Release notes for Simone, version V0.6.4

Release Information

Project	Simone
Milestone	Milestone 4 – Library delivered incl. demonstration in development complete quality level
Software Version	V0.6.4
Software Life Cycle Stage	Beta
Release Date	24 May 2016

The supplied software is beta software and is as such not at the quality level of the final product. Occasional faults and some inconsistency in operation may be experienced. Observations made during evaluation of this software are welcome. Please contact raf@rtx.dk for support or to report observations

Contents

Version info

RTX GIT server tag	ReleaseV0.6.4 13fed396f1d30f02ce966802c4c883e251f4d32a
Contiki snapshot ID	bf41de1be5432570dd7d84cc7e3e85d0515efb0c
Ubuntu	Ubuntu 14.04.3 LTS 64 bit

What's new

This is a release of the SIMONE project, which targets delivery of 6LoWPAN libraries for ULE for integration of 6LoWPAN functionality on top of existing ULE transport layer implementations.

This release includes editorial changes only – incl. addition of LICENSE file and replacement of RTX headers to UA headers.

Changes in this release

- Addition of V064\ReleaseV0.6.4.zip\ule6lowpanV0.6.4\LICENSE
- Removal of ule6lowpan\doc\dnsmasq\dnsmasq.conf
- Replacement of the headers of 'RTX developed code' with the proposed UA header

Known limitations

Hardware integration

The libraries are tested in a testbench, but not yet integrated into a fully functional ULE platform. Integration on top of existing ULE transport layer implementations is expected to require adaption to the library. Follow-up releases must be expected.

API implementation

All API's are defined with primitives and type definitions, but only a subset of the API's have been implemented with functionality. API's indicated with 'Stub' in the table below are not implemented.

6LBR Interface

ule6loGI_init	Full	
ule6loGI_getStatus	Full	
ule6loGI_getIp6addr	Full	
ule6loGI_getDomain	Deleted	
ule6loGI_addMulticastAddr	Full	
ule6loGI_removeMulticastAddr	Full	
ule6loGI_setMacAddress	Full	
ule6loNI_receive	Full	
ule6loNI_send	Stub	
ule6loNI_echoRequest	Full	
ule6loLLI_init	Stub	(The IPEI is currently obtained on a fake DECT registration call)
ule6loLLI_receive	Full	
ule6loLLI_send	Stub	
ule6loLLI_delivered	Stub	
ule6loOS_processRun	Full	
ule6loOS_getMACAddr	Full	
ule6loOS_getTimerTick	Stub	
ule6loTestIn_init	Full	
ule6loTestIn_deinit	Full	
ule6loTestIn_reset	Full	
ule6loTestIn_getNbListSize	Full	
ule6loTestIn_getNbList	Full	

ule6loTestIn_getnofSentPacket	Full	
ule6loTestIn_getnofReceivedPacket	Full	
ule6loTestIn_regRxHook	Full	
ule6loTestIn_regTxHook	Full	

6LN Interface

ule6loGI_init	Full	
ule6loGI_getStatus	Full	
ule6loGI_getIp6addr	Full	
ule6loGI_getDomain	Deleted	
ule6loGI_addMulticastAddr	Full	
ule6loGI_removeMulticastAddr	Full	
ule6loGI_setMacAddress	Full	
tcp_socket_register	Full	
tcp_socket_connect	Full	
tcp_socket_listen	Full	
tcp_socket_unlisten	Full	
tcp_socket_send	Full	
tcp_socket_send_str	Full	
tcp_socket_close	Full	
tcp_socket_unregister	Full	
udp_socket_register	Full	
udp_socket_close	Full	
udp_socket_bind	Full	
udp_socket_connect	Full	
udp_socket_send	Full	
udp_socket_sendto	Full	
resolv_query	Full	
resolv_lookup	Full	
ule6loLLI_init	Full	
ule6loLLI_receive	Full	
ule6loLLI_connected	Full	
ule6loLLI_check	Full	
ule6loLLI_send	Full	
ule6loLLI_delivered	Stub	
ule6loOS_processRun	Full	
ule6loOS_getMACAddr	Full	
ule6loOS_getTimerTick	Stub	

ule6loTestIn_init	Full	
ule6loTestIn_deinit	Full	
ule6loTestIn_reset	Full	
ule6loTestIn_getNbListSize	Full	
ule6loTestIn_getNbList	Full	
ule6loTestIn_getnofSentPacket	Full	
ule6loTestIn_getnofReceivedPacket	Full	
ule6loTestIn_regRxHook	Full	
ule6loTestIn_regTxHook	Full	

Installation / Upgrading

Please refer to the getting started document which is included in the source release archive:
ule6lowpan.zip\ule6lowpan\doc\ule6lo_getting_started.docx

Find the release on:
https://redmine.et-it.hs-offenburg.de/attachments/download/4208/Releasev064.zip

Contact information

For further information and support contact: Rasmus Fossá <raf@rtx.dk>

Release History

Milestone:	Milestone 4 – Library delivered incl. demonstration in development complete quality level
Software Version:	V0.6.2
Software Life Cycle Stage:	Beta
Release Date:	9 March 2016

Milestone:	Milestone 4 – Library delivered incl. demonstration in development complete quality level
Software Version:	V0.5.0
Software Life Cycle Stage:	Development
Release Date:	20 January 2016

Milestone:	Milestone 4 – Early interim release of border router for integration purposes
Software Version:	V0.4.0
Software Life Cycle Stage:	Development
Release Date:	16 December 2015

Milestone:	Milestone 3 – Early interim release of software for integration purposes
Software Version:	V0.3.0
Software Life Cycle Stage:	Development
Release Date:	6 November 2015

-

Appendix: RTX Standard Software Lifecycle Model

The software progresses through a series of life cycle stages as the software matures and quality is improved.

Development Stage

Releases made during the development stages are mainly for internal purposes only but can to some degree be shared with external development partners. Development software contains defects and has missing or broken features and possibly unresolved performance issues.

Alpha Stage

Alpha software is “Code Complete” thus all features are implemented for the planned feature release, integrated and functional. Alpha releases are the first releases made that enable measurement of overall product quality. Alpha software is verified against the product requirements with system testing. The alpha software contains defects but must at least live up to following conditions:

All planned features and functions are usable (no feature groups are blocked from testing)

Stability issues do not impact system testing significantly.

Beta Stage

Beta software is software that has been matured during alpha stage to a level where e.g. field testing is possible. I.e. the software is of a quality where unsupervised end-user trials can be made. System verification and defect fixing continues in the Beta stage but other trials can be commenced that verify other aspects of the product.

Release Candidate (RC)

RC software is the software delivered for e.g. product acceptance testing. It is the software that the project promotes as a candidate for the final software that fulfils the goals for this software product (for this feature revision of the product).

General Availability (GA)

RC software can be raised to General Availability (GA) software when the software has been approved by the customer.