APPLICATION NOTE

Implementation and validation of porting SILK audio codec on the RX210 microcontroller

Author: Julien ROQUETTE

Student in electrical engineering (embedded systems)

January 2015

Contents

List of figures

Introduction

This application note provides all that is needed to validate the operation of the SILK codec on the RX210.

The SILK codec is described briefly in the first part. Then, the steps to validate the SILK encoder and the SILK decoder operation on the RX210 are described.

Related document

This document is related to the application note “SILK encoder implementation on the RX210 microcontroller" which describes the main points to port the SILK codec on the RX210.

I. SILK codec

In this Section, it is provided a description of the SILK codec.

The SILK codec is a part of the free and open source audio code Opus. It is a speech codec design for real-time application and was initially developed by Skype. It consists of an encoder and a decoder.

To encode an input signal, the SILK codec uses many control parameters:

- Sampling rate (8, 12, 16 or 24 kHz)
- Packet rate (20, 40, 60, 80 or 100 ms)
- Bitrate (6 to 40 kbps)
- Packet Loss Resilience
- Forward Error Correction (FEC)
- Algorithmic complexity (3 levels of complexity)
- Discontinuous Transmission (DTX)

Each one must be defined before porting it on the RX210 (see Related Document page 5). When the porting is done, it is necessary to test and validate separately the proper functioning of the SILK encoder and of the SILK decoder.

The encoder and the decoder validation are described in part II and part III, respectively.

II. SILK encoder validation

In this part, all the steps necessary to validate the SILK encoder operation on the RX210 are described.

The first step consists in producing some speech samples. To test and validate its operation on RX210, it is advisable to realize short samples (about 1 s) since the RX210 does not have a very large memory. All these samples should be stored on computer in an uncompressed format (e.g. in a WAV or a PCM file).

Then, each one must be encoded with the parameters defined before (an example of program's arguments setting from Code::Blocks is given Fig.1).

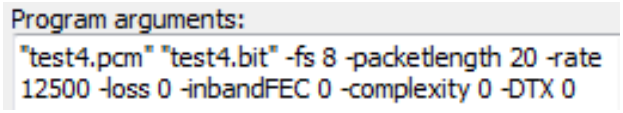

Figure 1: program's arguments setting

For this, the SILK encoder libraries will be compiled and executed on PC with the different vocal samples as input signals (see Fig. 2). The resulting BIT files will serve as references for testing.

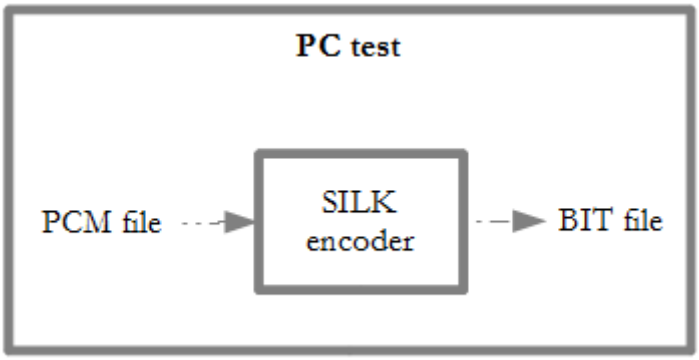

Figure 2: samples encoding

Thereafter, data contained in the PCM and the BIT files will be recovered and stored in the RX210 ROM memory (see Fig. 3). For this, data can be written in a text file and then defined as global in the main C program of the encoder.

```
19  42  8  21  178  247  28  68  13  85  218  165  155  143  218  144  33  61  15  139
10  41  218  210  221  50  213  239  41  179  63
33  54  187  154  247  250  165  9  16  72  71  80  112  5  183  58  163  62  45  20  243  104  3  7  123  39  221  219  97  160  9  50  166  243
(...)
25  52  134  34  14  200  249  7  27  171  41  91  17  120  10  218  142  243  79  124  241  82  204  101  107  123
-1
```

Figure 3: example of BIT file content

An example of a BIT file content is given Fig. 3. It is important to note that the first value of each line (variable named nBytes in the program) corresponds to the number of bytes that follow. The maximal value of nBytes is 160. The end of the BIT file is indicated by nBytes which takes a negative value (here -1).

The PCM file data will be encoded by the encoder ported on RX210 and the result of the encoding will be compared with the reference data from BIT file.

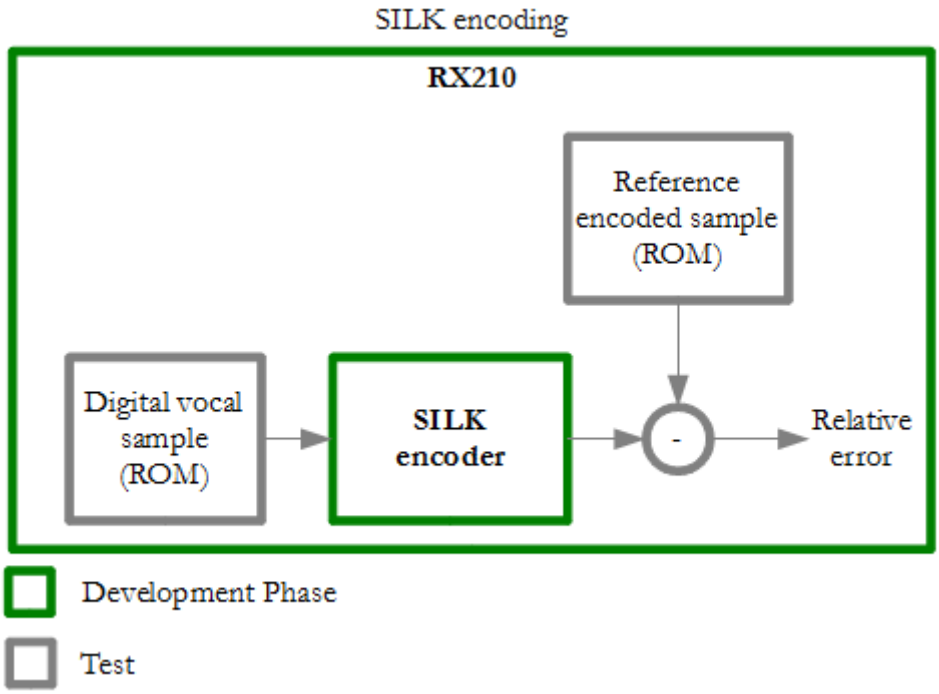

Figure 4: SILK encoder test

Thus, if the SILK parameters are the same on PC and RX210, the relative error should be zero.

III. SILK decoder validation

Here, the steps to test and validate the SILK decoder operation on the RX210 are presented.

Firstly, as for the test of the encoder, it is necessary to use audio speech samples. These samples must be encoded in the SILK format and stored in a BIT files. To obtain BIT files, the SILK encoder libraries must be compiled and executed on PC with the different speech audio samples as input signals (see Fig. 2).

Once the BIT files have been achieved, the SILK decoder libraries must be compiled and executed on PC in order to decode the compressed data stored in the BIT files (Fig. 5) and to store the output in a PCM file.

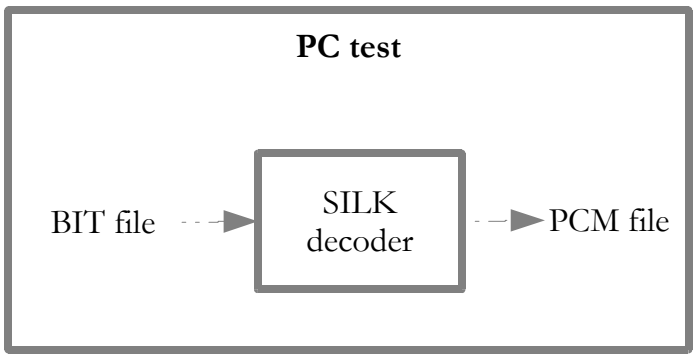

Figure 5: samples decoding

The BIT and PCM files' content of each sample can now be stored in the RX210 ROM memory. For this, data can be written in a text file and then defined as global in the main C program of the decoder.

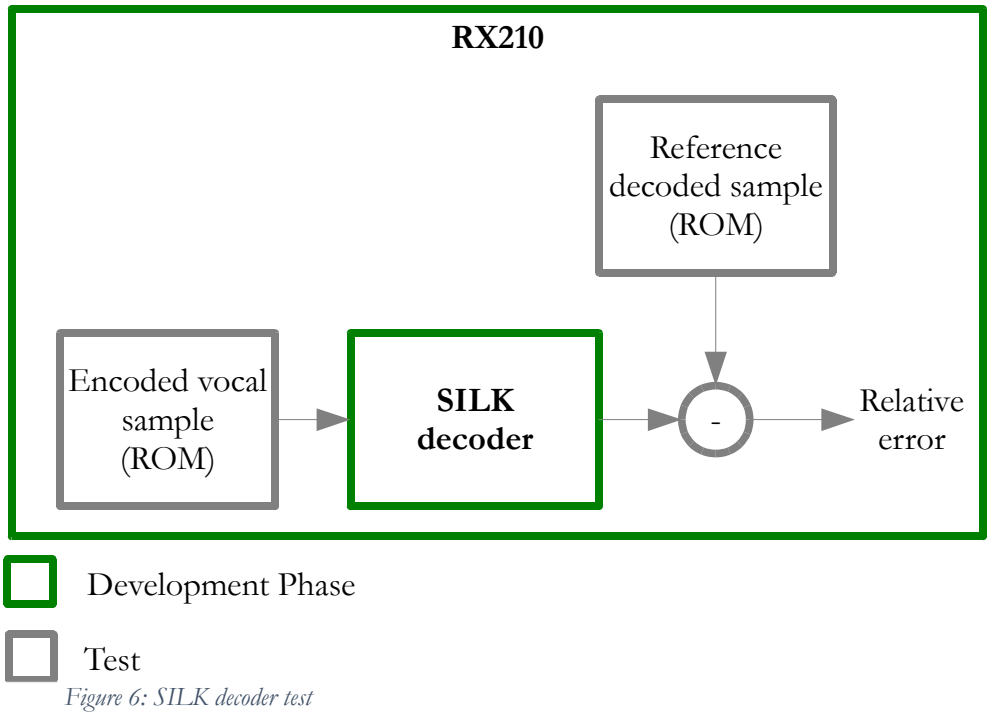

Figure 6: SILK decoder test

The BIT file's data will be encoded by the encoder ported on RX210 (Fig. 6) and the result of the decoding will be compared with the reference data from PCM file. If the port was done properly, the relative error between the reference data and the decoded data must be zero.