

Programación de tarjetas SIM sysmoISIM-SJA2

1. Instalación de dependencias

Antes de intentar conocer los parámetros de la tarjeta (o programarla), es recomendable utilizar una herramienta que permite verificar el estado de la conexión en el que se encuentra el programador y la tarjeta, además de indicar el tipo de tarjeta que se está utilizando y otros datos de la misma. Dicha herramienta lleva por nombre `pcsc_scan`, derivada de una serie de herramientas contenidas en el paquete `pcsc_tools`. Para instalar dicha herramienta, solo basta con introducir el siguiente comando (solo para sistemas operativos basados en Debian):

```
$ sudo apt install pcsc-tools pcscd libccid swig python3-dev python3-serial libpcsclite-dev git # Instalación de dependencias.
```

Por otro lado, antes de instalar el software que permite la programación de la tarjeta, es necesario instalar una dependencia, que es la librería de Python 3 *Módulo de tarjeta inteligente (Smartcard module for Python)* que lleva el nombre de `pyscard`. Como primera parte, es necesario poseer el gestor de paquetes de Python 3 denominado `pip`. Los comandos para realizar la instalación de estas dependencias son los siguientes:

```
$ sudo apt install python3-pip # Instalación del administrador de paquetes (pip).
$ sudo pip3 install pycard pytlv # Instalación de pycard.
# Si al instalar pycard presenta error, se podría solucionar instalándolo de una manera alternativa con el siguiente comando:
$ sudo apt install python3-pycard
```

1.1. Instalación del Software

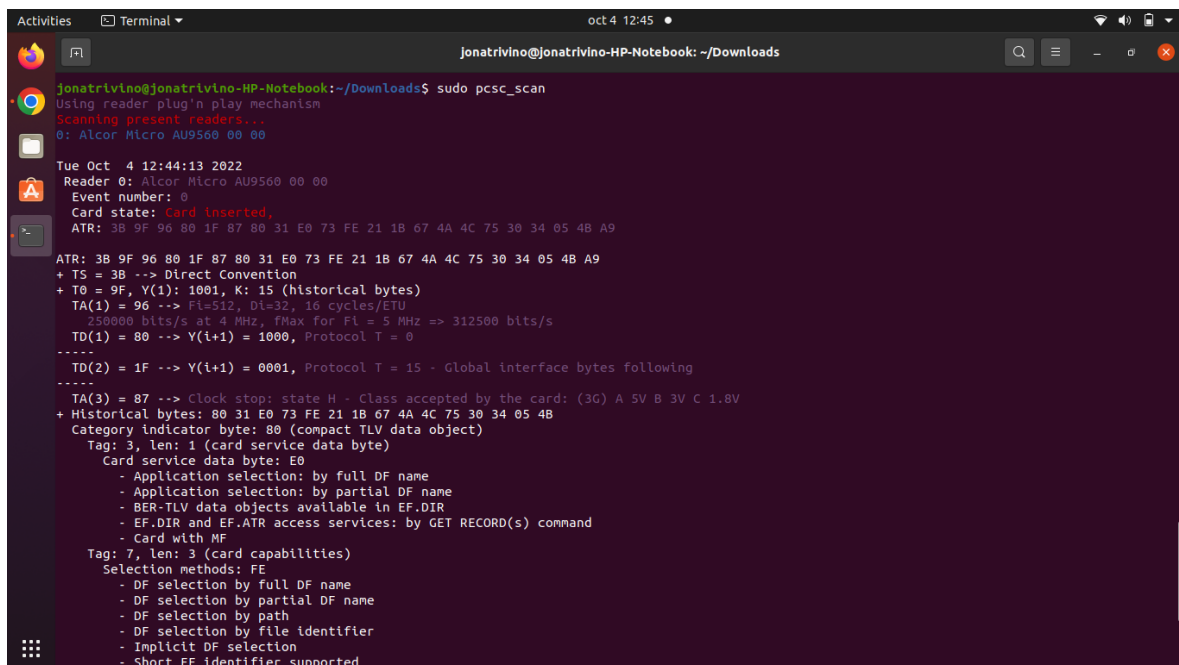
La empresa Sysmocom proporciona un manual de usuario que presenta la descripción de las tarjetas `sysmoUSIM-SJS1` y `sysmoISIM-SJA2`, además de la lectura y programación de las mismas. Este se puede encontrar en el siguiente link: <https://www.sysmocom.de/manuals/sysmousim-manual.pdf>

En el manual, se especifican 2 herramientas recomendadas por la empresa para la programación de las tarjetas, ambas basadas en python: `pysim` y `sysmo-usim-tool`. En esta ocasión, se utilizó `sysmo-usim-tool`, ya que presentó una mayor sencillez para su instalación y utilización. Para instalar la herramienta, basta con copiar el repositorio donde está alojada la misma mediante GIT:

```
$ git clone https://gitea.sysmocom.de/sysmocom/sysmo-usim-tool.git
```

1.2. Configuración de la Tarjeta

Para verificar el estado de la conexión en el que se encuentra la tarjeta es necesario usar `pcsc_scan` el cual verifica el estado entre el computador y el lector de tarjetas, el cual nos indica el tipo de tarjeta que se está utilizando, Figura 1.



```
Jonatrivino@Jonatrivino-HP-Notebook: ~/Downloads
Jonatrivino@Jonatrivino-HP-Notebook:~/Downloads$ sudo pcsc_scan
Using reader plug'n play mechanism
Scanning present readers...
0: Alcor Micro AU9560 00 00

Tue Oct  4 12:44:13 2022
Reader 0: Alcor Micro AU9560 00 00
Event number: 0
Card state: Card inserted.
ATR: 3B 9F 96 80 1F 87 80 31 E0 73 FE 21 1B 67 4A 4C 75 30 34 05 4B A9
+ TS = 3B --> Direct Convention
+ T0 = 9F, Y(1): 1001, K: 15 (historical bytes)
TA(1) = 96 --> F1=512, D1=32, 16 cycles/ETU
250000 bits/s at 4 MHz, FMax for F1 = 5 MHz => 312500 bits/s
TD(1) = 80 --> Y(i+1) = 1000, Protocol T = 0
-----
TD(2) = 1F --> Y(i+1) = 0001, Protocol T = 15 - Global interface bytes following
-----
TA(3) = 87 --> Clock stop: state H - Class accepted by the card: (3G) A 5V B 3V C 1.8V
+ Historical bytes: 80 31 E0 73 FE 21 1B 67 4A 4C 75 30 34 05 4B
Category indicator byte: 80 (compact TLV data object)
Tag: 3, len: 1 (card service data byte)
Card service data byte: E0
- Application selection: by full DF name
- Application selection: by partial DF name
- BER-TLV data objects available in EF.DIR
- EF.DIR and EF.ATR access services: by GET RECORD(s) command
- Card with MF
Tag: 7, len: 3 (card capabilities)
Selection methods: FE
- DF selection by full DF name
- DF selection by partial DF name
- DF selection by path
- DF selection by file identifier
- Implicit DF selection
- Short EF identifier supported
```

Figura 1. Lectura de la tarjeta sim.

Se puede observar que indica que la tarjeta está insertada y es reconocida. Adicionalmente nos muestra múltiples propiedades y configuraciones de la misma.

Posteriormente, al momento de realizar la programación de las tarjetas es necesario tener en cuenta los siguientes parámetros: conocer el **código IMSI** el cual es único para cada SIM card, el **Ki** y el **OPC**, que se tienen al momento de configurar el software. También es necesario poseer la **clave ADM1_KEY**, única para cada sim y que es proporcionada por el proveedor (en este caso Sysmocom). Por otro lado, los códigos **MCC** (Mobile Country Code) y **MNC** (Mobile Network Code) [43], los establece el usuario y estos indican el país y el operador de telefonía móvil.

Se procedió con la programación de la tarjeta, por lo que primero es necesario acceder a la carpeta donde clonamos el repositorio y ejecutamos el archivo `sysmo-isim-tool-sja2.py` especificando los parametros antes mencionados, así:

```
$ cd sysmo-usim-tool/           # Ingresar a la carpeta sysmo-usim-tool.
$ sudo python3 ./sysmo-isim-tool.sja2.py --help      # Parámetros
disponibles para la configuración de la sim.
$ sudo python3 ./sysmo-isim-tool.sja2.py -a 15281502 -J 901700123456789 -
K 00112233445566778899aabbccddeeff -O 63BFA50EE6523365FF14C1F45F88737D -N
70
```

Para especificar un comando es necesario usar un guión (-) y la letra que corresponde a dicho parámetro, en este caso con -a se indica la clave ADM_KEY que tiene la tarjeta sim que se está utilizando, con -J el IMSI que se le va a dar al usuario, la -K es el Ki, -O el OPC que son los que se ingresaron en srsRAN y por último -N que indica el MNC.

Una vez ejecutado el comando especificando los parámetros necesarios, se obtiene un mensaje que indica que la configuración de la tarjeta sim se ha hecho de forma satisfactoria, como se observa en la Figura 2.

```
Activities Terminal oct 4 16:06 Jonatrivino@Jonatrivino-HP-Notebook: ~/Downloads/sysmo-usim-tool

15 XOR
Jonatrivino@Jonatrivino-HP-Notebook:~/Downloads/sysmo-usim-tool$
Jonatrivino@Jonatrivino-HP-Notebook:~/Downloads/sysmo-usim-tool$ sudo python3 ./sysmo-usim-tool.sja2.py -a 15281502 -J 901700123456789 -K 00112233445566778899aabbccddeeff -O 63BFA50EE65
23365FF14C1F45F88737D -N 70
sysmoUSIM: SJA2 parameterization tool
Copyright (c)2019 Sysmocon S.f.r.l. GmbH

Trying to find card with ATR: 3B 9F 96 80 1F 87 80 31 E0 73 FE 21 18 67 4A 4C 75 30 34 05 4B A9
Initializing smartcard terminal...
* Detected Card IMSI: 901700123456789
  ISIM Application Installed
  USIM Application Installed

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

Writing IMSI value...
* Initializing...
* New IMSI setting:
  IMSI: 9901700123456789
* Programming...

Writing MNCLN value...
* Initializing...
* New MNCLN setting:
  MNCLN: 0x70
* Programming...

Writing KI value...
* Initializing...
* New KI setting:
  KI: 00112233445566778899aabbccddeeff
* Programming...

Writing OP value...
* Initializing...
* New OP setting:
  OP: 63bfa50ee6523365ff14c1f45f88737d
* Programming...

Done!
Jonatrivino@Jonatrivino-HP-Notebook:~/Downloads/sysmo-usim-tool$
```

Figura 2. Configuración de la tarjeta SIM.

Cabe recalcar, que si se llega a introducir una clave **ADM1** incorrecta, al momento de realizar la autenticación presentaba un error y nos informará que se redujo en 1 la cantidad de intentos que se poseen para programar la tarjeta antes de que esta sea bloqueada. Por lo que cada tarjeta sim cuenta con un total de tres intentos máximos.