection with adaptive filter

Martin Braun (martin.braun@ettus.com) suggests a paper [4] which introduced an algorithm for FCCH burst detection. The algorithm uses an adaptive filter to help distinguishing FCCH bursts from other bursts. I have implemented this algorithm

for test and upload to my github : https://github.com/hzhua/gr-fchdetection. As after GMSK modulation, the frequency correction bits is a sinusoid of frequency 67.7033KHz. The normal bursts at base-band are 100KHz. Therefore, I used a FIR band pass filter with center frequency 67.7033KHz with bandwidth of 18KHz. With the help of adaptive filter, it can detect the FCCH burst fast and accurately. Figure 7 shows the average error power of the adaptive filter. The data is gathered by my R820T with RTL2832U. The average error is very low at frequency burst and high at other bursts. A threshold can be easily find to separate two parts.

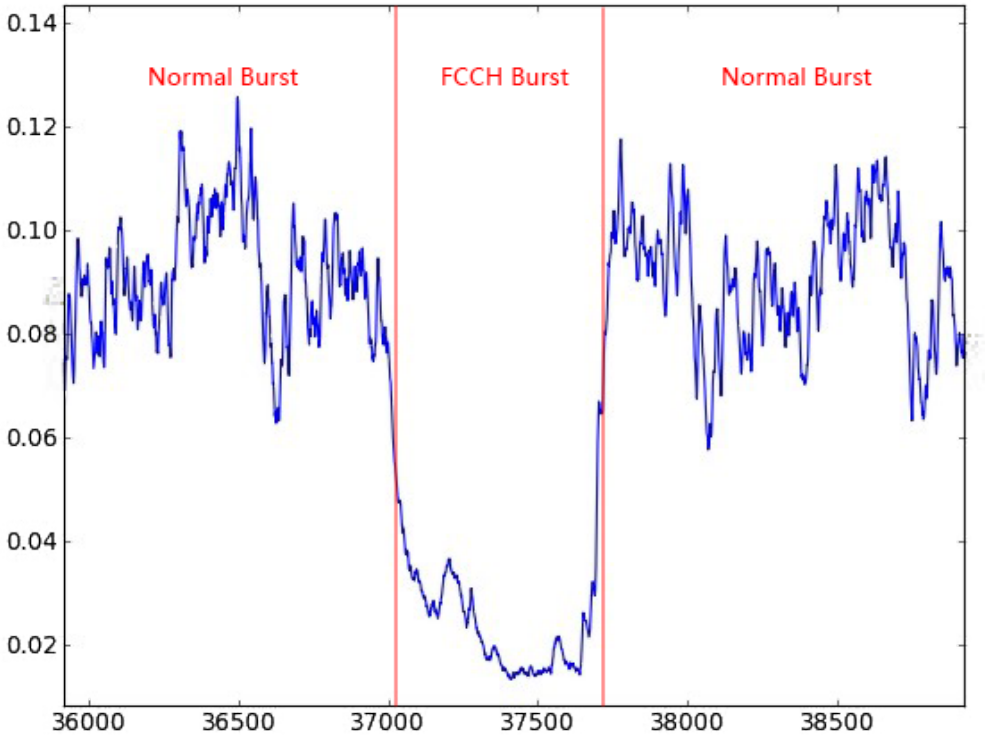

Figure 7: The average error of the adaptive filter

3.2 Improving the performance of frequency correlation

As Piot Krysik (perper@o2.pl) said, there is instability in the method of correction of the frequency offset in Airprobe. The reason is the lack of synchronization between computation of frequency offset and its correction inside of FIR filter. As Piot suggests, I will use stream tags to solve this problem.

3.3 Using GNU Radio 3.7 new features

- Correlate and Sync block is a new block added in GNU Radio 3.7.2. It is designed to search for a preamble by correlation and uses the results of the correlation to get a time and phase offset estimate. It can be used to help GSM signal to get time synchronization.
- The file sink in GNU Radio starts to support appending content into a file. This feature makes it easier to use with IPC. As GSM Analysis Tool will

support real-time decoding of GSM signal, it can be used to communicate with other tools for further analysis.

3.4 The Deliverables list

- Deliverable 0: Create on out-of-tree module for GSM analysis tool (gr-gsm), using gr-modtool;
- Deliverable 1: Get properly channelized / resampled data;
- Deliverable 2: Find frequency correction burst (FB) and realize frequency synchronization;
- Deliverable 3: Find synchronization burst (SB) and realize time synchronization;
- Deliverable 4: Decode and decrypt the downlink of 26-frame multiframes for the traffic channel (TCH);
- Deliverable 5: Decode and decrypt the downlink of 51-frame multiframes for 3 types of control channels ("BCCH+CCCH", "8SDCCH/8" and "BCCH+CCCH+4SDCCH/4");
- Deliverable 6: Integrate all parts with real-time capabilities.

4 Schedule

As the schedule of GSoC'14 shows, there are 14 weeks including midterm evaluation and final evaluations. This schedule is a rough plan which may change after discussions with the mentor. The timeline is structured as follows:

- **1st Phase (2 weeks)** : Initial discussions with the mentor. Learn how to write an out-of-tree module of GNU Radio. Learn the detailed protocol of GSM.
- **2nd Phase (1 week)** : OOT module is created. Data capturing is coded.
- **3rd Phase (2 weeks)** : Frequency synchronization and GMSK demodulator with Viterbi algorithm is coded.
- **4rd Phase (1 week)** : Start to code time synchronization.
- **5th Phase (1 week)** : Mid-term evaluation and time synchronization is finished.
- **6th Phase (2 weeks)** : TCH and Control Channel decoder is coded.
- **7th Phase (3 weeks)** : Integrate all parts with real-time capabilities.
- **8th Phase (2 weeks)** : QA test, code clean-up, adding documentation and final submission.

5 Expanding tasks

Because there is only 14 weeks in GSoC, it is too short to add other parts into this project during this 14 weeks. However I will keep improving gr-gsm after GSoC. Other expanding tasks will be added into gr-gsm after. Here is a list with part of them:

- frequency hopping protocol will be supported; (With high priority)
- A physical layer of GSM transmitter will be added. With this transmitter, one can disguise himself into a cellphone or a base station.
- GPRS will be supported;

6 Qualification

I'm a fourth year undergraduate in Electronic Information Engineering of University of Electronic Science and Technology of China. This fall, I will take up a PhD degree with a major in Computer Science in University of Hong Kong. I used to participate in the ACM International Collegiate Programming Contest (ICPC) and won two gold medals in Asia regional contest. My experience in ICPC greatly improved my algorithm and programming ability. Besides, I have been working on the research of Cognitive Radio Networks for a year. And I have submitted a paper on Cognitive Radio on WASA2014(The 9th International Conference on Wireless Algorithms, Systems, and Applications).

I got to know GNU Radio when I read a paper on cognitive radio which using GNU Radio and a USRP to analyse the algorithm in their paper. Then we bought a USRP to help our group in the research of cognitive radio. My easy access to USRP will help me in finishing the coding of GSM Analysis Tool.

On my coding experience, I was an intern in the Alibaba Group which is the biggest e-commerce company in China. I finished 60 percent codes of a Java project called Beacon. It uses Solr as searching engine to catch illegal users. My work was to optimize the searching engine and the coding of the back end. Unfortunately, the codes are under confidential. I also contributed my codes to an Open Source project SNSAPI (it is a Chinese project). You can find it in my github page: https://github.com/hzhua/snsapi.

As this is the biggest project I have dealt with, it may be a challenging task for me to complete in the given time span. If I cannot finish GSM Analysis Tool in 14 weeks, I will continue my works even after GSoC. For my personal motivation, I will continue my research on wireless networks during my PhD degree. Therefore I will use GNU Radio and USRP as powerful tools for many years. I would love to continue learning and working in GNU Radio community after GSoC. So, don't worry about if I will disappear during GSoC.

Please visit my personal page https://hzhua.github.io if you want to know me better. You can find my CV there.

I agree for my work to be published under the GPL license.

7 Conclusion

I hope I can win this chance to work in GNU Radio community. If you have any questions, don't be hesitated to contact me.

References

[1] "GNU Radio." http://gnuradio.org.

[2] G. ETSI, "05.01:" digital cellular telecommunications system (phase 2+), physical layer on the radio path," 1992.

[3] M. Glendrange, K. Hove, and E. Hvideberg, "Decoding gsm," 2010.

[4] G. Varma, U. Sahu, and G. Charan, "Robust frequency burst detection algorithm for gsm/gprs," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 6, pp. 3843–3846 Vol. 6, Sept 2004.