SIM Provisioning GUI for sysmoUSIM-SJS1

1. Brief Introduction

This application is a Graphical User Interface for SIM card Provisioning of sysmoUSIM-SJS1 SIM cards, it is used to program parameters of a SIM card e.g IMSI, ICCID, KI, OP, OPC, etc. It is composed of sysmo-usim-tool and pySim tools running in the background to program parameters of the SIM card. This application is a Linux(Ubuntu) based desktop application.

sysmocom
systems for mobile communications GmbH
custom-tailored GSM solutions from RAN to Core Network
Visit us at http://sysmocom.de/
• First-hand expertise in protocol R&D from A-bis to SS7/MAP
• Support, training and development for OpenBSC, OsmoSGSN
• Low-cost GSM pico-cell platform sysmoBTS
• Small, PBX-style autonomous GSM networks
sysmoUSIM-SJS1
sysmocom
systems for mobile communications GmbH
PIN 7280
PUK 98714018
IMSI 809901700000005019
custom-tailored GSM solutions from RAN to Core Network
info@sysmocom.de
http://sysmocom.de/
sysmoUSIM-SJS1

Figure 1.1 : sysmoUSIM-SJS1 SIM card

2. Dependencies & Application installation

2.1 Dependencies installation

(Run the following commands in Ubuntu terminal)

-
  ```
  sudo apt-get install pcscd pcsc-tools libccid libpcsclite-dev python3-pyscard python3-serial python3-pip
  ```

-
  ```
  pip3 install pytlv cmd2 jsonpath-ng construct bidict gsm0338 pyyaml pycrypto
  ```

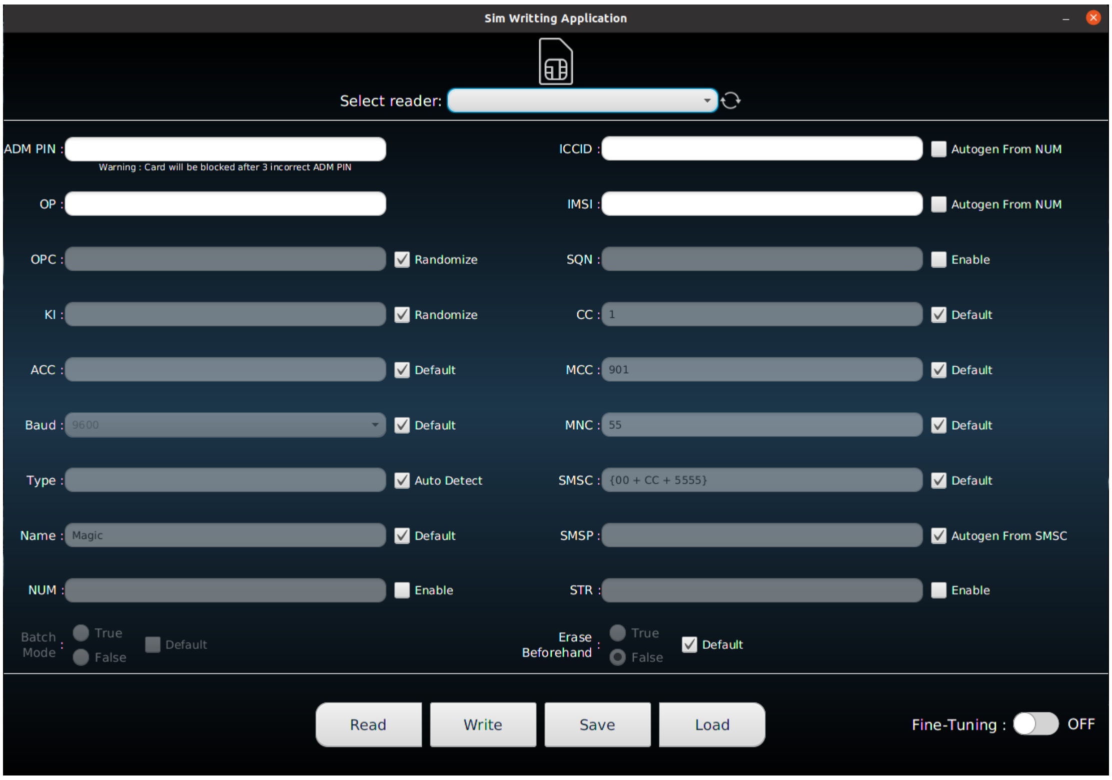
2.2 Application installation

- Copy pysim and sysmo-usim-tool folders to your Home directory in Ubuntu
  IMPORTANT: It is crucial that these two folders are placed in the HOME directory because this application references this path. Your Home directory path should be “**/home/<your-username>/**”.

- Copy the simwriting-1.0.deb debian package to any folder in your PC and open the terminal from the folder where you copied this debian package and execute the following command to install the application: `sudo dpkg -i simwriting-1.0.deb`
- To open the application click “Activies” on the top left corner and search “simwriting”, the application icon should appear, you can open by clicking that icon.

3. How to use the application

Figure 3.1 below shows the first interface you will see when you open your application. SIM card parameters will appear as shown in the figure, some of these parameters will be needed when writing/reading the card. When you hover the mouse over a parameter input field, a helper message will appear explaining more details about that parameter.



Sim Writting Application

Select reader:

ADM PIN :
Warning : Card will be blocked after 3 incorrect ADM PIN

OP :

OPC : ☑ Randomize

KI : ☑ Randomize

ACC : ☑ Default

Baud : 9600 ☑ Default

Type : ☑ Auto Detect

Name : Magic ☑ Default

NUM : □ Enable

Batch Mode : ○ True ○ False □ Default

ICCID : □ Autogen From NUM

IMSI : □ Autogen From NUM

SQN : □ Enable

CC : 1 ☑ Default

MCC : 901 ☑ Default

MNC : 55 ☑ Default

SMSC : {00 + CC + 5555} ☑ Default

SMSP : ☑ Autogen From SMSC

STR : □ Enable

Erase Beforehand : ○ True ● False ☑ Default

Fine-Tuning : OFF

Figure 3.1 : SIM card provisioning Graphical User Interface application

3.1 Reading Card

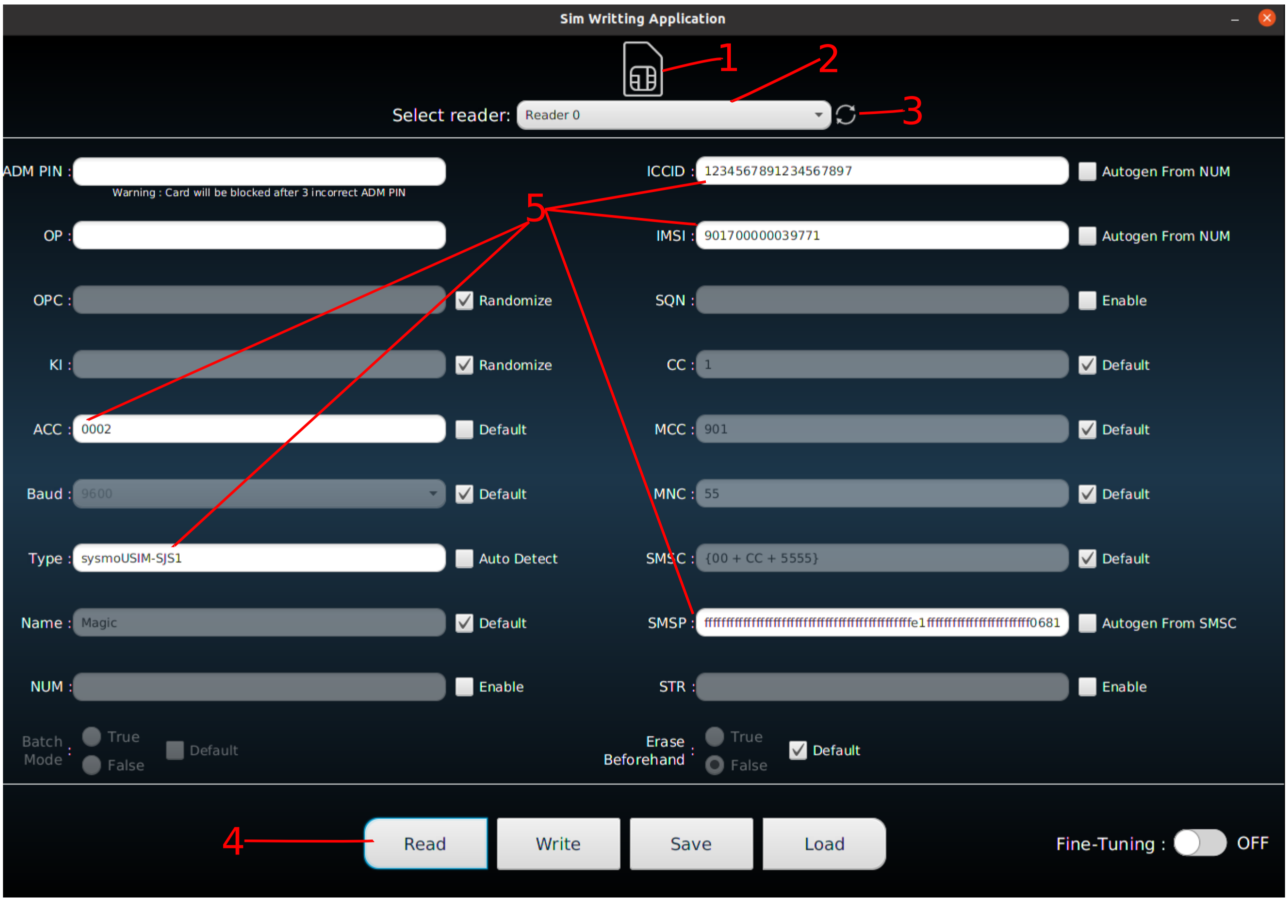
Number 1 in the figure 3.2 below shows the SIM card icon. If any SIM card is detected in any inserted reader, this icon will remain as it is in the below picture but if no SIM card was detected then this SIM card icon will appear with a diagonal slash over it showing that no SIM card was detected.

Number 2 in the figure 3.2 below shows the drop-down menu you must click to view all available readers with SIM cards in them.

Number 3 in figure 3.2 below shows a refreshing icon that you must click to refresh & search for available readers.

Number 4 in figure 3.2 below shows the read button you must click in order to read the card in the reader that you selected in 2. If you click the read button without selecting a reader you'll get an error. If the reader is selected then a dialogue (Shown in Figure 3.3) with card information will appear and some card parameters will be populated in the main window.

Number 5 in figure 3.2 below shows the parameter that have been read from the card after clicking the read button.



Select reader: Reader 0

ADM PIN :

Warning : Card will be blocked after 3 incorrect ADM PIN

OP :

OPC : ☑ Randomize

KI : ☑ Randomize

ACC : 0002 □ Default

Baud : 9600 ☑ Default

Type : sysmoUSIM-SJS1 □ Auto Detect

Name : Magic ☑ Default

NUM : □ Enable

Batch Mode : ○ True ○ False □ Default

ICCID : 1234567891234567897 □ Autogen From NUM

IMSI : 901700000039771 □ Autogen From NUM

SQN : □ Enable

CC : 1 ☑ Default

MCC : 901 ☑ Default

MNC : 55 ☑ Default

SMSC : {00 + CC + 5555} ☑ Default

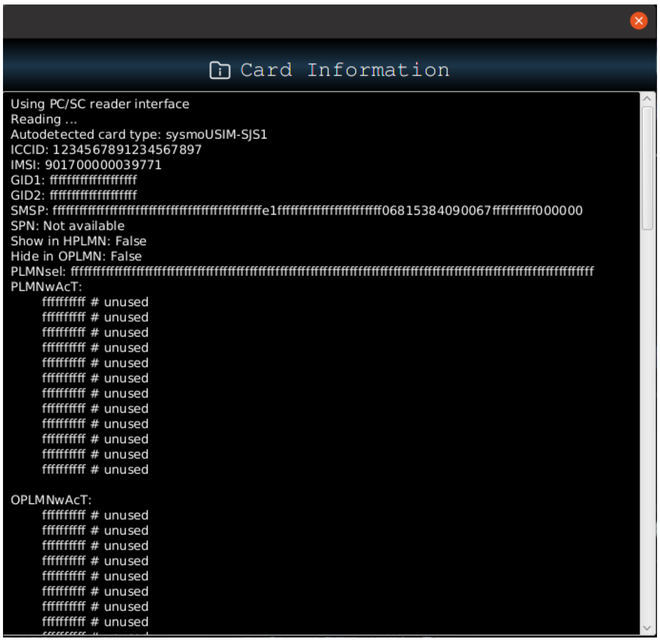
SMSP : ffffffffffffffffffffffffffffffffffffffffffffffffe1ffffffffffffffffffffffff0681 □ Autogen From SMSC

STR : □ Enable

Erase Beforehand : ○ True ● False ☑ Default

Fine-Tuning : OFF

Figure 3.2 : GUI Showing how to read card



```
Using PC/SC reader interface
Reading ...
Autodetected card type: sysmoUSIM-SJS1
ICCID: 1234567891234567897
IMSI: 901700000039771
GID1: ffffffffffffffff
GID2: ffffffffffffffff
SMSP: ffffffffffffffffffffffffffffffffffffffffffffffffe1ffffffffffffffffffffffff06815384090067ffffffffff000000
SPN: Not available
Show in HPLMN: False
Hide in OPLMN: False
PLMNsel: ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
PLMNwAcT:
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused

OPLMNwAcT:
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
        ffffffffff # unused
```

Figure 3.3 : Dialogue with Card Information

3.2 Writing Card

There are two modes that can be used to provision SIM card parameters :

1. Fine-Tuning OFF

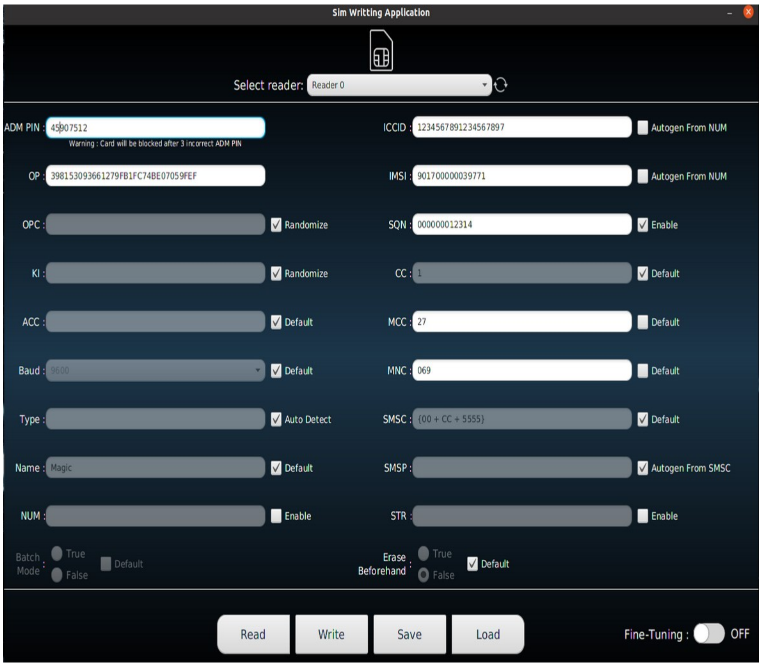
   This mode allows bulk provisioning of parameters, i.e. you have to specify all the parameters & some of them if not specified they can either be randomized/default/Auto-generated. ADM PIN & OP are compulsory in this mode. Figure 3.4 shows sample of how parameters looks like when fine-tuning mode is turned OFF

2. Fine-Tuning ON

   This mode allows allows you to fine-tune or tweak specific parameters. At this point, only 6 parameters can be fine-tuned(OP, OPC, KI, ICCID, IMSI,SQN). Please note it is possible to fine-tune other parameters as well, but in this application version only these ones can be fine-tuned. ADM PIN is compulsory in this mode. You will notice that other parameters that cannot be fine-tuned are disabled & you cannot edit them. Figure 3.5 shows sample of how parameters looks like when fine-tuning mode is turned ON

You can turn on/off fine-tuning by clicking the switch on the bottom right corner.

IMPORTANT: It is VERY crucial that you use the correct ADM PIN. Attempting to program with an invalid key can BRICK your card, rendering it useless!!!

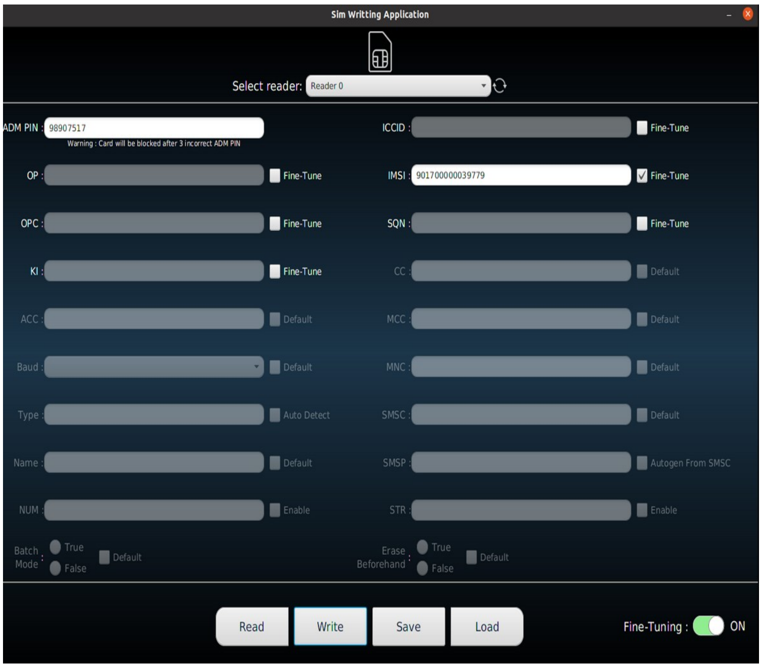


Select reader: Reader 0

ADM PIN : 45907512
Warning : Card will be blocked after 3 incorrect ADM PIN
ICCID : 1234567891234567897 □ Autogen From NUM
OP : 398153093661279FB1FC74BE07059FEF
IMSI : 901700000039771 □ Autogen From NUM
OPC : ☑ Randomize
SQN : 000000012314 ☑ Enable
KI : ☑ Randomize
CC : 1 ☑ Default
ACC : ☑ Default
MCC : 27 □ Default
Baud : 9600 ☑ Default
MNC : 069 □ Default
Type : ☑ Auto Detect
SMSC : {00 + CC + 5555} ☑ Default
Name : Magic ☑ Default
SMSP : ☑ Autogen From SMSC
NUM : □ Enable
STR : □ Enable
Batch Mode : ○ True ○ False □ Default
Erase Beforehand : ○ True ● False ☑ Default

Fine-Tuning : OFF

Figure 3.4 Fine-Tuning Mode OFF



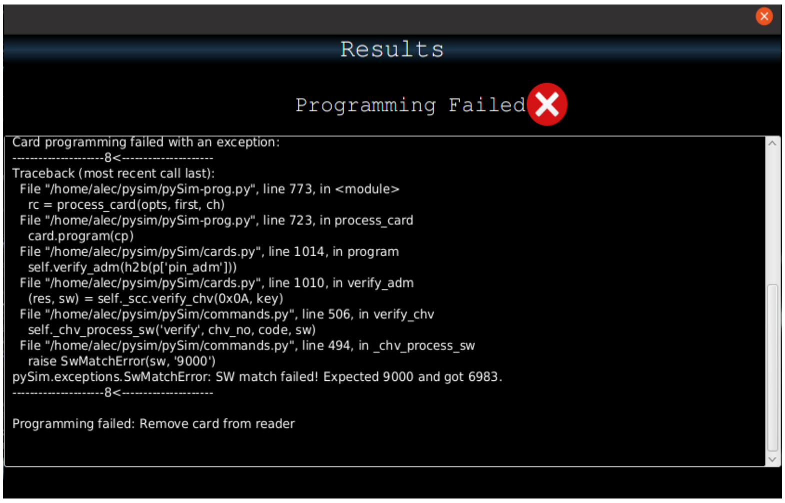
Select reader: Reader 0

ADM PIN : 98907517
Warning : Card will be blocked after 3 incorrect ADM PIN
ICCID : □ Fine-Tune
OP : □ Fine-Tune
IMSI : 901700000039779 ☑ Fine-Tune
OPC : □ Fine-Tune
SQN : □ Fine-Tune
KI : □ Fine-Tune
CC : □ Default
ACC : □ Default
MCC : □ Default
Baud : □ Default
MNC : □ Default
Type : □ Auto Detect
SMSC : □ Default
Name : □ Default
SMSP : □ Autogen From SMSC
NUM : □ Enable
STR : □ Enable
Batch Mode : ○ True ○ False □ Default
Erase Beforehand : ○ True ○ False □ Default

Fine-Tuning : ON

Figure 3.5 Fine-Tuning Mode ON

Figure 3.6 and Figure 3.7 below shows the results of after you clicked the write button on the above Figure 3.4 and Figure 3.5 respectively. Figure 3.6 shows the results of when Fine-Tuning mode is OFF and the ADM PIN is invalid. Figure 3.7 shows the results of when Fine-Tuning mode is ON and the ADM PIN is valid and we are Fine-Tuning the IMSI.

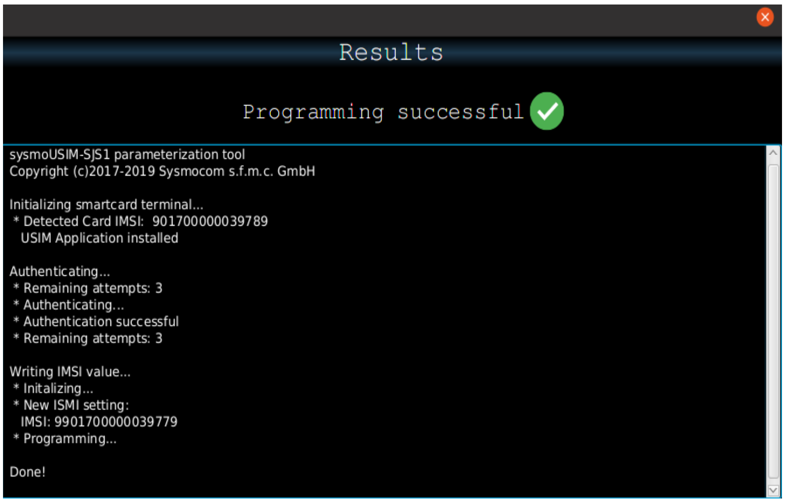


Results

Programming Failed

```
Card programming failed with an exception:
--------------------8<--------------------
Traceback (most recent call last):
  File "/home/alec/pysim/pySim-prog.py", line 773, in <module>
    rc = process_card(opts, first, ch)
  File "/home/alec/pysim/pySim-prog.py", line 723, in process_card
    card.program(cp)
  File "/home/alec/pysim/pySim/cards.py", line 1014, in program
    self.verify_adm(h2b(p['pin_adm']))
  File "/home/alec/pysim/pySim/cards.py", line 1010, in verify_adm
    (res, sw) = self._scc.verify_chv(0x0A, key)
  File "/home/alec/pysim/pySim/commands.py", line 506, in verify_chv
    self._chv_process_sw('verify', chv_no, code, sw)
  File "/home/alec/pysim/pySim/commands.py", line 494, in _chv_process_sw
    raise SwMatchError(sw, '9000')
pySim.exceptions.SwMatchError: SW match failed! Expected 9000 and got 6983.
--------------------8<--------------------

Programming failed: Remove card from reader
```

Figure 3.6 : Invalid ADM PIN



Results

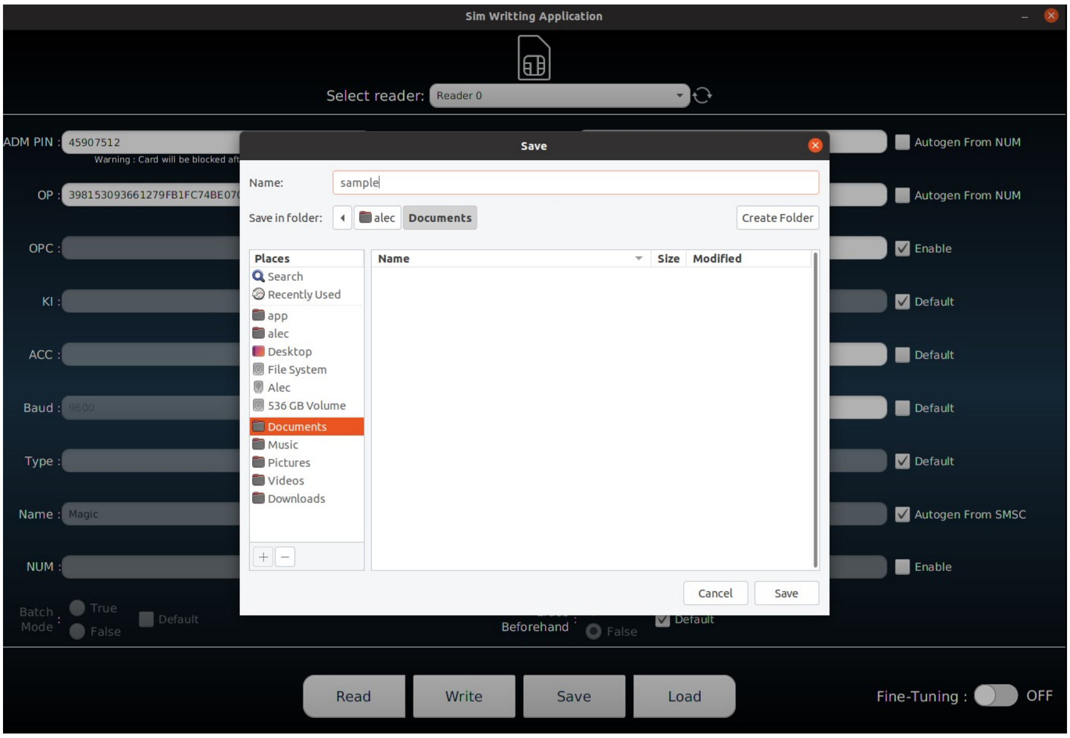
Programming successful

```
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017-2019 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Detected Card IMSI:  901700000039789
   USIM Application installed

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Writing IMSI value...
 * Initalizing...
 * New ISMI setting:
   IMSI: 9901700000039779
 * Programming...

Done!
```

Figure 3.7 : Valid ADM PIN

3.3 Save Data

This feature allows you to save the values of the parameters that you have inserted in the input field of all the parameters. This data will be saved in a .txt file with a name of your choice. This allows you to used this data at a later stage by loading the saved file into the application (You will see in 3.4). Figure 3.8 below shows the interface that appears when you click the save button, you can name your file and choose a location to save your file.



Sim Writting Application

Select reader: Reader 0

ADM PIN : 45907512

Warning : Card will be blocked aft

OP : 398153093661279FB1FC74BE07(

Baud : 9600

Name : Magic

Save

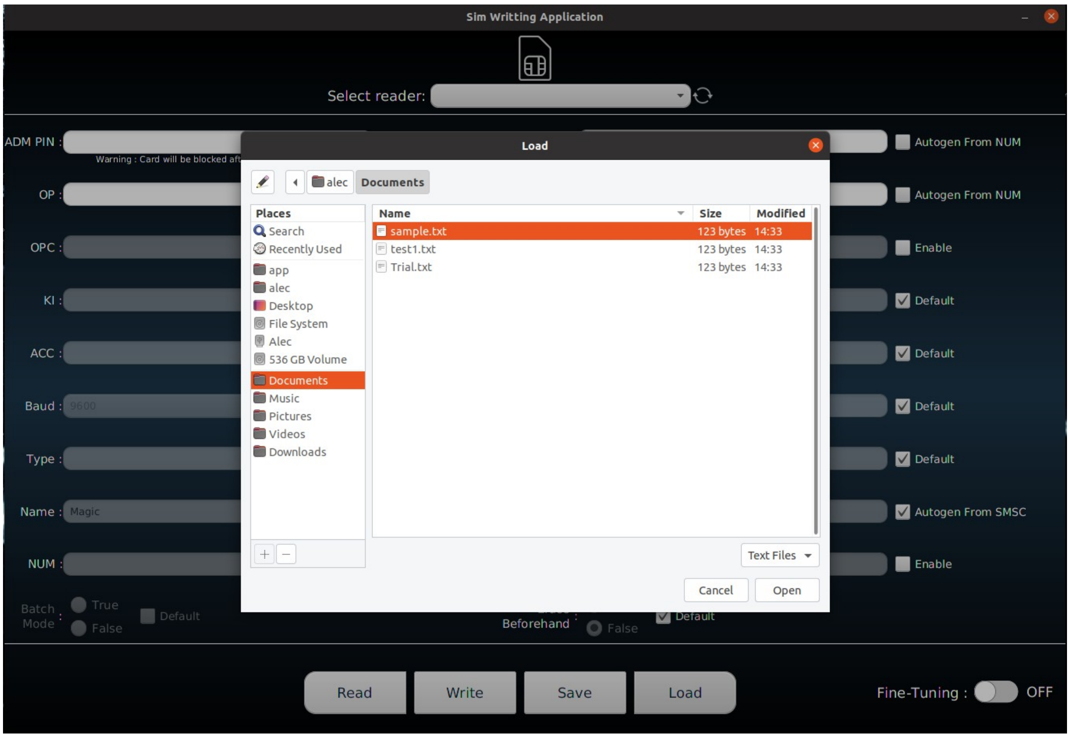
Name: sample

Save in folder: alec Documents

Figure 3.8 : Saving a file GUI

3.4 Load Data

This feature allows you to load up data from the file you have saved earlier (refer to 3.3). This will allowsyou to navigate to a folder where you would have saved your files, it will list all .txt files available in that selected folder. After you click open on that file, it will populate data from the file to the input field of every parameters that was saved in that file. Please note that you can only load files that you have saved using this Application only. Figure 3.9 below shows the interface that appears when you click the Load button.



Baud : 9600

Name : Magic

Batch Mode : ○ True ○ False □ Default

Beforehand : ○ False ☑ Default

Name	Size	Modified
sample.txt	123 bytes	14:33
test1.txt	123 bytes	14:33
Trial.txt	123 bytes	14:33

□ Autogen From NUM
□ Autogen From NUM
□ Enable
☑ Default
☑ Default
☑ Default
☑ Default
☑ Autogen From SMSC
□ Enable

Figure 3.9 : Loading a file GUI

4. Development Framework

- Java 8
- JavaFx Framework
- SceneBuilder
- NetBeans (IDE)

4. Future Features

- Help button – This feature will give you a brief explanation of SIM card parameters & how to use this application