lab.SCHC FullSDK
A SCHC implementation in C

Javier Fernandez
lab.SCHC

Actility

IPCore  SDK

OpenSCHC

IETF 120 - Vancouver
OpenSCHC

IETF 120 - Vancouver
Meet lab(dot)SCHC

Welcome to lab.SCHC, the research and development laboratory dedicated to advancing the SCHC mechanism for the Internet of Things (IoT). Our mission is to make SCHC accessible to everyone by developing innovative solutions that optimize communication and data transfer. Join us as we pave the way for a more connected and sustainable future.

FullSDK New! Resources Contact Us
Resources

lab.SCHC/fullSDK
Industrial-grade Open Source embedded SCHC implementation
lab.SCHC FullSDK GitLab project

lab.SCHC FullSDK Docs
Documentation for SCHC SDK
Requisites, building and debugging, API functions, and more!

Actility IPCore
SCHC Gateway implementation for business
High performance and stability

OpenSCHC
Open source SCHC implementation in Python
For SCHC Gateway and Device
lab.SCHC full-sdk-delivery

- lab-schc.fr

- CoAP CBOR, ...

- LoRaWAN lbm, loramac
- Sigfox, NB-IoT

- examples
- libs
- os

- full-sdk

- platforms

- AT Modem
- BareMetal, ...
- FreeRTOS

- NUCLEO-L476RG + SX1272 | SX1276
- B-L072Z-LRWAN1
- NRF9160 DevKit
## lab.SCHC Docs

- **lab-schc.github.io**

1. **The SCHC Mechanism**:
   - Learn the basics of SCHC
   - The SCHC adaptation layer

2. **Getting started**:
   - Requisites and compatibility
   - Building and debugging applications

3. **FullSDK Concepts**:
   - General architecture
   - Available interfaces
   - How to use

4. **FullSDK Reference Manual**:
   - Public API functions
   - Sample applications

### IETF 120 - Vancouver

<table>
<thead>
<tr>
<th>Requisites and Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>The hardware, operating systems, libraries, and third-party software in this section are recommended.</td>
</tr>
</tbody>
</table>

#### AT Modem UDP Client

This example illustrates an end-to-end communication from a client and downlink packet transmission from a device (a board for the example). The diagram below depicts the overall architecture of the lab.SCHC FullSDK.

**AT Modem UDP Client**

Example:

```
cd lab-schc-examples/at-modem/udp-client/
head README.md
```

#### Operation

This example considers a UDP server application with the IP address

```
cd full-sdk/delivery/
# Build:
cmake -S .. -B .. 
-DAPP_NAME=ATModem 
-DAPP_VERSION=1.0.0 -DFULLSDK_VERSION=3.0.0 
-DTOOLCHAIN=arm-none -DTARGET=m4 -DPLATFORM=STM32 
-DL2_STACK=mbedtls -DNUCLEO_LORA_SHIELDS=STM32L476 
```

```
cmake -S .. -B .. 
-DAPP_NAME=ATModem 
-DAPP_VERSION=1.0.0 -DFULLSDK_VERSION=3.0.0 
-DTOOLCHAIN=arm-none -DTARGET=m4 -DPLATFORM=STM32 
-DL2_STACK=mbedtls -DNUCLEO_LORA_SHIELDS=STM32L476 
```
lab.SCHC FullSDK

- Building & Flashing

- CMake
- Openocd
- GNU ARM Toolchain
  + LoRa Semtech stack source
- fpvgcc

Example output:

<table>
<thead>
<tr>
<th>FILE</th>
<th>VEC</th>
<th>FLASH</th>
<th>RAM</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>libfullsdk.a</td>
<td>26848</td>
<td>2884</td>
<td>29732</td>
<td></td>
</tr>
<tr>
<td>LoRaMac.c.obj</td>
<td>11532</td>
<td>1716</td>
<td>13248</td>
<td></td>
</tr>
<tr>
<td>libgcc.a</td>
<td>9880</td>
<td>9880</td>
<td></td>
<td>9880</td>
</tr>
<tr>
<td>command.c.obj</td>
<td>4279</td>
<td>2588</td>
<td>6867</td>
<td></td>
</tr>
</tbody>
</table>

```
cmake -S . -B ./build
-DAPP_NAME=<app_name>
-DTOOLCHAIN=<toolchain>
-DTARGET=<target>
-DL2_STACK=<l2_stack> && make -C ./build
```

```
OPENOCD_TARGET=<openocd.cfg>
BIN_FILE=build/<toolchain>/<target>/<app_name>.bin make -C openocd/flash
```
Conclusion

1. Go to lab-schc.fr
2. Clone the repo.
3. Look around & try it out.
4. Email us at contact@lab-schc.fr

FEC, OpenSCHC Frag Interoperability and more. Coming up!

DEMO?