THALES

EP SDK Overview

www.thalesgroup.com

Agenda

- Module info
- Tools Required
- SDK Setup
- SDK Folder Structure
- USB Enumeration
- Firmware Upgrade Process
- Compile your first program
- Download and Start Applications
- Limitations
- GINA Brief

Module info

- For EXSx2, Embedded Processing supported since Rel1.3
- For TXx2, Embedded Processing supported since Rel1.3
- For PLSx3, Embedded Processing supported since Rel1.0
- For detail, please help check with local Thales FAE

Tools Required

Python 2.7 (from doc\Getting_Started.pdf)

Linux

Install the Python 2.7.x package and pyserial 3.x.

```
# Install Python:
    $ sudo apt-get install python

# Show version of python:
    $ python -V

# Install pyserial:
    $ sudo apt-get install python-serial

# Show version of pyserial:
    $ sudo apt-cache policy python-serial
```

Windows

Install Python 2.7.x and pyserial 3.x.

```
# Download and install Python 2.7:
    https://www.python.org/downloads/release/python-2717/

# Add Python folder in PATH environment variable
    $ setx PATH "%PATH%;C:\Python27"

# Show version of Python:
    $ python -V

# Install pyserial:
    $ python -m pip install pyserial

# Show version of pyserial:
    $ python -m pip show pyserial
```

Tools Required - cont.

Cinterion EXS62-W/EXS82-W module

Module Type	LGA	Eval Board
EXS62-W	L30960-N6250-A130	L30960-N6251-A130
EXS82-W	L30960-N6200-A130	L30901-N6201-A130

Cinterion Development Kit

- B80 (L30960-N0040-A100)

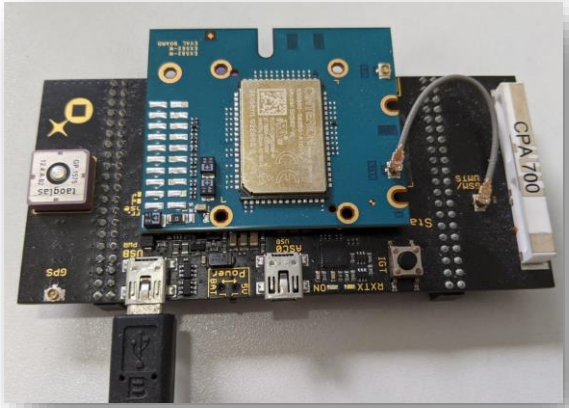
OR

- LGA Dev Kit (L30960-N0111-A100)
- LGA Dev Kit Socket (L30960-N0110-A100)

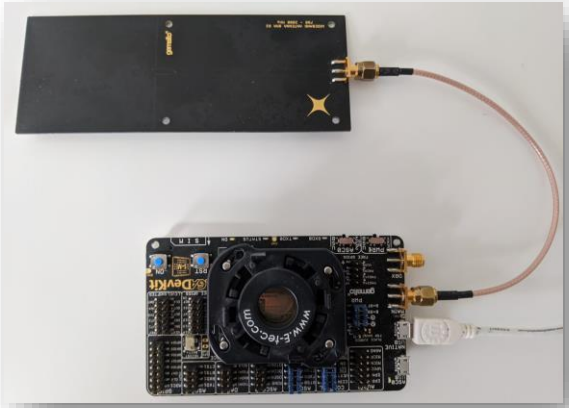
SIM Card

Host Machine running either

- Windows 7/10
- Linux (Ubuntu 18.04 LTS)



B80 Kit Setup



LGA Kit Setup

SDK Setup

- Copy the latest SDK release from the following folder:
  - *exs82_rev01.200_arn01.000.01_fw_048b\SDK*
- Version used for this demo is:
  - *exs82_rev01.200_arn01.000.01_fw_048b*
- EP SDK releases in the future may be handled differently after merging the EP building block with the Step 03 baseline.
  - Please check with Serval Step 03 Team for more details in the future

SDK Folder Structure

Root Folder	Sub folder	Description
SDK\	common\	Header files (GINA, ThreadX and QAPIs) + Precompiled libraries
	doc\	Documents
	drivers\	USB driver for EXSx2-W module
	examples\	Sample applications in C
	firmware\	EXSx2-W module firmware file
	tools\	LLVM compiler suite, firmware programming tool, application download tool
	README.txt	
	SDK_Release_Note.pdf	

USB Enumeration (Linux)

- Connect the native USB interface on the LGA DevKit to the PC
- Press the “ON” button on the LGA DevKit to start up the module.
- With a Linux PC please install the option driver for the vendor specific USB Diag port as follows:

```
# load the option driver
sudo modprobe option

# add new VID:PID to option driver
sudo sh -c "echo '1e2d 006B' > /sys/bus/usb-serial/drivers/option1/
new_id"
```

- The USB interfaces will then enumerate as follows:
- ttyACM0 Modem port for AT commands. No added driver necessary.
- ttyACM1 2nd USB port for AT commands. No added driver necessary.
- ttyUSB0 Diag port. Option driver required.

USB Enumeration (Windows)

It's necessary to install USB driver for Windows 7/10

- Device Manager enumeration
  - Cinterion USB Modem: Modem port for AT commands
  - Cinterion Ethernet Adapter: Ethernet adapter
  - Cinterion Diagnostics: Diagnostics port
  - Cinterion USB Com Port2: 2nd port for AT commands

Connect via USB

Open the connection via the native USB interface

```
# Connect to the target specifying your port (on windows) or your device
# (on linux) and speed. If successful, these communication parameters
# will be saved locally (config.db) for next commands.

# On linux (Note: Please add your user to the dialout group for accessing
# the tty devices without root privileges)
    $ python connect.py -p /dev/ttyACM0 -s 115200
    Connection with the target on /dev/ttyACM0 at 115200
    Done !

# On windows, specify the modem port as displayed in the figure shown
# in Section 2.4.
    $ python connect.py -p COM15 -s 115200
    Connection with the target on COM15 at 115200
    Done !
```

Firmware Update

- Use the firmware.py tool to read the boot loader version installed on your module. If the version is prior to the SBL1_118.usf file located under SDK/firmware, please download the new boot loader onto the module.
- Use the firmware.py tool to download the new firmware over ASC0 or USB. The new firmware is provided as *.usf file under SDK/firmware/ (e.g., exs82.usf).

Firmware Update - cont.

Windows 7/10	Linux	Comment
cd SDK\tools	**cd SDK/tools**	Go to the tool directory from SDK (on linux, use '/' instead of '\')
python connect.py -p COM1 -s 115200	**python connect.py -p /dev/ttyS0 -s 115200**	Connect to the target specifying your port (on windows) or your device #(on linux) and speed.
python firmware.py info Modem version (rev): 01.100 Applicative version (arn): 01.000.08 Bootloader version (sbl): SBL1_117	**python firmware.py info** Modem version (rev): 01.100 Applicative version (arn): 01.000.08 Bootloader version (sbl): SBL1_117	Read your current boot loader and firmware version
python firmware.py update ..\firmware\SBL1_118.usf	python firmware.py update ..\firmware\SBL1_118.usf	Download the new boot loader onto the module if required
python firmware.py update ..\firmware\exs82.usf	python firmware.py update ..\firmware\exs82.usf	Download the new firmware onto the module
python firmware.py --help	python firmware.py --help	For more firmware commands and options please consult help

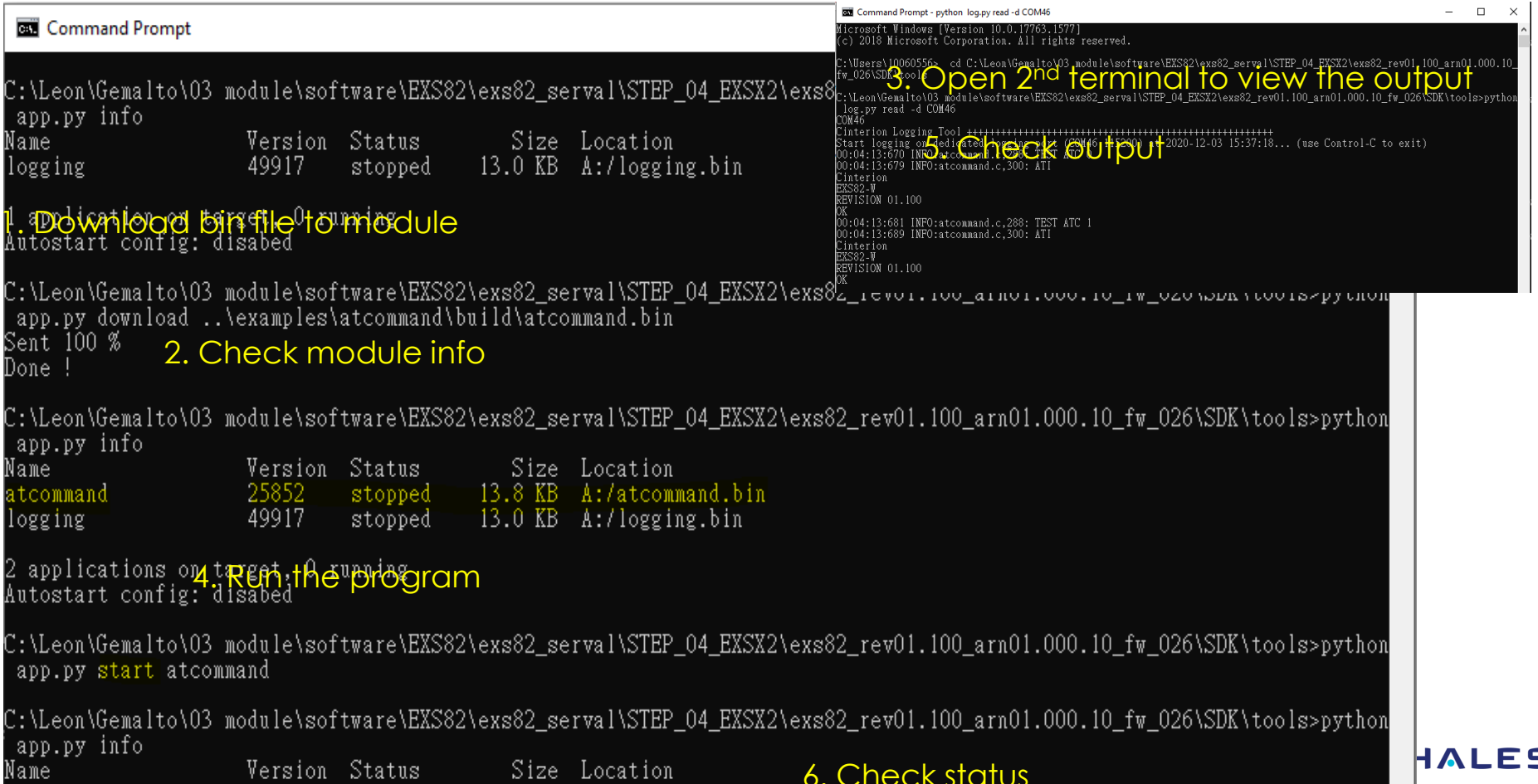
Compile your first program

- Goto any project under the examples folder
- Simply run the following script file
  - **Linux:** *./build.sh*
  - **Windows:** *build.bat*
- After the build is successful, a "build" folder is created with the *project.bin* file inside.
- The *project.bin* file is loaded in the module to run

Download and Start Applications

- Application binary is SDK\examples\xxxx\build\xxxx.bin
- List the application loaded on the target
  - python app.py info
- Download your application on the target
  - python app.py download ..\examples\xxxx\build\xxxx.bin
- To view the output, open a second terminal to start the logging tool on the 2nd USB port
  - python log.py read -d COM41
- Switch back to the first terminal and start the application
  - python app.py start xxxx
- Stop your application
  - python app.py stop xxxx

Downloading and Starting Applications - cont



DEMO

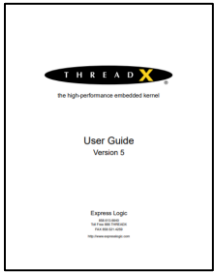
Limitations

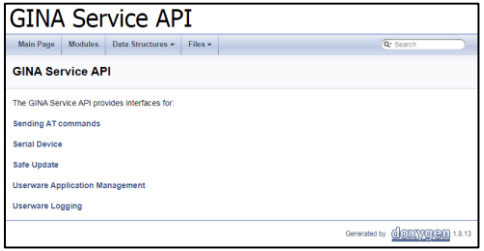
- Only 1 application can run at a time
- Stop 1 application to start the other again
- Auto start your application on modem power up by renaming the project.bin file to oem_app.bin

SDK Service APIs - GINA Brief

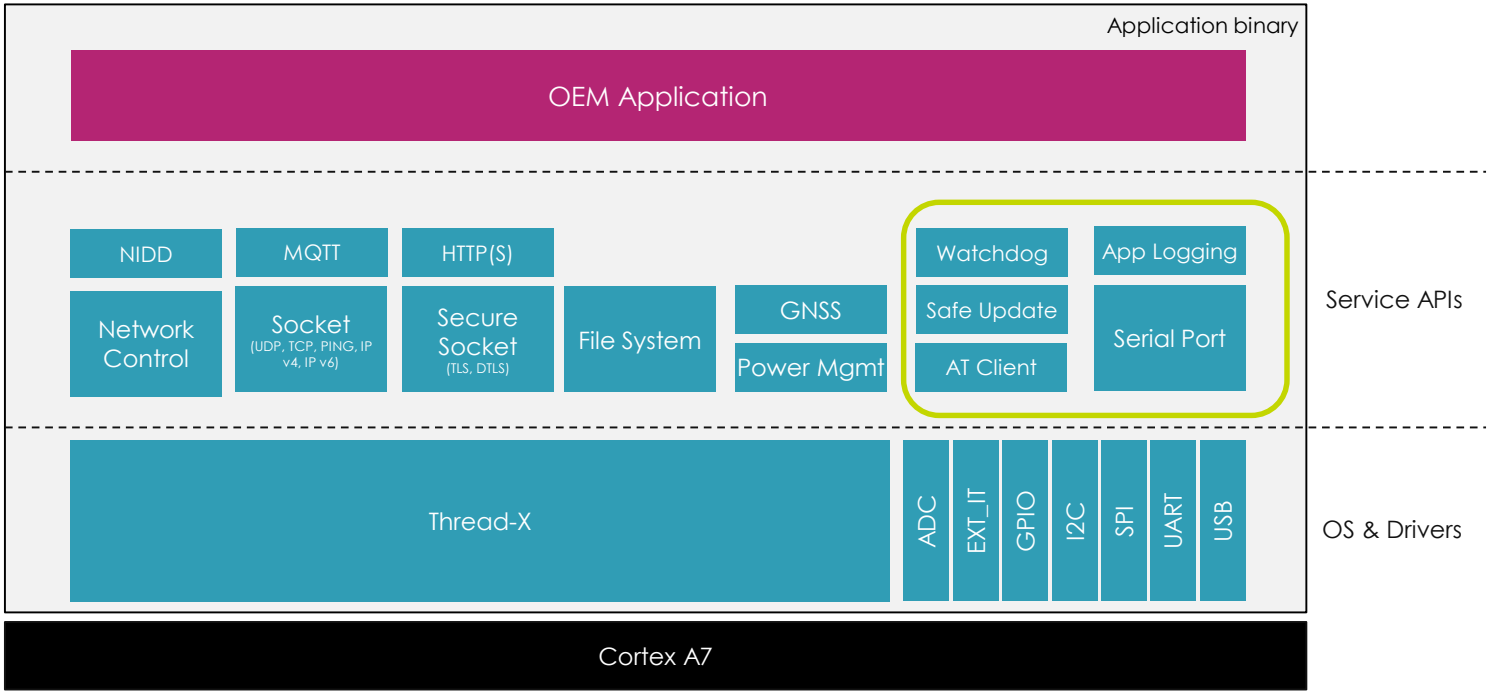
Thales is exposing following Service APIs for OEM app

- Thread-X RTOS API
  - For Operating System interface: ThreadX User Guide
- Qualcomm QAPI
  - For hardware peripherals (I2C, ADC...)
  - For Sockets, IP protocols & File System
  - For modem functionalities
- Thales GINA Service API (Generic Interface for Applications)
  - A (complete) C-based platform abstraction layer
    - Unified serial interfaces (for any UART or USB-CDC ACM)
    - Safety procedures for software update
    - Application management (watchdog, cleanup) and logs
    - Security services





RunTime Architecture



OEM app

Thales Services & Thread-X

Developing Embedded Applications

API Document	Chapter	Status	Example
Gina_Service_API	Sending AT Commands	Supported	SDK\examples\atcommand
Gina_Service_API	Serial Device	Supported	SDK\examples\serial
Gina_Service_API	Safe Update	Supported	SDK\examples\update_fw SDK\examples\update_app
Gina_Service_API	Userware Logging	Supported	SDK\examples\logging
80-P8102-1	4 - DSS Net Control APIs	Supported	SDK\examples\ping SDK\examples\non_ip
80-P8102-1	5 - QAPI Networking Socket	Supported	SDK\examples\sockv4 SDK\examples\sockv6
80-P8102-1	6 - QAPI Network Security APIs	Supported	SDK\examples\tls
80-P8102-1	7 - QAPI Networking Services	Supported	SDK\examples\ping
80-P8102-1	8 - Domain Name System Client Service APIs	Supported	SDK\examples\dns
80-P8102-1	9 - MQTT API	Supported	SDK\examples\mqtt
80-P8102-1	10 - HTTP(S) APIs	Supported	SDK\examples\http
80-P8102-1	13 - LWM2M APIs	Not supported	
80-P8102-1	14 - AT Forward Service Framework	Not supported	
80-P8102-1	15.1 - Driver Access APIs for the DAM Application Space	Not supported	
80-P8102-1	15.2 - Command Line Interface	Not supported	
80-P8102-1	16 - Sensor Manager APIs	Supported (roadmap)	
80-P8102-2	4.1 - GPIO Interrupt Controller APIs	Supported	SDK\examples\gpio SDK\examples\gpio_int
80-P8102-2	4.2 - PMM APIs	Not supported	
80-P8102-2	5 - Diagnostic Services Module	Not supported	
80-P8102-2	6 - Pulse Width Modulation	Not supported	
80-P8102-2	7.2 - File System APIs	Supported with known issues	
80-P8102-2	7.3 - FTL Data Types and APIs	Not supported	
80-P8102-2	8.2 - USB APIs	Not supported[1]	
80-P8102-2	9.1 - I2C Master APIs	Supported	SDK\examples\i2c
80-P8102-2	9.2 - SPI Master APIs	Supported (roadmap)	
80-P8102-2	9.3 - UART APIs	Not supported[1]	SDK\examples\serial
80-P8102-2	10.1 - Timer APIs	Supported	SDK\examples\helloworld
80-P8102-2	10.2 - PMIC RTC APIs	Supported (roadmap)	
80-P8102-2	10.3 - PMIC Battery Status Information	Supported (roadmap)	
80-P8102-2	11.2 - ADC APIs	Supported	SDK\examples\adc
80-P8102-2	11.4 - TSENS APIs	Supported (roadmap)	
80-p8102-3	4 - GNSS Location Driver	Supported	SDK\examples\location
80-p8102-4	3 - System Power Save Management	Supported (roadmap)	SDK\examples\psm

1. API has been replaced by Serial Device from Gina Service API.