Quick Start Guide

SmartAxiom Blockchain Secured,
Azure IoT Cloud Management Solution for
Renesas RA and RZ Evaluation Kits

RENESAS RA

RENESAS RZ

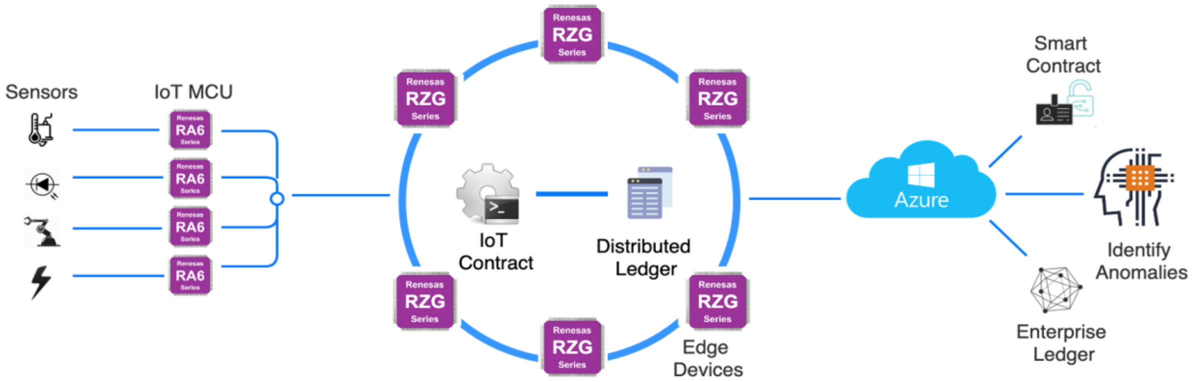
Contact:

SmartAxiom, Inc.
530 Technology Drive, Suite 100
Irvine, CA 92618, USA
www.smartaxiom.com
support@smartaxiom.com
Tel: +1 408 220 6028

Table of Contents

Solution Overview

SmartAxiom software manages and secures IoT devices through patented, lite blockchain technology running among those devices at the edge of the Internet - enabling them to defend themselves. Our peer-to-peer distributed ledgers improve security, latency, reliability and manageability. We uniquely create the first true endpoint-to-cloud blockchain solution, while our IoT Smart Contracts push intelligence to the edge. Our solution:



Beachhead
Endpoint Trust & ID

BlockLock
Edge Blockchain

Tenacious
Cloud Management

Key Features and Benefits

- **Easy evaluation and development:** Proven on RA MCUs and RZ MPUs
- **Low latency:** Local response much faster than cloud/server-based security.
- **Redundant and Reliable:** No single point failure at security server or its network.
- **Resilient & decentralized** ledger-based IoT security fabric.
- **Auto-provisioning** and **Tiered Clustering**
- **Scalable & Flexible:** keyless peer authentication reduces certificates and costs.
- **Secure:** To break in, majority of the nodes must be hacked between events.
- **Zero Trust framework model**
- **Stateful, hash-based signatures** on every data block.
- **Role based** access control policies.

Target Markets and Applications:

- Smart Buildings
- Logistics & Shipping
- In and between
- Industrial IoT
- Building Security
- EV Charging Transactions

Quick Start Guide Introduction

This Quick Start Guide includes step-by-step instructions to install and run SmartAxiom Beachhead endpoint firmware on RA6M3 with BlockLock blockchain security on RZG or in cloud containers, all synchronized and managed through Tenacious on Azure cloud.

Version History

Version	Date	Description
1.0	July 28, 2020	Launch
1.2	Feb 17, 2021	Adds BlockLock on RZG
1.3	Feb 22, 2021	Corrections and URLs.

References

[1] Evaluation Kit for RA6M3 Microcontroller Quick Start Guide R20QS0011EU0102

[2] RZ/G2 Reference Boards Start-up Guide R01TU0279EJ0105

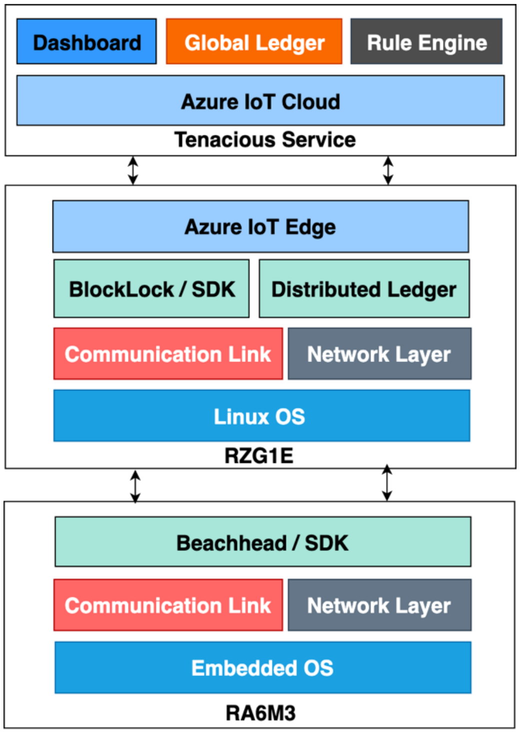
[3] Getting Started with the RZ/A2M Evaluation Board Kit R01QS0038EJ0103

Notice to User

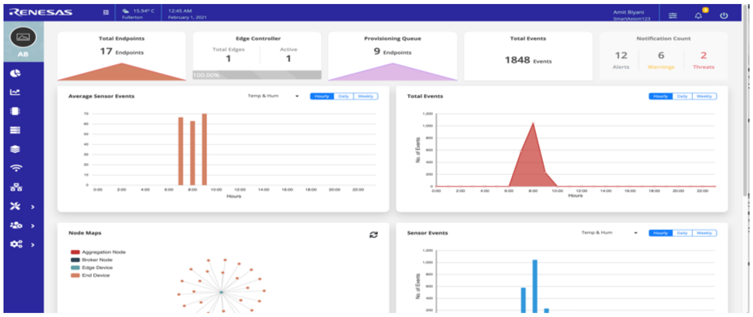
Use of the SDK-Beachhead, SDK-BlockLock and Tenacious Software is subject to terms and conditions included in the Terms of Use section below. Please read it before installation.

SmartAxiom® Beachhead, BlockLock and Tenacious are trademarks of SmartAxiom Inc.

SmartAxiom Demo Architecture



RENESAS
EVALUATION KIT
EK-RA6M3
Version 1
Evaluation Kit for RA6M3 MCU Group
Renesas RA Family of 32-bit MCUs with Arm Core
RA



The diagram above shows the three main components of the demo system:

1. SmartAxiom **Beachhead** endpoint firmware on RA6M3 or RZA2M, talking to
2. Our **BlockLock** blockchain security system on RZG (or in cloud containers), all synchronized and managed through
3. **Tenacious** on Azure cloud.

The SmartAxiom Beachhead and BlockLock SDK includes instructions, tools and sample code to allow a user to manage and secure wide variety of embedded and edge devices for different verticals like Industrial IoT, Smart Buildings and Logistics. It integrates with the Renesas RA and RZ platforms to provide a tailored, project-ready solution.

SmartAxiom's Tenacious is a multi-tenant, cloud-agnostic backend for device and system management. It also archives IoT data for reporting and big data analytics such as anomaly detection.

Beachhead Endpoint SDK

Beachhead SDK presents at the physical security layer and its main function is to generate encrypted blocks at the data source to protect devices. This is optimized with security measures including signed software images, root-keys, secure boots, and tamper-proofing, so sensors and devices are secured even if they are damaged or stolen.

Beachhead has been ported to the RA6M3 and integrates with RA platform. It supports FreeRTOS and will soon include Azure RTOS support.

The Beachhead SDK includes:

- Secure MQTT 3.1.X, HTTPS and TCP/UDP
- Crypto Engine – SHA256/SHA224/SHA512, ECDH, ECDSA (ED25519 and ED448)
- Handshaking and Self Registration - Encryption key establishment
- Sync-envelope – SmartAxiom's time stamping and verification protocol
- Root-identity
- Establishing Master Clock Time
- Secure Boot (as supported by RA families)

The sync-envelope message format wraps the payload with an authentication header and a signature. The complete message, including header, payload, timestamp and signature are combined in a serialized message pack array.

The section below Beachhead Endpoint Setup on RA6M3 includes step-by-step Beachhead integration instructions and sample code.

BlockLock Edge SDK

BlockLock SDK is a decentralized connectivity and security framework that provides IoT device management and autonomous secure connection and security at the edge. Its distributed ledger system implements intelligent IoT smart contracts between the devices and the IoT ecosystem. Its main functionality is to authenticate endpoint source identity and reputation and verify communication with its peers.

BlockLock has been ported to RZ/G2E and integrates with RZ platform. It is OS-agnostic software for edge devices but for applications without edge gateways or controllers it is also deployable in the cloud through docker images.

The BlockLock supports Azure through Azure runtime IoT edge. This Demo Kit installs the BlockLock SDK on RZ for Azure, and is fully integrated with Azure IoT Hub, Stream Analytics, and PowerBI Visualization. The BlockLock SDK includes:

- Device onboarding and provisioning
- Device authentication and identity
- Digital ownership and authorization
- Device and endpoints grouping
- Chord Peer-to-Peer network via LAN/WAN/Wi-Fi/BLE
- Decentralized and distributed ledger system
- IoT Smart Contracts – to push intelligent at edge
- Transaction Verification – Consensus algorithm with Proof-of-stake

It will support flexible network connectivity options, including wired Ethernet, Wi-Fi and LTE. BlockLock is integrated with Azure runtime IoT edge module for secure connectivity to the Azure IoT Hub.

The section below BlockLock Edge Setup on EK-874 with RZ/G2E includes step-by-step instructions for BlockLock integration, provisioning and setup through Azure runtime IoT edge to the Azure IoT Hub.

Tenacious Cloud Management SDK

Tenacious is a cloud-agnostic service for archiving, monitoring and managing IoT devices and their security. It’s includes a global ledger system, a customizable dashboard and a rule engine. The Tenacious management and visualization tool includes:

- Multi-tenant cloud service with customer management
- Customer sign-on and mapping of evaluation boards
- Interactive dashboard with data visualization
- Global Ledger - synchronizes with BlockLock edge nodes and aggregates the multiple distributed ledgers
- Rule Engine – Create rules for IoT contracts, for permissions, firmware updates, etc.

The global ledger offers a broad range of features to store and forward device sensor data for visualization. Using Stream Analytics and PowerBI, event payloads can be pictured through Visualization tiles to graphically display time-series, graph and meter views.

Tenacious includes customer onboarding, device and user management and sensor and device mapping. It will also provide a tutorial to create rules for IoT contracts and Tenacious SDK to provide access device and their events data. It will further integrate with enterprise AI and ML application.

System Requirements

You need a desktop or laptop PC running Windows 10 x64.

Install Segger J-link debug drivers and flash programming software https://www.segger.com/downloads/jlink/JLink_Windows_V688b.exe

In case of any issue please refer to J-Link installation guide and RA EK-RA6M3 evaluation guide.

Beachhead Endpoint setup on EK-RA6M3

The SDK-Beachhead Demo uses the following configuration as shown in the

Setup section below:

- Renesas RA EK-RA6M3 Evaluation Kit.

  https://www.renesas.com/us/en/products/microcontrollers-microprocessors/ra-cortex-m-mcus/ek-ra6m3-ra6m3-mcu-group-evaluation-board

- Renesas RA e2studio 2020-10, Download and install Renesas RA e2studio on the Windows PC.

  https://www.renesas.com/us/en/software-tool/e-studio

  Eclipse-based Renesas integrated development environment (IDE).

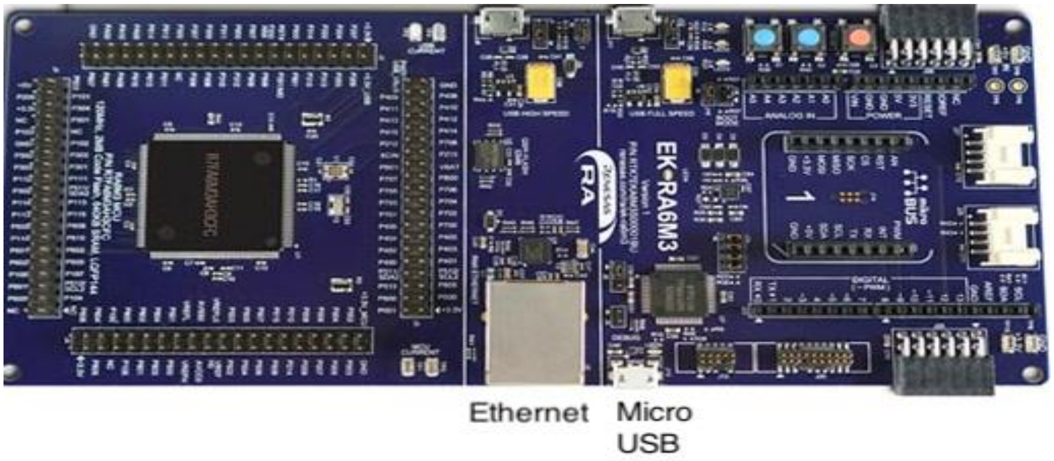
- Renesas RA FSP v2.1.0 and download and install on the Windows PC.

  https://www.renesas.com/us/en/software-tool/flexible-software-package-fsp

  FSP provides production ready peripheral drivers, FreeRTOS and middleware stacks to take advantage of the FSP ecosystem

- Micro USB cable for debugging and powering board- Debug port

- Ethernet cable for communication



Download and Unzip SDK-Beachhead Package

SDK-Beachhead and Sample code available by download zip or tar.xz files from:

https://www.smartaxiom.com/partners/renesas/ek-ra6m3

Unzip SDK-Beachhead package

Then double click SDK-Beachhead-ek-ra6m3.zip

or

```
$ tar -xzf SDK-Beachhead-ek-ra6m3.tar.xz
```

Or for those that prefer, you can download from a git repository. The link will be viewable after registering on the SmartAxiom download page here:

https://www.smartaxiom.com/partners/renesas/ek-ra6m3

SDK-Beachhead package includes the following files and sample source files

- SDK Library - libbeachhead.a
- Includes – beachhead.h and error.h
- User Developer Guide

ek-ra6m3 Sample code

- ek-ra6m3 firmware
- Sample code for ADC
- Sample code for push button
- Build script
- Flash script

Board Configuration and Setup

RA6M3 board configuration and setup steps to run SDK-Beachhead sample applications:

1. Connect micro-USB cable to PC and another end will connect to debug port J10 on RA6M3 to power the board
2. Ethernet port J5 connect to local network for communication
3. Run J-Link RTT Viewer for console messages
4. Program the sample application to the board through one of the two following methods:

   Through J-Flash Lite:

   a. Run J-Flash lite v6.94 for flash firmware or sample application

b. Configure J-Flash lite tools:
c. Device type: R7FA6M2AF
d. Interface (default): SWD, at 4000 kHz

Or:

Use a batch process to flash Beachhead-xxx-sample.srec using a script beachhead-flash-ra6m3.bat

You should see a J-Link RTT viewer console log after connecting to RA6M3 board such as this:

```
LOG: RTT Viewer disconnected.
LOG: Connecting to J-Link via USB...
LOG: Device "R7FA6M2AF" selected.
LOG: Found SW-DP with ID 0x5BA02477
LOG: Found SW-DP with ID 0x5BA02477
LOG: DPIDR: 0x5BA02477
LOG: Scanning AP map to find all available APs
LOG: AP[2]: Stopped AP scan as end of AP map has been reached
LOG: AP[0]: AHB-AP (IDR: 0x24770011)
LOG: AP[1]: APB-AP (IDR: 0x44770002)
LOG: Iterating through AP map to find AHB-AP to use
LOG: AP[0]: Core found
LOG: AP[0]: AHB-AP ROM base: 0xE00FF000
LOG: CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
LOG: Found Cortex-M4 r0p1, Little endian.
LOG: FPUnit: 6 code (BP) slots and 2 literal slots
LOG: CoreSight components:
LOG: ROMTbl[0] @ E00FF000
LOG: ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
LOG: ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
LOG: ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
LOG: ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
LOG: ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
LOG: ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
LOG: ROMTbl[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF
LOG: ROMTbl[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC
LOG: ROMTbl[0][8]: E0044000, CID: B105F00D, PID: 001BB101 TSG
LOG: RTT Viewer connected.
LOG: RTT Viewer disconnected.
LOG: Connecting to J-Link via USB...
LOG: Device "R7FA6M2AF" selected.
LOG: Found SW-DP with ID 0x5BA02477
LOG: Found SW-DP with ID 0x5BA02477
LOG: DPIDR: 0x5BA02477
LOG: Scanning AP map to find all available APs
LOG: AP[2]: Stopped AP scan as end of AP map has been reached
LOG: AP[0]: AHB-AP (IDR: 0x24770011)
LOG: AP[1]: APB-AP (IDR: 0x44770002)
LOG: Iterating through AP map to find AHB-AP to use
LOG: AP[0]: Core found
LOG: AP[0]: AHB-AP ROM base: 0xE00FF000
LOG: CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
LOG: Found Cortex-M4 r0p1, Little endian.
LOG: FPUnit: 6 code (BP) slots and 2 literal slots
LOG: CoreSight components:
LOG: ROMTbl[0] @ E00FF000
LOG: ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
LOG: ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
LOG: ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
LOG: ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
LOG: ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
LOG: ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
```

```
LOG: ROMTbl[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF
LOG: ROMTbl[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC
LOG: ROMTbl[0][8]: E0044000, CID: B105F00D, PID: 001BB101 TSG
```

Install SDK

Flash Beachhead-xxx-sample.srec sample application using J-link Lite or a batch file that comes with the package:

- Using J-Link Lite

1. Start J-Link Lite and select R7FA6M2AF
2. Select Data file Beachhead-xxx-sample.srec and program at default address
3. Program device
4. After flash sample code, press red reset button C65 on evaluation board
5. Start RA6M3 start booting up and LED light near debug port start flashing

- Using Batch File

  Install the SDK-Beachhead program to the EK-RA6M3 evaluation kit using the supplied scripts. From windows desktop explorer, go to the demo kit scripts folder

  C:/Renesas/RA6M3/SDK-Beachhead/ Beachhead-xxx-sample/scripts

  Then double click beachhead_ek-ra6m3 _demo.bat

  This runs the script to download and flash the Beachhead-xxx-sample demo to the EK-RA6M3 board.

  You should see a J-Link Commander console log similar to this:

```
SEGGER J-Link Commander V6.88b (Compiled Nov 27 2020 15:45:43)
DLL version V6.88b, compiled Nov 27 2020 15:44:11
J-Link Command File read successfully.
Processing script file...
J-Link connection not established yet but required for command.
Connecting to J-Link via USB...O.K.
Firmware: J-Link OB-S124 compiled Sep 21 2020 16:58:21
Hardware version: V1.00
S/N: 831004110
VTref=3.300V
Selecting SWD as current target interface.
Selecting 4000 kHz as target interface speed
Target connection not established yet but required for command.
Device "R7FA6M3AH" selected.
Connecting to target via SWD
Found SW-DP with ID 0x5BA02477
Found SW-DP with ID 0x5BA02477
DPIDR: 0x5BA02477
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
AP[1]: APB-AP (IDR: 0x44770002)
```

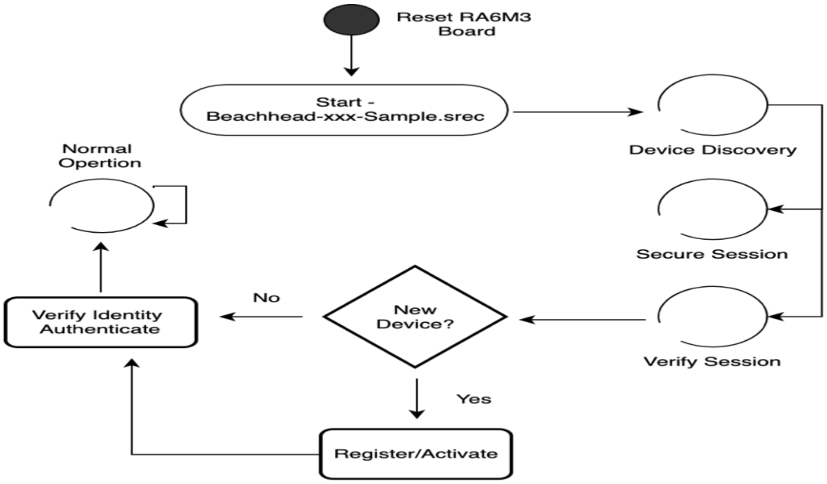
```
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
ROMTbl[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF
ROMTbl[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC
ROMTbl[0][8]: E0044000, CID: B105F00D, PID: 001BB101 TSG
Cortex-M4 identified.
Erasing selected range...
Comparing flash   [100%] Done.
Erasing flash     [100%] Done.
J-Link: Flash download: Total time needed: 2.397s (Prepare: 0.112s, Compare: 0.000s, Erase: 2.276s, Program: 0.000s, Verify:
0.000s, Restore: 0.008s)
J-Link: Flash download:
Flash sectors within Range [0x00000000 - 0x00200000] deleted.
Erasing done.
Downloading file [Release\Beachhead-adc-sample.srec]...
Comparing flash   [100%] Done.
J-Link: Flash download: Bank 0 @ 0x0100A150: Skipped. Contents already match
Erasing flash     [100%] Done.
Programming flash [100%] Done.
J-Link: Flash download: Bank 1 @ 0x00000000: 2 ranges affected (131072 bytes)
J-Link: Flash download: Total: 1.624s (Prepare: 0.180s, Compare: 0.010s, Erase: 0.000s, Program & Verify: 1.358s, Restore: 0.074s)
J-Link: Flash download: Program & Verify speed: 94 KB/s
O.K.
Found SW-DP with ID 0x5BA02477
Found SW-DP with ID 0x5BA02477
DPIDR: 0x5BA02477
AP map detection skipped. Manually configured AP map found.
AP[0]: AHB-AP (IDR: Not set)
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
ROMTbl[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF
ROMTbl[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC
ROMTbl[0][8]: E0044000, CID: B105F00D, PID: 001BB101 TSG
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
Script processing completed.
```

Run the Demo

- To register yourself or your organization in the cloud management system, visit this URL to access the Tenacious Signup page: https://renesas.smartaxiom.com/#/tenant-signup
- Enter the mandatory details and enter your official email ID, if using the application for your workplace.

- Upon completing the registration process, you will receive a verification email to verify your email ID, which will be treated as your username for subsequent logins.
- After the email verification process, you will be redirected to the Tenacious login page https://renesas.smartaxiom.com/
- Reset or power cycle ek-ra6m3 board

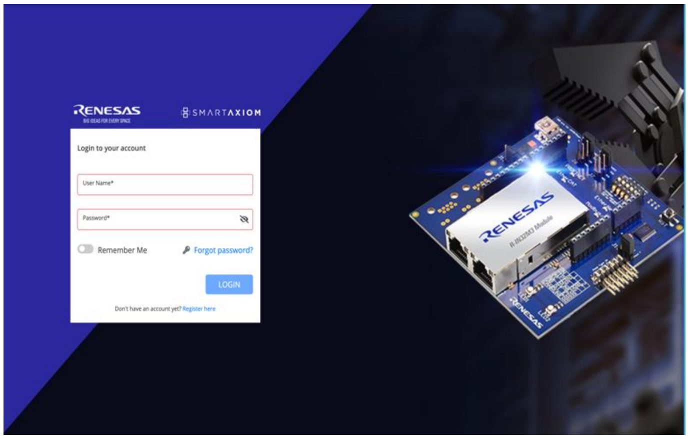
Console message

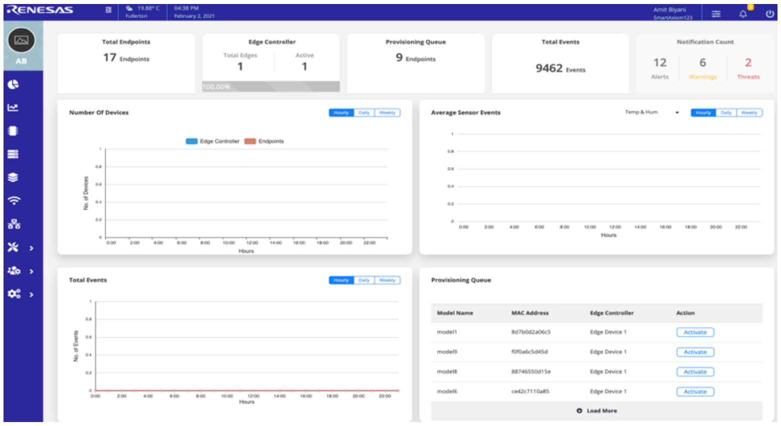


```
  *****************************************************************************************************
13> **** Welcome to Quick Start Beachhead example project for secure data transfer          ****
13> **** Example for Board EK-RA6M3 to demonstrate steps to use the secure library          ****
13> **** Module use: IO, UDP, FREERTOS etc                                         ****
13> **** For desired debug level set configuration file VARBOSE level                        ****
13> *****************************************************************************************************
13> *********************                  Initialize IP Information                  **********************
13> *****************************************************************************************************
13> .........Opening Communication Channel.......
13> Heap Size Socket Init: 28432
13> Initialize Beachhead Blockchain Engine......
13> Received total number of valid byte 196
13> ......Network Discovered.......
13> Retrieve block, state: 1......
13> Discovery complete WELCOME BEACHEAD .....
13> Remote Address 10.0.0.5 Port 18088 ....
13> Heap Size Socket Init: 27272
13> Discover responder.....
13> Received total number of valid byte 212
13> Complete Handshake process.......
13> Retrieve block, state: 2......
13> Setup a secure session.....
13> Received total number of valid byte 196
13> Verify secure session connection.....
13> Retrieve block, state: 7......
13> Verified Secure session...
13> Device has been provisioned and need to authentication...
13> Starting device Authenticate process.......
13> Received total number of valid byte 233
13> Retrieve block, state: 6......
13> Device Authenticate completed and ready to communicate.....
13> Discovery and Session Verification Complete 104......
13> Received total number of valid byte 232
13> Received total number of valid byte 232
13> Retrieve block, state: 6......
13> Device Received completed 6
```

```
13> Send Sensor Data
13> Received total number of valid byte 232
13> Received total number of valid byte 232
13> Retrieve block, state: 6......
```

- Use the email ID and password to login into Tenacious cloud management service https://renesas.smartaxiom.com/
- On Dashboard there is a Provision Queue option
- The Provision Queue shows the list of devices already registered to the system
- User can view endpoints devices in list and
- Click on Activate button for activating device and start receiving data from sensors





Total Endpoints: 17 Endpoints

Edge Controller: Total Edges 1, Active 1 (100.00%)

Provisioning Queue: 9 Endpoints

Total Events: 9462 Events

Notification Count: 12 Alerts, 6 Warnings, 2 Threats

Provisioning Queue

Model Name	MAC Address	Edge Controller	Action
model1	8d7b0d2a06c5	Edge Device 1	Activate
model9	f0f0a6c5d45d	Edge Device 1	Activate
model8	88746550d15e	Edge Device 1	Activate
model6	ce42c7110a85	Edge Device 1	Activate

Load More

BlockLock Edge setup on RZ/G2E Reference Board

The SDK-Beachhead Demo requires BlockLock Edge running in a local network. BlockLock can run on EK-RZG2E (64bit ARM Linux) or Windows or Linux PC.

BlockLock uses the following configuration as shown in the Setup section below:

- Silicon Linux RZ/G2E evaluation kit (EK874).

  https://www.si-linux.co.jp/index.php?CAT%2FCAT874

  https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-cortex-a-mpus/rzg-linux-platform/rzg-marketplace/board-solutions/rzgb000011

- 8GB or higher capacity microSD card.
- Software to program flash SD card image such as Balena Etcher.

  https://www.balena.io/etcher/

- Serial terminal software such as Tera Term.

  https://ttssh2.osdn.jp/index.html.en

- Ethernet cable for communication

- SD card image with BlockLock edge software.

https://www.smartaxiom.com/partners/renesas

blocklock_rzg2e_ek874_sd_8gb_ver_x.y.img.xz

Board Configuration and Setup

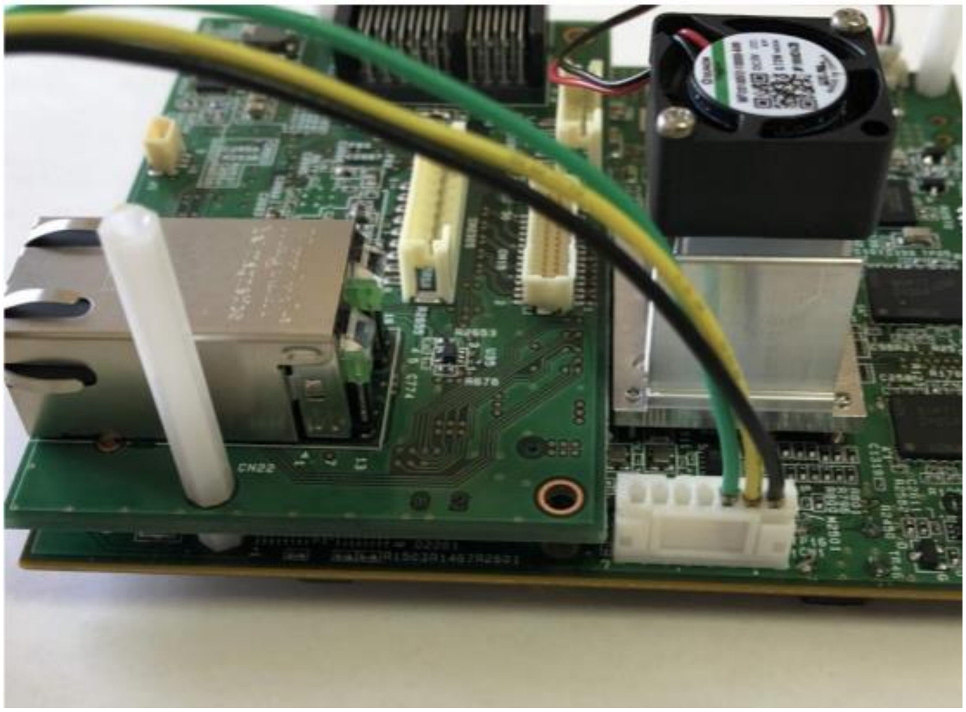
EK874 board configuration and setup steps to run BlockLock edge software:

1. No configuration is needed for board but refer to user guide if needed.

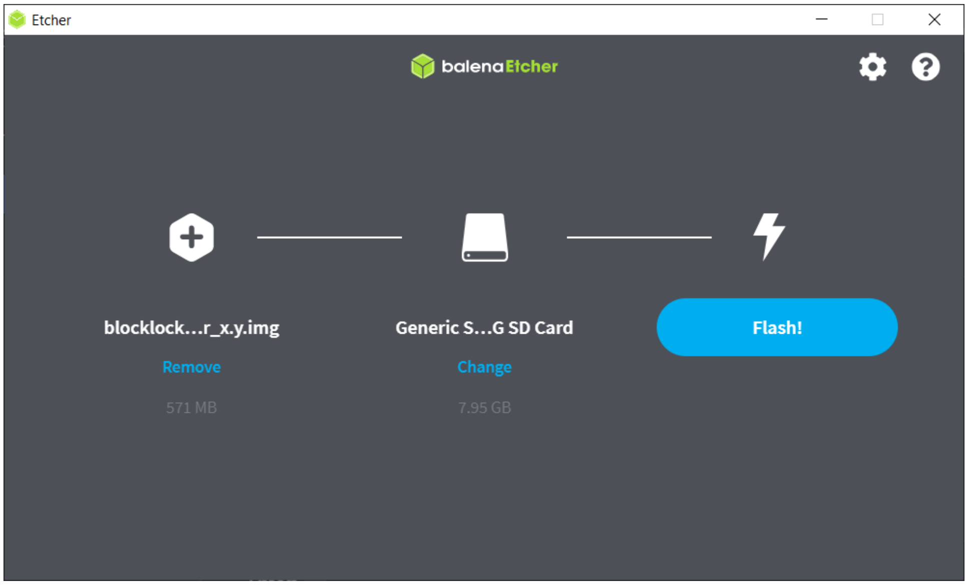
https://www.si-linux.co.jp/index.php?plugin=attach&refer=CAT%2FCAT874&openfile=CAT874_HardwareManual_20210112.pdf

2. Connect Ethernet cable for network communication.

3. Connect micro USB serial cable to access console.



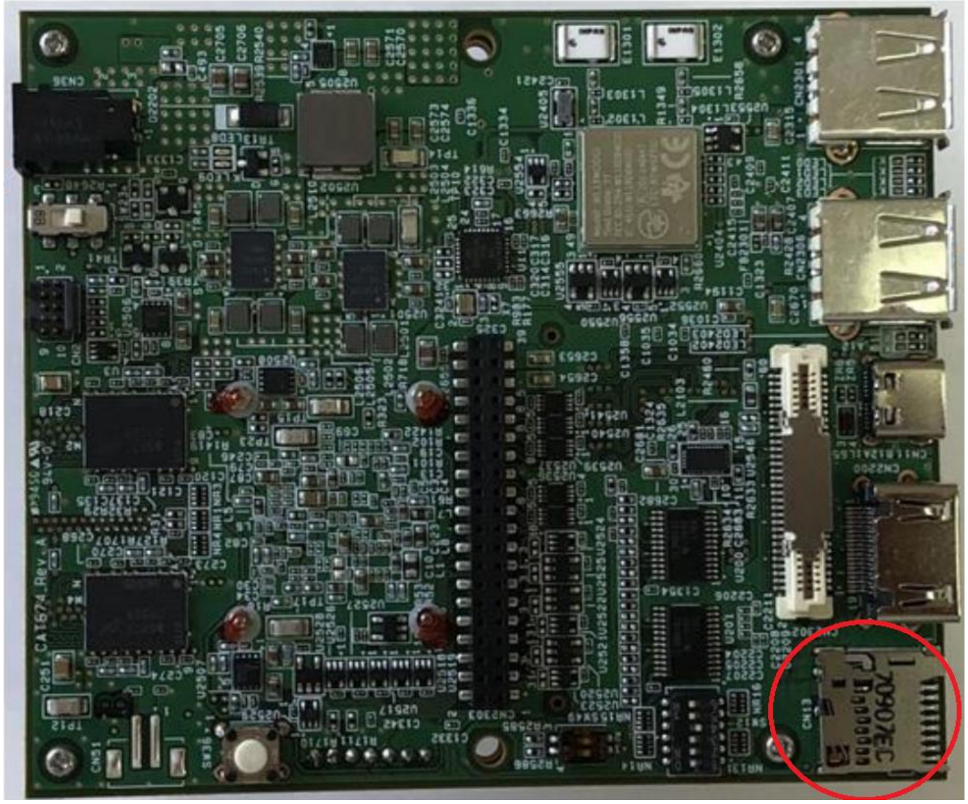
4. Flash the SD card image using Balena Etcher.



blocklock...r_x.y.img
Remove
571 MB

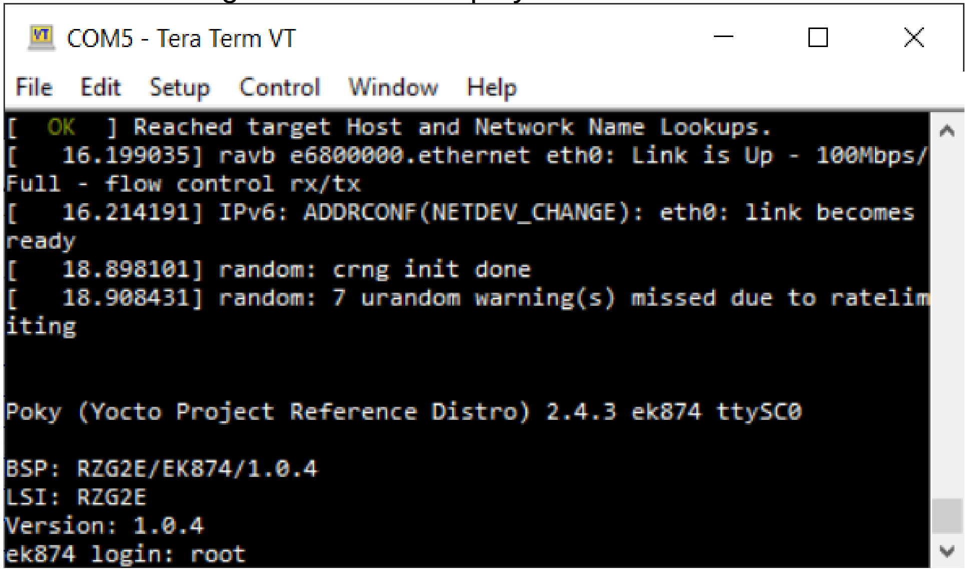
Generic S...G SD Card
Change
7.95 GB

5. Insert the SD card in socket and power up the board.



6. Run Tera Term and select com port for USB serial cable. Configure it for 115200 Baud rate, 8 data bits and No parity.

7. Boot messages should be displayed in the Tera Term window.



```
[  OK  ] Reached target Host and Network Name Lookups.
[   16.199035] ravb e6800000.ethernet eth0: Link is Up - 100Mbps/
Full - flow control rx/tx
[   16.214191] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes
ready
[   18.898101] random: crng init done
[   18.908431] random: 7 urandom warning(s) missed due to ratelim
iting


Poky (Yocto Project Reference Distro) 2.4.3 ek874 ttySC0

BSP: RZG2E/EK874/1.0.4
LSI: RZG2E
Version: 1.0.4
ek874 login: root
```

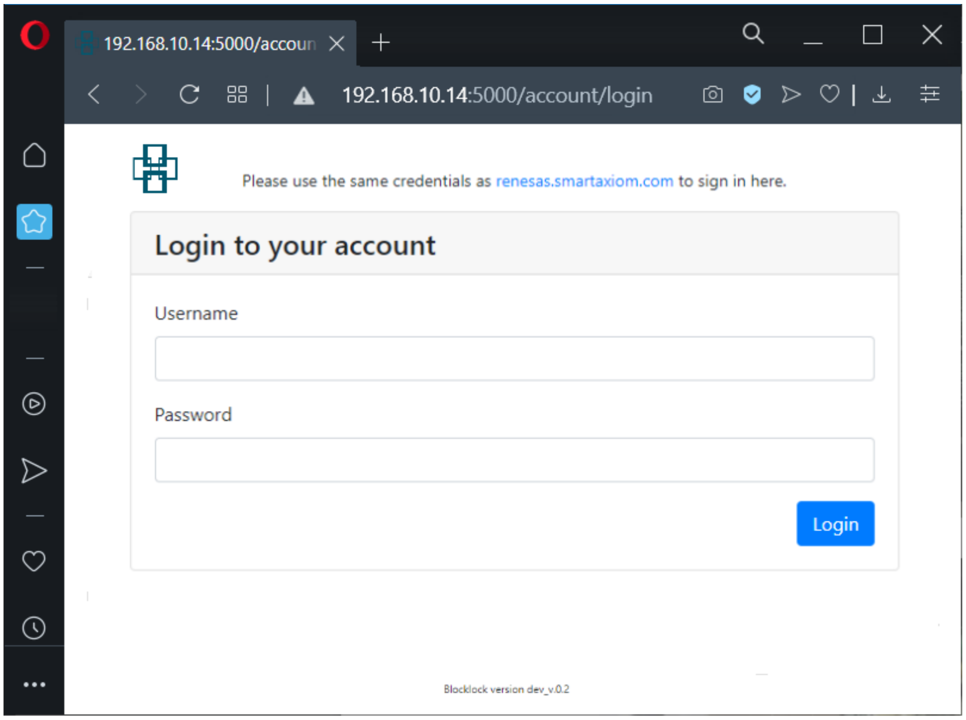
8. BlockLock software will automatically start in background. It will respond to Beachhead endpoint devices on the network.

9. Internet connectivity from the network where Beachhead endpoints and BlockLock edge devices are connected.

10. Configure port forwarding of 19000 and 19010 tcp/udp ports to EK874 in internet gateway/router.

11. Connect BlockLock to Tenacious Cloud. Open a browser window on computer connected to the same network and go to EK874_ip_address:5000. A login page will open. Enter Tenacious cloud login credentials to connect BlockLock Edge to cloud.

12. After successful login BlockLock will transmit data securely from Beachhead endpoints to Tenacious cloud.



Please use the same credentials as renesas.smartaxiom.com to sign in here.

Login to your account

Username

Password

Login

Blocklock version dev_v.0.2

BlockLock Edge setup on other devices

In case RZ/G2E Reference Board is not available, BlockLock software can also be run on Windows 10 64-bit, Linux 64-bit, Linux ARM-64v8 and Mac OSX computers. Download the executable package for your device from https://www.smartaxiom.com/blocklock.

BlockLock docker images are also available for amd64 and ARM-64v8 Linux platform from the Docker Hub. https://hub.docker.com/u/smartaxiom. Please contact support for more information.

- Devices running BlockLock edge software should be connected to the same network as Beachhead endpoints.

- Install MongoDB for your device. Select version 3.6 or higher.

  https://www.mongodb.com/try/download/community

- Details for the network Beachhead endpoints and BlockLock edge devices are connected:
  - Open 18090, 18091, 19000 and 19010 tcp/udp ports in device firewall. Allow BlockLock software to send/receive data on these ports. Configure port forwarding of 19000 and 19010 tcp/udp ports to device in internet gateway/router.

  - Windows 10 Firewall:

    https://support.microsoft.com/en-us/windows/turn-microsoft-defender-firewall-on-or-off-ec0844f7-aebd-0583-67fe-601ecf5d774f

  - Linux Firewall:

    https://ubuntu.com/server/docs/security-firewall

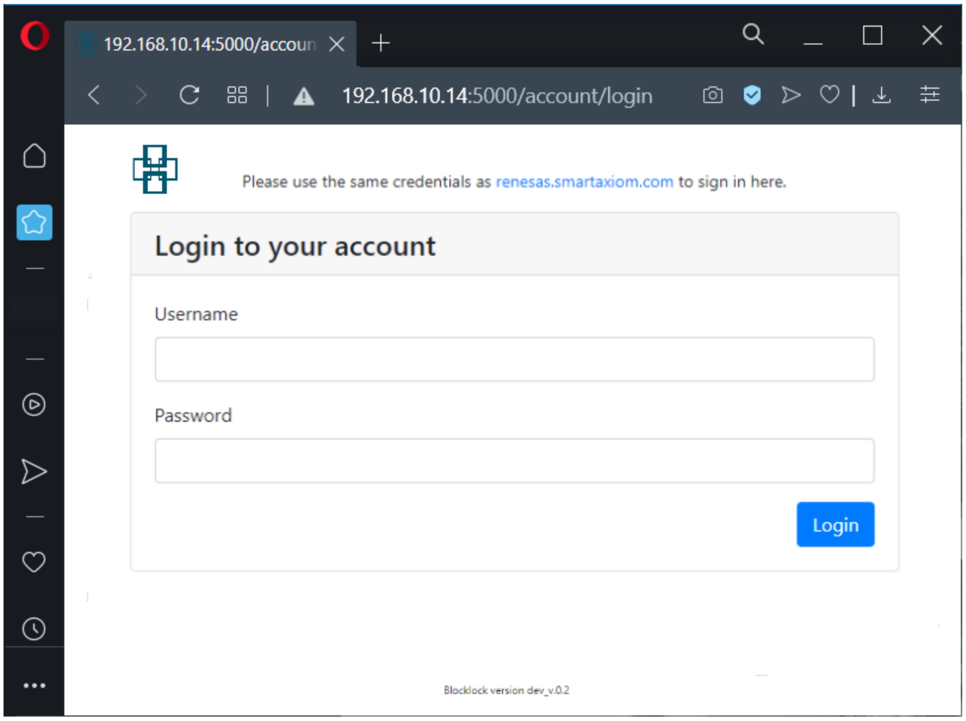
  - Mac Firewall:

    https://support.apple.com/guide/mac-help/block-connections-to-your-mac-with-a-firewall-mh34041/mac

- Extract contents of BlockLock package in a folder. Double click on blocklock executable binary to run software.

- Connect BlockLock to Tenacious Cloud. Open a browser window and go to localhost:5000.

- After successful login BlockLock will transmit data securely from Beachhead endpoints to Tenacious cloud.



192.168.10.14:5000/account/login

Please use the same credentials as renesas.smartaxiom.com to sign in here.

Login to your account

Username

Password

Blocklock version dev_v.0.2

Terms of Use

Please read this end-user license agreement carefully before installing or using the licensed programs accompanying this agreement. Installing the licensed programs constitutes acceptance of all terms and conditions of this agreement without modification. If you do not agree to any of these terms promptly remove the software and discontinue its use.

Time limited license. This EULA is time limited and terminates one year from date of first installation.

1. Licensed programs

Licensed program(s) include one or more desktop application programs, software development kits (SDKs), toolkits and other software libraries, in source or binary form, whether as firmware or otherwise, including any related documentation, whether executed on host computer or dedicated processor, which are collectively called the SmartAxiom Secure IoT Management System.

2. License grant, limitations, terms and conditions

Subject to all terms and conditions in the EULA, SmartAxiom Inc grants customer (the user and licensee) a limited license to use the licensed programs for demonstration and evaluation purposes only. This license is not a sale of the licensed program(s)or any rights thereto.

3. Title and ownership

The licensed program(s) including all title, copyright and other rights thereto, is owned by and shall remain the property of SmartAxiom Inc, and its use shall be subject to the EULA. Licensee agrees that it will use the licensed program(s)only as authorized, and that it will not modify, decompile, disassemble, translate or reverse engineer the licensed program. Licensee further agrees that it will retain all proprietary and copyright notices of SmartAxiom Inc in the licensed program(s).