Published by:

Atop Technologies, Inc.

2F, No. 146, Sec. 1, Tung-Hsing Rd,
30261 Chupei City,
Hsinchu County
Taiwan, R.O.C.

Tel: +886-3-550-8137
Fax: +886-3-550-8131
sales@atop.com.tw
www.atoponline.com
www.atop.com.tw
Important Announcement

The information contained in this document is the property of Atop Technologies, Inc., and is supplied for the sole purpose of operation and maintenance of Atop Technologies, Inc., products. No part of this publication is to be used for any other purposes, and it is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form, by any means, in whole or in part, without the prior explicit written consent of Atop Technologies, Inc., Offenders will be held liable for damages and prosecution. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Disclaimer

We have checked the contents of this manual for agreement with the hardware and the software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual is reviewed regularly and any necessary corrections will be included in subsequent editions. Suggestions for improvement are welcome.

All other product's names referenced herein are registered trademarks of their respective companies.

Documentation Control

<table>
<thead>
<tr>
<th>Author</th>
<th>Carlos Hsu, Willy Lin, Matteo Tabarelli</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision</td>
<td>1.1</td>
</tr>
<tr>
<td>Revision History</td>
<td>Released</td>
</tr>
<tr>
<td>Creation Date</td>
<td>5 February 2018</td>
</tr>
<tr>
<td>Last Revision Date</td>
<td>4 May 2018</td>
</tr>
<tr>
<td>Product Reference</td>
<td>SE5901 (SDK), SE5901B (SDK), SE5904D (SDK), SE5908 (SDK), SE5916 (SDK), SE5900A (SDK), SE5908A (SDK), SE5901A (SDK)</td>
</tr>
<tr>
<td>Document Status</td>
<td>Released</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Preface ................................................................................................................................. 9
  1.1 Purpose of the Manual ................................................................................................. 9
  1.2 Who Should Use This User Manual ........................................................................... 9
  1.3 Supported Platform ..................................................................................................... 9
  1.4 Warranty Period .......................................................................................................... 9
2 Introduction to Atop SDK .................................................................................................... 10
  2.1 Overview of SE59XX-SDK development environment .............................................. 10
  2.2 Description of SDK Folders ..................................................................................... 12
  2.3 Firmware upgrade ..................................................................................................... 15
  2.3.1 Use boot-loader update via console port ............................................................ 15
  2.3.2 Use Device Manager or Device Management Utility ........................................... 19
  2.4 Verify current firmware version.................................................................................. 20
  2.5 Installing a Cross-Compiler ....................................................................................... 21
  2.6 Compiling Procedure for Atop Applications ............................................................. 21
  2.7 Compiling new Applications ..................................................................................... 21
  2.8 Download new Applications to the device ................................................................ 22
  2.8.1 Using TFTP protocol ......................................................................................... 22
  2.8.2 Using FTP protocol ............................................................................................ 22
  2.9 Login or Remote Login to the device ......................................................................... 23
  2.9.1 Remote Login ...................................................................................................... 23
  2.9.2 Use a debug command line to Login .................................................................. 23
  2.10 Automatic execution on Startup of Custom-Applications .......................................... 23
  2.11 Startup messages ...................................................................................................... 24
3 Hardware Specifications ....................................................................................................... 29
  3.1 Packing List ................................................................................................................. 29
  3.2 Optional Accessories ................................................................................................. 29
  3.3 Hardware ..................................................................................................................... 31
  3.4 External Device’s Overview ........................................................................................ 34
  3.5 Serial Pin Assignments ............................................................................................... 38
  3.5.1 SE5901 Pin Assignments for Serial Interfaces ....................................................... 38
  3.5.2 SE5904D Pin Assignments .................................................................................. 39
  3.5.3 SE5901B Pin Assignments .................................................................................. 41
  3.5.4 SE5908A/ SE5916A Pin Assignments ................................................................. 42
4 Software Specifications ....................................................................................................... 44
  4.1 COM Port Interface ..................................................................................................... 44
  4.1.1 Program COM port interface .............................................................................. 44
  4.2 Network Interface ....................................................................................................... 45
  4.3 Other Interfaces ......................................................................................................... 45
  4.3.1 Buzzer .................................................................................................................. 46
  4.3.2 Digital Inputs ....................................................................................................... 46
  4.3.3 Digital Outputs .................................................................................................... 46
  4.3.4 Relay Outputs ..................................................................................................... 46
  4.3.5 LCM (SE5908 / SE5916 only) ............................................................................. 46
  4.3.6 Reset Button ....................................................................................................... 47
5 Testing interfaces

5.1 Test COM port interface – transmit and receive .............................................. 49
5.1.1 Test COM port interface by using atop_tcp_server ...................................... 49
5.1.1.1 Test Method ......................................................................................... 49
5.1.1.2 Test Execution ..................................................................................... 49
5.2 Test Buzzer interface .......................................................................................... 51
5.3 Test Digital Input ................................................................................................ 53
5.4 Test Digital Output .............................................................................................. 53
5.5 Test Hardware Relay Output .............................................................................. 53
5.6 Test Hardware Button ......................................................................................... 53
5.7 Test Hardware Watchdog Interface (WDT) ....................................................... 53
5.8 Test device LED .................................................................................................. 54
5.9 Test RTC interface ............................................................................................... 54
5.9.1 Setup RTC time: ............................................................................................ 54
5.9.2 Read RTC time: ............................................................................................. 55
5.9.3 RTC2system ................................................................................................. 55
5.9.4 system2RTC ................................................................................................. 55
5.10 Using NOR Flash – JFFS2 ................................................................................ 55
5.11 MQTT ................................................................................................................ 55
5.12 Firmware upgrade .............................................................................................. 56

6 Software API Reference ......................................................................................... 57

6.1 File List ............................................................................................................... 57

7 Data Structure Documentation ............................................................................... 58

7.1 sessiontag Struct Reference ................................................................................. 58
7.1.1 data fields ...................................................................................................... 58
7.2 Network APIs ...................................................................................................... 58
7.2.1 Functions ...................................................................................................... 58
7.3 Network APIs Function documentation ............................................................. 59
7.3.1 void AtopSDK4GHWReset (void) ................................................................. 59
7.3.2 void AtopSDKSet4GApn (char * apn) ............................................................. 60
7.3.3 void AtopSDKSet4GConnect (void) .............................................................. 60
7.3.4 void AtopSDKSet4GDialOnBoot (int value) ............................................... 60
7.3.5 void AtopSDKSet4GDisconnect (void) ......................................................... 61
7.3.6 void AtopSDKSet4GPinDisable (void) ......................................................... 61
7.3.7 void AtopSDKSet4GPinEnable (char * pinCode) ......................................... 61
7.3.8 void AtopSDKSet4GReConnect (int value) .................................................. 61
7.3.9 void AtopSDKSetNetDefGateway (int eth) ............................................... 61
7.3.10 void AtopSDKSetNetGateway (int eth, char *gw) .................................... 62
7.3.11 void AtopSDKSetNetIP (int eth, char *ip) .................................................. 62
7.3.12 void AtopSDKSetNetMask (int eth, char *mask) ....................................... 63
7.4 EEPROM User Name and Password Settings APIs .......................................... 63
7.4.1 Functions ..................................................................................................... 63
7.5 EEPROM User Name and Password Settings API Function documentation .... 64
7.5.1 void AtopSDKSetUserName (char name) ...................................................... 64
7.5.2 void AtopSDKSetUserPassword (int password) ............................................. 64
7.6 Run LED API Function documentation ......................................................... 64
7.6.1 Functions ................................................................................................. 64
7.6.2 void AtopSetRunLed (onMs u32,offMs u32) ........................................... 64
7.7 Alarm LED API Function documentation ...................................................... 65
7.7.1 Functions ................................................................................................. 65
7.7.2 void AtopSetAlarmLed (u32 value) .......................................................... 65
7.8 Read Reset Button API ................................................................................. 65
7.8.1 Functions ................................................................................................. 65
7.8.2 int AtopButton (void) ................................................................................ 65
7.9 Use Buzzer API documentation ................................................................. 66
7.9.1 Functions ................................................................................................. 66
7.9.2 Macros ....................................................................................................... 66
7.9.3 void AtopBuzzerOnOff (int value) ............................................................. 66
7.10 Read Digital Inputs API documentation .................................................... 66
7.10.1 Functions ................................................................................................. 66
7.10.2 int AtopGetDI (int index) ....................................................................... 67
7.11 Write Digital Output API documentation .................................................. 67
7.11.1 Functions ................................................................................................. 67
7.11.2 int AtopSetDO (int index, int value) ....................................................... 67
7.12 Hardware Watchdog API documentation ................................................... 67
7.12.1 Functions ................................................................................................. 67
7.12.2 Macros ..................................................................................................... 68
7.12.3 void atop_hwd_clear (void) ................................................................. 68
7.12.4 void atop_hwd_disable (void) ............................................................... 68
7.12.5 void atop_hwd_enable (void) ............................................................... 68
7.13 Wi-Fi USB Dongle control APIs documentation ...................................... 69
7.13.1 Functions ................................................................................................. 69
7.13.2 void create_default_ini_file (void) ...................................................... 70
7.13.3 void get_key_mgmt (char key_mgmt) .................................................... 70
7.13.4 int get_psk(char psk) ............................................................................. 70
7.13.5 int get_ssid (char ssid) ...................................................................... 71
7.13.6 int parse_setting (void) ...................................................................... 71
7.13.7 void run_connection (void) ................................................................. 72
7.13.8 int set_key_mgmt (int mode) ............................................................... 72
7.13.9 int set_psk (char psk_name) ............................................................... 73
7.13.10 int set_ssid (char ssid_name) ........................................................... 74
Table of Figures

Figure 2.1 Architecture of SE5904D SDK..............................................................10
Figure 2.2 Console firmware update- connections..................................................15
Figure 2.3 COM port Parameters for Console Firmware update................................16
Figure 2.4 TFTP32 appearance after execution.......................................................16
Figure 2.5 SE5904D Boot loader menu.................................................................17
Figure 2.6 LAN Settings.........................................................................................18
Figure 2.7 LAN1 settings .....................................................................................18
Figure 2.8 TFTP download menu ........................................................................18
Figure 2.9 SE59XX connection scheme (example on SE5904D) ...............................19
Figure 2.10 Firmware update prompt ....................................................................19
Figure 2.11 Firmware selection.............................................................................20
Figure 2.12 Firmware version - Console...............................................................20
Figure 2.13 Firmware version in Device Management Utility (English) .......................21
Figure 2.14 FTP access credentials .....................................................................22
Figure 2.15 FTP Download with FileZilla.............................................................22
Figure 2.16 Command line Login ........................................................................23
Figure 3.1 DB9 Pin Number...................................................................................38
Figure 3.2 TB5 Pin Number...................................................................................38
Figure 3.3 DB9 Pin Number...................................................................................39
Figure 3.4 Terminal Block (TB-5) Pin Number......................................................39
Figure 3.5 DB9 Pin Number...................................................................................41
Figure 3.6 2 x 7-pin Male Terminal Block.............................................................41
Figure 3.7 DB9 Pin Number...................................................................................42
Figure 3.8 Terminal Block (TB-5) Pin Number......................................................42
Figure 5.1 COM1 loopback test connection..........................................................49
Figure 5.2 Process execution on SE5904D, example ...........................................50
Figure 5.3 Setup TCPTest.exe for COM port loopback test ..................................50
Figure 5.4 Result of loopback test......................................................................51

List of Tables

Table 2.1 Content of 3rd party folder ....................................................................12
Table 2.2 Content of Software folder ....................................................................12
Table 2.3 List of programs in filesystem folder .....................................................14
Table 2.4 Content of build folder .........................................................................14
Table 3.1 Packing List ..........................................................................................29
Table 3.2 Optional Accessories ..........................................................................29
Table 3.3 Hardware features ................................................................................31
Table 3.4 SE5901 Pin Assignment for DB9 to RS-232/RS-422/RS-485 Connector ....38
Table 3.5 SE5901 Pin Assignment for TB5 to RS-232/RS-422/RS-485 Connector ........................................ 38
Table 3.6 MB5904D Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors ....................................... 39
Table 3.7 MB5904D Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors .............. 39
Table 3.8 SE5901B Pin Assignment for DB9 to RS-232/RS-485 Connector ..................................................... 41
Table 3.9 SE5901B 2 x 7-pin Male TB for RS-232/485(COM 1),RS-232(COM 2) Relay and DI pin-assignment. 41
Table 3.10 SE5908A/16A Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors .......................... 42
Table 3.11 SE5908A/16A Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors... 42
Table 4.1 Sample programs for COM port interface ................................................................. 44
Table 4.2 SE59XX device node ........................................................................................................... 44
Table 4.3 SE59XX Programming commands per device node ..................................................................... 44
Table 4.4 SE59XX ioctl command of COM Port ..................................................................................... 45
Table 4.5 Sample programs for TCP server connection to COM port communication .......................... 45
Table 4.6 Sample program for Buzzer .................................................................................................... 46
Table 4.7 Sample program for Digital Input .......................................................................................... 46
Table 4.8 Sample program for Digital Output ......................................................................................... 46
Table 4.9 Sample program for Relay Output ......................................................................................... 46
Table 4.10 Sample program for LCM ..................................................................................................... 47
Table 4.11 Sample program for Reset Button ....................................................................................... 47
Table 4.12 Sample program for WDT .................................................................................................. 47
Table 4.13 Sample program for LEDs ................................................................................................ 47
Table 4.14 Sample program for Cellular functions ............................................................................. 48
Table 5.1 TCP-port to COM-port mapping .............................................................................................. 51
1 Preface

1.1 Purpose of the Manual

This manual supports you in understanding the software SDK architecture of ATOP's SE59XX Series and should be a reference guide for application development on this platform.

1.2 Who Should Use This User Manual

This manual is to be used by qualified programmers, network personnel or support technicians who are familiar with network operations and C Language programming. For any related problems, please contact your local distributor. If they are unable to assist you, please redirect your inquiries to www.atop.com.tw or www.atoponline.com.

1.3 Supported Platform

This manual is designed for the SE5901, SE5901B, SE5904D, SE5908, SE5916, SE5900A, SE5908A, and SE5916A Industrial Serial and Ethernet controller and that model only.

1.4 Warranty Period

ATOP provides a 5-year limited warranty for SE59XX Series.
# Introduction to Atop SDK

## 2.1 Overview of SE59XX-SDK development environment

**Notice:** Please upgrade to the Firmware version on which this SDK document is based.

Figure 2.1 shows the whole architecture of SE59XX SDK. Three types of Applications are provided in user's layer:

1. **ATOP applications:** providing multiple sample SDK programs to hardware devices
2. **ATOP utility:** providing firmware upgrade, network settings and storage mounting tools
3. **Third-party:** providing 3rd parties software required such as SNMP / Apache / SQLite

In Kernel Layer, Linux 3.14.26 is customized to provide complete networking protocols. In Driver Layer, device drivers for all Industrial communication interfaces are provided. In hardware Layer, Customized ARM Cortex-A8 platform and Atop FPGA management core are provided.

![Figure 2.1 Architecture of SE5904D SDK](image-url)

<table>
<thead>
<tr>
<th>User Layer</th>
<th>Atop Applications</th>
<th>Atop Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCAN</td>
<td>Date adjust</td>
</tr>
<tr>
<td></td>
<td>LED</td>
<td>Buzzer</td>
</tr>
<tr>
<td></td>
<td>Reboot</td>
<td>Restore</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kernel Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
</tr>
<tr>
<td>ATOP customized Linux</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driver Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>atop Technologies</td>
</tr>
<tr>
<td>ATOP RS232 / 422 / 485</td>
</tr>
<tr>
<td>EEPROM</td>
</tr>
<tr>
<td>GPS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOP FPGA</td>
</tr>
</tbody>
</table>
2.2 Description of SDK Folders

Extract `sdk_release_YYYYMMDD.tar.bz2` and refer `SDK_Release/` folder (please note that YYYY corresponds to the release year, MM to the release month and DD to the release Day).

There are 4 sub-folders:
- **build**: this folder includes build done firmware and merge utilities.
- **filesystem**: this folder includes root file system and boot up script.
- **software**: this folder includes ATOP library, sample code and header file.
- **3rd party**: 3rd party utilities

The followings are the list of document in "3rdparty" folder:

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Busybox-1.23.1</td>
<td>Busybox source</td>
</tr>
<tr>
<td>c-ares</td>
<td>C library for asynchronous DNS requests</td>
</tr>
<tr>
<td>dhcp-4.1-esv-R13</td>
<td>IPv6 dhcp utilities</td>
</tr>
<tr>
<td>email-3.1.3</td>
<td>E-mail utility</td>
</tr>
<tr>
<td>gmp-6.1.2</td>
<td>gmp-6utility – for arbitrary precision arithmetic</td>
</tr>
<tr>
<td>Hostap</td>
<td>user space daemon for access point and authentication servers.</td>
</tr>
<tr>
<td>i2c-tools-3.1.2</td>
<td>I2C tools to manage I2C Bus</td>
</tr>
<tr>
<td>iniparser</td>
<td>Ini file parser library</td>
</tr>
<tr>
<td>iptables-1.6.1</td>
<td>Tool to manage IP tables</td>
</tr>
<tr>
<td>libmodbus-3.1.2</td>
<td>Modbus stack</td>
</tr>
<tr>
<td>libnl-3.2.25</td>
<td>libnl suite is a collection of libraries</td>
</tr>
<tr>
<td>libpcap-1.7.4</td>
<td>a portable C/C++ library for network traffic capture</td>
</tr>
<tr>
<td>libuuid</td>
<td>to generate unique ident for obj to be accessible beyond local system</td>
</tr>
<tr>
<td>monit-5.18</td>
<td>Daemon monitor utility</td>
</tr>
<tr>
<td>mosquitto-1.4.14</td>
<td>MQTT stack</td>
</tr>
<tr>
<td>ncftp-3.2.5</td>
<td>FTP utility</td>
</tr>
<tr>
<td>openssl-1.0.2</td>
<td>Openssl library</td>
</tr>
<tr>
<td>rtl8192EU_linux_v4.4.1.1</td>
<td>Wi-Fi dongle driver.</td>
</tr>
<tr>
<td>strongswan-5.5.2</td>
<td>IPsec VPN</td>
</tr>
<tr>
<td>ucarp-1.5.2</td>
<td>allows 2 host share common virtual IP to provide automatic failover</td>
</tr>
<tr>
<td>wireless_tools.29</td>
<td>Wifi tools</td>
</tr>
<tr>
<td>zlib-1.2.8</td>
<td>Zip library</td>
</tr>
</tbody>
</table>

The followings are the list of application programs in "software" folder:

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include</td>
<td>Reference header file</td>
</tr>
<tr>
<td>atop_library</td>
<td>ATOP library</td>
</tr>
<tr>
<td>atop_application</td>
<td>Sample code</td>
</tr>
<tr>
<td>libatop.so.1.0.0</td>
<td>ATOP library binary</td>
</tr>
</tbody>
</table>
The following are the list of application programs in "filesystem" folder:

Table 2.3 List of programs in filesystem folder

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>etc</td>
<td>Bootup script, no need to modify under this folder</td>
</tr>
<tr>
<td>rootfs.tar.bz2</td>
<td>Pre-build root file system.</td>
</tr>
</tbody>
</table>

The following are the list of application programs in "build" folder:

Table 2.4 Content of build folder

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image.dld</td>
<td>Build done FW image.</td>
</tr>
<tr>
<td>initrd.uboot</td>
<td>Root file system package</td>
</tr>
<tr>
<td>composer</td>
<td>Merge image utility</td>
</tr>
<tr>
<td>u-boot.bin</td>
<td>Bootloader image for rescuing device</td>
</tr>
<tr>
<td>u-boot.dld</td>
<td>Bootloader image for rescuing device</td>
</tr>
<tr>
<td>zlimage</td>
<td>Linux kernel raw image</td>
</tr>
</tbody>
</table>
2.3 Firmware upgrade

There are two ways to upgrade the firmware on the SE59XX platform:

2.3.1 Use boot-loader update via console port

Prepare a Debug Cable (RJ45 to Serial) and a CAT5E Ethernet cable. Then, follow below figure to connect the Debug port to PC’s COM and CAT5E cable to connect to the Device’s LAN1 Ethernet port to any Host PC’s Ethernet port.

On your PC, run Windows’ “Super Terminal” setup COM port parameters as follows:

- Port: the connected COM port
- Baud Rate: 115200 bps
- Parity: none
- Data: 8 data bits
- Stop: 1 stop bit
- Flow control: none
With this method, TFTP protocol is used. The TFTP client is already set-up and running inside the SE59XX platform. Thus, the user needs to execute TFTP server in Windows. An open source version is available for download and can be found as "tftpd32". Screenshot below shows "tftpd32.exe" after running the application.

Now, setup the IP address of the TFTP server. The current folder is the one where "tftpd32.exe" is located. After executing TFTP server, reboot the target SE59XX platform and press the Escape ("Esc") key immediately. A boot-loader menu will be shown as Figure 2.5.
Figure 2.5 SE5904D Boot bade menu
Select item 1 to enter "LAN Setting" menu as Figure 2.6, and setup IP/Netmask/Gateway of LAN1 as Figure 2.7

![Figure 2.6 LAN Settings](image)

Enter 0 to exit to upper layer menu and select 5 to enter the "TFTP Download" menu, then select 1 to setup TFTP server IP as Figure 2.8

![Figure 2.7 LAN1 settings](image)

![Figure 2.8 TFTP download menu](image)

After the setup of the server IP is completed, select 2 to download the firmware image.

**Note: the extension of the firmware should be .dld**
2.3.2 Use Device Manager or Device Management Utility

Please use a CAT5E cable to connect SE59XX to a PC running Windows where ATOP Device Management utility is already installed. To install Device Management Utility, please download the latest release from ATOP Website and follow its dedicated user manual for the installation.

The device doesn't have necessarily to be directly connected to the PC, as long as it is inside the same LAN. Atop Management Utility will scan the whole network automatically.

![SE59XX connection scheme](image)

Now, please power on the device and run Atop's Device Management Utility from your Host PC. Once the device is running, the utility will list all devices found. If the device doesn't show up, push the leftmost button (Rescan function). Once identified, select the device by mouse left button and select "Firmware" >> "Download Firmware" as per Figure 2.10.

![Firmware update prompt](image)

Select the firmware (Kernel or AP) from this dialog and select the upgraded file as Figure 2.11. Then, click on the "Upgrade" button to upgrade the firmware selected.
Note: This example is made with SE5901A. All other models of SE59XX family share the same method.

![Image of firmware selection](image)

**Figure 2.11 Firmware selection**

Note that the extension file name of upgraded firmware should be `.dld`.

### 2.4 Verify current firmware version

There are two methods to verify the firmware version:

1. Use a debug line as per Paragraph 2.3.1 above to connect console port of the device. After boot up, type “atop.sh ver” in the console command line to check current version as Figure 2.13 shown. The red rectangle shows information of bootloader (V1.00), Kernel (V1.00) and AP (V1.00) version number.

   ![Image of firmware version - Console](image)

   **Figure 2.12 Firmware version - Console**

2. Use DeviceManager or Device Management Utility (Serial Manager) to check version number as per Figure 2.13. (DeviceManager is currently supported to Simplified Chinese release)
2.5 Installing a Cross-Compiler

1) Copy `ti-am335x-linux-devkit-08.00.00.00.tar.gz` to `opt` folder and extract it. Be sure that you have and use the root account to do it. This user manual is made with this version. If a newer, stable version is available, the SDK package will include it.

```
tar xzf ./ti-am335x-linux-devkit-08.00.00.00.tar.gz /opt
```

2) Add these environment variables

```
export ARCH=arm
export PATH=/opt/ti-am335x-linux-devkit-08.00.00.00/bin:$PATH
export CROSS_COMPILE=arm-linux-gnueabihf-
```

2.6 Compiling Procedure for Atop Applications

To compile ATOP application, in SDK root folder, please enter

```
makedepend
make clean
dear all .o object and executable files
make release platform=arm335x_v8
compile and link the source code
```

After build done, you can find your image under build folder be named `image.dld`.

2.7 Compiling new Applications

1) Put the source code under `./software/atop_application/utils/<YOUR_APP_FOLDER>` folder. `<YOUR_APP_FOLDER>` is a name chosen by yourself. (such as “Test”)

2) Follow Paragraph 2.6 above to build your application. Modify “Makefile” following the `atop_sdk`.
2.8 Download new Applications to the device

New applications can be downloaded in two ways:

2.8.1 Using TFTP protocol

- Please execute `tftpd32.exe` in the remote PC and modify target folder and IP address as Figure 2.8
- Login into the target device (under Linux console) and enter:

```
tftp -gr YOUR_APP_NAME YOUR_TFTP_SERVER_IP
```

Remember to use `chmod` command to modify the access attributes of these files. If transmission failed, please check the networking connection.

2.8.2 Using FTP protocol

- Setup or read FTP account and password from A to B boot-loader menu as per image below.

  ![FTP access credentials](image)

  **Figure 2.14 FTP access credentials**

- Login the Linux system in order to make sure the network connection is fine.
  
  Use any 3rd party ftp software to transfer the files. For example, use FileZilla as

  ![FTP Downlad with FileZilla](image)

  **Figure 2.15 FTP Download with FileZilla**

- Input FTP account/password of SE59XX and login to the FTP server.
Note: Make sure the binarymode to be set during the transmission. Remember to use `chmod` command to modify the access attribute of these files. If transmission failed, please check the networking connection key or not between SE59XX platform and Host PC.

2.9 Login or Remote Login to the device

2.9.1 Remote Login
1) Setup or read FTP account and password from boot-loader menus as Figure 1-16
2) Use any tools supporting the telnet protocol such as "ssh" inside of Windows.
3) Enter `SE59XX_TARGET_IP` via ssh using software putty.
4) Login account as first step shown.

2.9.2 Use a debug command line to Login
If you’re not pressing “Esc” button within 3 seconds from boot-up, the device will enter Linux login mode as per screenshot below.

```
Welcome to ATOP system
ATOP login: [ 8.978831] libphy: 4a101000.mdio:02 - Link is Up - 100/Full
Welcome to ATOP system
ATOP login:
```

Figure 2.16 Command line Login

2.10 Automatic execution on Startup of Custom-Applications

1) Put your startup script "user_pre.sh" or "user_post.sh" into jffs2 of root file system via FTP or TFTP.
2) SE5904D will execute both "user_pre.sh" and "user_post.sh" after startup from next reboot.
2.11  Startup messages

The following is the standard startup message from SE5904D (as example):


I2C: ready
DRAM: 512 MiB
Flash: 32 MiB
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
*** Warning - bad CRC, using default environment

Net:  cpsw
Hit ESC to execute ATOP menu:
Wait ... 0
Booting from ramdisk ...

Starting kernel ...

Starting kernel ...

Total of 128 interrupts on 1 active controller
OMAP clockevent source: timer2 at 24000000 Hz
sched_clock: 32 bits at 24MHz, resolution 41ns, wraps every 17895696942ns
OMAP clocksource: timer1 at 24000000 Hz
Calibrating delay loop... 794.62 BogoMIPS (lpj=397312)
OMAP GPIO hardware version 0.1
Platform 49000000.edma: alias fck already exists
Platform 50000000.gpmc: could not find pctldev for node /pinmux@44e10800/pinmux_i2c0_pins, deferring probe
platform 44e0b000.i2c: Driver omap_i2c requests probe deferral
platform 4802a000.i2c: could not find pctldev for node /pinmux@44e10800/pinmux_i2c1_pins, deferring probe
platform 4802a000.i2c: Driver omap_i2c requests probe deferral
Trying to unpack rootfs image as initramfs...
Freeing initrd memory: 6628K (c7986000-c7fff000)
hw perfevents: enabled with ARMv7 Cortex-A8 PMU driver, 5 counters available
UDP hash table entries: 256 (order: 0, 4096 bytes)
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
RPC: Registered named UNIX socket transport module.
RPC: Registered tcp transport module.
RPC: Registered udp transport module.
RPC: Registered tcp NSV4.1 backchannel transport module.
Trying to unpack rootfs image as initramfs...
rootfs image is not initramfs (no cpio magic); looks like an initrd
Freeing initrtd memory: 6628K (c7986000-c7fff000)
Starting network...
[  6.877040] device eth0 entered promiscuous mode
[  6.909977] net eth0: initializing cpsw version 1.12 (0)
[  6.971560] net eth0: phy found : id is : 0x2000a212
[  6.979199] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[  6.985310] 8021q: adding VLAN 0 to HW filter on device eth0
net.ipv6.conf.eth0.accept_ra = 1
[  7.048551] device eth1 entered promiscuous mode
[  7.090144] net eth1: initializing cpsw version 1.12 (0)
[  7.150504] net eth1: phy found : id is : 0x2000a212
[  7.156080] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
[  7.162240] 8021q: adding VLAN 0 to HW filter on device eth1
net.ipv6.conf.eth1.accept_ra = 1
Starting atop_n
Starting dropbear sshd: OK
Starting ProFTPD: [  7.332967] warning: `proftpd' uses 32-bit capabilities (legacy support in use)
done
Starting stunnel: [ ] stunnel 5.09 on arm-buildroot-linux-gnueabihf platform
[ ] Compiled/running with OpenSSL 1.0.2 22 Jan 2015
[ ] Threading:FORK Sockets:POLL,IPv6 TLS:ENGINE,FIPS,OCSP,PSK,SNI
[ ] errno: (*__errno_location ())
[ ] Reading configuration from file /etc/stunnel/stunnel.conf
[ ] UTF-8 byte order mark detected
[ ] Enabling support for engine "capi"
[!] error queue: 2606A074: error:2606A074:engine routines:ENGINE_by_id:no such engine
[!] Line 18: "engine = capi": Failed to open the engine
FAIL
Starting network management services: snmpd.

Welcome to ATOP system
ATOP login:
3 Hardware Specifications

3.1 Packing List

Inside the purchased package, you will find the following items:

Table 3.1 Packing List

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE59XX</td>
<td>1</td>
<td>Industrial Serial Device Server</td>
</tr>
<tr>
<td>Mounting Kit</td>
<td>1</td>
<td>On SE5908 / SE5916 / SE5908A / SE5916A Rack Mounting Type-L angles x 2(</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Screws ) x 6( On SE5901 / SE5904D / SE5901B - DIN Rail Kit</td>
</tr>
<tr>
<td>Terminal Block</td>
<td></td>
<td>Power Supply/ Relay output:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB3 x 1: 3-pin 5.08mm lockable Terminal Block (SE5901, SE5901B)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB3 x 2: 3-pin 5.08mm lockable Terminal Block (SE5908-DC,SE5916-DC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB7 x1: 7-pin 5.08mm lockable Terminal Block (SE5904D only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial ports: Terminal block is included only on TB model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB5 x 1: 5-pin 5.08mm lockable Terminal Block (SE5901)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB5 x 4: 5-pin 5.08mm lockable Terminal Block (SE5904D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB5 x 8: 5-pin 5.08mm lockable Terminal Block (SE5908A)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB5 x 16: 5-pin 5.08mm lockable Terminal Block (SE5916A)</td>
</tr>
<tr>
<td>Documentation</td>
<td>1</td>
<td>Hardware Installation Guide ) Warranty card is included(</td>
</tr>
<tr>
<td>Mounting Kit</td>
<td>1</td>
<td>DIN-Rail Kit (Already mounted on the device)</td>
</tr>
</tbody>
</table>

Note: Please notify your sales representative if any of the above items is missing or damaged in any form upon delivery. If your sales representative is unable to satisfy your enquiries, please contact us directly.

3.2 Optional Accessories

The following table lists optional accessories for SE59XX SDK series.

Table 3.2 Optional Accessories

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UN315-1212(US-LDC)</td>
<td>Y-Type (5.08mm) power adapter, 100-240VAC input, 1.25A @ 12VDC output, US plug</td>
</tr>
<tr>
<td>UNE315-1212(EU-LDC)</td>
<td>Y-Type (5.08mm) power adapter, 100-240VAC input, 1.25A @ 12VDC output, EU plug</td>
</tr>
<tr>
<td>ADP-DB9(F)-TB5</td>
<td>Female DB9 to Female 3.81 TB5 Converter</td>
</tr>
<tr>
<td>CBL-RJ45(8P)-DB9(F)</td>
<td>8-pin RJ45-DB9 debug cable, 90cm</td>
</tr>
<tr>
<td>GDC-120</td>
<td>120mm copper woven grounding cable</td>
</tr>
<tr>
<td>Model</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LM28-C3S-TI-N</td>
<td>SFP Transceiver, 1250Mbps, 850nm VCSEL, Multi-mode, 550m, 3.3V, -20~85°C</td>
</tr>
<tr>
<td>LM38-C3S-TI-N</td>
<td>SFP Transceiver, 1250Mbps, 1310nmFP, Multi-mode, 2km, 3.3V, -40~85°C</td>
</tr>
<tr>
<td>LS38-C3S-TI-N</td>
<td>SFP Transceiver, 1250Mbps, 1310nmFP, Single-mode, 10km, 3.3V, -40~85°C</td>
</tr>
<tr>
<td>LS38-C3L-TI-N</td>
<td>SFP Transceiver, 1250Mbps, 1310nm DFB, Single-mode, 30km, 3.3V, -40~85°C</td>
</tr>
<tr>
<td>WMK-450-Black</td>
<td>Black Aluminum Wall Mount Kit (DIN-rail items only)</td>
</tr>
</tbody>
</table>
## 3.3 Hardware

### Table 3.3 Hardware features

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>32-bit ARM Based TI CPU AM3354 800MHz (except SE5908A/SE5916A use AM3352 1GHz)</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32MB</td>
</tr>
<tr>
<td>RAM</td>
<td>SE5901 DDR2 128MB&lt;br&gt;SE5901B DDR2 256MB&lt;br&gt;SE5904D DDR3 256MB&lt;br&gt;SE5900A/08A/16A/MB5908/16 DDR3 256MB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>8 KB</td>
</tr>
<tr>
<td>Reset</td>
<td>Built-in Recessed Key (Restore to Factory Defaults)</td>
</tr>
<tr>
<td>Watchdog</td>
<td>Hardware built-in</td>
</tr>
</tbody>
</table>

### Network

<table>
<thead>
<tr>
<th>Ethernet Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IEEE 802.3X BaseT&lt;br&gt;IEEE 802.3u 100BaseT(X)&lt;br&gt;IEEE 802.3ac 1000BaseT(X) – SFP version of SE5904D only&lt;br&gt;IEEE 802.3af (PoE PD) – selected SE5901 and SE5904D versions can be powered through PoE</td>
</tr>
<tr>
<td>Connection</td>
<td>SFP or RJ45</td>
</tr>
</tbody>
</table>

### Serial

<table>
<thead>
<tr>
<th>Serial Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS-232/RS-422/RS-485 Software Selectable (Default: RS-232)&lt;br&gt;• The first port available on SE5901B is RS-232/RS-485&lt;br&gt;• The second port available on SE5901B-I0-X is only RS-232&lt;br&gt;• The isolation version (-SiS) on SE5908/SE5916/SE5908A/SE5916A supports only RS-422/RS-485</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serial Connector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Connector Type&lt;br&gt;• SE5916 - 16 Serial Ports (RJ45)&lt;br&gt;• SE5908 - 8 Serial Ports (RJ45)&lt;br&gt;• SE5916A – 16 Serial Ports (TB-5 or DB-9)&lt;br&gt;• SE5908A – 8 Serial Ports (TB-5 or DB-9)&lt;br&gt;• SE5904 – 4 Serial Ports (TB-5 or DB-9)&lt;br&gt;• SE5901 – 1 Serial Port (TB-5 or DB-9)&lt;br&gt;• SE5901B – 1 Serial Port (TB-14 or DB-9) – includes I/O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SE5901/SE5901B no isolation&lt;br&gt;SE5904D/ SE5908A/16A (optional 3V)&lt;br&gt;SE5908/16 (optional 2.5kV)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serial Port Communication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baud-rate: 1200 bps ~ 921600 bps&lt;br&gt;Parity: None, Even, Odd, Mark, or Space&lt;br&gt;Data Bits: 5, 6, 7, 8&lt;br&gt;Stop Bits: 1, 2 Software Selectable&lt;br&gt;Flow Control: RTS/CTS (RS-232 only), XON/XOFF, None</td>
</tr>
</tbody>
</table>
## LED Indicator

<table>
<thead>
<tr>
<th>LED indication</th>
<th>Power x 2 (SE5901-SE5901B – SE5908 – SE5916 x 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RUN x 1</td>
</tr>
<tr>
<td></td>
<td>ALARM x 1</td>
</tr>
<tr>
<td></td>
<td>LAN:</td>
</tr>
<tr>
<td></td>
<td>- x 2 (all versions except SE5908A and SE5916A)</td>
</tr>
<tr>
<td></td>
<td>- x 6 (SE5908A and SE5916A only)</td>
</tr>
<tr>
<td>COM port:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- x 16 (SE5916 and SE5916A);</td>
</tr>
<tr>
<td></td>
<td>- x 8 (SE5908 and SE5908A);</td>
</tr>
<tr>
<td></td>
<td>- x 4 (SE5904D);</td>
</tr>
<tr>
<td></td>
<td>- x 1 (SE5901 and SE5901B)</td>
</tr>
</tbody>
</table>

## Power Requirement & EMC

<table>
<thead>
<tr>
<th>Input</th>
<th>SE5908/SE5916:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Single 100~240 VAC (EU/US versions)</td>
</tr>
<tr>
<td></td>
<td>- Single 24~48 VDC (DC version)</td>
</tr>
<tr>
<td></td>
<td>SE5908A/SE5916A:</td>
</tr>
<tr>
<td></td>
<td>- Redundant 100<del>240 VAC or 100</del>370 VDC (TB) – HV vers.</td>
</tr>
<tr>
<td></td>
<td>- Redundant 24~48 VDC-DC version</td>
</tr>
<tr>
<td></td>
<td>SE5901/SE5901B:</td>
</tr>
<tr>
<td></td>
<td>- Single 9~48 VDC</td>
</tr>
<tr>
<td></td>
<td>SE5904D:</td>
</tr>
<tr>
<td></td>
<td>- Redundant 9~48 VDC</td>
</tr>
<tr>
<td>Consumption</td>
<td>Max. 17.5 W(SE5908/SE5916)</td>
</tr>
<tr>
<td></td>
<td>Max. 6 W(SE5901)</td>
</tr>
<tr>
<td></td>
<td>Max. 7.8 W(SE5904D)</td>
</tr>
<tr>
<td></td>
<td>Max. 17.5 W(SE5908A/SE5916A)</td>
</tr>
<tr>
<td></td>
<td>Max. 7.2 W(SE5901B)</td>
</tr>
<tr>
<td>EMI/EMC</td>
<td>FCC Part 15, Subpart B, Class A</td>
</tr>
<tr>
<td></td>
<td>EN 55032, Class B, EN 61000-6-2, Class B</td>
</tr>
<tr>
<td></td>
<td>EN 61000-3-2, EN 61000-3-3</td>
</tr>
<tr>
<td></td>
<td>EN 5504, EN 61000-6-4</td>
</tr>
<tr>
<td></td>
<td>IEC 61850-3 / IEEE 1613 (SE5908A and SE5916A only)</td>
</tr>
</tbody>
</table>

## Mechanical

<table>
<thead>
<tr>
<th>Dimensions (W x H x D, mm)</th>
<th>SE5901: 32 mm x 110 mm x 90 mm (1.26 x 4.33 x 3.54 in)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SE5901B: 32 mm x 122 mm x 92 mm (1.26 x 4.8 x 3.62 in)</td>
</tr>
<tr>
<td></td>
<td>SE5904D: 55 mm x 145 mm x 113 mm (2.17 x 5.7 x 4.45 in)</td>
</tr>
<tr>
<td></td>
<td>SE5908: 436 mm x 43.5 mm x 200 mm (17.17 x 1.71 x 7.87 in)</td>
</tr>
<tr>
<td></td>
<td>SE5916: 436 mm x 43.5 mm x 200 mm (17.17 x 1.71 x 7.87 in)</td>
</tr>
<tr>
<td></td>
<td>SE5908A: 440.6 mm x 44 mm x 309 mm (17.35 x 1.73 x 12.17 in)</td>
</tr>
<tr>
<td></td>
<td>SE5916A: 440.6 mm x 44 mm x 309 mm (17.35 x 1.73 x 12.17 in)</td>
</tr>
<tr>
<td>Enclosure</td>
<td>IP30 protection, metal housing</td>
</tr>
</tbody>
</table>

## Environmental

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Operations</th>
<th>-40°C ~ 85°C (-40°F ~ 185°F)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(except SE5901B -40°C ~ 70°C and SE5908/SE5916 -20°C ~ 70°C)</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>Storage</td>
<td>-40°C ~ 85°C (-40°F ~ 185°F)</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>5% ~ 95%, 55°C Non-condensing</td>
<td></td>
</tr>
</tbody>
</table>
3.4 **External Device's Overview**

The following figures show particular SE59XX series device’s front and rear panels.

**SE5901**

**SE5904D**
3.5 Serial Pin Assignments

3.5.1 SE5901 Pin Assignments for Serial Interfaces

DB9 to RS-232/RS-422/RS-485 connectors

![DB9 Pin Number](image)

Figure 3.1 DB9 Pin Number

Table 3.4 SE5901 Pin Assignment for DB9 to RS-232/RS-422/RS-485 Connector

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232 Full Duplex</th>
<th>RS-422/4-Wire RS-485 Full Duplex</th>
<th>2-Wire RS-485 Half Duplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>RxD</td>
<td>TXD+</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>RXD+</td>
<td>Data+</td>
</tr>
<tr>
<td>4</td>
<td>DTR</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>RXD-</td>
<td>Data-</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
<td>TXD-</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>RI</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

1 x 5-pin (Male Terminal Block) for RS-232/RS-422/RS485 Connector

![TB5 Pin Number](image)

Figure 3.2 TB5 Pin Number

Table 3.5 SE5901 Pin Assignment for TB5 to RS-232/RS-422/RS-485 Connector

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232 Full Duplex</th>
<th>RS-422/4-Wire RS-485 Full Duplex</th>
<th>2-Wire RS-485 Half Duplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RxD</td>
<td>T+</td>
<td>NC</td>
</tr>
<tr>
<td>2</td>
<td>CTS</td>
<td>T-</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>R+</td>
<td>Data+</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>R-</td>
<td>Data-</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
</tbody>
</table>
3.5.2  **SE5904D Pin Assignments**

DB9 to RS-232/RS-485/RS-422 connectors

![DB9 Pin Number](image)

**Table 3.6 MB5904D Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232 Full Duplex</th>
<th>RS-422 Full Duplex</th>
<th>RS-485 Half Duplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>RxD</td>
<td>TxD+</td>
<td>Data+</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>RxD+</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>DTR</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>RxD-</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
<td>TxD-</td>
<td>Data-</td>
</tr>
<tr>
<td>9</td>
<td>RI</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

5-Pin Terminal Block to RS-485/RS-422 connectors

![Terminal Block (TB-5) Pin Number](image)

**Table 3.7 MB5904D Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232</th>
<th>RS-422 4-Wire RS-485</th>
<th>2-W RS-485</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RxD</td>
<td>TxD+</td>
<td>Data+</td>
</tr>
<tr>
<td>2</td>
<td>CTS</td>
<td>TxD-</td>
<td>Data-</td>
</tr>
<tr>
<td></td>
<td>TxD</td>
<td>RxD+</td>
<td>N/A</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>RxD+</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>RxD-</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
</tbody>
</table>
3.5.3 **SE5901B Pin Assignments**

**DB9 to RS-232/RS-485/RS-422 connectors**

![DB9 Connector Diagram](image)

Figure 3.5 DB9 Pin Number

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232 Full Duplex</th>
<th>RS-485 Half Duplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>RxD</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>Data+</td>
</tr>
<tr>
<td>4</td>
<td>DTR</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>Data-</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>RI</td>
<td>N/A</td>
</tr>
</tbody>
</table>

2 x 7-pin Male Terminal Block for RS-232/485(COM 1),RS-232(COM 2) Relay and DI

![Terminal Block Diagram](image)

Figure 3.6 2 x 7-pin Male Terminal Block

<table>
<thead>
<tr>
<th>Pin#</th>
<th>DI and Relay</th>
<th>COM1 (RS-232)</th>
<th>COM1 (RS-485)</th>
<th>COM2 (RS-232)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DI1</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>2</td>
<td>DI2</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>3</td>
<td>Relay 1 -</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>4</td>
<td>Relay 1+</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>5</td>
<td>Relay 2 -</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>6</td>
<td>Relay 2+</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
<td>Dedicated for DI/DO</td>
</tr>
<tr>
<td>7</td>
<td>Dedicated for COM</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>Dedicated for COM</td>
<td>Rx</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>Dedicated for COM</td>
<td>CTS</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>Dedicated for COM</td>
<td>Tx</td>
<td>Data +</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>Dedicated for COM</td>
<td>RTS</td>
<td>Data -</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>Dedicated for COM</td>
<td>-</td>
<td>-</td>
<td>SG (Signal Ground)</td>
</tr>
</tbody>
</table>
3.5.4 SE5908A/SE5916A Pin Assignments

DB9 to RS-232/RS-485/RS-422 connectors

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232</th>
<th>RS-422</th>
<th>RS-485</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>RxD</td>
<td>TxD+</td>
<td>Data+</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>RxD+</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>RxD-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
<td>TxD-</td>
<td>Data-</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

5-Pin Terminal Block to RS-232/RS-485/RS-422 connectors

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS-232</th>
<th>RS-422 4-Wire RS-485</th>
<th>2-W RS-485</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RxD</td>
<td>TxD+</td>
<td>Data+</td>
</tr>
<tr>
<td>2</td>
<td>CTS</td>
<td>TxD-</td>
<td>Data-</td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>RxD+</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>RxD-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
<td>SG (Signal Ground)</td>
</tr>
<tr>
<td>---</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4 Software Specifications

The device node is the communication interface between user space and hardware device in Linux. Each chapter is divided into two parts:

1. **How to program these interfaces** – The main purpose is to provide the way to access device node with sample code.
2. **How to test the interface** – The main purpose is to describe the way to use Linux internal or Atop supporting tools to test the interfaces.

### 4.1 COM Port Interface

SE59XX Series (Except SE5900A) are equipped with COM ports. Each COM port is registered as a TTY terminal interface with the kernel.

- Maximum baud rate: 921600
- Minimum baud rate: 300
- Serial interface supported: RS232/RS485/RS422

The sample programs are in the software/atop_application/utils/atop_loopback folder:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs232_loopback.c</td>
<td>Loopback test program for RS232 ports</td>
</tr>
<tr>
<td>rs422_loopback.c</td>
<td>Loopback test program for RS422 ports</td>
</tr>
<tr>
<td>rs485_loopback.c</td>
<td>Loopback test program for RS485 ports</td>
</tr>
</tbody>
</table>

**4.1.1 Program COM port interface**

The following tables list the device node of COM port for each EVM model.

<table>
<thead>
<tr>
<th>Device node</th>
<th>Major &amp; Minor number</th>
<th>Device Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttyATOP0</td>
<td>266 0</td>
<td>Character</td>
<td>ATOP COM port 1</td>
</tr>
<tr>
<td>ttyATOP1</td>
<td>266 1</td>
<td>Character</td>
<td>ATOP COM port 2</td>
</tr>
<tr>
<td>ttyATOP2</td>
<td>266 2</td>
<td>Character</td>
<td>ATOP COM port 3</td>
</tr>
<tr>
<td>ttyATOP3</td>
<td>266 3</td>
<td>Character</td>
<td>ATOP COM port 4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Table 4.3 SE59XX Programming commands per device node

<table>
<thead>
<tr>
<th>Device node</th>
<th>ioctl command</th>
<th>Command Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttyATOP0-3</td>
<td>0x9000</td>
<td>Configure SE59XX COM port as one of RS232 / RS485 / RS422</td>
</tr>
</tbody>
</table>

Table 4.4 SE59XX ioctl command of COM Port

<table>
<thead>
<tr>
<th>ioctl command</th>
<th>parameter type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x9000</td>
<td>integer</td>
<td>1</td>
<td>Configure to RS232 mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Configure to RS422 mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Configure to RS485 mode</td>
</tr>
</tbody>
</table>

In Linux system, user can use POSIX standard terminal interface to configure baud rate, data length, etc. It is called Termios and defined in system document <termios.h>. Please refer related Linux system document to configure it.

### 4.2 Network Interface

SE59XX Series are equipped with Network ports. The sample program in the folder `software/atop_application/utils/atop_tcpserver` describes how to use COM ports in combination with TCP server connections:

Table 4.5 Sample programs for TCP server connection to COM port communication

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>atop_tcp_server.c</td>
<td>A sample program to use TCP server connection and COM port to make data communication.</td>
</tr>
</tbody>
</table>

### 4.3 Other Interfaces

There are multiple other interfaces available on SE59XX platform, depending on the actual hardware in use. Some devices are equipped with 4G connectivity, others with Relays and Digital inputs and soon. ATO P’s convenient Software Development Kit is standardized for the whole family. We put at disposal simple programs that you can easily copy or simulate to make the best use of all interfaces.

All sample programs are in `/atop_application/utils/atop_sdk` folder.
4.3.1 Buzzer
There is one Buzzer in each SE59XX device. The sample program is available in the software/atop_application/utils/atop_sdk folder:

Table 4.6 Sample program for Buzzer

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>buzzer.c</td>
<td>A sample program to use the device’s Buzzer.</td>
</tr>
</tbody>
</table>

4.3.2 Digital Inputs
There are 2 Digital inputs on SE5901B-10. The sample program is available in the software/atop_application/utils/atop_sdk folder:

Table 4.7 Sample program for Digital Input

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>di_test.c</td>
<td>A sample program to use the device’s Digital Inputs.</td>
</tr>
</tbody>
</table>

4.3.3 Digital Outputs
There are 2 Digital Outputs on SE5901B-10. The sample program is available in the software/atop_application/utils/atop_sdk folder:

Table 4.8 Sample program for Digital Output

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dh_test.c</td>
<td>A sample program to use the device’s Digital Outputs.</td>
</tr>
</tbody>
</table>

4.3.4 Relay Outputs
There are Relay outputs on SE5904D, SE5908, SE5916, SE590A, SE5908A and SE5916A. The sample program is available in the software/atop_application/utils/atop_sdk folder:

Table 4.9 Sample program for Relay Output

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>relay.c</td>
<td>A sample program to use the device’s Relay Outputs.</td>
</tr>
</tbody>
</table>

4.3.5 LCM (SE5908 / SE5916 only)
There is an LCM in SE5908 and SE5916. The sample program is available in the
software/atop_application/utils/atop_sdk folder:

Table 4.10 Sample program for LCM

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcm_test.c</td>
<td>A sample program to use the device's LCM.</td>
</tr>
</tbody>
</table>

4.3.6  **Reset Button**
All SE59XX hardware platforms have a reset button. The sample program is available in the software/atop_application/utils/atop_sdk folder:

Table 4.11 Sample program for Reset Button

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button.c</td>
<td>A sample program to use the device's reset button.</td>
</tr>
</tbody>
</table>

4.3.7  **Hardware Watchdog Timer**
There is a hardware watchdog IC on each CPU board. If this IC is not reset within 1.6 seconds, then the system will reboot. This implementation allows the hardware to automatically understand if the system is crashing, for whatever reason. During a System crash, the OS won't reset the IC within the deadline and therefore the system will automatically reboot. All SE59XX hardware platforms do have an integrated hardware watchdog timer. The sample programs available in the software/atop_application/utils/atop_sdk folder:

Table 4.12 Sample program for WDT

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hwd.c</td>
<td>A sample program to use the device's Hardware Watchdog timer.</td>
</tr>
</tbody>
</table>

4.3.8  **LEDs**
Different devices in SE59XX family have different LEDs based on the number of ports. But all devices are equipped with a RUN/Fault LED. The sample programs available in the software/atop_application/utils/atop_sdk folder:

Table 4.13 Sample program for LEDs

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alarmLed.c</td>
<td>A sample program to use the device's Alarm (RED) LED</td>
</tr>
<tr>
<td>runLed.c</td>
<td>A sample program to use the device's Run (GREEN) LED</td>
</tr>
</tbody>
</table>
4.3.9 3G/4G Cellular (SE5901B only)
The sample programs are available in the software/applicationutils/atop_sdk folder:

Table 4.14 Sample program for Cellular functions

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>atop_4G_apn.c</td>
<td>A sample program to set the cellular Access Point</td>
</tr>
<tr>
<td>atop_4G_connect.c</td>
<td>A sample program to connect to 3G/4G</td>
</tr>
<tr>
<td>atop_4G.DialOnBoot.c</td>
<td>A sample program to set the device to dial on boot</td>
</tr>
<tr>
<td>atop_4G_PinDisable.c</td>
<td>A sample program to disable the SIM PIN</td>
</tr>
<tr>
<td>atop_4G_PinEnable.c</td>
<td>A sample program to enable the SIM PIN</td>
</tr>
<tr>
<td>atop_4G_reconnect.c</td>
<td>A sample program to reconnect to the cellular network</td>
</tr>
<tr>
<td>atop_4G_reset.c</td>
<td>A sample program to reset the cellular module</td>
</tr>
</tbody>
</table>

4.3.10 RTC Interface

There is one RTC clock via I2C interface, and it supports time unit to seconds, minutes, hours, days, months, and years up to year 2099.
5 Testing interfaces

ATOP provides some simple text programs. Please follow the below instructions to test the interfaces when it's needed.

5.1 Test COM port interface – transmit and receive

RS232/RS422/RS485 loopback test:

Execute rs232_loopback under the kernel shell. Be sure that you have connect the testing COM ports connected. Be sure to make TXD & RXD pins connected.

rs232_loopback
rs422_loopback
rs485_loopback

The baud rate is set at 115200

5.1.1 Test COM port interface by using atop_tcp_server

5.1.1.1 Test Method

The setup of the testing is shown as Figure 5.1.

![Figure 5.1 COM loopback test connection](image)

5.1.1.2 Test Execution

1) Execute the command "atop_tcp_server" as next line to test RS232 with baud rate 115200.
2) Type "ps -ef" from super terminal program to check if atop_tcp_server is executed as Figure 5.2

```
957 admin proftp: [accepting connections]
964 admin -sh
965 admin /usr/sbin/snmpd -ld -f /dev/null -p /var/run/snmpd.pid -c /etc
971 admin dhclient -6 -nw -pf /var/run/dhclient.eth0.pid -lf /var/lib/dhcp
1898 admin sleep 10
1899 admin **atop_tcp_server rs232 115200**
1904 admin ps
```

Figure 5.2 Process execution on SE5904D, example

3) Connect loopback for COM1 and execute TCPtest from MS-Windows as Figure 5.3

```
Figure 5.3 Setup TCPtest.exe for COM port loopback test
```

4) Select the TCP_Server mode and input IP address and TCP port number for COM1.
5) Click "Connect" to make TCP connections. The data keyed in "Send" box will be sent through the COM port. "Send Loop" is used to send continuously every certain period of time.
6) Click "Send One" to start the data transmission from PC to COM port. The data received from loopback link will be shown on the lower part "Receive" box of the Window as Figure 3-4.
7) You should be able to see "Hello" as hexadecimal display of each character "48 65 6C 6C 6F 0D 0A" shown on "Receive" box.
The default mapping table between TCP port number and COM port number:

<table>
<thead>
<tr>
<th>COM port</th>
<th>TCP port</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM 1</td>
<td>4660</td>
</tr>
<tr>
<td>COM 2</td>
<td>4661</td>
</tr>
<tr>
<td>COM 3</td>
<td>4662</td>
</tr>
<tr>
<td>COM 4</td>
<td>4663</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>COM 16</td>
<td>4660 + (16-1)</td>
</tr>
</tbody>
</table>

### 5.2 Test Buzzer interface

Upload the test file “buzzer” from `.software/atop_application folder into /jffs2 and execute the following command on the kernel shell:

```
#!/rootfs/ordinon/buzzer

on: enable
off: disable
```

You should hear the buzzer
5.3 Test Digital Input

Upload the test file "di_test" from ./software/atop_application folder into /jffs2 and execute the following command on the kernel shell:

```
./di_test
```

You can see the message print DI0/1 value.

5.4 Test Digital Output

Upload the test file "do_test" from ./software/atop_application folder into /jffs2 and execute the following command on the kernel shell:

```
./do_test
```

You can use multimeter to check the DO0/D1 turn on 5 sec then turn off.

5.5 Test Hardware Relay Output

Use test tool "relay" to test HW relay device.

```
./relay
```

You can use multimeter to check the relay turn on then turn off after 10 sec.

5.6 Test Hardware Button

Use test tool "button" to get “press” then “release” event.

```
./button
```

5.7 Test Hardware Watchdog Interface (WDT)

Upload test file "hwd" from ./software/atop_application folder into /jffs2 and execute the following command on the kernel shell:
If watchdog is not cleared or disabled in the source code, then system will restart automatically after 1.6 sec.

5.8 Test device LED

Upload test file "runLed" from .software/atop_application folder to /jffs2 and execute the following command on the kernel shell:

```
./runLed <on/off>
```

on: enable
def: disable
You should see the RUN Led turn on or off.

Use test tool "alarmLed" to test HW alarm LED.

```
./alarmLed
./alarmLed off
```

You should see the ALARM(Red) Led turn on or off.

5.9 Test RTC interface

Upload test file "rtc" from .software/atop_application folder to /jffs2 and execute the following command on the kernel shell:

Set link file:

```
ln -s rtc get_rtc
ln -s rtc set_rtc
ln -s rtc rtc2system
ln -s rtc system2rtc
```

5.9.1 Setup RTC time:

Execute the following command on the kernel shell:

```
/set_rtc 2017/02/15:18:00:00
```

It will process both commands "date -s 2017-02-15 18:00:00" and "hwclock -w /dev/rtc1".
5.9.2 **Read RTC time:**
Execute the following command on the kernel shell:

```
./get_rtc
```

It will process command "hwclock –r –f /dev/rtc1".
The console will display the current RTC time such as "Wed Feb 11 11:50 2017"

5.9.3 **RTC2system**
Execute the following command on the kernel shell:

```
./rtc2system
```

It will process command "hwclock –s –f /dev/rtc1".
rtc2system: set system time from hardware clock.

5.9.4 **system2RTC**
Execute the following command on the kernel shell:

```
./system2rtc
```

It will process command "hwclock –w –f /dev/rtc1".
system2rtc: set hardware clock from system time.

In order to make sure the clock was set correctly, turn off the power and restart the system. After startup is completed, check the RTC time.

5.10 **Using NOR Flash – JFFS2**

There is a NOR flash on each device. 16MB of it is reserved for user applications mounted on JFFS2 file system. This will be mounted automatically on system start-up. The user can put all application programs and the related data into JFFS2.

All data in the JFFS2 will be kept when system is shut down.

5.11 **MQTT**

You can use http://test.mosquitto.org/MQTT broker (server) for testing.
Subscriber and Publisher example:

mosquitto_sub -h test.mosquitto.org -t "atop" -v &
mosquitto_pub -h test.mosquitto.org -t "atop" m"HelloWorld"

MQTT with example (You can download test certificates from test.mosquitto.org):

mosquitto_sub -h test.mosquitto.org -p 8883 -t "atop" --cafile /jffs2/mosquitto.org.crt &
mosquitto_pub -h test.mosquitto.org -p 8883 -t "atop" --cafile ./mosquitto.org.crt -d -m "test"

MQTT with username and password example:

mosquitto_sub -h 192.168.4.238 -u atop -P 123456 -d -t atop &
mosquitto_pub -h 192.168.4.238 -u atop -P 123456 -d -t atop m"test123"


5.12 Firmware upgrade

Use test tool "frw-upgrd" to upgrade kernel & rootfs.

./frw-upgrd xxx.dld

Note: The upgrade program only support dld file format.
6 Software API Reference

Software API is to be referred by the software application to configure system environment, include user name, password and network setting. The user can configure and then restart the system to make the new environment effective.

6.1 File List

Here is a list of all documented files with brief descriptions:

atop_application/utils/atop_libsdk/network.c (Network APIs) .................................................................
atop_application/utils/atop_libsdk/system.c (EEPROM User Name and Password settings) ........
include/atop_alarmled.h (Set alarmLED on/off) ......................................................................................
include/atop_button.h (Read reset button status) ......................................................................................
include/atop_buzzer.h (Control buzzer on/off) ..........................................................................................
include/atop_di.h (Get DI on/off) ..............................................................................................................
include/atop_do.h (Control DO on/off) ....................................................................................................
include/atop_hwd.h (Watch dog control API) ...........................................................................................
include/atop_libwl.h (Wireless control APIs) ...........................................................................................
include/atop_runled.h (RunLed control API) ............................................................................................


7 Data Structure Documentation

7.1 sessiontag Struct Reference

7.1.1 data fields

The documentation for this struct was generated from the following file:
atop_application/utils/atop_tcp_server/atop_tcp_server.c

- `pthread_t thandler`
- `int tid`
- `int uartfd`
- `int serv_sockfd`
- `int serv_acceptsockfd`
- `struct sockaddr_in serv_addr`
- `struct sockaddr_in client_addr`
- `int serv_link_state`
- `int serv_socket_init`
- `int err_count`

7.2 Network APIs

```c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "atop_common.h"
#include "atop_eeprom.h"
#include "mobile.h"
```

7.2.1 Functions

```c
void AtopSDKSetNetIP (int eth, char *ip)

AtopSDKSetNetIP
  Sets the device’s network IP on the selected Ethernet port

void AtopSDKSetNetMask (int eth, char *mask)

AtopSDKSetNetMask
  Sets the device’s subnet mask on the selected Ethernet port
```
void AtopSDKSetNetGateway (int eth, char *gw)

AtopSDKSetNetGateway

Sets the device's default Gateway IP on the selected Ethernet port

void AtopSDKSetNetDefGateway (int eth)

AtopSDKSetNetDefGateway

Sets the device's network IP on the selected Ethernet port

void AtopSDKSet4G DialOnBoot (int value)

AtopSDKSet4G DialOnBoot

Enables or disables automatic 4G dial on boot (SE5901B only)

void AtopSDKSet4G Apn (char *apn)

AtopSDKSet4G Apn

Sets the device's cellular Access Point (SE5901B only)

void AtopSDKSet4G PinEnable (char *pinCode)

AtopSDKSet4G PinEnable

Sets the SIM card's PIN code (SE5901B only)

void AtopSDKSet4G PinDisable (void)

AtopSDKSet4G PinDisable

Disables the SIM PIN code

void AtopSDKSet4G Connect (void)

AtopSDKSet4G Connect

Connects 3G/4G

void AtopSDKSet4G ReConnect (int value)

AtopSDKSet4G ReConnect

Sets automatic reconnect in case of connection loss

void AtopSDK4G HWReset (void)

AtopSDK4G HWReset

Resets the 3G/4G module

void AtopSDKSet4G Disconnect (void)

AtopSDKSet4G Disconnect

Disconnects from the 3G/4G Cellular network

7.3 Network APIs Function documentation

7.3.1 void AtopSDK4G HWReset (void )

API: AtopSDK4G HWReset
Arguments: nothing (void)
Function: resets 3G/4G module
Returns: nothing (void)

Example code:
```c
218 {
219     int fno;
220     int value;
221     fno = open("/dev/atop_3g_reset", O_WRONLY);
222     if (fno > 0)
223     {
224         value = 0;
225         write(fno, &value, 1);
226         sleep(1);
227         value = 1;
228         write(fno, &value, 1);
229     } 
230     else
231     printf("error (%d)\n", fno);
232     close(fno);
233     //printf("reset 3g module\n");
234     return;
235 }
```

7.3.2  void AtopSDKSet4GApn (char * apn)

API: AtopSDKSet4GApn
Arguments: apn (char) – string of APN (access point name)
Function: writes APN string to Module ini file
Returns: nothing (void)

Example code:
```c
145 {
146     set_apn_info((uint8_t *)apn);
147 }
```

7.3.3  void AtopSDKSet4GConnect (void)

API: AtopSDKSet4GApn
Arguments: nothing (void)
Function: triggers a connection
Returns: nothing (void)

Example code:
```c
190 {
191     threeg_connect();
192 }
```

7.3.4  void AtopSDKSet4GDialOnBoot (int value)

API: AtopSDKSet4GDialOnBoot
Arguments: value (integer)
Function: enables or disables 3G/4G cellular dial-on-boot function
Returns: nothing (void)

Example code:
```c
129 {
130     set_dial_when_bootup(value);
131 }
```
7.3.5  **void AtopSDKSet4GDisconnect (void)**

**API:** AtopSDKSet4GDisconnect  
**Arguments:** nothing (void)  
**Function:** closes a 3G/4G cellular connection and resets the 3G/4G cellular module  
**Returns:** nothing (void)  
**Example code:**
```
247 {
248    threeg_disconnect();
249    AtopSDK4GHWReset();
250 }
```

7.3.6  **void AtopSDKSet4GPinDisable (void)**

**API:** AtopSDKSet4GPinDisable  
**Arguments:** nothing (void)  
**Function:** disables 3G/4G cellular module SIM PIN check  
**Returns:** nothing (void)  
**Example code:**
```
176 {
177    set_pin_enable(0);
178 }
```

7.3.7  **void AtopSDKSet4GPinEnable (char * pinCode)**

**API:** AtopSDKSet4GPinEnable  
**Arguments:** new PIN code (char)  
**Function:** writes SIM PIN code to the 3G/4G cellular module to the ini file  
**Returns:** nothing (void)  
**Example code:**
```
161 {
162    set_pin_enable(1);
163    set_pinCode((uint8_t *)pinCode);
164 }
```

7.3.8  **void AtopSDKSet4GReConnect (int value)**

**API:** AtopSDKSet4GReConnect  
**Arguments:** recon value (int)  
**Function:** writes 3G/4G cellular reconnection settings to the ini file  
**Returns:** nothing (void)  
**Example code:**
```
204 {
205    set_reconn_enable(value);
206 }
```

7.3.9  **void AtopSDKSetNetDefGateway (int eth)**
API: AtopSDKSetNetDefGateway
Arguments: eth: index of the Eth interface (ethX) - (int)
Function: writes default gateway settings to EEPROM
Returns: nothing (void)
Example code:

```
110 {
111     char defGW[16];
112     sprintf(defGW, "%d", eth);
113     AtopEESetDefaultGW(NULL, defGW);
114 }
```

7.3.10 void AtopSDKSetNetGateway (int eth, char *gw)

API: AtopSDKSetNetGateway
Arguments:
- eth: index of the Eth interface (ethX) - (int)
- gw: gateway address (char)
Function: writes gateway address settings to EEPROM
Returns: nothing (void)
Example code:

```
88 {
89     struct in_addr inp;
90     char info[16];
91     sprintf(info, "%dN", eth);
92     if (inet_aton(gw, &inp) > 0)
93         AtopEESetLanGateway(info, inet_ntoa(inp));
94 }
```

7.3.11 void AtopSDKSetNetIP (int eth, char *ip)

API: AtopSDKSetNetIP
Arguments:
- eth: index of the Eth interface (ethX) - (int)
- ip: new IP address of port ethX (char)
Function: writes new IP address settings of specified port to EEPROM
Returns: nothing (void)
Example code:

```
34 {
35     struct in_addr inp;
36     char info[16];
37     sprintf(info, "%dN", eth);
38     if (!strncmp("DHCP", ip, strlen("DHCP")))
39         { AtopEESetLanIPMode(info, "DHCP");
40         } else
41         { if (inet_aton(ip, &inp) > 0)
42             { AtopEESetLanIPMode(info, "Static");
43                 AtopEESetLanIP(info, inet_ntoa(inp));
44             }
45         }
```
### 7.3.12 void AtopSDKSetNetMask (int eth, char *mask)

**API:** AtopSDKSetNetMask

**Arguments:**
- `eth`: index of the Eth interface (ethX) - (int)
- `mask`: subnet mask of port ethX (char)

**Function:** writes subnet mask settings to EEPROM

**Returns:** nothing (void)

**Example code:**
```c
66 {
67     struct in_addr inp;
68     char info[16];
69     sprintf(info, "%dN", eth);
70     if (inet_aton(mask, &inp) > 0)
71         AtopEESetLanNetmask(info, inet_ntoa(inp));
72 }
```

### 7.4 EEPROM User Name and Password Settings APIs

atop_application/utils/atop_libsdk/system.c File Reference

```c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include "atop_common.h"
#include "atop_eeprom.h"
```

#### 7.4.1 Functions

- `void AtopSDKSetUserName (char *name)`
  
  **AtopSDKSetUserName**
  
  Saves the device’s Administrator Username to the EEPROM

- `void AtopSDKSetUserPassword (char *password)`
  
  **AtopSDKSetUserPassword**
  
  Saves the device’s Administrator Password to the EEPROM
## 7.5 EEPROM User Name and Password Settings API Function documentation

### 7.5.1 void AtopSDKSetUserName (char name)

**API:** AtopSDKSetUserName  
**Arguments:** name – new administrator username (char)  
**Function:** writes new System User Name to the EEPROM  
**Returns:** nothing (void)  
**Example code:**

```c
27 {  
28     char info[16];  
29     sprintf(info, "N");  
30     AtopEESetSysUserName(info, name);  
31 }
```

### 7.5.2 void AtopSDKSetUserPassword (int password)

**API:** AtopSDKSetUserPassword  
**Arguments:** password (char)  
**Function:** writes new password to the EEPROM  
**Returns:** nothing (void)  
**Example code:**

```c
45 {  
46     char info[16];  
47     sprintf(info, "N");  
48     AtopEESetSysUserPassword(info, password);  
49 }
```

## 7.6 Run LED API Function documentation

include/atop_runled.h File Reference

### 7.6.1 Functions

Void AtopSetRunLed (u32 onMs, u32 offMs)  

AtopSetRunLed  
Sets RUN Led ON and OFF time settings

### 7.6.2 void AtopSetRunLed (onMs u32, offMs u32)

**API:** AtopSetRunLed  
**Arguments:**
• onMs: delay on in ms - (u32)
• offMs: delay off in ms - (u32)

**Function**: sets RUN Led ON and OFF time settings  
**Returns**: nothing (void)  
**Example code**: not available

### 7.7 Alarm LED API Function documentation

include/atop_alarmled.h File Reference

#### 7.7.1 Functions

void **AtopSetAlarmLed** (u32 value)

`AtopSetAlarmLed`

Turns the Alarm LED Off and On

#### 7.7.2 **void AtopSetAlarmLed (u32 value)**

**API**: `AtopSetAlarmLed`

**Arguments**: value – on or off (u32)

• 0: set to OFF  
• 1: set to ON

**Function**: Turns the Alarm LED ON or OFF  
**Returns**: nothing (void)

### 7.8 Read Reset Button API

include/atop_button.h File Reference

#### 7.8.1 Functions

int **AtopButton** (void)

`AtopButton`

Reads reset button status

#### 7.8.2 **int AtopButton (void)**

**API**: `AtopButton`
Arguments: nothing (void)
Function: Reads reset button status
Returns: button status (int)
  • 0: button not pushed
  • 1: button pushed

7.9 Use Buzzer API documentation

include/atop_buzzer.h File Reference

7.9.1 Functions

void AtopBuzzeronOff (int value)

  AtopBuzzeronOff
  Starts or stops the buzzer

7.9.2 Macros

#define BUZZER_OFF  0
#define BUZZER_ON   1

7.9.3 void AtopBuzzeronOff (int value)

  API: AtopBuzzeronOff
  Arguments: value – whether the buzzer should be on or off (0=OFF; 1=ON)
  Function: starts or stops the buzzer
  Returns: nothing (void)

7.10 Read Digital Inputs API documentation

include/atop_di.h File Reference

7.10.1 Functions

int AtopGetDI (int index)

  AtopGetDI
  Returns Digital Input value of DI index
7.10.2  `int AtopGetDI (int index)`

**API:** AtopGetDI  
**Arguments:** index – defines the number of Digital Input to read value from  
**Function:** Reads digital input value (high or low)  
**Returns:** Digital Input value (int)

---

### 7.11 Write Digital Output API documentation

include/atop_do.h File Reference

#### 7.11.1 Functions

int `AtopSetDO (int index, int value)`  
**AtopSetDO**  
Writes to Digital Output channel index, the value

#### 7.11.2 `int AtopSetDO (int index, int value)`

**API:** AtopSetDO  
**Arguments:**  
- index: defines the number of Digital Output selected for control  
- value: defines the status set to the selected Digital output (1 or 0)  
**Function:** Controls digital output to be off or on  
**Returns:** error code (int)

---

### 7.12 Hardware Watchdog API documentation

include/atop_hwd.h File Reference

#### 7.12.1 Functions

Void `atop_hwd_enable (void)`  
`atop_hwd_enable`  
enables Hardware Watchdog  

Void `atop_hwd_disable (void)`  
`atop_hwd_disable`  
disables Hardware Watchdog
Void atop_hwd_clear (void)
   atop_hwd_clear
   clears Hardware Watchdog

7.12.2 Macros

#define HWD_DEV "/dev/atop_hwd0"

7.12.3 void atop_hwd_clear (void)

   API: atop_hwd_clear
   Arguments: nothing (void)
   Function: kicks hardware watchdog timer (reset)
   Returns: nothing (void)

7.12.4 void atop_hwd_disable (void)

   API: atop_hwd_disable
   Arguments: nothing (void)
   Function: disables Hardware watchdog
   Returns: nothing (void)

7.12.5 void atop_hwd_enable (void)

   API: atop_hwd_enable
   Arguments: nothing (void)
   Function: enables hardware watchdog control
   Returns: nothing (void)
7.13 Wi-Fi USB Dongle control APIs documentation

These APIs can be used when SE59XX is used with one of the Approved USB to Wi-Fi dongles. Please contact ATOP for more information on the supported models.

include/atop_libwl.h File Reference

7.13.1 Functions

Void create_default_ini_file (void)

create_default_ini_file
creates Wi-Fi INI configuration file. To be done before establishing a new connection

Int get_ssid (char **ssid)

get_ssid
reads the current SSID from the INI file and saves to **ssid

Int set_ssid (char **ssid_name)

set_ssid
sets new SSID to the INI file

Int get_key_mgmt (char **key_mgmt)

get_key_mgmt
reads the security configuration from the INI file and saves to **key_mgmt

Int set_key_mgmt (int mode)

set_key_mgmt
writes the security configuration to the INI file

Int get_psk (char **psk)

get_key_mgmt
reads the security key from the INI file and saves to **psk

Int set_psk (char **psk_name)

set_key_mgmt
writes the new security key to the INI file

Int parse_setting (void)

parse_setting
defines the parse settings

Void run_connection (void)

run_connection
initiates the connection
7.13.2  void create_default_ini_file (void)

API: create_default_ini_file
Arguments: nothing (void)
Function: creates default INI configuration file for Wi-Fi dongle settings
Returns: nothing (void)
Example code:

```
13 {
14     FILE *fout;
15     fout = fopen(ini_file_name, "w");
16     fprintf(fout,
17         "\n"
18         "[network]\n"
19         "ssid = default;\n"
20         "key_mgmt = WPA2-PSK;\n"
21         "psk = 12345678;\n"
22 } fclose(fout);
```

7.13.3  void get_key_mgmt (char key_mgmt)

API: get_key_mgmt
Arguments: key_mgmt – buffer for the management mode read value (char)
Function: reads Wi-Fi dongle key management mode from the INI configuration file
Returns: Error code (int)
  - -1 : ERROR – Cannot open file; cannot read key management from file
  - 0 : Successful
Example code:

```
86 {
87     char *str;
88 // read key from ini
89     dictionary *ini = iniparser_load(ini_file_name);
90     if (ini==NULL) {
91         fprintf(stderr, "Cannot open %s\n",ini_file_name);
92         return -1;
93     }
94 // read key_mgmt from ini
95     str = iniparser_getstring(ini, "network:key_mgmt", NULL);
96     // malloc a space and copy key_mgmt to it
97     if ( str!=NULL )
98         *key_mgmt = strdup(str);
99     else {
100         printf("Get key_mgmt error!\n");
101         iniparser_freedict(ini);
102         return -1;
103     }
104 // free dictionary space
105     iniparser_freedict(ini);
106     return 0;
107 }
```

7.13.4  int get_psk(char psk)

API: get_psk
Arguments: psk : buffer for PSK (string)
Function: Reads current passphrase (PSK) from INI file and writes it to the argument
Returns: Error code (int)
-1 : ERROR – Cannot open file, or get PSK error  
0 : Successful

Example code:

```c
157 { 
158     char *str;
159
160     // read key from ini
161     dictionary *ini = iniparser_load(ini_file_name);
162     if (ini==NULL) {
163         fprintf(stderr, "Cannot open \%s\n",ini_file_name);
164         return -1;
165     }  
166
167     // read psk from ini
168     str = iniparser_getstring(ini, "network:PSK", NULL);
169     // malloc a space and copy psk to it
170     if ( str!=NULL )
171     *psk = strdup(str);
172 else {
173         printf("Get PSK error!
");
174         iniparser_freedict(ini);
175         return -1;
176     }
177
178     // free dictionary space
179     iniparser_freedict(ini);
180     return 0;
181 }
```

7.13.5  

`int get_ssid (char ssid)`

API: get_ssid  
Arguments: ssid : buffer for reading SSID (string)  
Function: reads SSID from ini file.

Returns: Error code (int)  
-1 : ERROR: Cannot open file or function error  
0 : Successful

Example code:

```c
27 { 
28     char *str;
29
30     // read key from ini
31     dictionary *ini = iniparser_load(ini_file_name);
32     if (ini==NULL) {
33         fprintf(stderr, "Cannot open \%s\n",ini_file_name);
34         return -1;
35     }
36
37     // read ssid from ini
38     str = iniparser_getstring(ini, "network:SSID", NULL);
39
40     // malloc a space and copy ssid to it
41     if ( str!=NULL )
42     * ssid = strdup(str);
43 else {
44         printf("Get ssid error!
");
45         iniparser_freedict(ini);
46         return -1;
47     }
48
49     // free dictionary space
50     iniparser_freedict(ini);
51     return 0;
52 }
```
7.13.6  \textit{int parse\_setting} (void)

\textbf{API:} parse\_setting  \\
\textbf{Arguments:} nothing (void)  \\
\textbf{Function:} resets Wi-Fi connection settings (via USB-dongle)  \\
\textbf{Returns:} nothing (void)  \\
\textbf{Example code:}

\begin{verbatim}
226 {  
227     char *ssid, *psk, *key_mgmt;
228  
229     // read key from ini
230     dictionary *ini = iniparser_load(ini_file_name);
231     if (ini==NULL) {
232         fprintf(stderr, "Cannot open %s\n", ini_file_name);
233         return -1;
234     }
235  
236     // read ssid.psk.key_mgmt from ini
237     ssid = iniparser_getstring(ini, "network:SSID", NULL);
238     psk  = iniparser_getstring(ini, "network:PSK", NULL);
239     key_mgmt = iniparser_getstring(ini, "network:key_mgmt", NULL);
240  
241     // parse ini into config format
242     FILE *fout;
243     fout = fopen(parse_conf_name, "w");
244     fprintf(fout,"network={
245     fprintf(fout,"\"tscan_ssid=1\n"");
246     if (ssid != NULL) fprintf(fout,"\"tssid=\"%s\n", ssid);
247     if ((psk != NULL) && (strcmp(key_mgmt,"None") != 0)) fprintf(fout,"\"tpsk=\"%s\n", psk);
248     if (key_mgmt != NULL) fprintf(fout,"\"tkey_mgmt=\"%s\n", key_mgmt);
249     fclose(fout);
250     iniparser_freedict(ini);
251     return 0;
252 }
\end{verbatim}

7.13.7  \textbf{void run\_connection} (void)

\textbf{API:} run\_connection  \\
\textbf{Arguments:} nothing (void)  \\
\textbf{Function:} initiate a Wi-Fi connection through the USB Wi-Fi dongle. It runs authentication via \texttt{wpa_supplication}.  \\
\textbf{Returns:} nothing (void)  \\
\textbf{Example code:}

\begin{verbatim}
256 {
257     char cmd[60];
258     system("killall wpa_supplicant");
259     sprintf(cmd,"wpa_supplicant -Dnl80211 -iwlans0 -c%s \n", parse_conf_name);
260     system(cmd);
\end{verbatim}

7.13.8  \textit{int set\_key\_mgmt} (int \textit{mode})

\textbf{API:} set\_key\_mgmt  \\
\textbf{Arguments:} \textit{mode}: - (int)  \\
\begin{itemize}
\item 0: Disabled  
\item 1: WPA-PSK  
\item 2: WPA2-PSK
\end{itemize}  \\
\textbf{Function:} sets security key management mode to INI file for Wi-Fi dongle connection  \\
\textbf{Returns:} error code (int)  \\
\begin{itemize}
\item -1 : Cannot open file  
\item 0 : Successful
\end{itemize}  \\
\textbf{Example code:}
### 7.13.9 `int set_psk (char psk_name)`

**API:** `set_psk`

**Arguments:** `psk_name`: string of psk to be saved in ini file - (char)

**Function:** writes psk to ini file

**Returns:** int:
- `-1`: Error – cannot open file, Invalid command, wrong length
- `0`: Successful

**Example code:**

```c
    185 {  
    186     char *key_mgmt;
    187    188     // read key from ini
    189     dictionary *ini = iniparser_load(ini_file_name);
    190     if (ini==NULL) {
    191         fprintf(stderr, "Cannot open %s\n", ini_file_name);
    192         return -1;
    193     }
    194
    195     // cannot set psk, if key_mgmt is disable
    196     key_mgmt = iniparser_getstring(ini, "network:key_mgmt", NULL);
    197     if ( strcmp(key_mgmt,"None")==0 ) {
    198         printf("Invalid commend!\nkey_mgmt is disable\n");
    199         iniparser_freedict(ini);
    200         return -1;
    201     }
    202     // write psk into ini, if it's length between 8 and 63
```
if ( (8<=strlen(*psk_name)) && (strlen(*psk_name)<=63) ) {
    iniparser_set(ini, "network:PSK", *psk_name);
    FILE *fin = fopen("/jffs2/tmpFile", "w");
    iniparser_dump_ini(ini, fin);
    fclose(fin);
    remove(ini_file_name);
    rename("/jffs2/tmpFile", ini_file_name);
} else {
    printf("Set PSK error!\nPSK length must be between 8 and 63.\n");
    iniparser_freedict(ini);
    return -1;
}

// free dictionary space
iniparser_freedict(ini);
return 0;

7.13.10  int set_ssid (char ssid_name)

API: set_ssid
Arguments: ssid_name: string of SSID - (char)
Function: writes SSID to ini file
Returns: int:
    -1 : error – SSID length problem or cannot open file
    0 : successful

Example code:
```c
if ( (1<=strlen(*ssid_name)) && (strlen(*ssid_name)<=32) ) {
    iniparser_set(ini, "network:SSID", *ssid_name);
    FILE *fin = fopen("/jffs2/tmpFile", "w");
    iniparser_dump_ini(ini, fin);
    fclose(fin);
    remove(ini_file_name);
    rename("/jffs2/tmpFile", ini_file_name);
} else {
    printf("Set SSID error!\nSSID length must be between 1 and 32.\n");
    iniparser_freedict(ini);
    return -1;
}
```
Atop Technologies, Inc.

www.atoponline.com
www.atop.com.tw

TAIWAN HEADQUARTER:
2F, No. 146, Sec. 1, Tung-Hsing Rd,
30261 Chupei City, Hsinchu County
Taiwan, R.O.C.
Tel: +886-3-550-8137
Fax: +886-3-550-8131

ATOP CHINA BRANCH:
3F, 75th, No. 1066 Building,
Qingzhou North Road,
Shanghai, China
Tel: +86-21-64956231

ATOP INDIA OFFICE:
Abhishek Srivastava
Head of India Sales
Atop Communication Solution(P) Ltd.
No. 22, Kensington Terrace,
Kensington Rd,
Bangalore, 560008, India
Tel: +91-80-4920-6363
E-mail: Abhishek.S@atop.in

ATOP INDONESIA BRANCH:
Jopson Li
Branch Director
Wisma Lampung Jl.
No. 40, Tomang Raya
Jakarta, Barat, 11430, Indonesia
Tel: +62-857-10595775
E-mail: jopsonli@atop.com.tw

ATOP EMEA OFFICE:
Bhaskar Kailas (BK)
Vice President (Business Development)
Atop Communication Solution(P) Ltd.
No. 22, Kensington Terrace,
Kensington Rd,
Bangalore, 560008, India
Tel: +91-988-0788-559
E-mail: Bhaskar.k@atop.in

ATOP AMERICAs OFFICE:
Venke Char
Sr. Vice President & Head of Business
11811 North Tatum Blvd, Suite 3031
Phoenix, AZ 85028,
United States
Tel: +1-602-953-7669
E-mail: venke@atop.in