



Atop Technologies, Inc.

SE59XX-SDK Family

Node-RED user guide

User Manual

V1.0

March 9th, 2018

This PDF Document contains internal hyperlinks for ease of navigation.
For example, click on any item listed in the [Table of Contents](#) to go to that page.

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.atponline.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

| | |
|----------------------------|--|
| Author: | Carlos Hsu, Matteo Tabarelli |
| Revision: | 1.0 |
| Revision History: | Draft |
| Creation Date: | 13 March 2018 |
| Last Revision Date: | 13 March 2018 |
| Product Reference: | SE5901 (SDK-NodeRED), SE5901B (SDK-NodeRED), SE5904D (SDK-NodeRED), SE5908 (SDK-NodeRED), SE5916 (SDK-NodeRED), SE5900A (SDK-NodeRED), SE5908A (SDK-NodeRED), SE5901A (SDK-NodeRED) |
| Document Status: | Released |

Table of Contents

| | | |
|-------|---|----|
| 1 | Preface | 8 |
| 1.1 | Purpose of the Manual | 8 |
| 1.2 | Notice | 8 |
| 1.3 | Who Should Use This User Manual | 8 |
| 1.4 | Supported Platform | 8 |
| 1.5 | Warranty Period..... | 8 |
| 2 | Introduction to Atop SDK with Node-RED | 9 |
| 2.1 | Overview of SE59XX product line..... | 9 |
| 2.2 | Node-RED | 10 |
| 3 | Hardware Specifications..... | 11 |
| 3.1 | Packing List | 11 |
| 3.2 | Optional Accessories..... | 11 |
| 3.3 | Hardware..... | 13 |
| 3.4 | External Device's Overview..... | 15 |
| 3.5 | Serial Pin Assignments..... | 19 |
| 3.5.1 | SE5901 Pin Assignments for Serial Interfaces | 19 |
| 3.5.2 | SE5904D Pin Assignments | 20 |
| 3.5.3 | SE5901B Pin Assignments..... | 21 |
| 3.5.4 | SE5908A/ SE5916A Pin Assignments | 22 |
| 4 | Setting up Node-RED | 23 |
| 4.1 | Firmware upgrade..... | 23 |
| 4.1.1 | Use boot-loader update via console port | 23 |
| 4.1.2 | Use Device Manager or Device Management Utility..... | 25 |
| 4.2 | Verify current firmware version..... | 27 |
| 4.3 | Login or Remote Login to the device | 28 |
| 4.3.1 | Factory default settings | 28 |
| 4.3.2 | Remote Login | 28 |
| 4.3.3 | Use a debug command line to Login | 28 |
| 4.3.4 | Check and Modify the network settings..... | 28 |
| 4.4 | Install Node-RED on SD card or USB pen-drive | 29 |
| 4.4.1 | Login to SE59XX | 29 |
| 4.4.2 | Show SD card/ USB pen-drive partitions to discover the device name..... | 30 |
| 4.4.3 | Delete existing partitions on the SD card/ USB pen-drive..... | 31 |
| 4.4.4 | Create a new partition | 31 |
| 4.4.5 | Prepare to upload Node-RED | 32 |
| 4.4.6 | Prepare the Node-RED installation script..... | 33 |
| 5 | Using Node-RED..... | 38 |
| 5.1 | Accessing Node-RED flow-editor and dashboard..... | 38 |
| 5.1.1 | Login to Node-RED | 40 |
| 5.2 | Node-RED overview..... | 41 |
| 5.2.1 | Node-RED flow | 41 |
| 5.2.2 | Node palette selector | 42 |
| 5.2.3 | Flow Workspace..... | 43 |

| | |
|--|-----------|
| 5.2.4 Information Panel..... | 46 |
| 5.2.5 User Menu | 47 |
| 6 Using Node-RED..... | 48 |
| 6.1 Create a new flow..... | 48 |
| 6.2 Node-RED flow example | 49 |
| 6.3 Dashboard-specific settings | 54 |
| 6.3.1 Dashboard: layout settings..... | 55 |
| 6.3.2 Dashboard: theme settings..... | 56 |
| 6.3.3 Dashboard: site settings..... | 57 |
| 6.4 Dashboard- user inputs..... | 59 |
| 6.5 Accessing and controlling ATOP SE59XX Hardware with Node-RED | 60 |
| 6.5.1 Configure Serial Port mode..... | 60 |
| 6.5.2 Read and Write data to Serial Ports | 61 |
| 6.5.3 Modbus TCP/RTU/ASCII..... | 64 |
| 6.5.4 Read data to Serial Ports using Modbus RTU/ASCII..... | 65 |
| 6.5.5 Read data from Ethernet ports using Modbus TCP | 67 |
| 6.5.6 Write data using Modbus TCP/RTU/ASCII | 68 |
| 6.5.7 Acting as a passive Modbus TCP/RTU/ASCII Slave/Server..... | 70 |
| 6.5.8 Access other interfaces..... | 70 |
| 6.5.8.1Buzzer..... | 70 |
| 6.5.8.2Turn the LEDs on or off | 70 |
| 6.5.8.3Digital Inputs..... | 71 |
| 6.5.8.4Digital Outputs..... | 71 |
| 7 Global Nodes list | 72 |
| 8 Appendix..... | 90 |

Table of Figures

| | |
|---|----|
| Figure 2.1 Architecture of SE59XX SDK..... | 9 |
| Figure 3.1 DB9 Pin Number..... | 19 |
| Figure 3.2 TB5 Pin Number..... | 19 |
| Figure 3.3 DB9 Pin Number..... | 20 |
| Figure 3.4 Terminal Block (TB-5) Pin Number | 20 |
| Figure 3.5 DB9 Pin Number..... | 21 |
| Figure 3.6 2 x 7-pin Male Terminal Block..... | 21 |
| Figure 3.7 DB9 Pin Number..... | 22 |
| Figure 3.8 Terminal Block (TB-5) Pin Number | 22 |
| Figure 4.1 Console firmware update- connections..... | 23 |
| Figure 4.2 COM port Parameters for Console Firmware update..... | 24 |
| Figure 4.3 TFPD32 appearance after execution | 24 |
| Figure 4.4 SE5904D Boot loader menu..... | 24 |
| Figure 4.5 LAN Settings..... | 25 |
| Figure 4.6 LAN1 settings..... | 25 |
| Figure 4.7 TFTP download menu | 25 |

| | |
|--|----|
| Figure 4.8 SE59XX connection scheme (example on SE5904D) | 26 |
| Figure 4.9 Firmware update prompt..... | 26 |
| Figure 4.10 Firmware selection..... | 27 |
| Figure 4.11 Firmware version - Console | 27 |
| Figure 4.12 Firmware version in Device Management Utility (English)..... | 27 |
| Figure 4.13 Command line Login..... | 28 |
| Figure 4.14 Content of node-red_files_YYYYMMDD.tar.bz2..... | 33 |
| Figure 4.15 Example of Tftpd64 configuration for uploading files..... | 33 |
| Figure 4.16 Node-RED Auto_Node_Red.sh installation script result | 37 |
| Figure 5.1 Node-RED login page | 38 |
| Figure 5.2 Node-RED flow example | 40 |
| Figure 5.3 Node-RED dashboard example | 40 |
| Figure 5.4 Node-RED flow window | 41 |
| Figure 5.5 Node-RED nodes categories..... | 42 |
| Figure 5.6 Node-RED Flow workspace | 43 |
| Figure 5.7 Node information panel (example on "switch" function) | 44 |
| Figure 5.8 Node configuration panel (example on "Modbus-Read" Node)..... | 45 |
| Figure 5.9 Information panel for "Switch" node..... | 46 |
| Figure 5.10 Dashboard configuration | 46 |
| Figure 5.11 User Menu..... | 47 |
| Figure 5.12 Drop-down menu | 47 |
| Figure 6.1 New Node-RED flow created | 48 |
| Figure 6.2 Flow options..... | 49 |
| Figure 6.3 Node-RED flow example | 50 |
| Figure 6.4 Inject node properties | 52 |
| Figure 6.5 Function node properties | 52 |
| Figure 6.6 Flow options..... | 53 |
| Figure 6.7 Sample Flow dashboard | 54 |
| Figure 6.8 Dashboard settings in Node-RED flow editor | 54 |
| Figure 6.9 Dashboard Layout Settings | 55 |
| Figure 6.10 Dashboard Theme settings | 55 |
| Figure 6.11 Dashboard site settings | 55 |
| Figure 6.12 Dashboard Groups and tabs Flow example | 56 |
| Figure 6.13 Dashboard Groups and tabs Dashboard result on Tab 1..... | 56 |
| Figure 6.14 Dashboard Groups and tabs Dashboard result on Tab 2..... | 56 |
| Figure 6.15 Dashboard Dark Theme settings | 57 |
| Figure 6.16 Dashboard showing Tab 2 with customized title and hidden title bar | 57 |
| Figure 6.17 Dashboard showing all different available user inputs | 59 |
| Figure 6.18 Exec Node | 60 |
| Figure 6.19 Exec Node Configuration Parameters | 60 |
| Figure 6.20 Serial Read/Write Nodes..... | 61 |
| Figure 6.21 Serial Read Node options..... | 62 |
| Figure 6.22 Serial Read Node port configuration options | 62 |
| Figure 6.23 Serial Write Node options | 63 |
| Figure 6.24 Serial Write Node port configuration options | 64 |
| Figure 6.25 Main Modbus Nodes..... | 65 |
| Figure 6.26 Modbus Serial Read example flow | 65 |
| Figure 6.27 Modbus Read Node Settings | 66 |
| Figure 6.28 Modbus Settings – RTU/TCP/ASCII etc..... | 67 |
| Figure 6.29 Modbus RTU/ASCII Read Node Settings | 67 |
| Figure 6.30 Modbus TCP Read Node Settings | 68 |

| | |
|---|----|
| Figure 6.31 Modbus TCP/RTU/ASCII Write Node Settings..... | 69 |
| Figure 6.32 Modbus TCP Server Settings..... | 70 |

List of Tables

| | |
|--|----|
| Table 3.1 Packing List..... | 11 |
| Table 3.2 Optional Accessories | 11 |
| Table 3.3 Hardware features | 13 |
| Table 3.4 SE5901 Pin Assignment for DB9 to RS-232/RS-422/RS-485 Connector | 19 |
| Table 3.5 SE5901 Pin Assignment for TB5 to RS-232/RS-422/RS-485 Connector..... | 19 |
| Table 3.6 MB5904D Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors | 20 |
| Table 3.7 MB5904D Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors | 20 |
| Table 3.8 SE5901B Pin Assignment for DB9 to RS-232/RS-485 Connector..... | 21 |
| 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..... | 21 |
| Table 3.10 SE5908A/16A Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors | 22 |
| Table 3.11 SE5908A/16A Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors | 22 |
| Table 5.1 SE5908A/16A Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors..... | 42 |
| Table 6.1 SE59XX Programming commands per device node | 60 |
| Table 6.2 SE59XX ioctl command of COM Port | 61 |
| Table 6.3 SE59XX device node..... | 61 |
| Table 6.4 Sample program for Buzzer | 70 |
| Table 6.5 Sample program for LEDs..... | 70 |
| Table 6.6 Sample program for LCM | 71 |
| Table 6.7 Sample program for Reset Button..... | 71 |
| Table 6.8 Sample program for Digital Input..... | 71 |
| Table 6.9 Sample program for Digital Output | 71 |

1 Preface

1.1 Purpose of the Manual

This manual supports you in understanding how to use NodeRED add-on on ATOP's SE59XX Series and should be a reference guide for application development on this platform.

1.2 Notice

- (a) Node-Red is a open-source freeware for IoT developments, designed in cooperation with IBM. Node-RED is a browser-based logical flow building-block editor, and it embeds a web-based dashboard
- (b) Node-Red requires a large amount of storage memory that is not available on SE59XX embedded computer. In order to run it, it should be plugged in to the device as a pre-loaded SD card or USB storage pen drive.
- (c) If you purchased the pre-loaded SD card or USB pen drive from ATOP, there is no configuration necessary: it's enough for you to plug in the SD card or the USB pen drive in SE59XX hardware before powering on. Otherwise, if you are downloading the library from ATOP's website and loading it in the USB pen drive or SD card yourself, you need create data & swap partitions in SD card or USB storage first.
- (d) All the details that require users to input or modify are highlighted in this document.

1.3 Who Should Use This User Manual

This manual is intended to be used by qualified programmers, network personnel, support technicians or from hands-on people that are familiar with Javascript. Familiarity with network operations and Javascript Language programming may be necessary. 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.

1.4 Supported Platform

This manual is designed for the SE5901, SE5901B, SE5904D, SE5908, SE5916, SE5900A, SE5908A, and SE5916A Industrial Edge computers and for these models only.

1.5 Warranty Period

ATOP provides a **5-year limited warranty** for SE59XX Series.

2 Introduction to Atop SDK with Node-RED

2.1 Overview of SE59XX product line

ATOP's SE59XX Embedded computers are industrial grade, wide temperature embedded computers running Linux. All devices are powered by a powerful 800MHz or 1000MHz ARM Cortex A8 Texas Instruments Sitara AM3354 or AM3352 CPU. The Embedded Linux operating System, properly customized to better fit inside ATOP's rugged hardware is already running on the device and is the backbone on which the Node-RED application is running.

Figure 2.1 shows the whole architecture of SE59XX SDK. The device can also be used as a C-programmable embedded computer, but this is not the scope of this user manual. For more information, please refer to SE59XX SDK user manual. 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 Node-RED /SNMP / Apache / SQLite

In Kernel Layer, Linux 4.5 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.

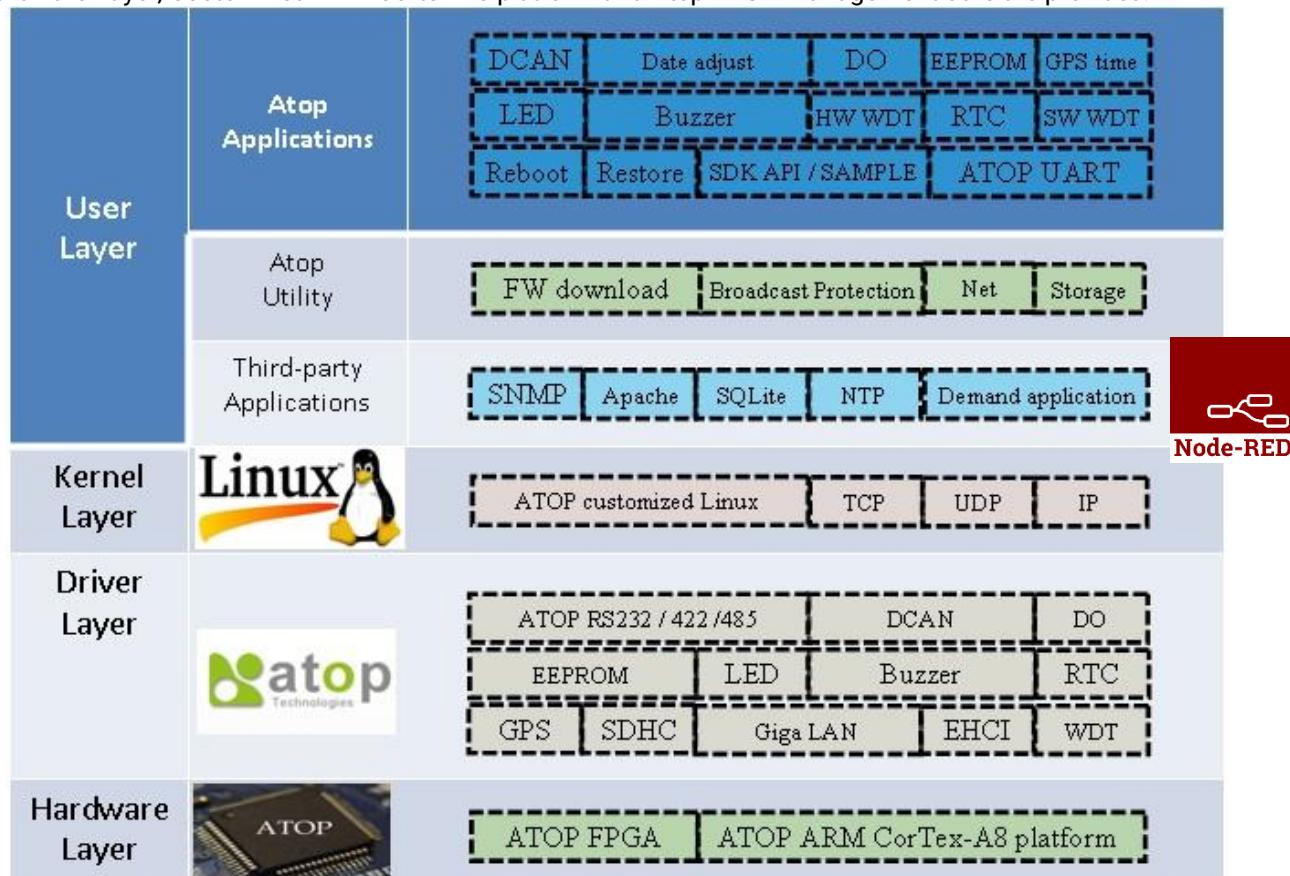
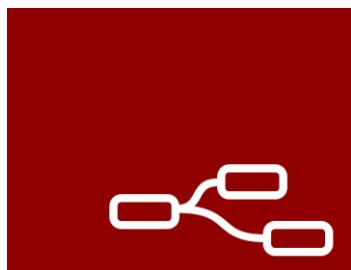


Figure 2.1 Architecture of SE59XX SDK

2.2 *Node-RED*



Node-RED is a freeware, open source building block programming tool, developed by IBM. It requires a large amount of storage memory that is not available on SE59XX embedded computer. In order to run it, it should be plugged in to the device as a pre-loaded SD card or USB storage pen drive. Once the device has been turned on, please move directly to Section 5 below.

If you purchased the pre-loaded SD card or USB pen drive from ATOP, there is no configuration necessary: it's enough for you to plug in the SD card or the USB pen drive in SE59XX hardware before powering on. Otherwise, if you are downloading the library from ATOP's website and loading it in the USB pen drive or SD card yourself, you need to create data & swap partitions in SD card or USB storage first. Please see chapter 4.4 below for setting up partitions and installing Node-RED properly.

3 Hardware Specifications

3.1 Packing List

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

Table 3.1 Packing List

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

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

| Item | Description |
|---------------------|--|
| UN315-1212(US-LDC) | Y-Type (5.08mm) power adapter, 100-240VAC input, 1.25A @ 12VDC output, US plug |
| UNE315-1212(EU-LDC) | Y-Type (5.08mm) power adapter, 100-240VAC input, 1.25A @ 12VDC output, EU plug |
| ADP-DB9(F)-TB5 | Female DB9 to Female 3.81 TB5 Converter |
| CBL-RJ45(8P)-DB9(F) | 8-pin RJ45-DB9 debug cable, 90cm |
| GDC-120 | 120mm copper woven grounding cable |
| LM28-C3S-TI-N | SFP Transceiver, 1250Mbps, 850nmVCSEL, Multi-mode, 550m, 3.3V, -20~85°C |
| LM38-C3S-TI-N | SFP Transceiver, 1250Mbps, 1310nmFP, Multi-mode, 2km, 3.3V, -40~85°C |
| LS38-C3S-TI-N | SFP Transceiver, 1250Mbps, 1310nmFP, Single-mode, 10km, 3.3V, -40~85°C |

| | |
|---------------|---|
| LS38-C3L-TI-N | SFP Transceiver, 1250Mbps, 1310nmDFB, Single-mode, 30km, 3.3V, -40~85°C |
| WMK-450-Black | Black Aluminum Wall Mount Kit (DIN-rail items only) |

3.3 Hardware

Table 3.3 Hardware features

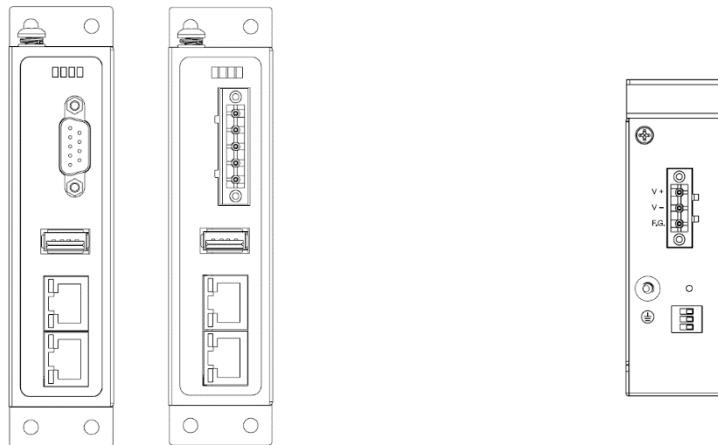
| System | |
|---------------------------|---|
| CPU | 32-bit ARM Based TI CPU AM3354 800MHz (except SE5908A/SE5916A use AM3352 1GHz) |
| Flash Memory | 32MB |
| RAM | SE5901 DDR2 256MB (NodeRED version only) SE5901B DDR2 256MB SE5904D DDR3 256MB SE5900A/08A/16A/MB5908/16 DDR3 256MB |
| EEPROM | 8 KB |
| Reset | Built-in Recessed Key (Restore to Factory Defaults) |
| Watchdog | Hardware built-in |
| Network | |
| Ethernet Interface | IEEE 802.3 10BaseT IEEE 802.3u 100BaseT(X) IEEE 802.3ac 1000BaseT(X) – SFP version of SE5904D only IEEE 802.3af (PoE PD) – selected SE5901 and SE5904D versions can be powered through PoE Connection: SFP or RJ45 |
| Serial | |
| Serial Interface | RS-232/RS-422/RS-485 Software Selectable (Default: RS-232) <ul style="list-style-type: none"> • The first port available on SE5901B is RS-232/RS-485 • The second port available on SE5901B-IO-X is only RS-232 • The isolation version (-SiS) on SE5908/SE5916/SE5908A/SE5916A supports only RS-422/ RS-485 |
| Serial Connector | Connector Type <ul style="list-style-type: none"> • SE5916 -16 Serial Ports (RJ45) • SE5908 - 8 Serial Ports (RJ45) • SE5916A – 16 Serial Ports (TB-5 or DB-9) • SE5908A – 8 Serial Ports (TB-5 or DB-9) • SE5904 – 4 Serial Ports (TB-5 or DB-9) • SE5901 – 1 Serial Port (TB-5 or DB-9) • SE5901B – 1 Serial Port (TB-14 or DB-9) – includes I/O |
| Protection | SE5901/SE5901B no isolation SE5904D/ SE5908A/16A (optional 3V) SE5908/16 (optional 2.5kV) |
| Serial Port Communication | Baud-rate: 1200 bps ~ 921600 bps Parity: None, Even, Odd, Mark, or Space Data Bits: 5, 6, 7, 8 Stop Bits: 1, 2 Software Selectable Flow Control: RTS/CTS (RS-232 only), XON/XOFF, None |
| LED Indicator | |

| | |
|------------------------------------|---|
| LED indication | <p>Power x 2 (SE5901- SE5901B – SE5908 – SE5916 x 1)</p> <p>RUN x 1</p> <p>ALARM x 1</p> <p>LAN:</p> <ul style="list-style-type: none"> • x 2 (all versions except SE5908A and SE5916A) • x 6 (SE5908A and SE5916A only) <p>COM port:</p> <ul style="list-style-type: none"> • x 16 (SE5916 and SE5916A); • x 8 (SE5908 and SE5908A); • x 4 (SE5904D); • x 1 (SE5901 and SE5901B) |
| Power Requirement & EMC | |
| Input | <p>SE5908/ SE5916 :</p> <ul style="list-style-type: none"> • Single 100~240 VAC (EU/US versions) • Single 24~48 VDC (DC version) <p>SE5908A/ SE5916A</p> <ul style="list-style-type: none"> • Redundant 100~240 VAC or 100~370 VDC (TB) – HV vers. • Redundant 24~48 VDC- DC version <p>SE5901/SE5901B : Single 9~48 VDC</p> <p>SE5904D : Redundant 9~48 VDC</p> |
| Consumption | <p>Max.17.5 W (SE5908 /SE5916)</p> <p>Max. 6W (SE5901)</p> <p>Max. 7.8W(SE5904D)</p> <p>Max. 17.5W(SE5908A/SE5916A)</p> <p>Max. 7.2W(SE5901B)</p> |
| EMI/EMC | <p>FCC Part 15, Subpart B, Class A</p> <p>EN 55032, Class B, EN 61000-6-2, Class B</p> <p>EN 61000-3-2, EN 61000-3-3</p> <p>EN 55024, EN 61000-6-4</p> <p>IEC 61850-3 / IEEE 1613 (SE5908A and SE5916A only)</p> |
| Mechanical | |
| Dimensions (W x H x D,mm) | <p>SE5901: 32 mm x 110 mm x 90 mm (1.26 x 4.33 x 3.54 in)</p> <p>SE5901B: 32 mm x 122mm x 92 mm (1.26 x 4.8 x 3.62 in)</p> <p>SE5904D: 55 mm x 145 mm x 113mm (2.17 x 5.17 x 4.45 in)</p> <p>SE5908: 436 mm x 43.5 mm x 200 mm (17.17 x 1.71 x 7.87 in)</p> <p>SE5916: 436 mm x 43.5 mm x 200 mm (17.17 x 1.71 x 7.87 in)</p> <p>SE5908A: 440.6mm x 44 mm x 309 mm (17.35 x 1.73 x 12.17 in)</p> <p>SE5916A: 440.6mm x 44 mm x 309 mm (17.35 x 1.73 x 12.17 in)</p> |
| Enclosure | IP30 protection, metal housing |
| Environmental | |
| Temperature | Operations -40°C ~ 85°C (-40°F ~ 185°F) (except SE5901B -40°C ~ 70°C and SE5908/SE5916 -20°C ~ 70°C) |
| | Storage -40°C ~ 85°C (-40°F ~ 185°F) |
| Relative Humidity | 5% ~ 95%, 55°C Non-condensing |

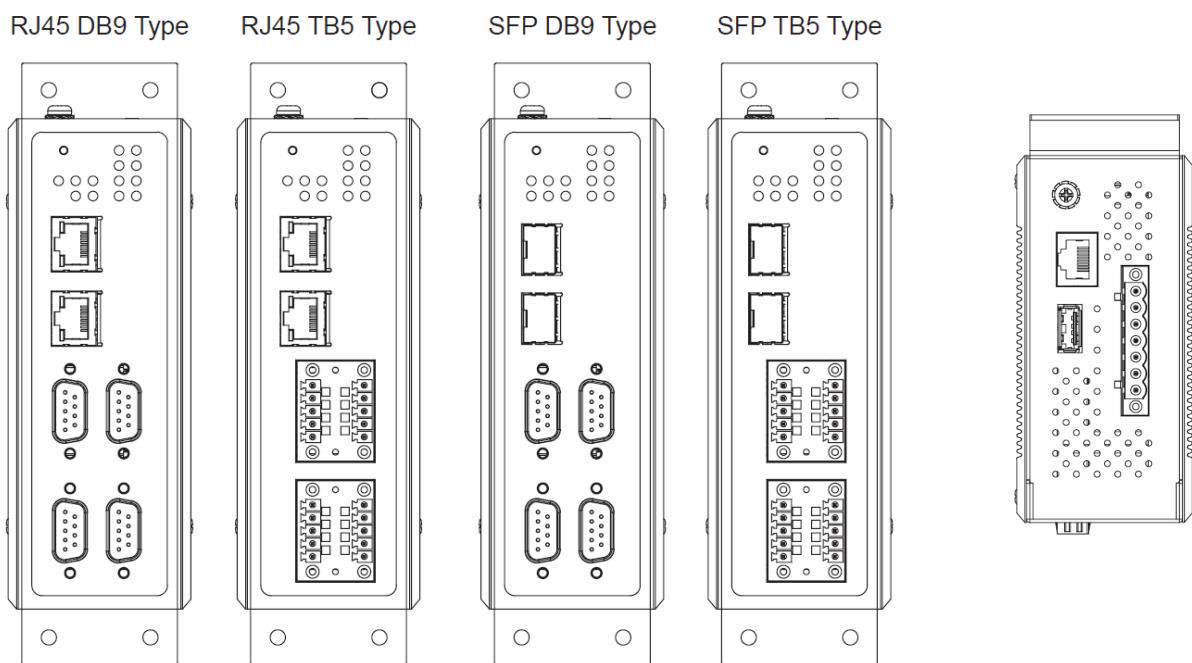
3.4 External Device's Overview

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

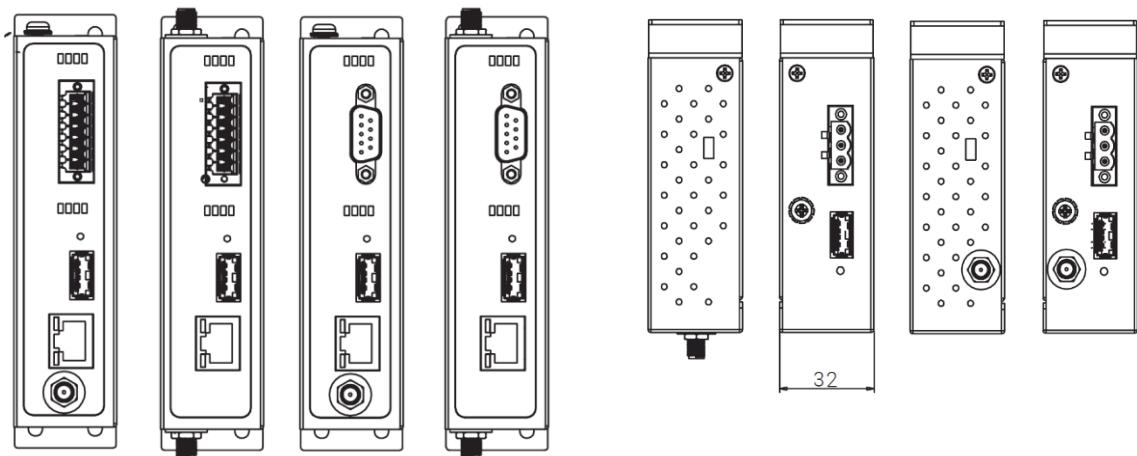
SE5901



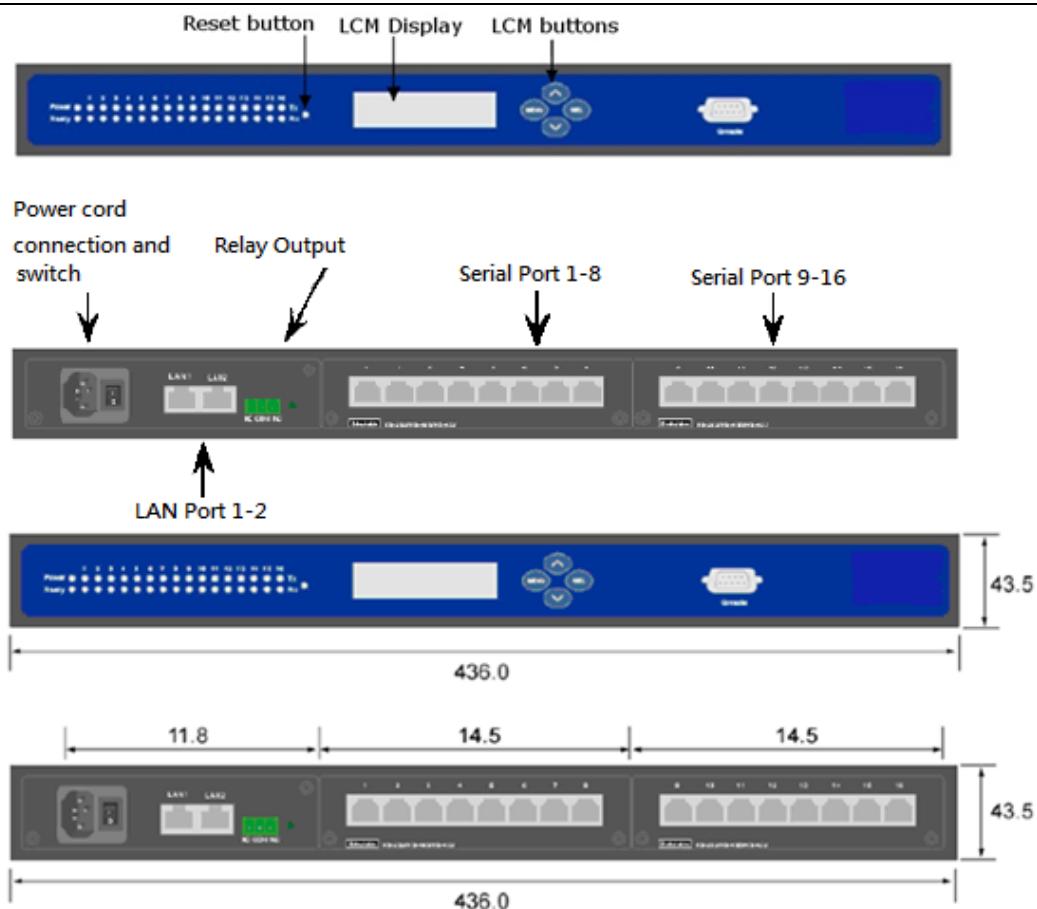
SE5904D



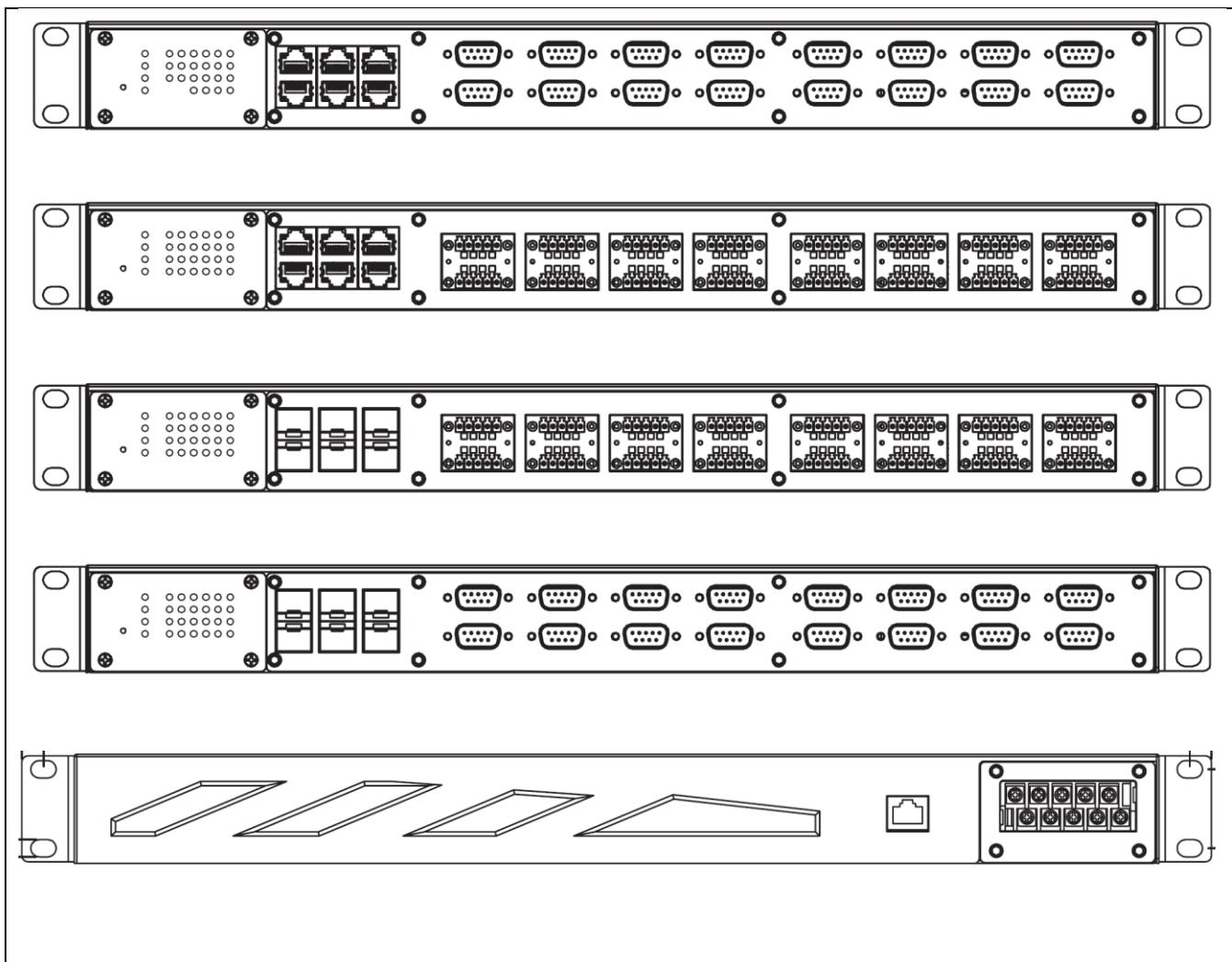
SE5901B



SE5908/16



SE5908A/16A



3.5 Serial Pin Assignments**3.5.1 SE5901 Pin Assignments for Serial Interfaces**

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

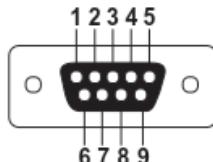


Figure 3.1 DB9 Pin Number

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

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

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

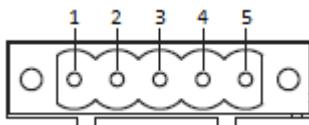


Figure 3.2 TB5 Pin Number

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

| Pin# | RS-232 Full Duplex | RS-422/4-Wire RS-485 Full Duplex | 2-Wire RS-485 Half Duplex |
|------|-----------------------|-------------------------------------|------------------------------|
| 1 | RxD | T+ | NC |
| 2 | CTS | T- | NC |
| 3 | TxD | R+ | Data+ |
| 4 | RTS | R- | Data- |
| 5 | SG (Signal Ground) | SG (Signal Ground) | SG (Signal Ground) |

3.5.2 SE5904D Pin Assignments

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

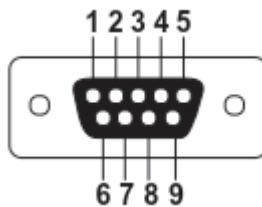


Figure 3.3 DB9 Pin Number

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

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

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

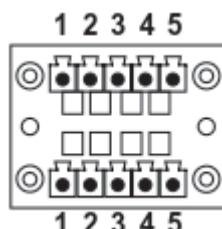


Figure 3.4 Terminal Block (TB-5) Pin Number

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

| Pin# | RS-232 | RS-422 4-Wire RS-485 | 2-W RS-485 |
|------|--------------------|-------------------------|--------------------|
| 1 | RxD | TxD+ | Data+ |
| 2 | CTS | TxD- | Data- |
| 3 | TxD | RxD+ | N/A |
| 4 | RTS | RxD- | N/A |
| 5 | SG (Signal Ground) | SG (Signal Ground) | SG (Signal Ground) |

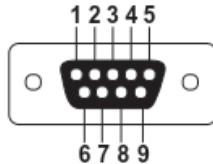
3.5.3 SE5901B Pin Assignments**DB9 to RS-232/RS-485/RS-422 connectors**

Figure 3.5 DB9 Pin Number

Table 3.8 SE5901B Pin Assignment for DB9 to RS-232/RS-485 Connector

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

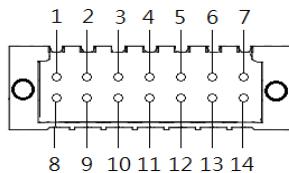
2 x 7-pin Male Terminal Block for RS-232/485(COM 1),RS-232(COM 2) Relay and DI

Figure 3.6 2 x 7-pin Male Terminal Block

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

| Pin# | DI and Relay | COM1 (RS-232) | COM1 (RS-485) | COM2 (RS-232) |
|------|-------------------|---------------------|---------------------|---------------------|
| 1 | DI1 | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 2 | DI2 | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 3 | Relay 1 - | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 4 | Relay 1 + | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 5 | Relay 2 - | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 6 | Relay 2 + | Dedicated for DI/DO | Dedicated for DI/DO | Dedicated for DI/DO |
| 7 | Dedicated for COM | SG (Signal Ground) | SG (Signal Ground) | - |
| 8 | Dedicated for COM | Rx | - | - |
| 9 | Dedicated for COM | CTS | - | - |
| 10 | Dedicated for COM | Tx | Data + | - |
| 11 | Dedicated for COM | RTS | Data - | - |
| 12 | Dedicated for COM | - | - | SG (Signal Ground) |
| 13 | Dedicated for COM | - | - | Rx |
| 14 | Dedicated for COM | - | - | Tx |

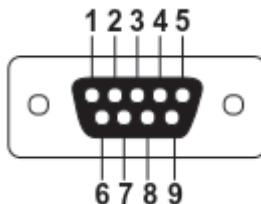
3.5.4 SE5908A/ SE5916A Pin Assignments**DB9 to RS-232/RS-485/RS-422 connectors**

Figure 3.7 DB9 Pin Number

Table 3.10 SE5908A/16A Pin Assignment for DB9 to RS-232/RS422/RS-485 Connectors

| Pin# | RS-232 | RS-422 | RS-485 |
|-------------|--------------------|--------------------|--------------------|
| 1 | - | - | - |
| 2 | RxD | TxD+ | Data+ |
| 3 | TxD | RxD+ | - |
| 4 | - | - | - |
| 5 | SG (Signal Ground) | SG (Signal Ground) | SG (Signal Ground) |
| 6 | - | - | - |
| 7 | RTS | RxD- | - |
| 8 | CTS | TxD- | Data- |
| 9 | - | - | - |

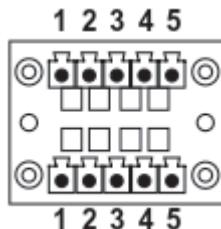
5-Pin Terminal Block to RS-232/RS-485/RS-422 connectors

Figure 3.8 Terminal Block (TB-5) Pin Number

Table 3.11 SE5908A/16A Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors

| Pin# | RS-232 | RS-422 4-Wire RS-485 | 2-W RS-485 |
|-------------|--------------------|---------------------------------|--------------------|
| 1 | RxD | TxD+ | Data + |
| 2 | CTS | TxD- | Data - |
| 3 | TxD | RxD+ | - |
| 4 | RTS | RxD- | - |
| 5 | SG (Signal Ground) | SG (Signal Ground) | SG (Signal Ground) |

4 Setting up Node-RED

4.1 Firmware upgrade

It may be necessary to upgrade firmware from time to time. There are two ways to upgrade the firmware on the SE59XX platform:

4.1.1 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.



Figure 4.1 Console firmware update- connections

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

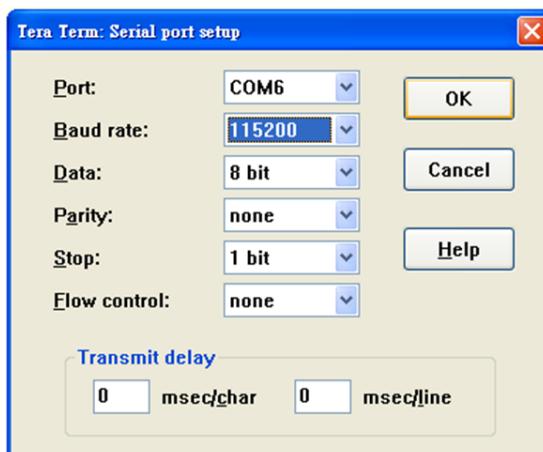


Figure 4.2 COM port Parameters for Console Firmware update

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.

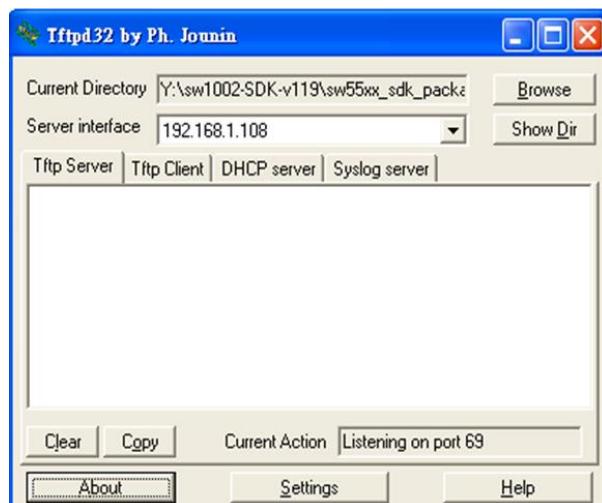


Figure 4.3 TFPD32 appearance after execution

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 4.4.

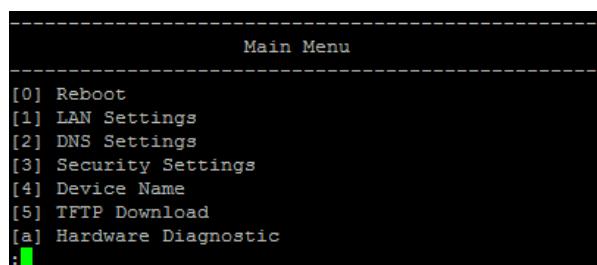


Figure 4.4 SE5904D Boot loader menu

Select item 1 to enter "LAN Setting" menu as Figure 4.5, and setup IP/Netmask/Gateway of LAN1 as Figure 4.6

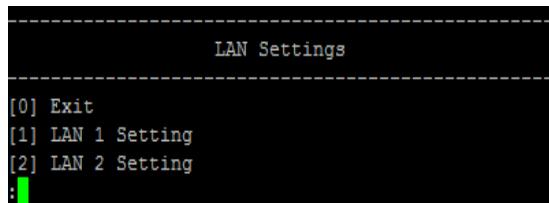


Figure 4.5 LAN Settings

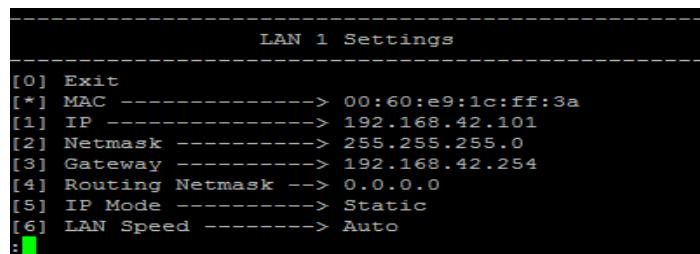


Figure 4.6 LAN1 settings

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 4.7

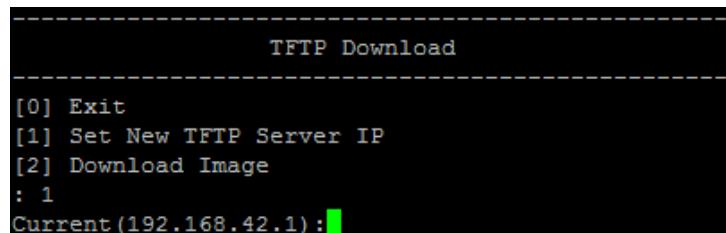


Figure 4.7 TFTP download menu

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

4.1.2 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.

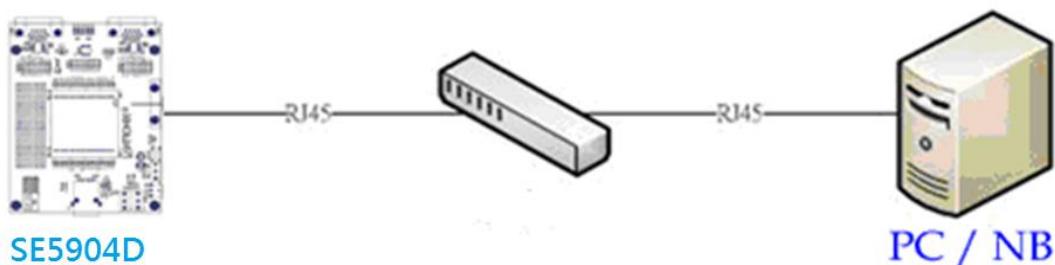


Figure 4.8 SE59XX connection scheme (example on SE5904D)

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 4.9.

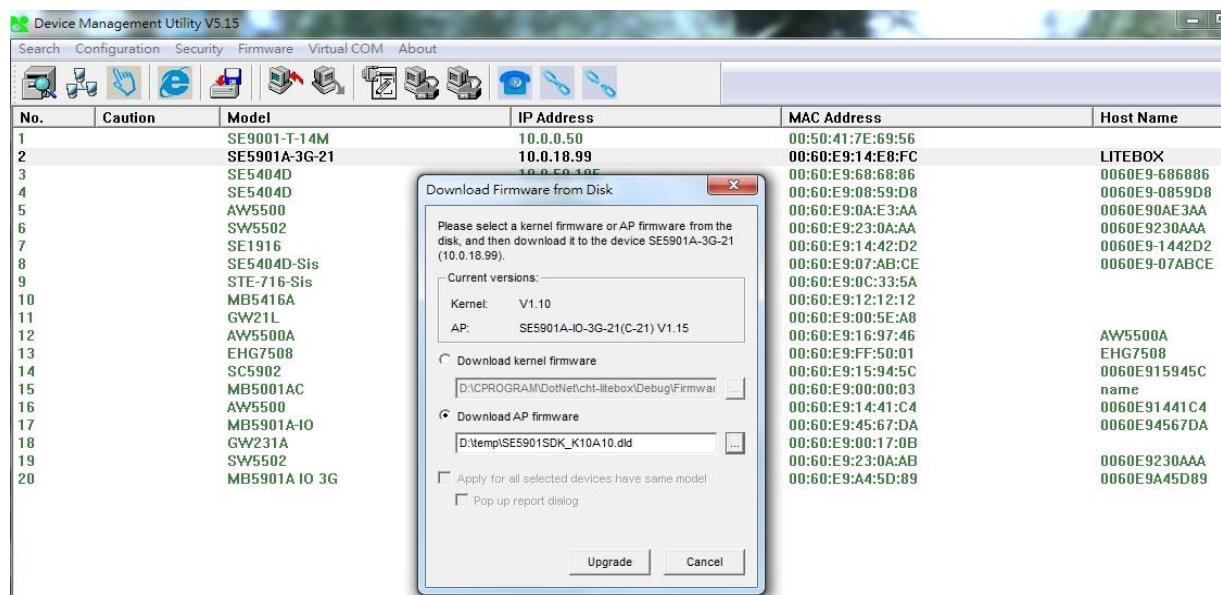


Figure 4.9 Firmware update prompt

Select the firmware (Kernel or AP) from this dialog and select the upgraded file as Figure 4.10. 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.

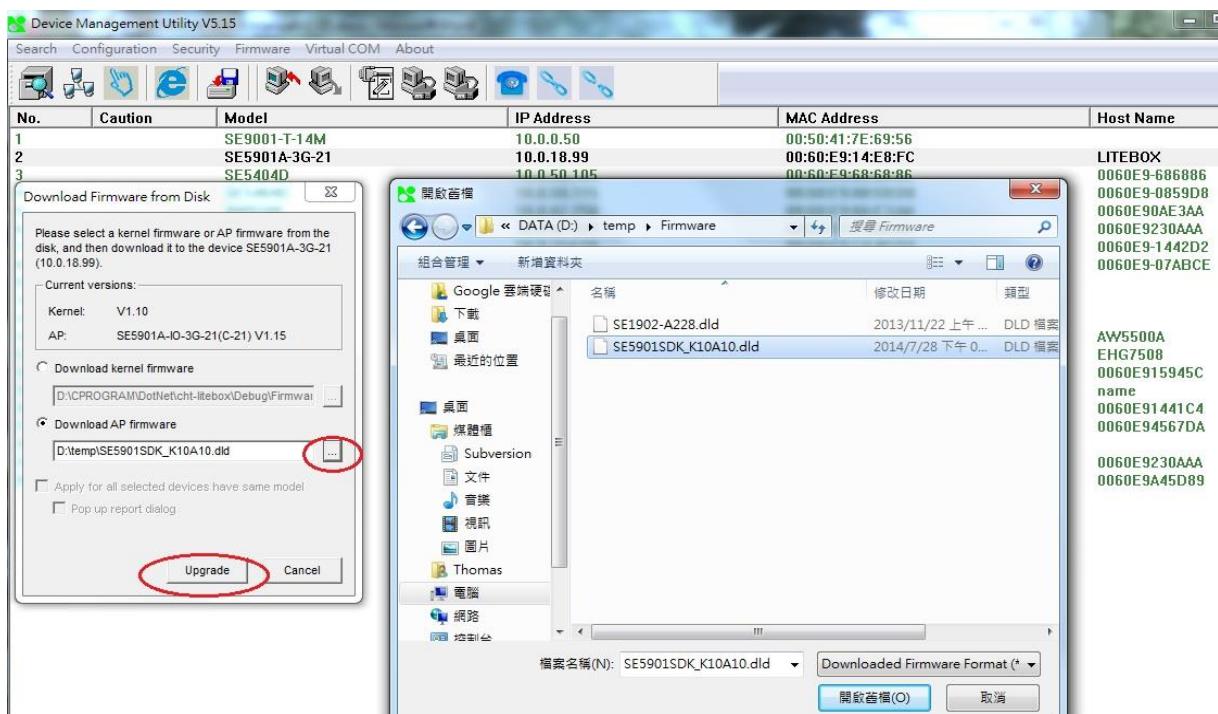


Figure 4.10 Firmware selection

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

4.2 Verify current firmware version

There are two methods to verify the firmware version:

- 1) Use a debug line as per Paragraph 4.1.1 above to connect console port of the device. After boot up, type "atop_show_ver" in the console command line to check current version as Figure 4.11 shown. The red rectangle shows information of boot-loader (V1.00), Kernel (V1.00) and AP (V1.00) version number.

```
Welcome to ATOP system
ATOP login: admin
Password:
# atop_show_ver
Bootloader Version: 1.00
Kernel Version: 1.00
AP Version: 1.00
Core Module CPLD Version: 1.02
#
```

Figure 4.11 Firmware version - Console

- 2) Use Device Manager or Device Management Utility (SerialManager) to check version number as per Figure 4.12. (Device Manager is currently supported to Simplified Chinese release)

| No. | Caution | Model | IP Address | MAC Address | Host Name | Kernel | AP Information |
|-----|---------|---------|--------------|-------------------|--------------|--------|----------------|
| 1 | | SE5904D | 192.168.4.13 | 00:60:E9:1C:FF:3A | yyyyyyyyyyyy | V1.0 | SE5904D V1.00 |

Figure 4.12 Firmware version in Device Management Utility (English)

4.3 Login or Remote Login to the device

4.3.1 Factory default settings

- IP address: 10.0.50.100
- Username: admin
- Password: default

4.3.2 Remote Login

- 1) Setup or read FTP account and password from ATOP boot-loader menu 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 putty utility.
- 4) Login account as first step shown

4.3.3 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
[ 8.985240] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

Welcome to ATOP system
ATOP login: [
```

Figure 4.13 Command line login

4.3.4 Check and Modify the network settings

If you don't need to change the default network settings, you can skip this step. In order to change IP address settings, DNS, Default Gateway and Subnetmask please follow the steps below.

- 1) Login to the device with SSH/Telnet
- 2) Use **ifconfig** command to check current settings and to modify IP address and subnetmask

```
# ifconfig          <-- show ethernet infomations

eth0    Link encap:Ethernet HWaddr 00:60:E9:1E:6C:48
        inet addr:192.168.4.33 Bcast:192.168.4.255 Mask:255.255.255.0 <- Example
        inet6 addr: fe80::260:e9ff:fe1e:6c48/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
          RX packets:391236 errors:0 dropped:3314 overruns:0 frame:0
          TX packets:85095 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:84832013 (80.9 MiB) TX bytes:5144093 (4.9 MiB)
          Interrupt:56
....
```

```
# ifconfig eth0 10.0.50.1 netmask 255.255.0.0 <- change IP & subnet of eth0 to 10.0.50.1 and 255.255.0.0
```

```
eth0    Link encap:Ethernet HWaddr 00:60:E9:1E:6C:48
        inet addr:10.0.50.100 Bcast:192.168.4.255 Mask:255.255.0.0 <- Changed
          inet6 addr: fe80::260:e9ff:fe1e:6c48/64 Scope:Link
             UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
             RX packets:391236 errors:0 dropped:3314 overruns:0 frame:0
             TX packets:85095 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:84832013 (80.9 MiB) TX bytes:5144093 (4.9 MiB)
             Interrupt:56
....
```

3) Use vi editor to modify DNS settings by changing `/etc/resolv.conf`

```
# vi /etc/resolv.conf      <- setup DNS

nameserver 8.8.8.8           <- modify your DNS IP (Google)
nameserver 168.95.1.1         <- modify your DNS IP (CHT)
nameserver 0.0.0.0
nameserver 0.0.0.0
nameserver 0.0.0.0
nameserver 0.0.0.0
nameserver 0.0.0.0
nameserver 0.0.0.0
nameserver fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
nameserver fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

4.4 *Install Node-RED on SD card or USB pen-drive*

Please follow the steps described below if you didn't purchase the pre-loaded SD-card or USB pen-drive with Node-RED installed on it and if you downloaded the Node-RED library from ATOP's website.

Note: if you purchased the pre-loaded SD-card or USB pen-drive from ATOP you can skip this step

4.4.1 *Login to SE59XX*

- Plug in the empty USB pen-drive or SD card in the SD card slot or in the USB socket. It is recommended to have more than 1GB storage available
- Turn on the device and wait until the RUN LED is blinking
- Connect the LAN cable to your computer. Please make sure the IP address of your computer is on the same subnet of the device.
- If working on Windows Operating System environment, use any tool supporting telnet protocol such as "ssh" inside of Windows (for example, putty).
- Enter SE59XX_IP_address via ssh.

- Login account as first step shown

Factory default settings

- IP address: 10.0.50.100
- Username: admin
- Password: default

4.4.2 Show SD card/ USB pen-drive partitions to discover the device name

Once the device is turned on, logged in, with the SD-card or USB plugged in properly, please input the following command and press "Enter". The example below shows what the result can be, but the actual result depends on the capacity of the drive used.

```
# fdisk -l <-- lowercase L

Disk /dev/mtdblock0: 0 MB, 524288 bytes
255 heads, 63 sectors/track, 0 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock0 doesn't contain a valid partition table

Disk /dev/mtdblock1: 0 MB, 131072 bytes
255 heads, 63 sectors/track, 0 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock1 doesn't contain a valid partition table

Disk /dev/mtdblock2: 0 MB, 393216 bytes
255 heads, 63 sectors/track, 0 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock2 doesn't contain a valid partition table

Disk /dev/mtdblock3: 5 MB, 5242880 bytes
255 heads, 63 sectors/track, 0 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock3 doesn't contain a valid partition table

Disk /dev/mtdblock4: 18 MB, 18874368 bytes
255 heads, 63 sectors/track, 2 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock4 doesn't contain a valid partition table

Disk /dev/mtdblock5: 41 MB, 41943040 bytes
255 heads, 63 sectors/track, 5 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/mtdblock5 doesn't contain a valid partition table
Disk /dev/mmcblk0: 7932 MB, 7932477440 bytes
4 heads, 16 sectors/track, 242080 cylinders
Units = cylinders of 64 * 512 = 32768 bytes
```

| Device | Boot | Start | End | Blocks | Id | System |
|----------------|------|--------|--------|---------|----|------------|
| /dev/mmcblk0p1 | | 1 | 183106 | 5859384 | 83 | Linux |
| /dev/mmcblk0p2 | | 183107 | 242080 | 1887168 | 82 | Linux swap |

4.4.3 Delete existing partitions on the SD card/ USB pen-drive

If the external storage drive has a partition, this should be deleted before proceeding.

Note: Please make sure that you have backed up all your personal data saved on the USB pen-drive or the SD-card. The following activities may result in a permanent loss on all information on the external device.

```
# fdisk /dev/ mmcblk0 <- please select your storage device name
```

The number of cylinders for this disk is set to 242080. There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): **d**

Partition number (1-4):**1** <- delete a partition 1

Command (m for help): **d**

Selected partition 2 <- delete a partition 2 (auto select)

Command (m for help): **p** <- print partition information

Disk /dev/mmcblk0: 7932 MB, 7932477440 bytes

4 heads, 16 sectors/track, 242080 cylinders

Units = cylinders of 64 * 512 = 32768 bytes

| Device | Boot | Start | End | Blocks | Id | System |
|--------|------|-------|-----|--------|----|--------|
|--------|------|-------|-----|--------|----|--------|

Command (m for help): **w** <- write table to disk and exit

4.4.4 Create a new partition

Please follow the steps below in order to prepare the partition for installing Node-RED. It is recommended to create one partition for the primary files and one partition for Swap.

```
# fdisk /dev/ mmcblk0 <- please select your storage device name
```

The number of cylinders for this disk is set to 242080. There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): **n** <----- add a partition 1 (for example, 6G)

Command action

 e extended

 p primary partition (1-4)

```
p
Partition number (1-4): 1
First cylinder (1-242080, default 1): Using default value 1 (press enter)
Last cylinder or +size or +sizeM or +sizeK (1-242080, default 242080): +6000M

Command (m for help): n <-- add a partition 2 (Extra space)
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (183107-242080, default 183107):Using default value 183107 (press enter)
Last cylinder or +size or +sizeM or +sizeK (3635-3814, default 3814):Using default value 3814
(press enter)

Command (m for help): p <- print partition information
Disk /dev/mmcblk0: 7932 MB, 7932477440 bytes
  4 heads, 16 sectors/track, 242080 cylinders
  Units = cylinders of 64 * 512 = 32768 bytes

  Device Boot      Start        End    Blocks   Id  System
  /dev/mmcblk0p1        1    183106   5859384   83  Linux
  /dev/mmcblk0p2    183107    242080   1887168   83  Linux

Command (m for help): t <- change partiton 2 to swap type
Partition number (1-4): 2
Hex code (type L to list codes): 82

Command (m for help): p <- print partition information
Disk /dev/mmcblk0: 7932 MB, 7932477440 bytes
  4 heads, 16 sectors/track, 242080 cylinders
  Units = cylinders of 64 * 512 = 32768 bytes

  Device Boot      Start        End    Blocks   Id  System
  /dev/mmcblk0p1        1    183106   5859384   83  Linux
  /dev/mmcblk0p2    183107    242080   1887168   82  Linux swap

Command (m for help): w <- write table to disk and exit
```

4.4.5 Prepare to upload Node-RED

There are several ways to load Node-RED libraries into the SD Card. In this manual, we will use FTP with the open-Source tool called "ftpd64.exe". "ftpd64.exe" is a third party application that is recommended. More information about "ftpd64.exe", its availability URL is available in Chapter 8 below

On ATOP's website there will be two different archives available (YYYY, MM, DD represent the release year, month and day respectively):

- 1) node-red_sdk_release_YYYYMMDD.tar.bz2 -> SE59XX SDK and image
- 2) node-red_files_YYYYMMDD.tar.bz2 >> Node-RED libraries

Extract the contents of the node-red_files_YYYYMMDD.tar.bz2 downloaded from ATOP's website in any folder of your liking. The contents are shown in Figure 4.14 below. The node-red files archive contains 2 more files:

- 1) Auto_Node_Red.sh >> NodeRED installation script
- 2) home_root_node-red.tar.bz2 >> includes the NodeRED blocks
- 3) node-v6.11.3-linux-armv7l.tar.bz2 >> includes Node Binaries

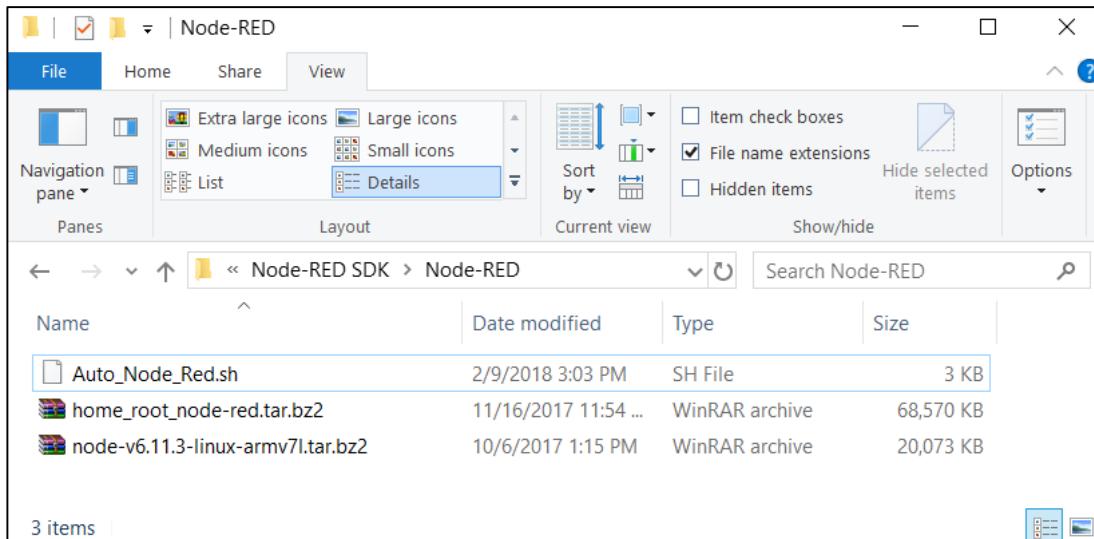


Figure 4.14 Content of node-red_files_YYYYMMDD.tar.bz2

Then, run Tftpd64.exe and select the folder in which Node-RED files have been extracted and the Network interface connected to SE59XX.

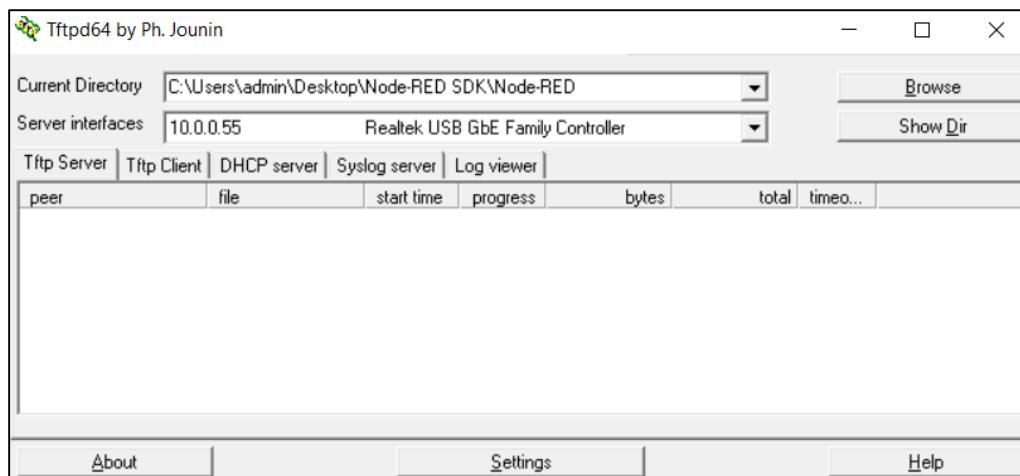


Figure 4.15 Example of Tftpd64 configuration for uploading files

4.4.6 Prepare the Node-RED installation script

Once the device is turned on, logged in, with the SD-card or USB plugged in properly, please connect to the device using SSH/telnet. In order to install Node-RED automatically, it will be necessary to have SE59XX to run an installation script that will:

- prepare the SD card/USB pen drive partition by formatting it

- copy the contents of node-red_files_YYYYMMDD.tar.bz2 archive that has been put in a folder of your choice in step 0 to the SD card/USB pen drive
- decompress those files into the storage
- install the software
- make some basic settings

An example of the installation script is `Auto_Node_Red.sh` that is located in `node-red_files_YYYYMMDD.tar.bz2` archive and can be modified on the computer before being uploaded to the device via FTP to jffs2 file system. Alternatively, you can create the file through your SSH/telnet command line directly.

When creating or modifying the file please make sure to provide the correct partition names used, as well as the correct .tar.bz2 file names as they are stored in the folder and the IP address of your computer.

```
# cd /jffs2
# vi Auto_Node_Red.sh  <-- Create a script via VI editor

#!/bin/sh

MountDir=/media
DirName=mmcblk0p1 <-- destination partition name
Ext2Name=mmcblk0p1 <-- SD card/ USB drive partition1 (ext2)
SwapName=mmcblk0p2 <-- SD card/ USB drive partition2 (swap)
NodeRedMain=node-v6.11.3-linux-armv7l <-- Node-RED binary archive (as is in the folder of 0)
UserNode=home_root_node-red <-- Nodes file name (as is in the folder of 0)
ServerIP=10.0.0.55 <-- Local IP address of the PC that will upload the files (as Figure 4.15)

# Format partition
read -p "Format partition ? (Y/N): " yn
if [ "${yn}" == "Y" ] || [ "${yn}" == "y" ]; then
    echo " Format partition ... "
    mke2fs /dev/${Ext2Name}
    mkswap /dev/${SwapName}
fi

# Mount SD card and extract node-red files
if [ -d "${MountDir}/${DirName}" ]; then
    echo " make dir already! "
else
    mkdir ${MountDir}/${DirName}
fi

if ! mount /dev/${Ext2Name} ${MountDir}/${DirName}; then
    echo " Mount Error !"
    exit 1
else
    swapon /dev/${SwapName}
fi

echo "*****"
```

```
echo "***** mount and swap done *****"

# tar Node-Red main program
cd ${MountDir}/${DirName}

if [ -d "${NodeRedMain}" ]; then
    echo " tar Node-Red main program already! "
    break;
elif [ -f "${NodeRedMain}.tar.bz2" ]; then
    echo " tar Node-Red main program ... "
    tar xf ${NodeRedMain}.tar.bz2

else
    echo " download & tar Node-Red main program ... "
    tftp -gr ${NodeRedMain}.tar.bz2 ${ServerIP};
    tar xf ${NodeRedMain}.tar.bz2
fi

echo "***** tar Node-Red main program done *****"

# tar user-defined node
if [ -d "${MountDir}/${DirName}/home" ]; then
    echo " make home dir already! "
    cd ${MountDir}/${DirName}/home
else
    mkdir ${MountDir}/${DirName}/home
    cd ${MountDir}/${DirName}/home
fi

if [ -d "root" ]; then
    echo " tar user-defined node already! "
    break;
elif [ -f "${UserNode}.tar.bz2" ]; then
    echo " tar user-defined node ... "
    tar xf ${UserNode}.tar.bz2

else
    echo " download & tar user-defined node ... "
    tftp -gr ${UserNode}.tar.bz2 ${ServerIP};
    tar xf ${UserNode}.tar.bz2
fi

echo "***** tar user-defined node done *****"
# Setup environmental parameters
export HOME=${MountDir}/${DirName}/home/root
export PATH=${MountDir}/${DirName}/${NodeRedMain}/bin:$PATH
export NODE_PATH=${MountDir}/${DirName}/${NodeRedMain}/bin

echo "***** Setup environmental parameters done *****"
```

```
# Run node-red server demo profile
cd ${MountDir}/${DirName}/home/root/.node-red/
npm config set strict-ssl false
node-red flows_ATOP.json &
echo "***** Starting install Node-Red *****"
```

After the file is saved or uploaded on /jffs2 with the correct parameters, please change the rwx rights (read-write-execute) to 777 using the following command on the command line.

```
# chmod 777 Auto_Node_Red.sh
```

Then, execute the script using the following command on the command line. If the installation script has been renamed, please use the new file name

```
# ./ Auto_Node_Red.sh  <- Execute script
```

After the script has been executed, the overall command line interface should show the record as shown in below Figure 4.16.

```
# ./Auto_Node_Red.sh
Format partition ? (Y/N): y
Format partition ...
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
366480 inodes, 1464846 blocks
73242 blocks (5%) reserved for the super user
First data block=0
Maximum filesystem blocks=4194304
45 block groups
32768 blocks per group, 32768 fragments per group
8144 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736
Setting up swap space version 1, size = 1932455936 bytes
[ 159.176023] Adding 1887164k swap on /dev/mmcblk0p2. Priority:-1 extents:1 across:1887164k SS
***** mount and swapon done *****
download & tar Node-Red main program ...
***** tar Node-Red main program done *****
download & tar user-defined node ...
***** tar user-defined node done *****
***** Setup environmental parameters done *****
***** Starting install Node-Red *****
# 24 Oct 09:22:56 - [info]

Welcome to Node-RED
=====

24 Oct 09:22:56 - [info] Node-RED version: v0.17.5
24 Oct 09:22:56 - [info] Node.js  version: v6.11.3
24 Oct 09:22:56 - [info] Linux 3.14.26-svn1196 arm LE
24 Oct 09:23:00 - [info] Loading palette nodes
24 Oct 09:23:15 - [info] Dashboard version 2.5.1 started at /ui
24 Oct 09:23:20 - [warn] -----
24 Oct 09:23:20 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
24 Oct 09:23:20 - [warn] -----
24 Oct 09:23:20 - [info] Settings file  : /media/sd/home/root/.node-red/settings.js
24 Oct 09:23:20 - [info] User directory : /media/sd/home/root/.node-red/
24 Oct 09:23:20 - [info] Flows file   : /media/sd/home/root/.node-red/flows_ATOP.json
24 Oct 09:23:20 - [info] Server now running at http://127.0.0.1:1880/
24 Oct 09:23:21 - [info] Starting flows
24 Oct 09:23:22 - [info] Started flows
24 Oct 09:23:23 - [info] [mqtt-broker:aa2acbb8.35bad8] Connected to broker: mqtt://test.mosquitto.org:1883
#
#
```

Figure 4.16 Node-RED Auto_Node_Red.sh installation script result

5 Using Node-RED

5.1 Accessing Node-RED flow-editor and dashboard

This chapter will explain you how to access Node-RED after this has been properly set-up and installed on the SD-Card or USB pen drive on SE59XX. If you have purchased ATOP's Node-RED version with preinstalled SD-card or USB Pen-drive, or if you have already followed all steps described in Section 4 above then Node-RED will start up automatically as soon as the device is powered on and Operating System is loaded.

Note: If you're installing the Node-RED library on the device by yourself, please read Section 4 above first.

Node-RED flow editor is accessible on the following path: http://DEVICE_IP_ADDRESS:1880/#

Node-RED dashboard is accessible on the following path: http://DEVICE_IP_ADDRESS:1880/ui/#/0

For example: <http://10.0.50.100:1880/#> and <http://10.0.50.100:1880/ui/#/0>

Node-RED is designed to work on all major browsers, such as Google Chrome, Mozilla Firefox, Safari (both mobile and desktop) and Internet Explorer

Note: Please make sure your computer/mobile 's IP address is in the same subnet of SE59XX.

After inputting the address above in the address bar of the browser, you'll be redirected to the Node-RED login page, as Figure 5.1. Please note that in below example the IP address has been changed to avoid conflict.

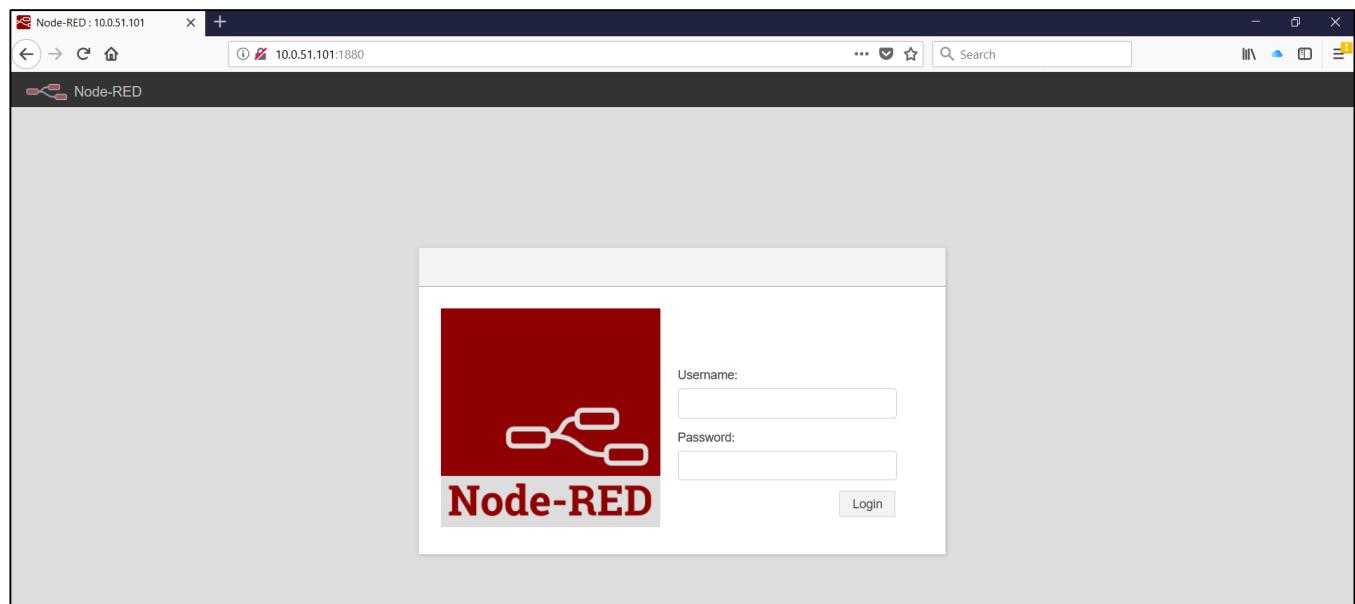


Figure 5.1 Node-RED login page

5.1.1 Login to Node-RED

Access to Node-RED flow editor and dashboard requires a user name and password. The default user name is "admin" and the default password is "password". After logging in is successful, you will be presented a sample flow generated by ATOP or, if you're logging in on the UI, with a sample UI based on ATOP's flow. This can be deleted and replaced with your own flow.

In order to change your access credentials, please check Section...

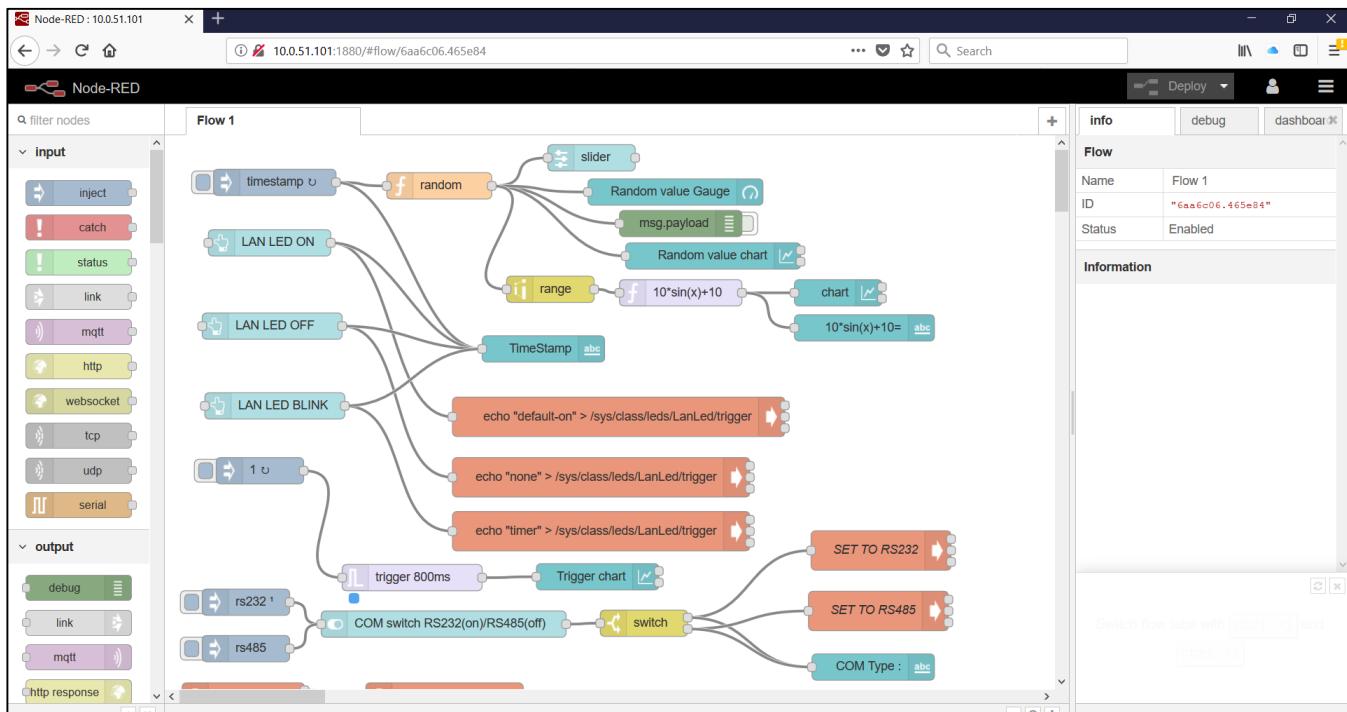


Figure 5.2 Node-RED flow example

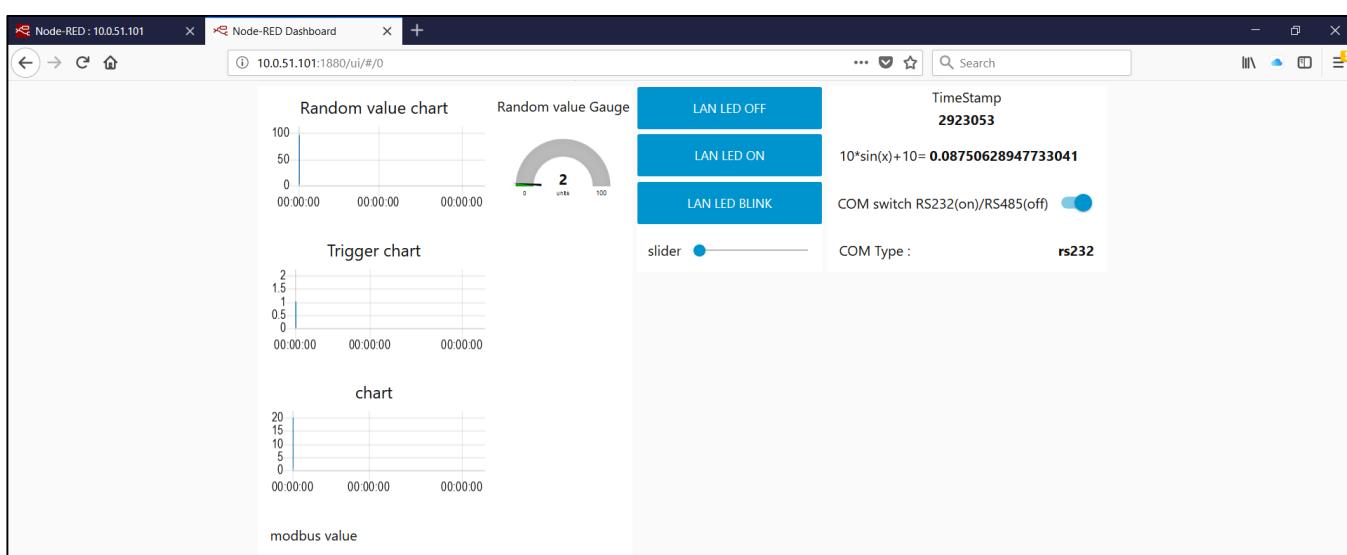


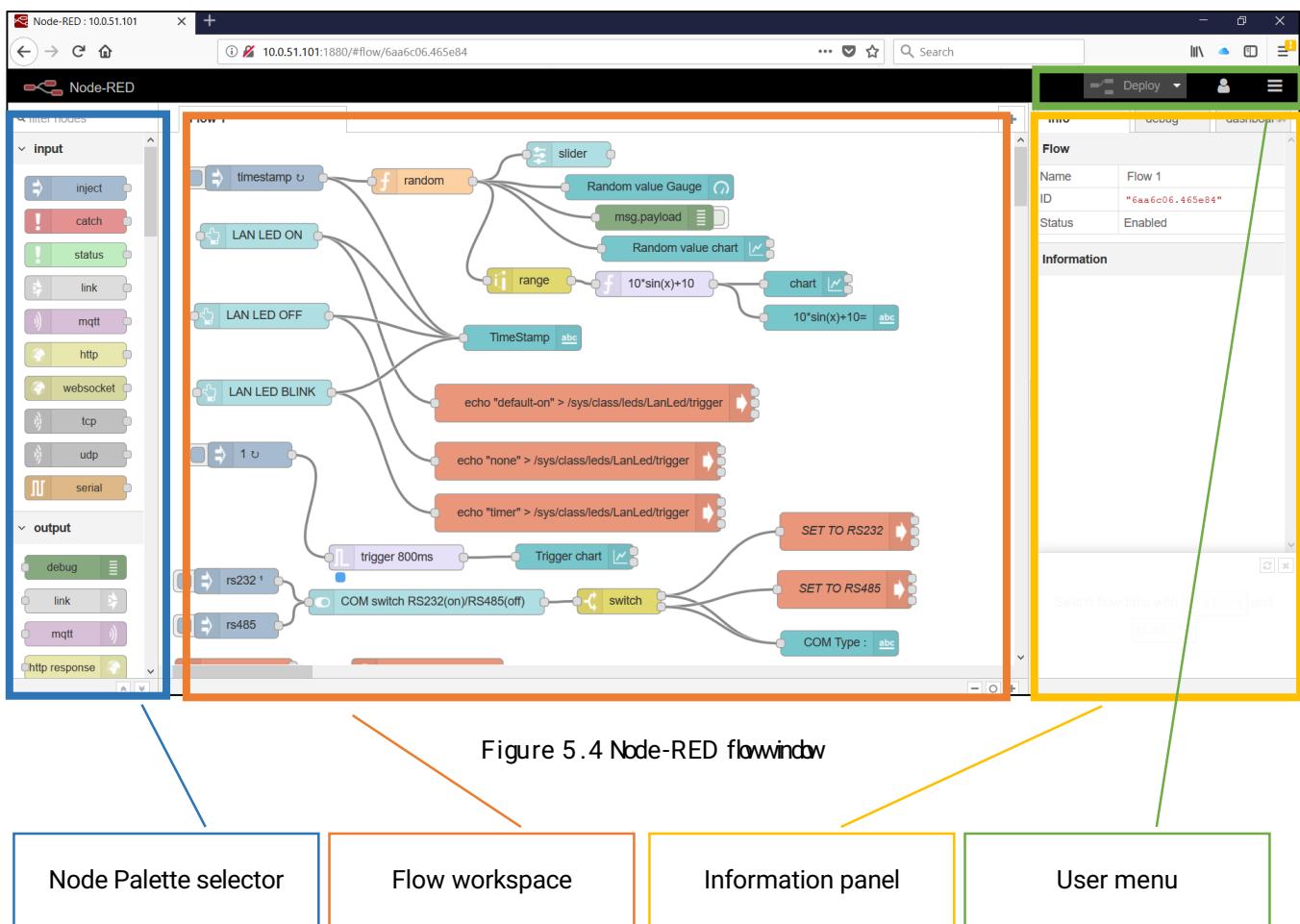
Figure 5.3 Node-RED dashboard example

5.2 Node-RED overview

5.2.1 Node-RED flow

Figure 5.4 shows the sample Node-RED flow that is pre-loaded in the device. This is for allowing the user to get a hands-on understanding on how it works. The Flow-editor page is divided into 4 sections:

- Node palette selector : displays the available building block "nodes" that can be used inside the flow
- Flow workspace : displays the application running on Node-RED
- Information panel : provides access to information related to the Node selected on the flow-area, to the debug information or to the dashboard-related settings (ordering, size, etc..)
- User Menu : provides access to user configuration (such as change of access credentials), import and export of flows, generation of flows or sub-flows and application deployment



5.2.2 **Node palette selector**

ATOP's SE59XX Node-RED has several Nodes that are pre-installed on the device. The user is able to add its own customized Nodes or function specific nodes working directly on the SD-card or on the USB pen-drive filesystem. How to add or install additional nodes is outside the scope of this user manual. For more information, please refer to Node-RED documentation on www.node-red.org. Figure 5.5 shows the different Node Categories available on Node-RED. They are split in 8 categories, based on the function.

- 1) INPUT: represents an input node (or a fixed variable) – this can also be an MQTT subscriber.
- 2) OUTPUT: represents an output node (or a fixed variable) – this can also be an MQTT publisher, an HTTP POST, etc.
- 3) Function: represents a function node that is used to process data or introduce a delay. Such as function, delay, trigger, switch, sort, split, join
- 4) Social: represents a node that can interact with social networks, email, etc..
- 5) Storage: represents a node that can read/write on the filesystem. This is useful for debugging
- 6) Analysis: mainly not used
- 7) Advanced: Exec function is useful for allowing SE59XX running binary applications on the device. Please note that ATOP provides simple I/O programs that allow the use of hardware such as Digital Inputs, Digital Outputs, Relays, Buzzer, LEDs etc.. via a binary application pre-installed on the device.
- 8) Modbus: function nodes that integrate and parse a Modbus Protocol Stack
- 9) Dashboards: graphical elements linked to the NodeRED dashboards. These can be input buttons, fields, sliders or output gauges, graphs, etc..

Figure 5.5 Node-RED nodes categories

In order to use a specific node, please select it from the Node Palette selector and drag it to the Flows page. Afterwards, please set the related node parameters and make the necessary connections in order to allow the node to receive the proper inputs and/or generate the correct outputs.

Table 5.1 SE5908A/16A Pin Assignment for 5-Pin Terminal Block to RS-232/RS-422/RS-485 Connectors

| | | | | | |
|------------------|-----------------|-------------------|-----------------------|--------------------|--------------|
| input | inject | catch | status | link | mqtt |
| | http | websocket | tcp | udp | serial |
| output | debug | link | mqtt | http response | websocket |
| | tcp | udp | serial | | |
| function | function | template | delay | trigger | comment |
| | http request | tcp request | switch | change | range |
| | split | join | csv | html | json |
| | xml | yaml | f(x) curve | rbe | |
| social | e mail | twitter | | | |
| storage | tail | file | | | |
| analysis | sentiment | | | | |
| advanced | watch | feedparse | exec | | |
| modbus | modbus response | modbus read | modbus getter | modbus flex getter | modbus write |
| | modbus server | modbus queue info | modbus flex connector | | |
| dashboard | button | dropdown | switch | slider | numeric |

| | text input | date picker | colour picker | form | text |
|--|------------|-------------|---------------------------|------------|----------|
| | gauge | chart | audio out notification | ui control | template |

5.2.3 Flow Workspace

The flow workspace is where the real embedded application is designed. The user-friendly interface allows to drag the nodes necessary to the application from the Node Palette selector and drop them into the FlowWorks page, set them up and wire them generating the application you need.

A section of ATOP's factory-default Flow is the following:

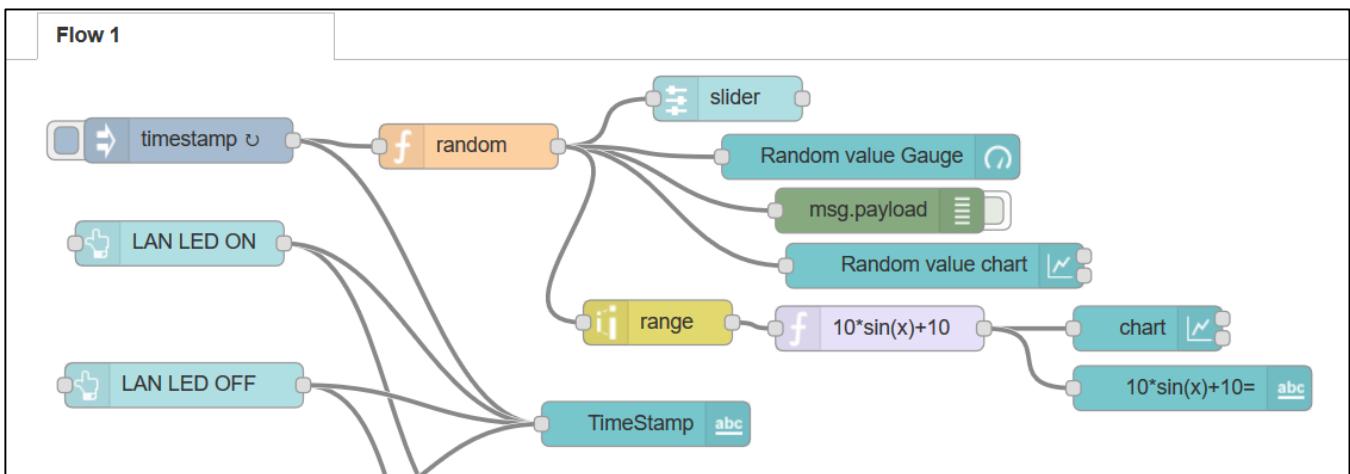


Figure 5.6 Node-RED Flow workspace

By clicking on any node, the Information panel on the right side of the browser will highlight, in the "Info" section, the information related to the function and the current settings as Figure 5.7.

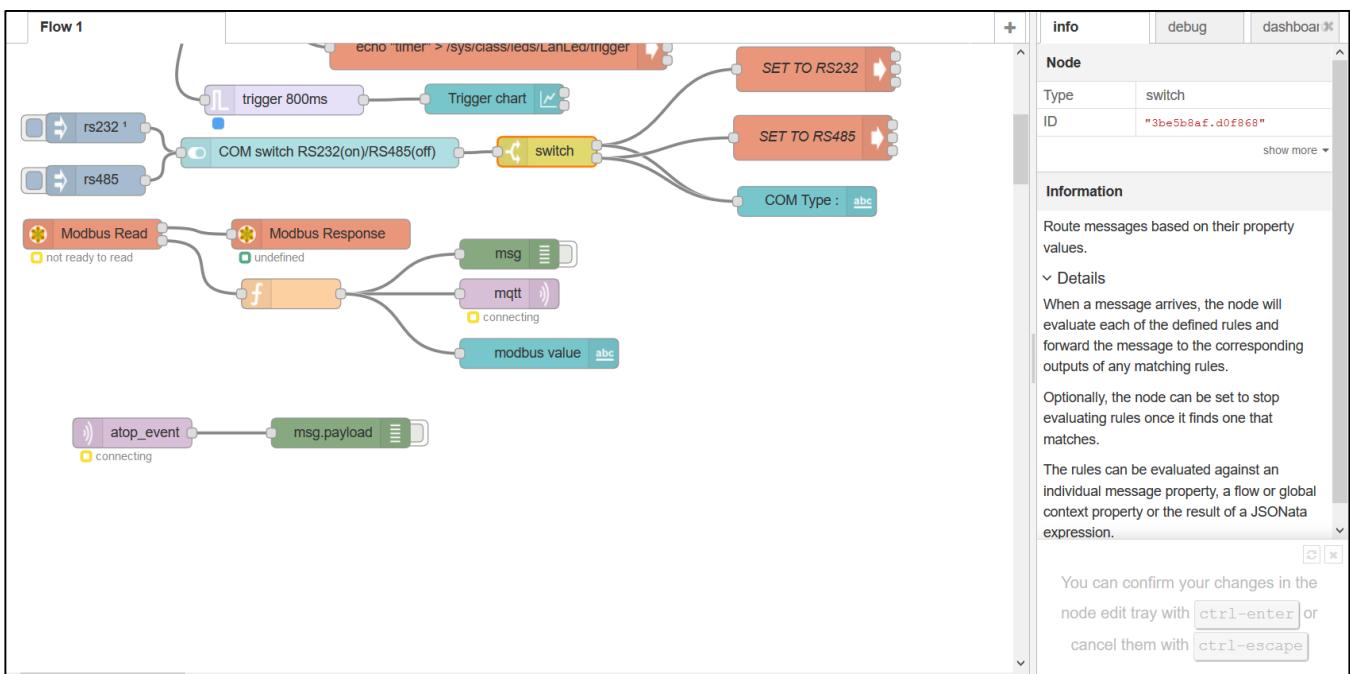


Figure 5.7 Node information panel (example on “switch” function)

By double-clicking the node, a configuration panel will open up allowing the user to input the Node-specific parameters for smooth and effective configuration. An example is provided in Figure 5.8. The parameters are:

- Node Name (useful for remembering what is the Node about). Examples in Section 6.2 below
- Node-specific settings (related to which node you're using). Examples in Section 6.2 below
- Dashboard-specific settings (related to the appearance of the element on the dashboard page) Examples in Section 6.3 below

Note: Dashboard settings will appear on dashboard-related nodes only

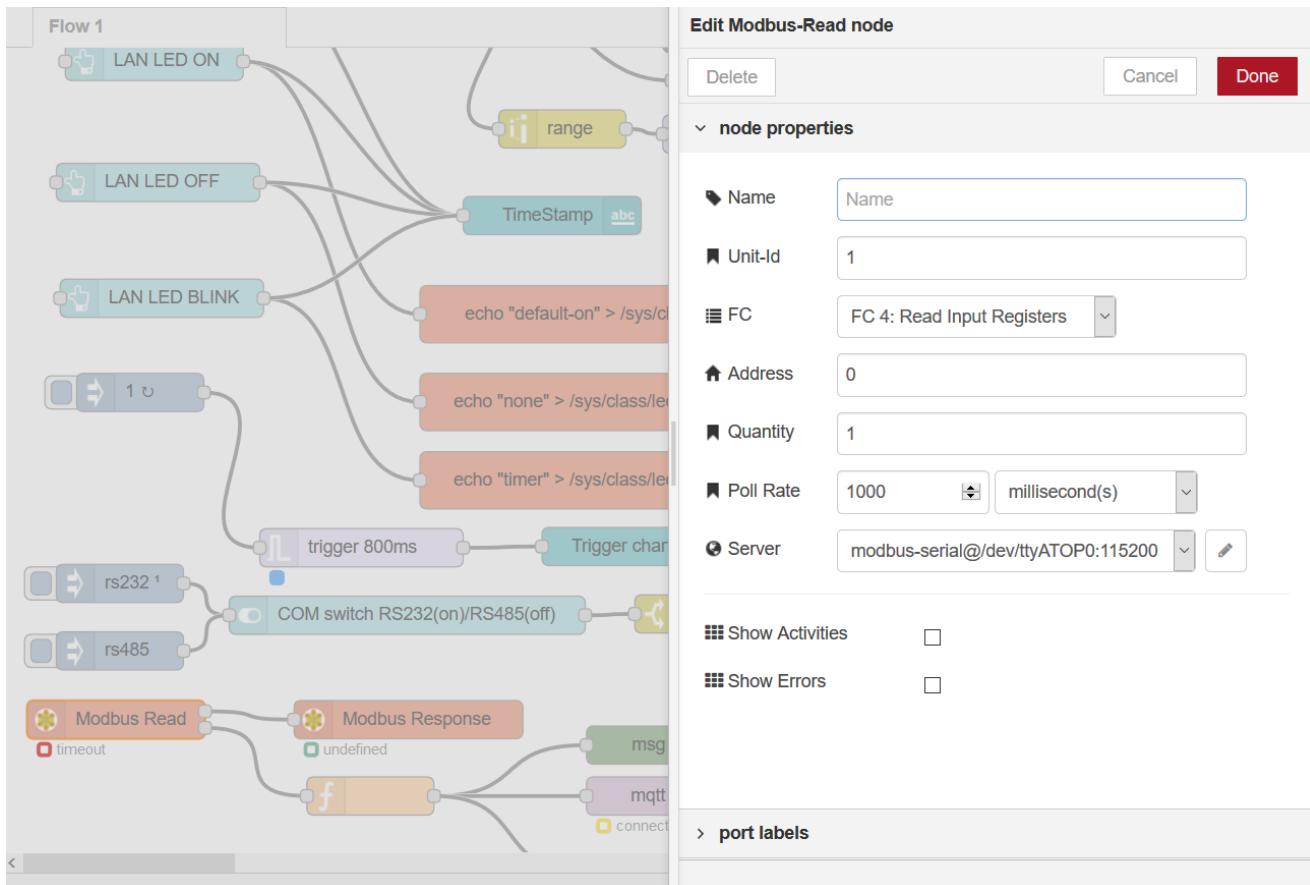


Figure 5.8 Node configuration panel (example on "Modbus-Read" Node)

5.2.4 Information Panel

The information panel is made of three different tabs: info, debug and dashboard.

The info tab is designed to show additional information that can be helpful for the developer to use the Node properly. It will display what is the fundamental parameters to have the function to run, the format of the result and the current configuration settings. Example available in Figure 5.7

info **debug** **dashboard** x

Node

| | |
|--------------|--------------------|
| Type | switch |
| ID | "3be5b8af.d0f868" |
| property | "payload" |
| propertyType | "msg" |
| rules | [object, object] |
| checkall | "true" |
| outputs | 2 |

[show less ▾](#)

Information

Route messages based on their property values.

▼ Details

When a message arrives, the node will evaluate each of the defined rules and forward the message to the corresponding outputs of any matching rules.

Optionally, the node can be set to stop evaluating rules once it finds one that matches.

The rules can be evaluated against an individual message property, a flow or global context property or the result of a JSONata expression.

Figure 5.9 Information panel for "Switch" node

The debug tab allows the user to review real-time data passing through the Node, in order to make debug operations easier.

The Dashboard tab is split into three different sub-tabs. The purpose of Dashboard section is to allow the user to define the order and the appearance of the graphical elements. This tab is selection-independent and from here the user can select any element and configure it for Layout (position), Theme (colors), Site (title, formats, etc.). Click on the circled item in order to open a Dashboard tab. This is an alternative to key in the dashboard path manually (http://IP_ADDRESS:1881/ui/#0). Examples and details are available in Section 6.3 below

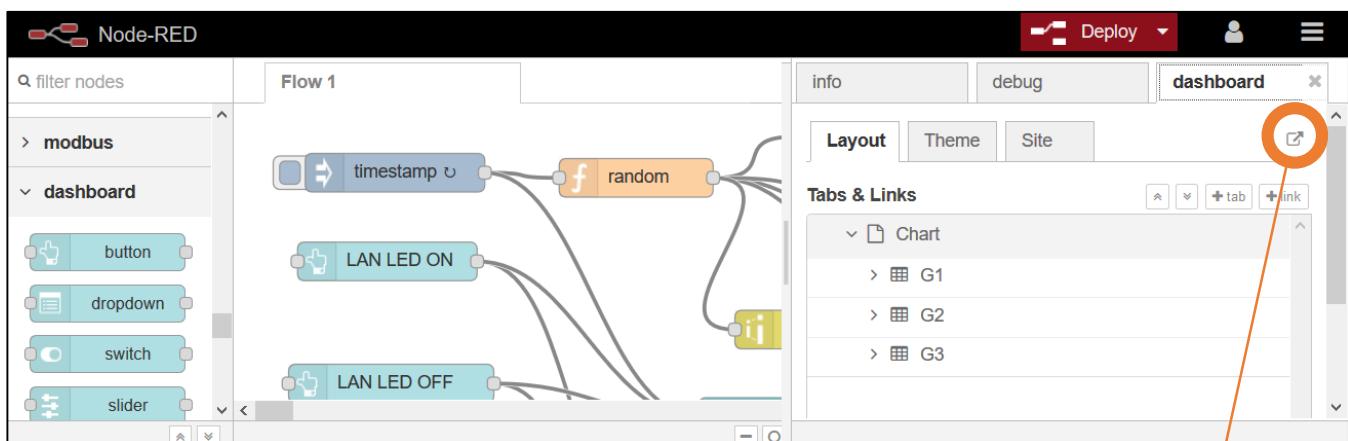
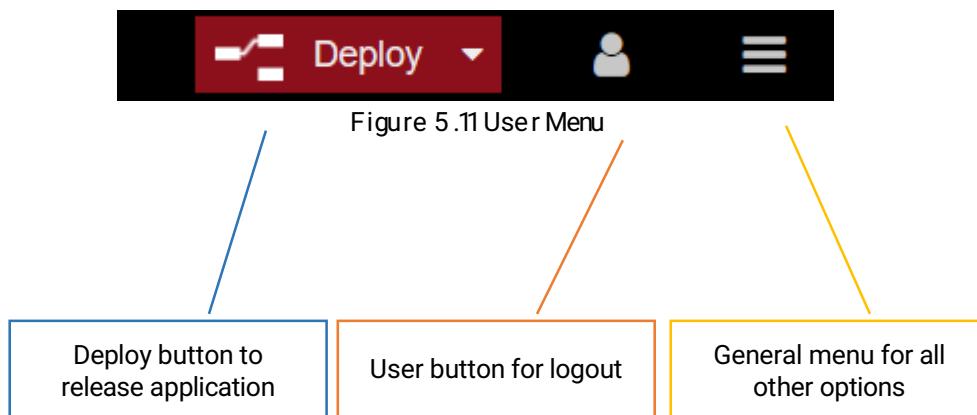


Figure 5.10 Dashboard configuration

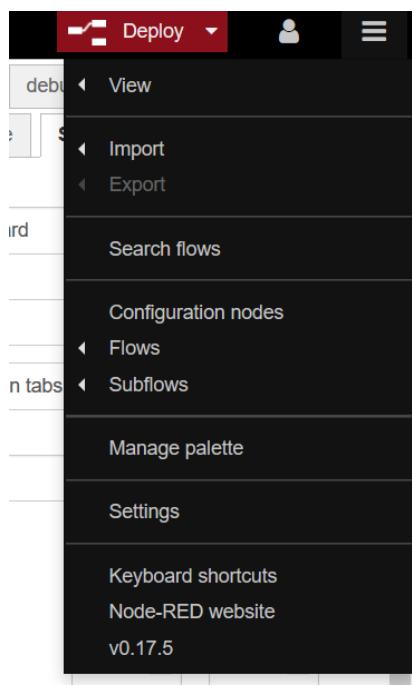
Open Dashboard in a new tab

5.2.5 User Menu

The user menu is located on the top-right corner of the screen and allows the user to define settings, save and export flows, import flows, etc... The red button "deploy" on the left side will be clickable as soon as some changes in the flow editor are made. When the user would like to make the changes operational, clicking this button will restart the application in the new way.



Clicking on the general menu will open a submenu that allows the user to carry out other activities:



For example:

1. **VIEW**: open sidebar, open Dashboard, open debug window
2. **IMPORT**: import new nodes, by copy and paste the code.
3. **EXPORT**: export nodes or library to filesystem or clipboard
4. **CONFIGURATION NODES**: opens a configuration tab into the info area
5. **FLOWS**: allows the user to create, rename or delete flows
6. **SUBFLOWS**: allows the user to create, rename or delete subflows. Subflows can be used inside a normal flow as a node. Once saved, the subflow will appear inside the Node Palette Menu
7. **MANAGE PALETTE**: allows the user to add nodes, update nodes etc.. This is very useful when willing to use third party nodes
8. **SETTINGS**: allows the user to customize NodeRED

Figure 5.12 Drop-down menu

6 Using Node-RED

This chapter aims to help you to design a simple flow, and manage its related Dashboard. Node-RED is Node.JS based, written in Javascript. Therefore, advanced implementations require a knowledge of Javascript programming language. Some nodes especially may require the user to input part of their own script inside of it.

6.1 Create a new flow

Creating a new flow with Node-RED is extremely simple and intuitive. If you'd like to remove the default flow set up by ATOP, select with the mouse all Nodes on the flow and then press the <delete> button on your keyboard. All the content of the Node-RED flow workspace will be emptied, and you'll be ready to create a new flow. If, instead, you'd like to create a new flow running in parallel, then left-click on the Menu on the User menu bar, select Flows, and then Add. Figure 6.1 shows the successful creation of a new flow, where the empty flow-workspace will be shown.

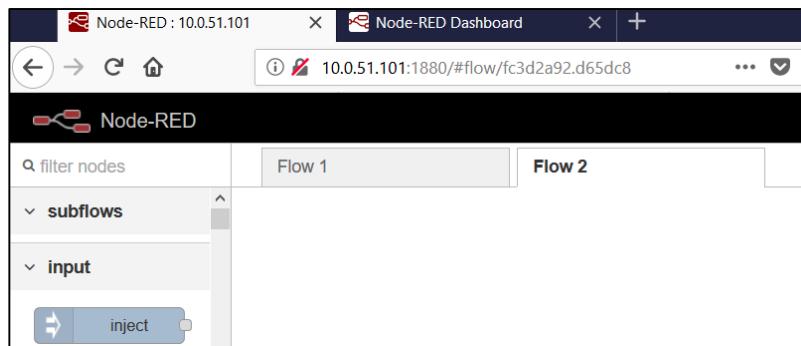


Figure 6.1 New Node-RED flow created

In order to change the flow name, add some descriptive text or temporarily disable the flow, please double-click on the "Flow2" tab (or the tab of the flow you'd like to edit), as shown in Figure 6.2. Click "Done" to save and exit.

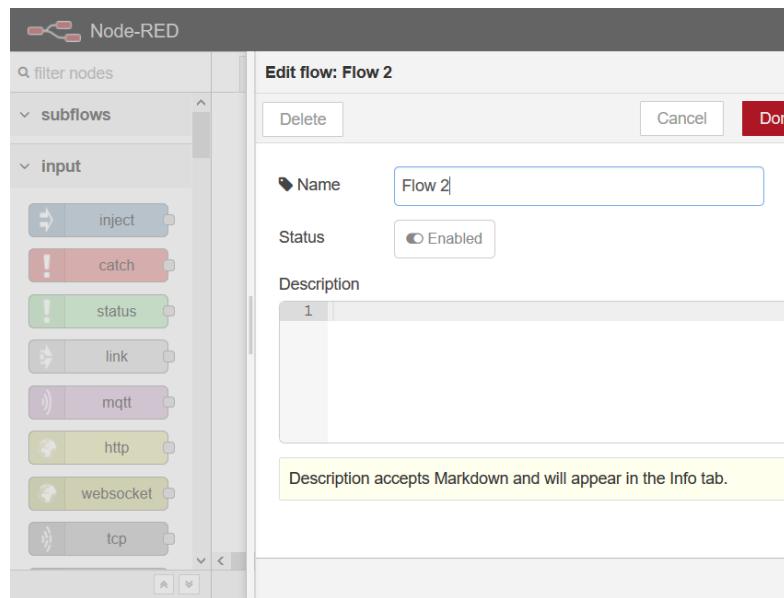


Figure 6.2 Flow options

6.2 Node-RED flow example

In general, Node-RED flows start on inputs (no matter fixed constants or inputs from hardware, web and soon), have data processed by functions (no matter pre-configured or customized with a Javascript script), and output.

The example shown below will generate a simple flow where a random number is generated every second and displayed on a Gauge on the Node-RED dashboard. Some functions like "inject" allows the user to input some constants or, if constants are set-up to be empty, to carry out an action in a specific amount of time that can be set by the user.

In order to design this example, drag from the Node-Palette section to the Node-RED flow workspace the inject node (Input section), the function node (function section) and the Gauge node (dashboard section).

Then, wire the nodes together as per Figure 6.3 by connecting the grey dots on the right of the node (outputs) to the grey dots to the left of the node (inputs).

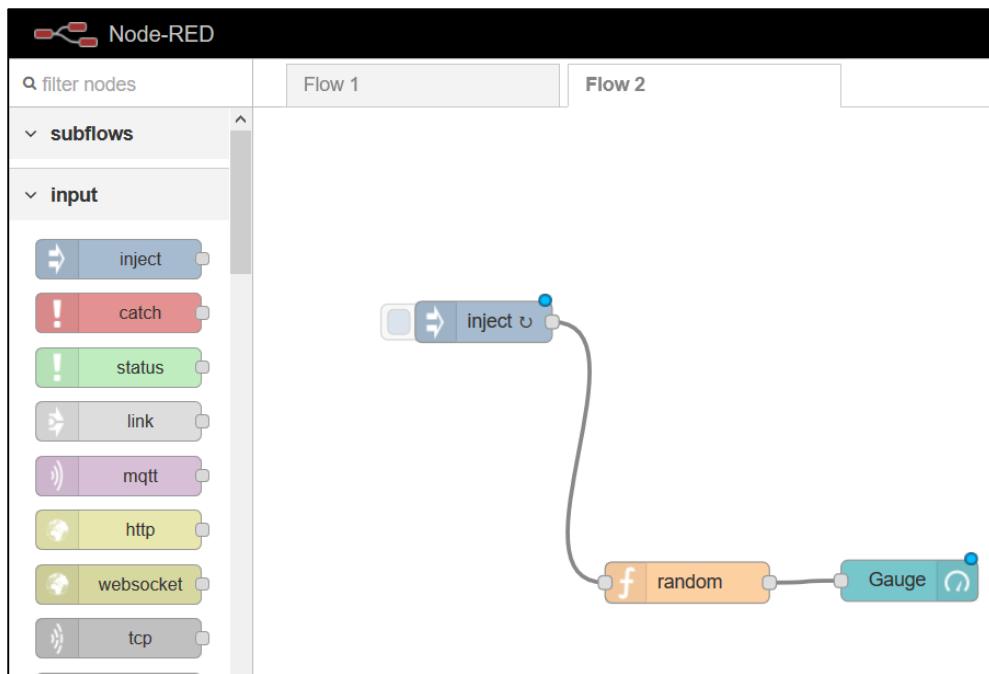


Figure 6.3 Node-RED flow example

Single-clicking a node will show the information panel section the description of that specific node. Double-clicking a node, will open up a node-configuration section that is specific to the selected node. Figures Figure 6.4, Figure 6.5 and Figure 6.6 show the different configurations possible for the inject, function and gauge nodes.

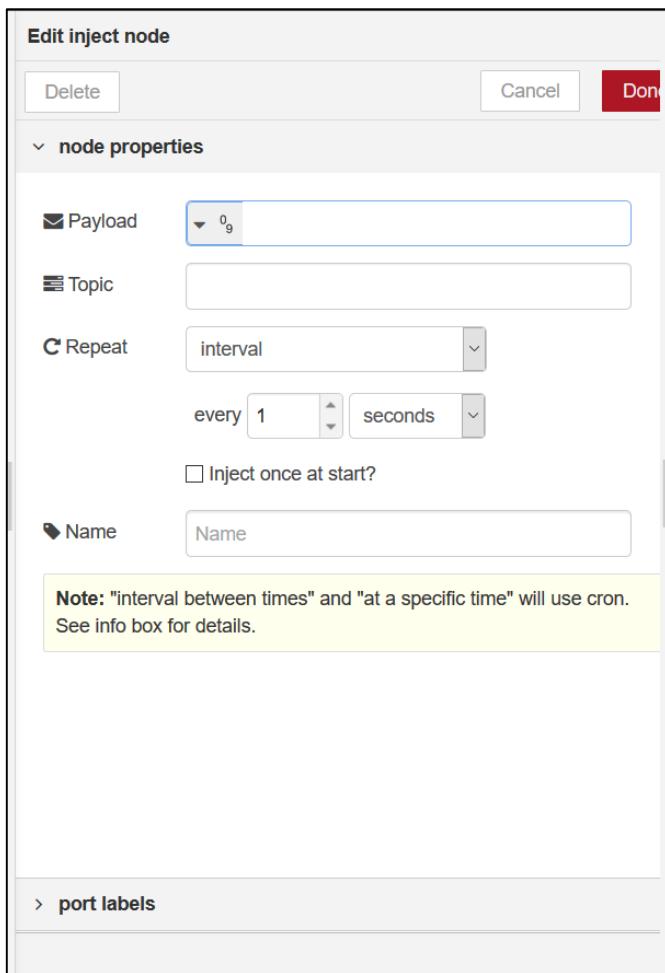


Figure 6.4 Inject node properties

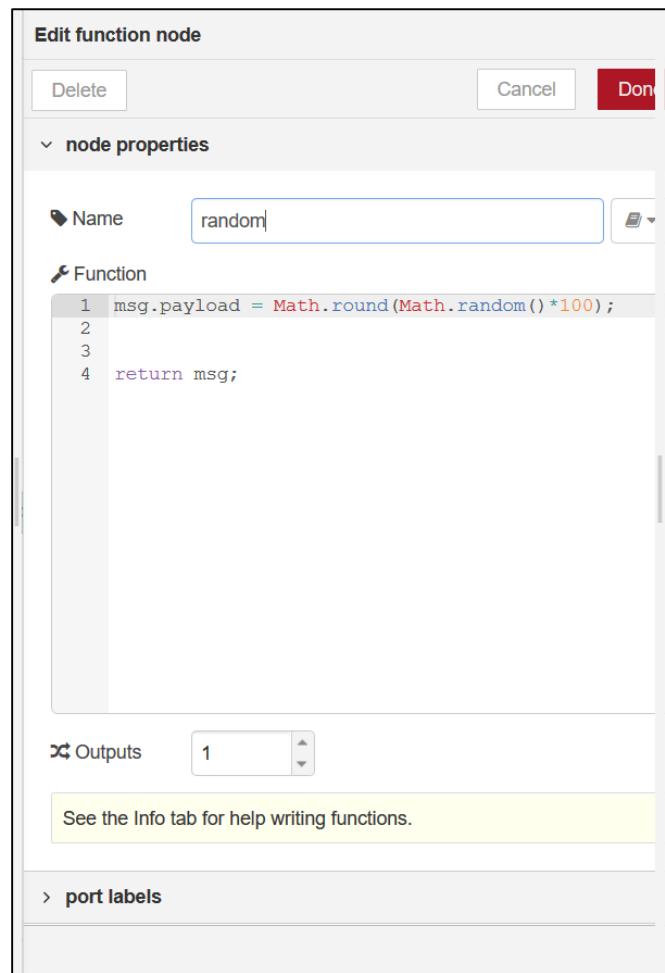


Figure 6.5 Function node properties

Figure 6.4 shows the properties of inject node. This function is usually used in injecting a fixed constant. Here, the example is generating a *null* string every 1 second, and sent to the subsequent node.

Figure 6.5 shows the properties of the function node. This function, one of the most used, doesn't include a specific function, but allows the user to write, in Javascript language, the code for the function itself. The "name" field allows the user to define a label of the function, making it easy to remember. The "function" field is the function itself, written in Javascript. In this specific example, the function generates a random number and multiplies it times 100. The User can also have a function to generate more than one output. When changing the "Outputs" number, automatically the node will have an additional dot, which another node can be wired to.

Javascript and the explanation on how to write functions for Node-RED is outside the scope of this document. Anyway, a very detailed documentation is accessible online on Node-RED website. Please refer to the following URL: <https://nodered.org/docs/writing-functions.html>

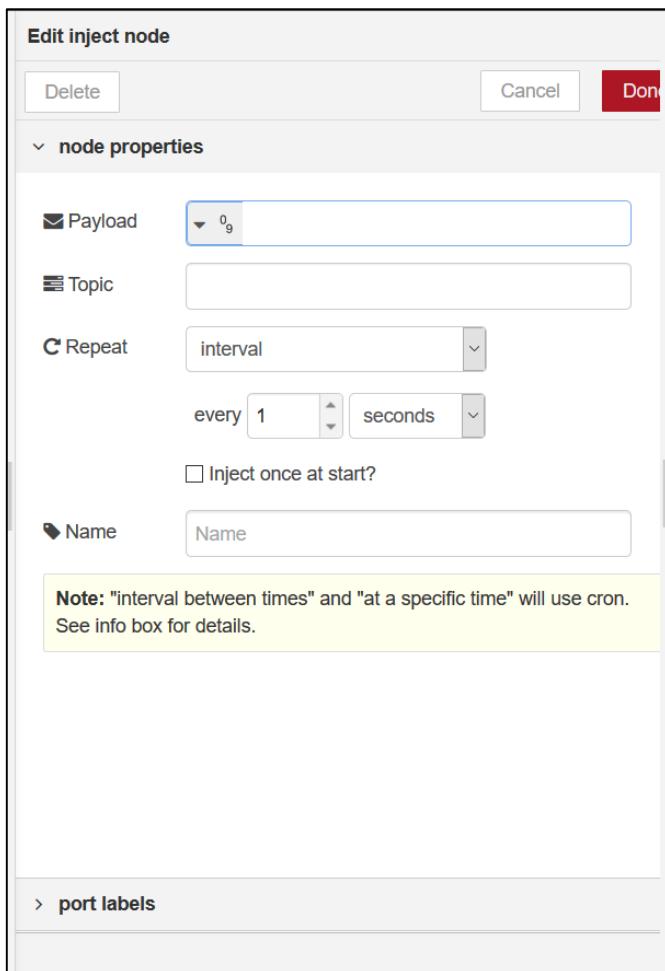


Figure 6.4 Inject node properties

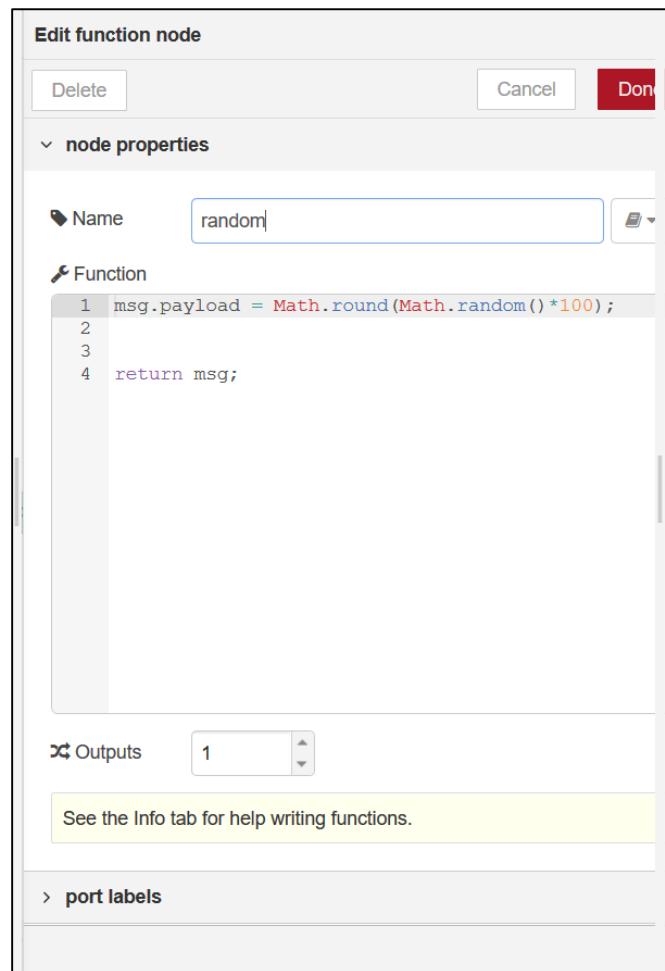


Figure 6.5 Function node properties

Figure 6.6 shows the properties of the Gauge node. This function is one of the many output functions provided by Node-RED in the Node-RED Dashboard. The options allow the user to configure the Group (area of the graphical element inside of the dashboard page), the size, the label, the format of the value shown, the unit of measure (if applicable), the range and the color gradient. The latter, is very useful if it's needed to show ok, alert and alarm values.

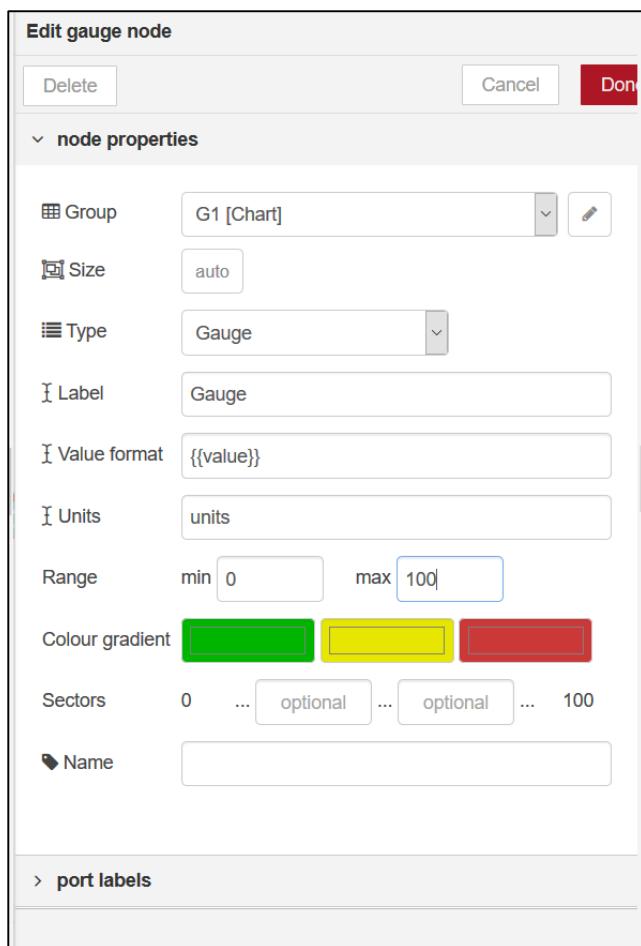


Figure 6.6 Flow options

Once all the parameters are set, click on the “deploy” button in the User Menus page (Figure 5.11). The System will stop existing flows, and run the new application completely. The Node-RED dashboard, if opened, will also refresh. The dashboard of the sample flow described in the previous pages will look approximately like Figure 6.7, with the value refreshing randomly every second.

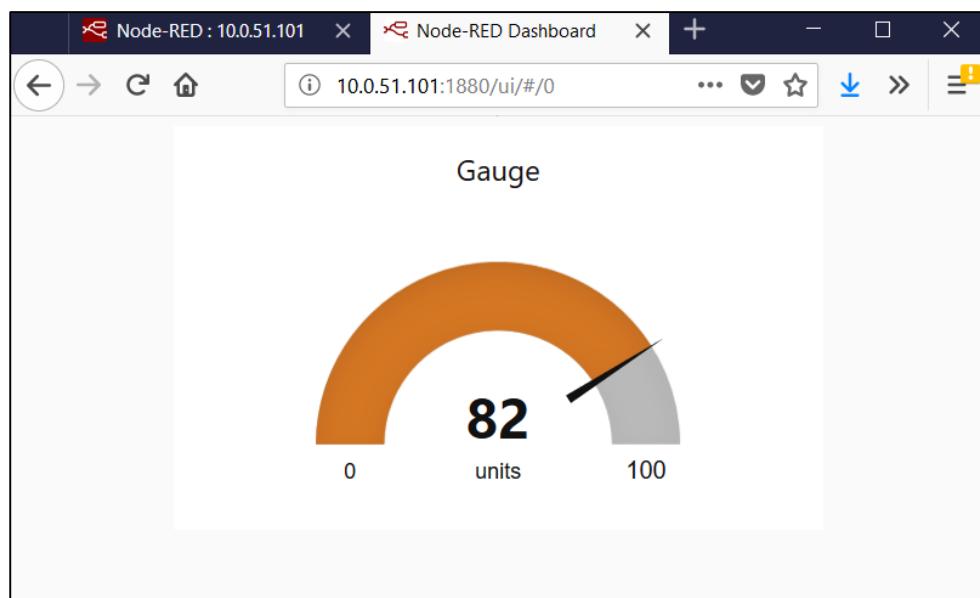


Figure 6.7 Sample Flashboard

6.3 Dashboard-specific settings

The dashboard is a powerful UI that can be used for monitoring the application residing on SE59XX-SDK Node-RED. It is not fundamental: the application runs on the device and doesn't have the need to have a Web-Client (such as Desktop or Laptop computer, Smartphone or Tablet) connected to it to work.

When accessing the Node-RED flow editor, on the right hand side of the screen there is a dashboard tab that can be selected. This is divided into three different sub-tabs, called "Layout", "Theme", "Site". Figure 6.8 shows the dashboard settings how they appear on the Node-RED editor once "dashboard" tab is selected. In Orange highlight the button that, when clicked, opens a separate tab on the browser displaying the dashboard itself.

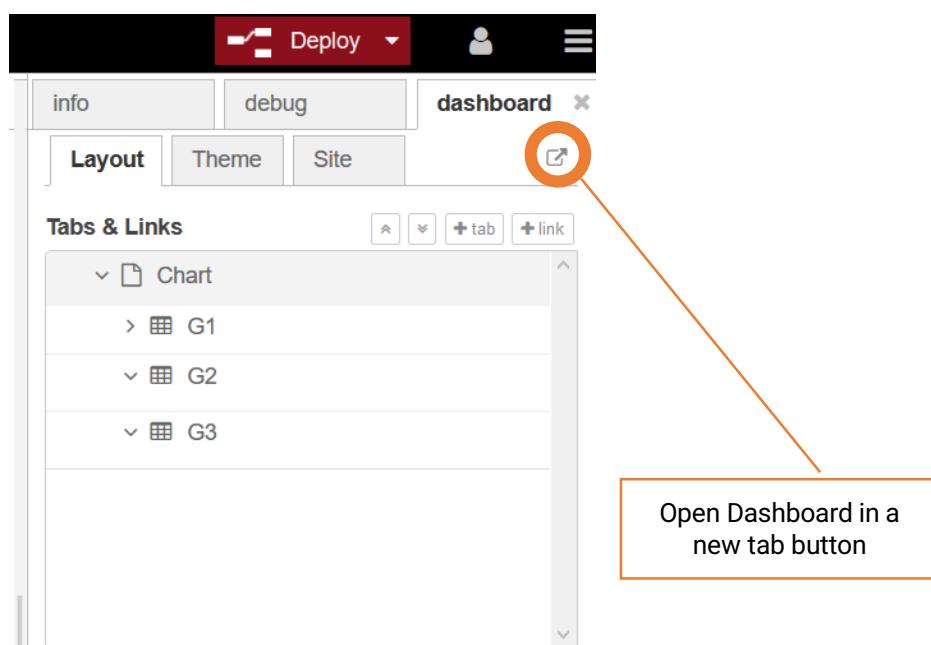


Figure 6.8 Dashboard settings in Node-RED floweditor

The three different sub-tabs have the following meaning:

- **LAYOUT** sub-tab: defines where the Dashboard objects will be located, in which dashboard tab and in which group of objects. The meaning of tabs and groups is explained in Figure 6.12 below.
- **THEME** sub-tab: allows the user to define the color theme and/or the font of the Dashboard. The different results are shown in Figure 6.13 below.
- **SITE** sub-tab: allows the user to customize other parameters, such as the webpage title, date formats, etc..

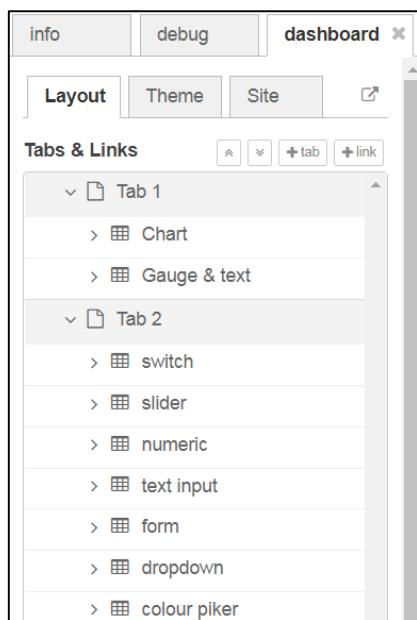


Figure 6.9 Dashboard Layout Settings

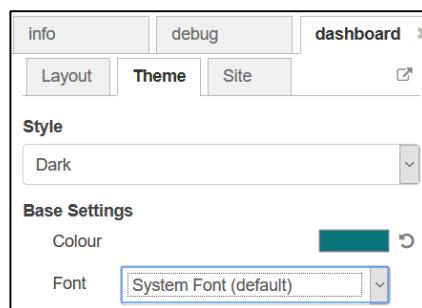


Figure 6.10 Dashboard Theme settings

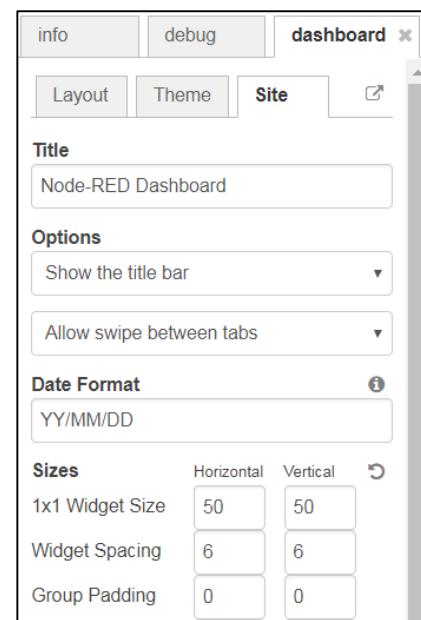


Figure 6.11 Dashboard site settings

6.3.1 Dashboard: layout settings

A **group** is a logical and graphical grouping of similar elements, or of elements that graphically make sense to be put nearby. For example, a dashboard may monitor temperatures from 3 sensors and pressure from three other sensors. It makes sense to have all temperatures grouped together or all pressures grouped together, on the same page. Or instead, pressure and temperature of location A grouped together and soon. This is at discretion of the user and its heavily application-oriented.

A **tab** is a logical and graphical grouping of elements, same as group. The difference with the group is that while the group shows the elements in different "groups" of the same page, when defining different tabs, the webpage displayed on the browser will show only the elements assigned to that specific tab only. Inside a Tab, groups can be defined.

Figure 6.12 shows the same flow designed as example (Figure 6.3), connecting to the "random number" generator function totally 4 different Dashboard output elements. 3 Elements are assigned to Tab1 (2 items to Group1, 1 item to Group 2) and 1 Element is assigned to Tab2, Group 1.

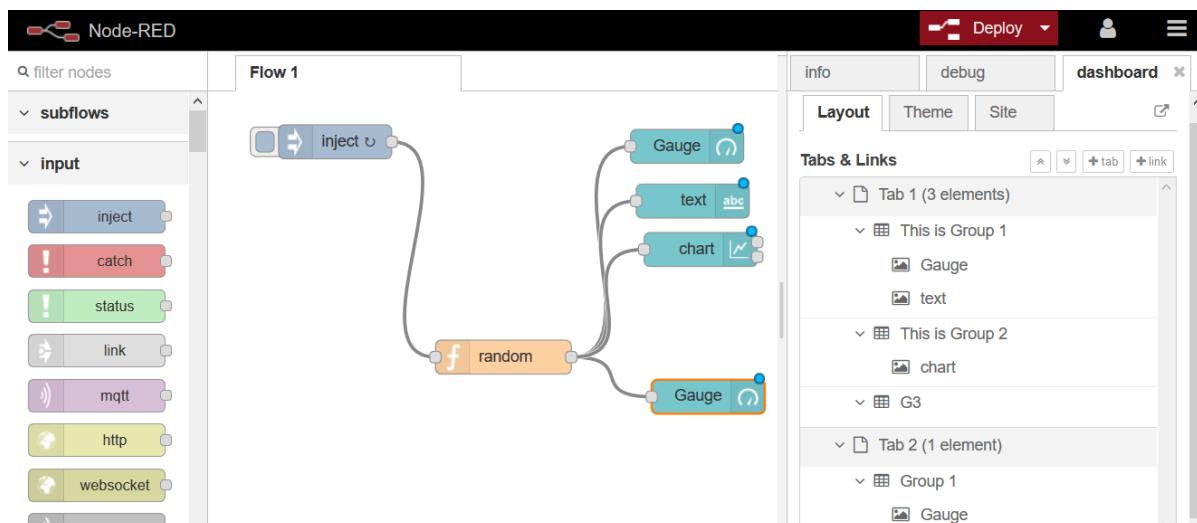


Figure 6.12 Dashboard Groups and tabs Flowexample

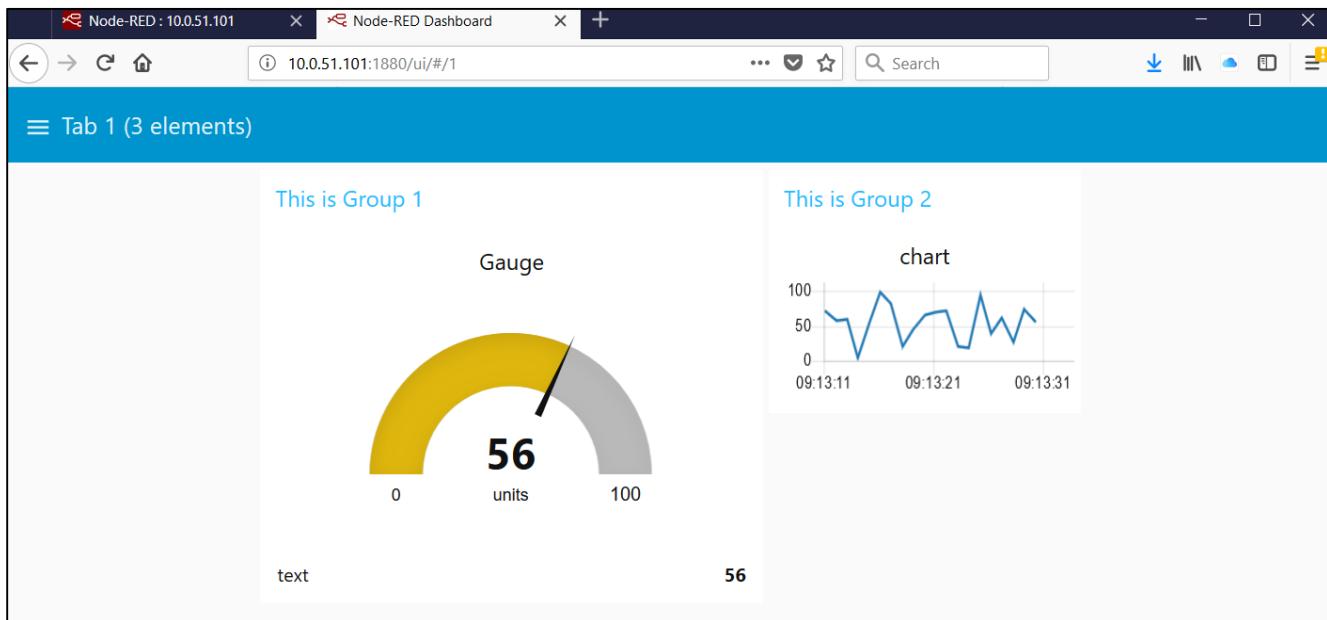


Figure 6.13 Dashboard Groups and tabs Dashboard result on Tab1

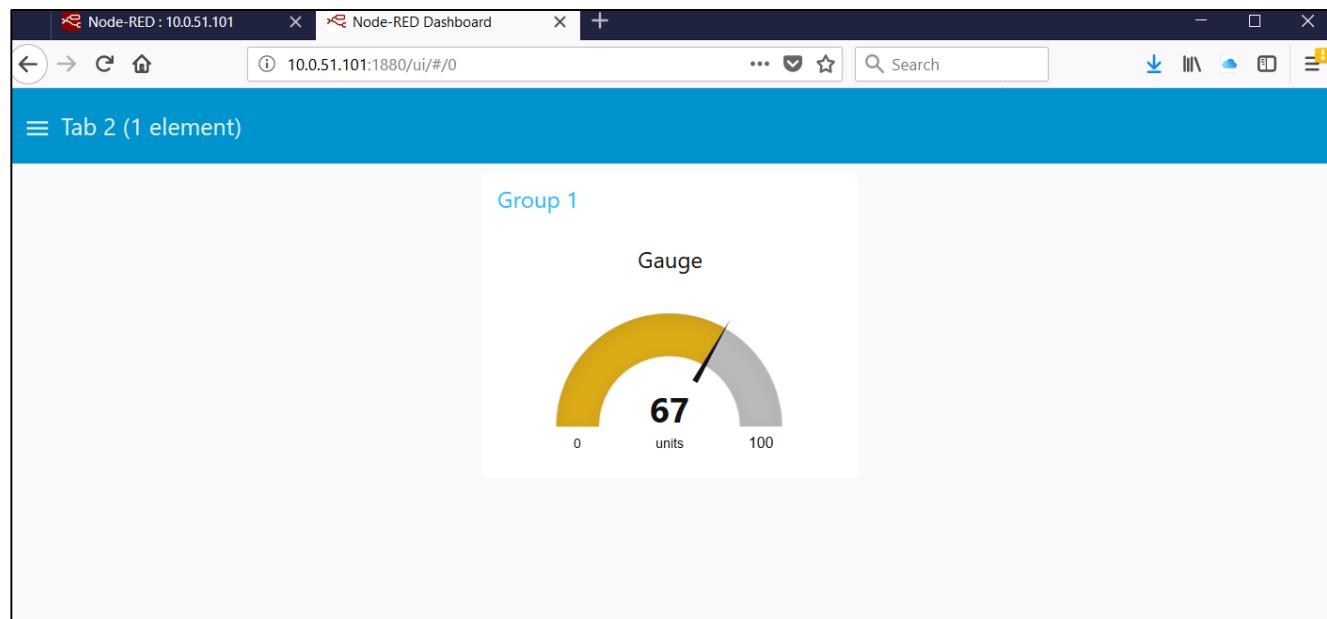


Figure 6.14 Dashboard Groups and tabs Dashboard result on Tab2

Note: in order to show the tab switching bar on top, it is necessary to set “Show title bar” in Dashboard/Site options as explained in Section 6.3.3 below.

6.3.2 Dashboard: theme settings

It is possible to change the theme of the dashboard. There are three options available:

- Light (default) : white background, and colors as in all dashboard previous examples

- Dark : dark grey background, and colors as in Figure 6.15
- Custom: gives the possibility to customize the appearance settings, including font, colors, etc..

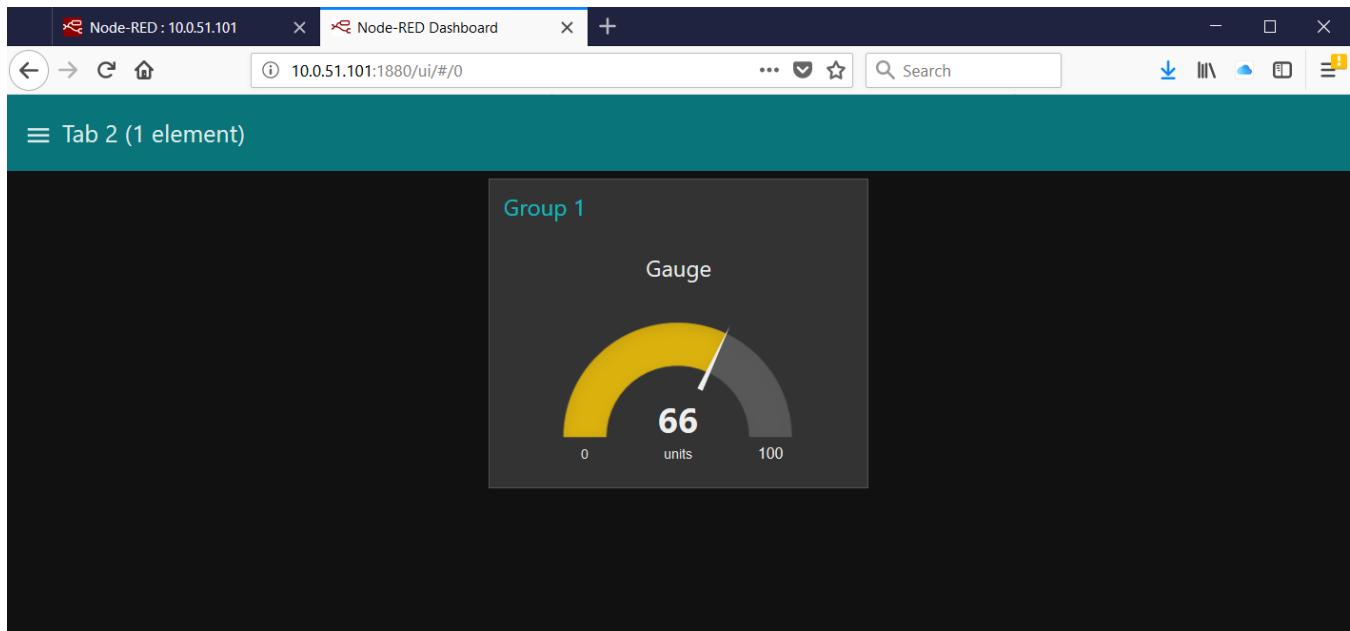


Figure 6.15 Dashboard Dark Theme settings

6.3.3 Dashboard: site settings

The site settings can be used to further customize the Node-RED dashboard

- Title: allows to modify the website title (Replaces “Node-RED Dashboard”, set by default)
- Show/hide title bar: allows to display the title bar that allows the user to switch between tabs. It is recommended to use “show title bar” option for easier user management.
- Allow/hide swipe between tabs: allows the user to switch between tabs from the top menu
- Date format: allows the user to set the regional date format standard.
- Spacing and sizing option: allows the user to specify size and spacing options.

Figure 6.16 below shows an example where the site settings have “ATOP TEST” as Title and hidden title bar.

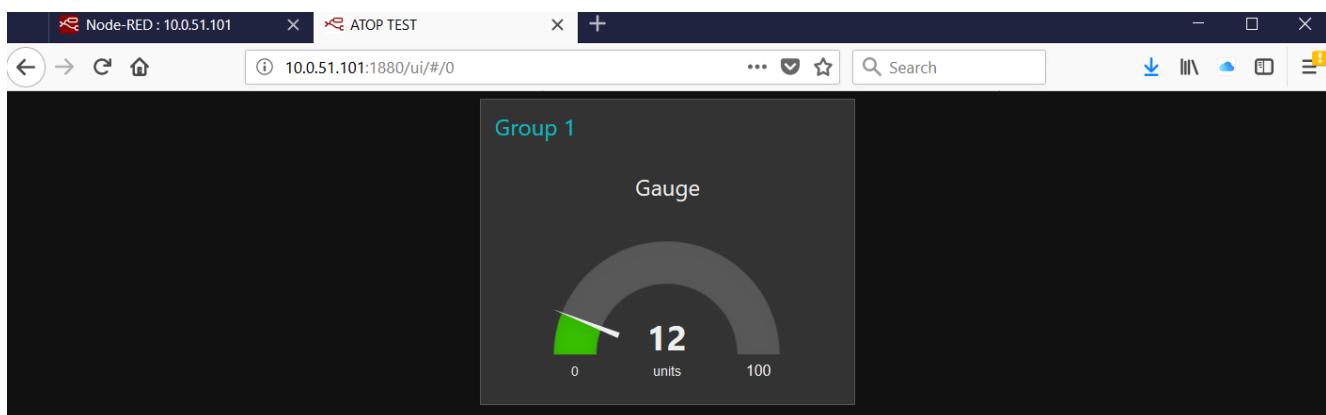


Figure 6.16 Dashboard showing Tab2 with customized title and hidden title bar

6.4 Dashboard- user inputs

Node-RED allows, along with displaying data elements, also to have data-input elements on the dashboard. These can be wired as desired to the other building blocks inside the flow in order to make it interactive. For example, a switch can be used to enable or disable an automatic algorithm, a slider can be used to make changes in a variable, and so on.

Note: Node-RED has the capability to run embedded applications but ATOP does not recommend do use it for mission-critical process control, automation, utilities and so on. Being a Javascript-based application, the run-time performance is not as good as a binary application running on SE59XX-SDK. Node-RED is very useful to be used for monitoring, but it may be dangerous if deployed in applications where a wrong behavior can put people's live in danger.

Figure 6.17 shows, on the dashboard, the different user inputs supported in Node-RED. The configuration of each single node is outside the scope of this document. If there are questions, please consult documentation on www.nodered.org

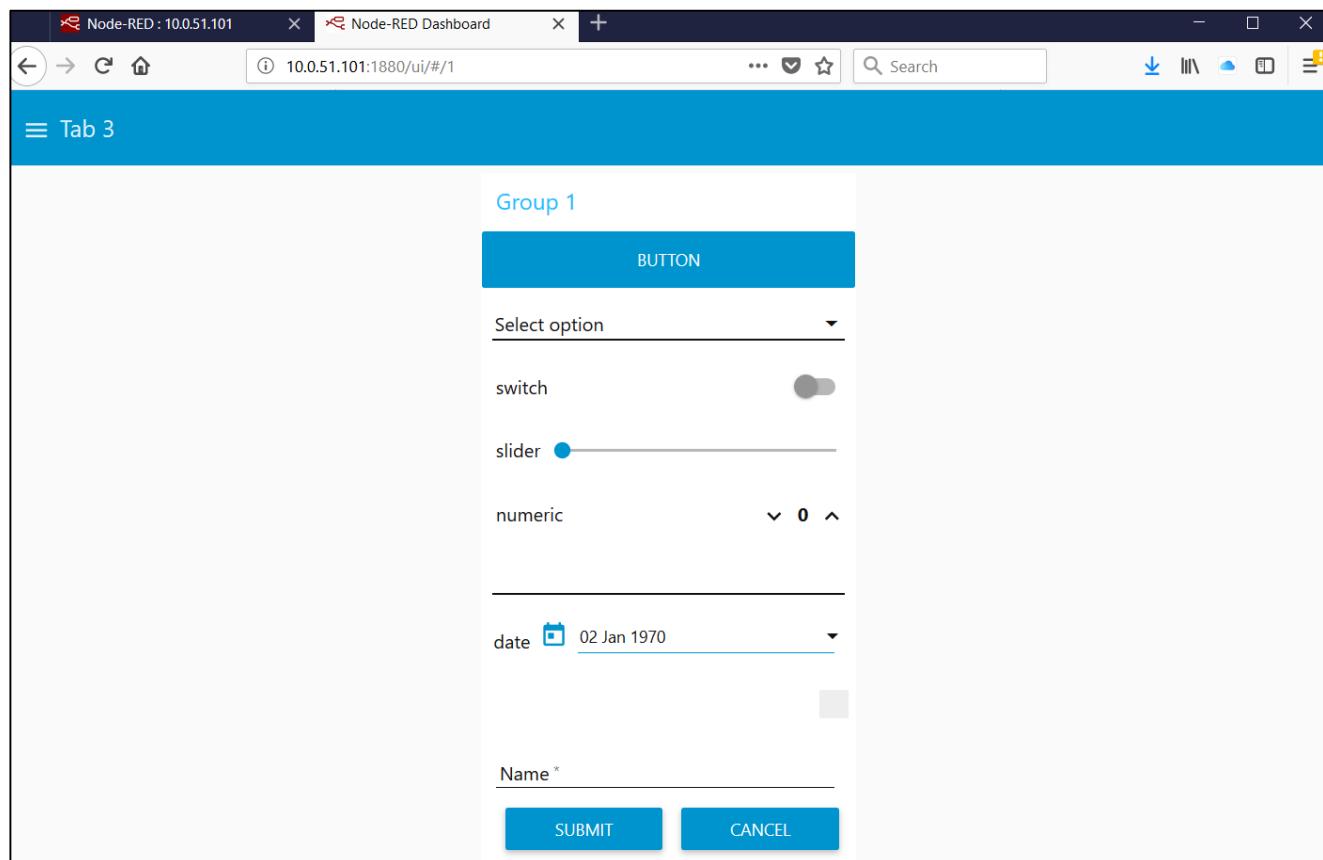


Figure 6.17 Dashboard showing all different available user inputs

6.5 Accessing and controlling ATOP SE59XX Hardware with Node-RED

Using Node-RED as embedded application on SE59XX-SDK Node-RED will allow you to get the best out of ATOP hardware and access all hardware interfaces available on it, no matter Serial, Ethernet, Digital Inputs, Digital Outputs, Buzzer, Relays and so on. In this chapter we will go through the methods to be used to access data.

6.5.1 Configure Serial Port mode

The configuration of the Serial Mode on SE59XX-SDK Node-RED is using the “exec” Node, shown in Figure 6.18. The “exec” Node, basically runs a Linux binary program that is stored in the Filesystem or a Linux command. The “exec” Node is located in the “advanced” section of the Node palette on the left hand side of the screen.



Figure 6.18 Exec Node

The configuration of the “exec” node requires some parameters, as shown in Figure 6.19 below and as explained in the following paragraph.

- **Command:** defines the linux command or the binary application to be run
- **Append “msg.payload” checkbox:** allows the execution of the application or the command to have appended the input wired to the “exec” node. If the checkbox is not checked, then the command will be executed without any input parameters, as it is written
- **Append extra input parameters:** allows the user to input additional constant parameters to the command or to the binary application
- **Output:** allows the user to define whether issue the output only when the command execution is finished or during run-time
- **Timeout:** allows the user to define the execution timeout time in ms before having the process killed.

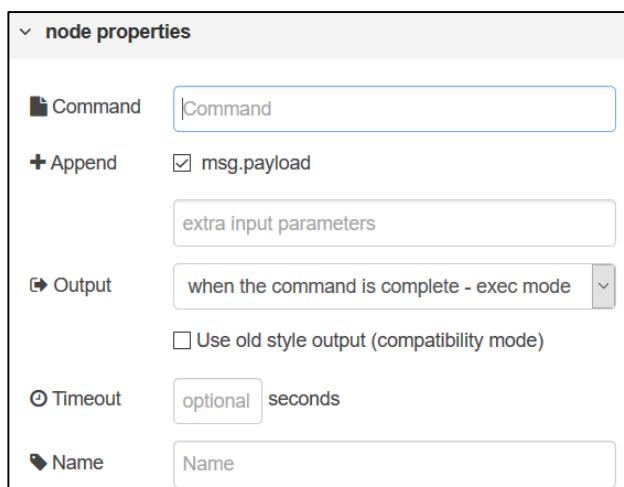


Figure 6.19 Exec Node Configuration Parameters

Use exec Node to configure the COM port Mode. The tables below list device node of COM port for each model.

Table 6.1 SE59XX Programming commands per device node

| Device node | ioctl command | Command Description |
|-----------------------|---------------|---|
| ttyATOP0 ~ttyATOPX | 0x9000 | Configure SE59XX COM port as one of RS232 / RS485 / RS422 |

Table 6.2 SE59XX ioctl command of COM Port

| ioctl command | parameter type | Value | Description |
|---------------|----------------|-------|-------------------------|
| 0x9000 | integer | 1 | Configure to RS232 mode |
| | | 2 | Configure to RS422 mode |
| | | 3 | Configure to RS485 mode |

6.5.2 Read and Write data to Serial Ports

The nodes shown below in Figure 6.20 are used to read and write from the serial port

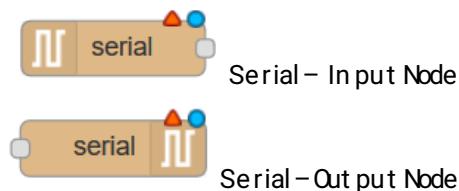


Figure 6.20 Serial ReadWrite Nodes

Besides the configuration, shown in Figure 6.21 and Figure 6.23 below, the Serial Input Node will produce an output, and the Serial Input Node will require an input.

Inside the serial port configuration, the default Serial port will automatically be selected (COM1). This will be marked as /dev/ttyATOP0. To select a different serial port, please cross-reference to Table 6.3 below.

Table 6.3 SE59XX device node

| Device node | Major & Minor number | Device Type | Description |
|-------------|----------------------|-------------|-----------------|
| ttyATOP0 | 266 0 | Character | ATOP COM port 1 |
| ttyATOP1 | 266 1 | Character | ATOP COM port 2 |
| ttyATOP2 | 266 2 | Character | ATOP COM port 3 |
| ttyATOP3 | 266 3 | Character | ATOP COM port 4 |
| ... | ... | ... | ... |

Figure 6.21 below shows the Serial Read Node configuration options.

- **Serial port:** defines on which serial port Node-RED should listen to. Click on the “pencil” icon to open up the window shown in Figure 6.22.
- **Name:** defines the node name, easy way to remember which port/application the node is referring to.



Figure 6.21 Serial Read Node options

Figure 6.22 shows the Serial port configuration options. The meaning of the fields is explained below:

- **Serial port:** Defines which device serial port Node-RED should listen to. Refer to Table 6.3
- **Baud rate:** Defines the baud rate which the sending device is transmitting to SE59XX
- **Data bits:** Defines how many data bits are inside each frame. This should be the same of the one set on the device that is transmitting to SE59XX
- **Parity:** defines whether there's a parity bit to check data consistency
- **Stop Bits:** defines whether there's a stop bit to mark the end of the frame
- **Split input:** defines how the data received should be split in different output messages.
- **And deliver:** defines the data output format
- **Add split character to output messages:** defines whether the split character should be appended to the output message or shouldn't be considered

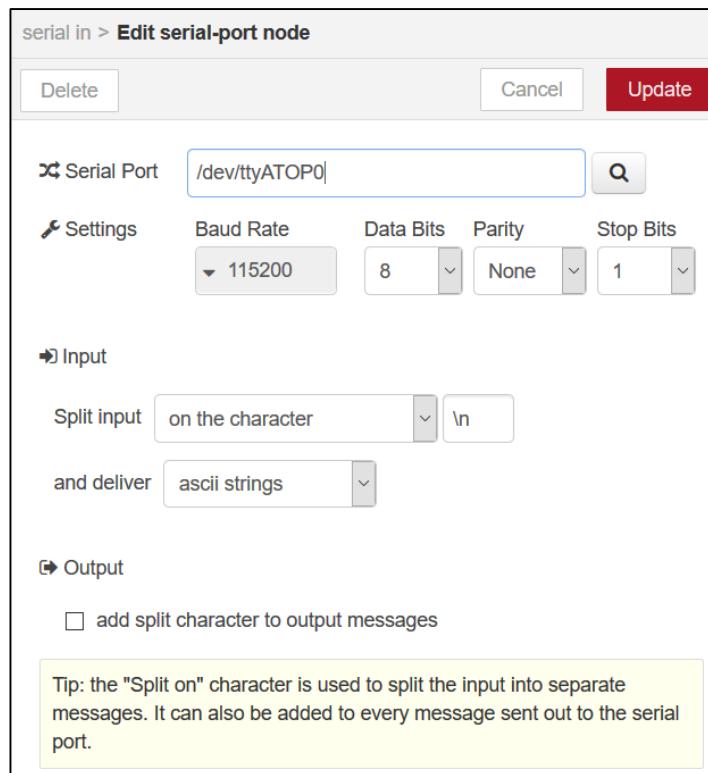


Figure 6.22 Serial Read Node port configuration options

Figure 6.23 below shows the Serial Read Node configuration options.

- **Serial port:** defines on which serial port Node-RED should write data to. Click on the “pencil” icon to open up the window shown in Figure 6.22.

- **Name:** defines the node name, easy way to remember which port/application the node is referring to.



Figure 6.23 SerialWrite Node options

Figure 6.24 shows the Serial port configuration options. The meaning of the fields is explained below:

- **Serial port:** Defines which device serial port Node-RED should write data to. Refer to Table 6.3
- **Baud rate:** Defines the baud rate which the SE59XX should be transmitting
- **Data bits:** Defines how many data bits are inside each frame. This should be the same of the one set on the device that is receiving data from SE59XX
- **Parity:** defines whether there's a parity bit to check data consistency
- **Stop Bits:** defines whether there's a stop bit to mark the end of the frame
- **Split input:** defines how the data received by the input node should be split in different output messages.
- **And deliver:** defines the data output format
- **Add split character to output messages:** defines whether the split character should be appended to the serial write message or shouldn't be considered

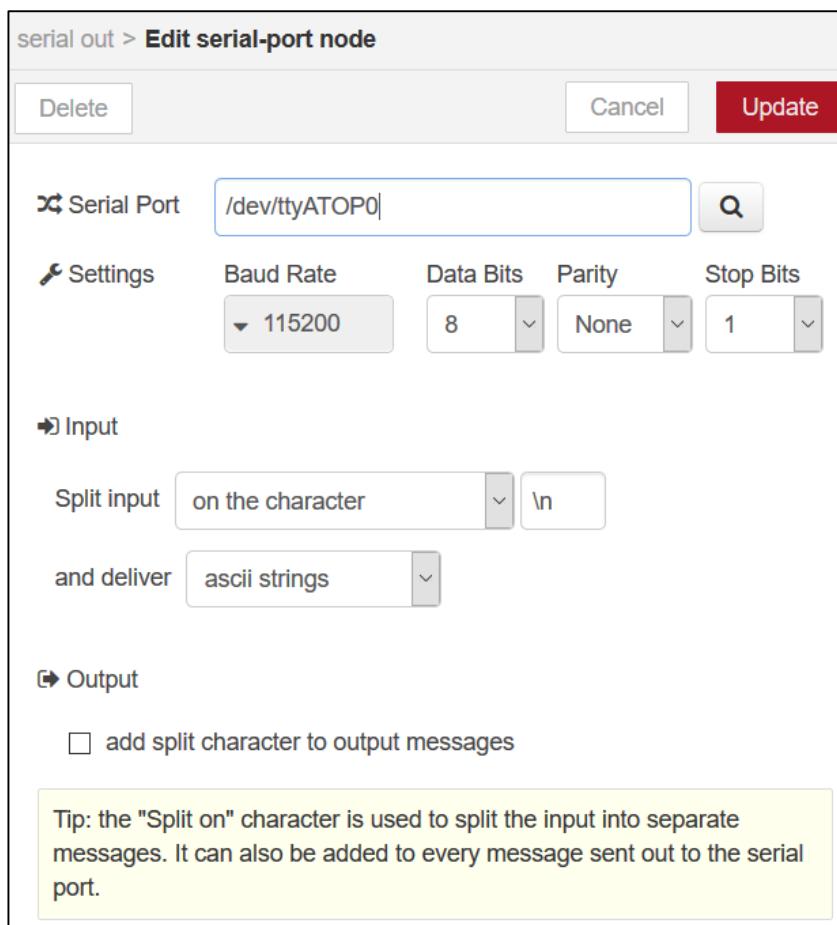


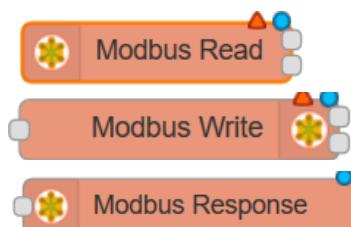
Figure 6.24 SerialWrite Node port configuration options

6.5.3 Modbus TCP/RTU/ASCII

The nodes shown below in Figure 6.25 are used to read and write from the serial port using Modbus RTU/ASCII. There is one node (Modbus Server) specifically designed to open a server on SE59XX-SDK Node-RED, while “Modbus Read”, “Modbus Write” and “Modbus response” nodes can be used either in connection with “Modbus Server” node or separately for data polling or command writing (in Modbus Master mode).

All Modbus Nodes support Modbus TCP, RTU, ASCII and can be used either to control serial ports or to use the device's LAN ports. In addition to the below mentioned Nodes, Node-RED provides additional Modbus functions, very useful when carrying out multiple pollings or multiple writings.

The examples below listed, and the configuration is not inside the scope of this document. If there are any questions or doubt, please review Node-RED official documentation on www.nodered.org



Modbus – Read Node

Modbus – Out put Node

Modbus – Write Node



Figure 6.25 Main Modbus Nodes

6.5.4 Read data to Serial Ports using Modbus RTU/ASCII

In order to read data from Serial ports using Modbus RTU/ASCII, please set-up the flow as shown in below Figure 6.26. This flow uses 1 “Modbus read” node, 1 “Modbus Response Node” and 1 “Dashboard text output” node, that here is used to show the data on the screen.



Figure 6.26 Modbus Serial Read example flow

The Modbus Read node has the following arguments, as per Figure 6.27:

- **Name:** arbitrary name
- **FC:** Modbus read function number to use (drop-down menu) – please see Modbus specifications
- **Address:** Modbus ID from which the information should be read
- **Quantity:** Quantity of coils or words that should be read starting from address mentioned
- **Poll-rate:** Frequency of data polling. This can be either in milliseconds, seconds, minutes, hours
- **Show activities flag:** shows the progress of the function
- **Show errors flag:** shows the polling errors, if any.
- **Server:** allows the user to choose which interface should be used for data polling. The default value is COM1 (/dev/ttyATOP0). By clicking on the arrow icon, an additional window will open up, as per Figure 6.28. The parameters of this window are the following
 - **Type:** Defines whether use Modbus over Serial (RTU or ASCII) or over Ethernet (TCP). The Serial Expert mode allows a more detailed parameter settings explanation.
 - **Serial port:** Defines which device serial port Node-RED should write data to. Refer to Table 6.3
 - **Serial type (only if Type is “Serial” or “Serial Expert”):** Defines whether use Modbus RTU or ASCII
 - **Baud rate:** Defines the baud rate which the SE59XX should be transmitting
 - **Data bits (“Serial Expert” only):** Defines how many data bits are inside each frame. This should be the same of the one set on the device that is transmitting to SE59XX
 - **Stop Bits (“Serial Expert” only):** defines whether there's a stop bit to mark the end of the frame
 - **Parity:** defines whether there's a parity bit to check data consistency
 - **Timeout:** Defines after how many millisecond the system should generate an error
 - **Reconnect timeout:** defines time to wait on reconnect before making next polling
 - **Serial Connection delay** (default 500 ms) - time to delay first command sending after reconnect

Edit Modbus-Read node

Delete Cancel Done

node properties

Name: Name

Unit-Id:

FC:

Address: 0:65535

Quantity: 1-65535

Poll Rate: 1-65535

Server: modbus-serial@/dev/ttyATOP0:115200

Show Activities:

Show Errors:

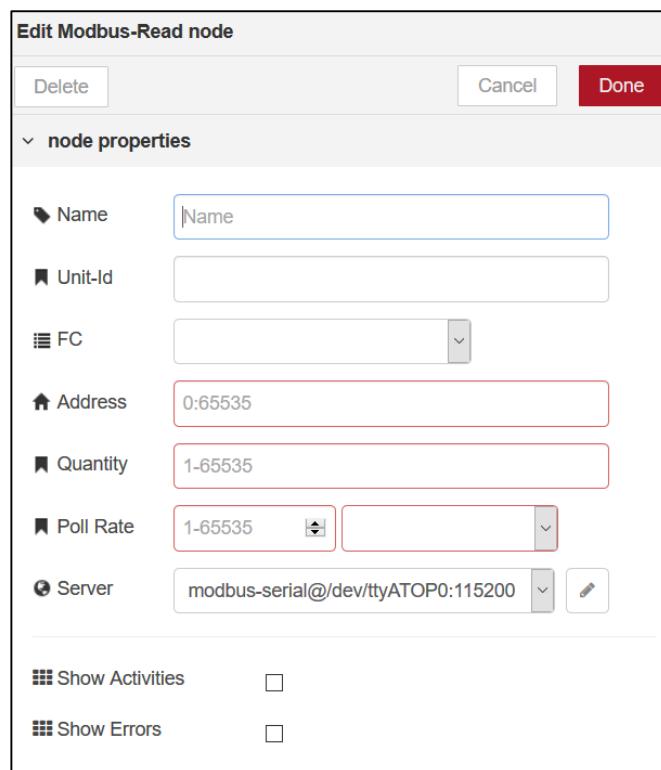


Figure 6.27 Modbus Read Node Settings

Modbus-Read > Add new modbus-client config node

Cancel Add

Name: Name

Type: Serial

Serial port: /dev/ttyATOP0

Serial type: RTU-BUFFERD

Baud rate: 9600

Unit-Id: 1

Timeout (ms): 1000

Reconnect timeout (ms): 2000

Log states changes:

Queue commands:

Queue delay (ms): 1

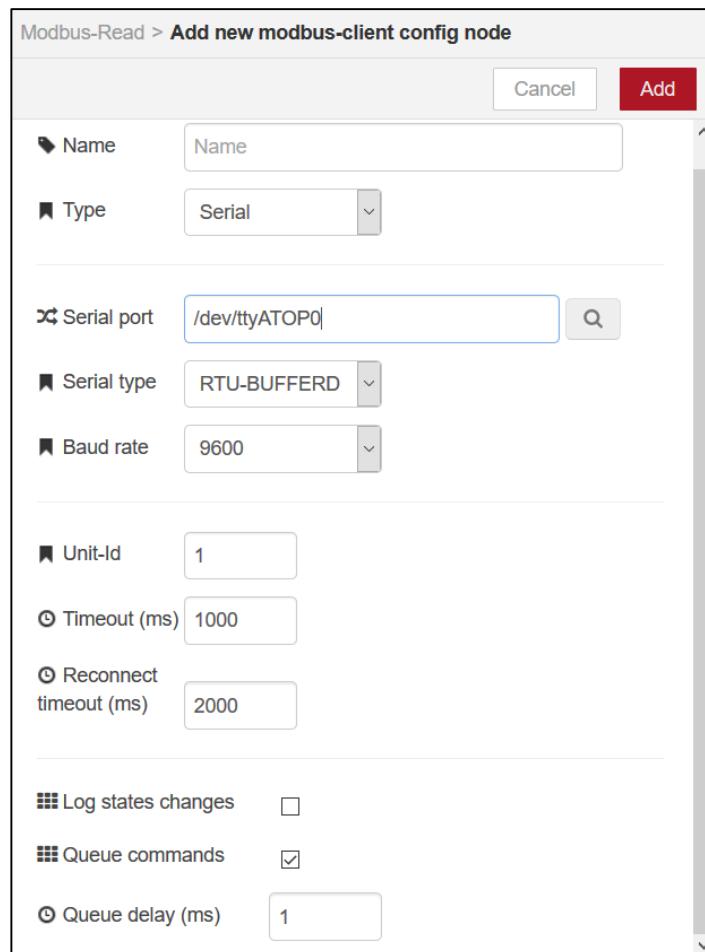


Figure 6.28 Modbus Settings – RTU/TCP/ASCII etc..

Please also set up the Modbus-response node as shown in Figure 6.23, defining only the name, for easy remembering what kind of polling it refers to.

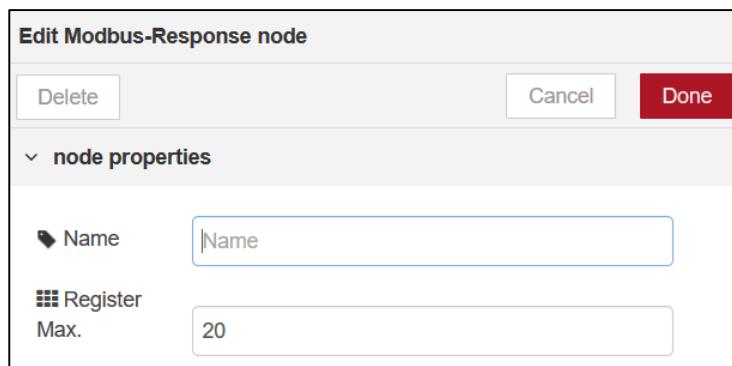


Figure 6.29 Modbus RTU/ASCII Read Node Settings

6.5.5 **Read data from Ethernet ports using Modbus TCP**

The way Modbus TCP should be configured is very similar to set up of Modbus RTU/ASCII. The only difference lies in the configuration of the "Server settings" located in Figure 6.28. Please select Type as "TCP". When this is done, the possible configuration parameters underneath change to what is shown in Figure 6.30. The parameters are explained as follows:

- **Type:** Defines whether use Modbus over Serial (RTU or ASCII) or over Ethernet (TCP). Select TCP
- **Host:** Defines the Modbus Server IP Address
- **Port (default 502):** Defines the TCP port used for Modbus TCP communication (usually 502)
- **Timeout, Reconnect timeout:** same as Chapter 6.5.4 above

The screenshot shows a configuration dialog for a Modbus TCP Read node. At the top right are three buttons: 'Delete' (gray), 'Cancel' (gray), and 'Update' (red). The main area contains the following settings:

- Name:** Name (text input)
- Type:** TCP (dropdown menu)
- Host:** 127.0.0.1 (text input)
- Port:** 502 (text input)
- Unit-Id:** 1 (text input)
- Timeout (ms):** 1500 (text input)
- Reconnect timeout (ms):** 1000 (text input)
- Log states changes:**
- Queue commands:**

Figure 6.30 Modbus TCP Read Node Settings

6.5.6 Write data using Modbus TCP/RTU/ASCII

The functioning of Modbus Write function is very similar to the Modbus read function previously explained in Chapters 6.5.4 and 6.5.5. Please follow the exact same proceedings for setting it up. The only difference lies in the FC (Function Code) drop-down menu, that will show different Modbus write function numbers. Figure 6.31 below shows the configuration options of Modbus-Write node.

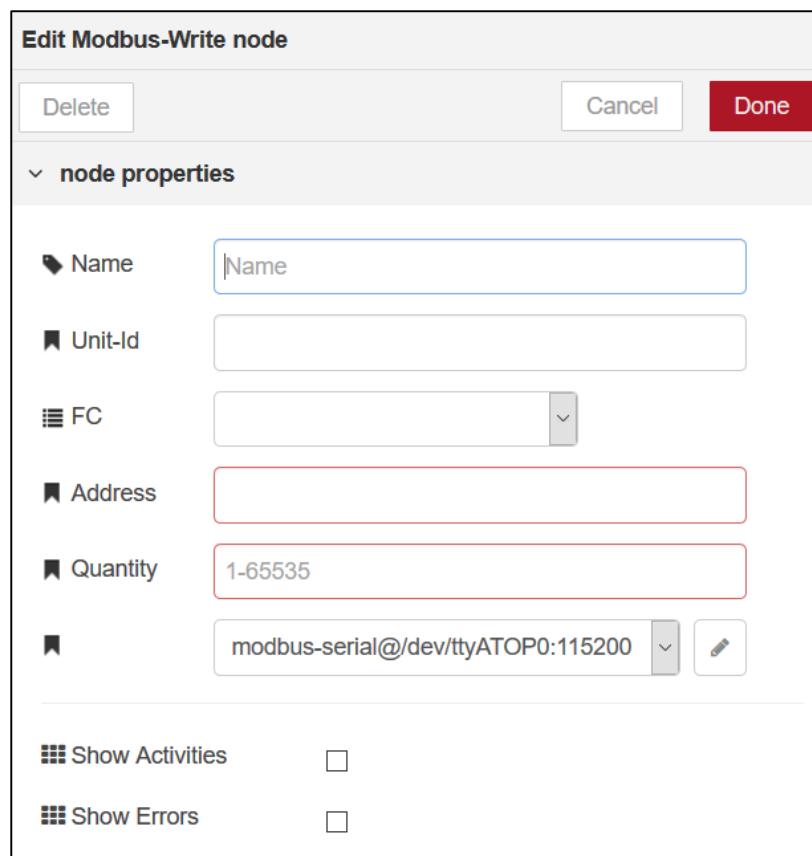


Figure 6 .31 Modbus TCP/RTU/ASCII Write Node Settings

6.5.7 Acting as a passive Modbus TCP/RTU/ASCII Slave/Server

SE59XX-SDK Node-RED can also act as a Modbus TCP Server. Please fill in the Modbus Server as below, in order to activate it. This will work in combination with other Modbus nodes

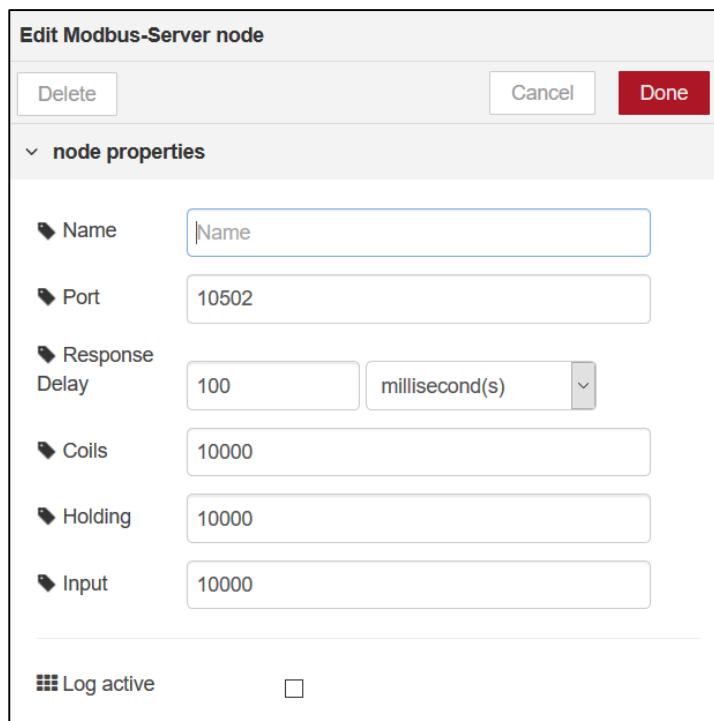


Figure 6.32 Modbus TCP Server Settings

6.5.8 Access other interfaces

The access of other interfaces on SE59XX-SDK Node-RED is using only the “exec” Node, shown in below **Error! Reference source not found..** The “exec” Node, basically runs a Linux binary program that is stored in the Filesystem or a Linux command. The “exec” Node is located in the “advanced” section of the Node palette on the left hand side of the screen. The detailed explanation on the way the “exec” Node works, is described in Chapter 6.5.1 above.

6.5.8.1 Buzzer

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

Table 6.4 Sample program for Buzzer

| Command | Description | Attributes |
|---------|--|------------|
| buzzer | A sample program to use the device's Buzzer. | on or off |

6.5.8.2 Turn the LEDs on or off

Table 6.5 Sample program for LEDs

| Command | Description | Attributes |
|---------|-------------|------------|
|---------|-------------|------------|

| | | |
|----------|--|-----------|
| alarmLed | A sample program to use the device's Alarm (RED) LED | on or off |
| runLed | A sample program to use the device's Run (GREEN) LED | on or off |

There is an LCM in SE5908 and SE5916. The sample program is available in the software/atop_application/utils/atop_sdk folder. Since the application is very strictly customer-dependent, there's no standard Binary application to manage the LCM. It is suggested to use the following sample programs as a reference and compile it. For this specific issue, a deep programming knowledge is needed. Users can refer to SE59XX-SDK user manual.

Table 6.6 Sample program for LCM

| File Name | Description |
|------------|---|
| lcd_test.c | A sample program to use the device's LCM. |

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

Table 6.7 Sample program for Reset Button

| File Name | Description |
|-----------|--|
| button.c | A sample program to use the device's reset button. |

6.5.8.3 Digital Inputs

There are 2 Digital inputs on SE5901B-IO:

Table 6.8 Sample program for Digital Input

| Command | Description | Attributes |
|---------|--|------------------------|
| di_test | A sample program to use the device's Digital Inputs. | 0 or 1 (read DI0/ DI1) |

6.5.8.4 Digital Outputs

There are 2 Digital Outputs on SE5901B-IO.

Table 6.9 Sample program for Digital Output

| Command | Description | Attributes |
|---------|---|---|
| do_test | A sample program to use the device's Digital Outputs. | 0 or 1 (write DI0/ DI1) 0 or 1 (on, off) |

7 Global Nodes list

Besides web nodes, ATOP SE59XX-SDK Node-RED has pre-installed the following Global nodes

```
+-- node-red@0.17.5
| +- basic-auth@1.1.0
| +- UNMET OPTIONAL DEPENDENCY bcrypt@~1.0.1
| +- bcryptjs@2.4.3
| +- body-parser@1.17.2
| | +- bytes@2.4.0
| | +- content-type@1.0.4
| | +- debug@2.6.7
| | | `-- ms@2.0.0
| | +- depd@1.1.1
| | +- http-errors@1.6.2
| | | +- depd@1.1.1 deduped
| | | +- inherits@2.0.3 deduped
| | | +- setprototypeof@1.0.3 deduped
| | | `-- statuses@1.3.1 deduped
| | +- iconv-lite@0.4.15
| | +- on-finished@2.3.0
| | | `-- ee-first@1.1.1
| | | +- qs@6.4.0
| | | +- raw-body@2.2.0 deduped
| | | `-- type-is@1.6.15
| | | +- media-typer@0.3.0 deduped
| | | `-- mime-types@2.1.17
| | |   `-- mime-db@1.30.0
| | +- cheerio@0.22.0
| | | +- css-select@1.2.0
| | | | +- boolbase@1.0.0
| | | | +- css-what@2.1.0
| | | | +- domutils@1.5.1
| | | | | +- dom-serializer@0.1.0 deduped
| | | | | `-- domelementtype@1.3.0 deduped
| | | | `-- nth-check@1.0.1
| | | |   `-- boolbase@1.0.0 deduped
| | | +- dom-serializer@0.1.0
| | | | +- domelementtype@1.1.3
| | | | | `-- entities@1.1.1 deduped
| | | | +- entities@1.1.1
| | | | +- htmlparser2@3.9.2
| | | | | +- domelementtype@1.3.0
| | | | | +- domhandler@2.4.1
| | | | | | `-- domelementtype@1.3.0 deduped
| | | | | +- domutils@1.5.1 deduped
| | | | | +- entities@1.1.1 deduped
| | | | | +- inherits@2.0.3 deduped
| | | | | `-- readable-stream@2.3.3 deduped
| | | +- lodash.assignin@4.2.0
| | | +- lodash.bind@4.2.1
| | | +- lodash.defaults@4.2.0
| | | +- lodash.filter@4.6.0
| | | +- lodash.flatten@4.4.0
| | | +- lodash.foreach@4.5.0
| | | +- lodash.map@4.6.0
| | | +- lodash.merge@4.6.0
| | | +- lodash.pick@4.4.0
| | | +- lodash.reduce@4.6.0
| | | +- lodash.reject@4.6.0
| | | `-- lodash.some@4.6.0
| | +- clone@2.1.1
```

```
|++ cookie@0.3.1
|++ cookie-parser@1.4.3
||+- cookie@0.3.1 deduped
||`-- cookie-signature@1.0.6
|+- cors@2.8.3
||+- object-assign@4.1.1
||`-- vary@1.1.2
|+- cron@1.2.1
||`-- moment-timezone@0.5.13
||`-- moment@2.18.1
|+- express@4.15.3
||+- accepts@1.3.4
|||+- mime-types@2.1.17 deduped
|||`-- negotiator@0.6.1
||+- array-flatten@1.1.1
||+- content-disposition@0.5.2
||+- content-type@1.0.4 deduped
||+- cookie@0.3.1 deduped
||+- cookie-signature@1.0.6 deduped
||+- debug@2.6.7
|||`-- ms@2.0.0 deduped
||+- depd@1.1.1 deduped
||+- encodeurl@1.0.1
||+- escape-html@1.0.3
||+- etag@1.8.1
||+- finalhandler@1.0.6
|||+- debug@2.6.9 deduped
|||+- encodeurl@1.0.1 deduped
|||+- escape-html@1.0.3 deduped
|||+- on-finished@2.3.0 deduped
|||+- parseurl@1.3.2 deduped
|||+- statuses@1.3.1 deduped
|||`-- unpipe@1.0.0 deduped
||+- fresh@0.5.0
||+- merge-descriptors@1.0.1
||+- methods@1.1.2
||+- on-finished@2.3.0 deduped
||+- parseurl@1.3.2
||+- path-to-regexp@0.1.7
||+- proxy-addr@1.1.5
|||+- forwarded@0.1.2
|||`-- ipaddr.js@1.4.0
||+- qs@6.4.0
||+- range-parser@1.2.0
||+- send@0.15.3
|||+- debug@2.6.7
|||`-- ms@2.0.0 deduped
|||+- depd@1.1.1 deduped
|||+- destroy@1.0.4
|||+- encodeurl@1.0.1 deduped
|||+- escape-html@1.0.3 deduped
|||+- etag@1.8.1 deduped
|||+- fresh@0.5.0 deduped
|||+- http-errors@1.6.2 deduped
|||+- mime@1.3.4
|||+- ms@2.0.0 deduped
|||+- on-finished@2.3.0 deduped
|||+- range-parser@1.2.0 deduped
|||`-- statuses@1.3.1 deduped
||+- serve-static@1.12.3
|||+- encodeurl@1.0.1 deduped
|||+- escape-html@1.0.3 deduped
|||+- parseurl@1.3.2 deduped
|||`-- send@0.15.3 deduped
||+- setprototypeof@1.0.3
||+- statuses@1.3.1
```

```
|| +- type-is@1.6.15 deduped
|| +- utils-merge@1.0.0
|| `-- vary@1.1.2 deduped
| +- express-session@1.15.2
| +- cookie@0.3.1 deduped
| +- cookie-signature@1.0.6 deduped
| +- crc@3.4.4
| +- debug@2.6.3
|| `-- ms@0.7.2
|| +- depd@1.1.1 deduped
|| +- on-headers@1.0.1 deduped
|| +- parseurl@1.3.2 deduped
|| +- uid-safe@2.1.5
|| | `-- random-bytes@1.0.0
|| | `-- utils-merge@1.0.0 deduped
|| +- follow-redirects@1.2.4
|| `-- debug@2.6.9
||   `-- ms@2.0.0 deduped
| +- fs-extra@1.0.0
| +- graceful-fs@4.1.11
| +- jsonfile@2.4.0
| | `-- graceful-fs@4.1.11 deduped
| | `-- klaw@1.3.1
| | `-- graceful-fs@4.1.11 deduped
| +- fs.notify@0.0.4
| +- async@0.1.22
| | `-- retry@0.6.1
| +- hash-sum@1.0.2
| +- i18next@1.10.6
| +- cookies@0.7.1
| | +- depd@1.1.1 deduped
| | `-- keygrip@1.0.2
| +- i18next-client@1.10.3
| | `-- json5@0.2.0
| +- is-utf8@0.2.1
| +- js-yaml@3.8.4
| | `-- argparse@1.0.9
| | | `-- sprintf-js@1.0.3
| | `-- esprima@3.1.3
| +- json-stringify-safe@5.0.1
| +- jsonata@1.2.6
| +- media-typer@0.3.0
| +- mqtt@2.9.0
| | +- commist@1.0.0
| | | +- leven@1.0.2
| | | | `-- minimist@1.2.0
| | | +- concat-stream@1.6.0
| | | +- inherits@2.0.3 deduped
| | | +- readable-stream@2.3.3 deduped
| | | `-- typedarray@0.0.6
| | +- end-of-stream@1.4.0
| | | `-- once@1.4.0
| | | `-- wrappy@1.0.2
| | +- help-me@1.1.0
| | | +- callback-stream@1.1.0
| | | | +- inherits@2.0.3 deduped
| | | | `-- readable-stream@2.3.3 deduped
| | | +- glob-stream@6.1.0
| | | +- extend@3.0.1 deduped
| | | +- glob@7.1.2
| | | | +- fs.realpath@1.0.0
| | | | | +- inflight@1.0.6
| | | | | | `-- once@1.4.0 deduped
| | | | | | `-- wrappy@1.0.2 deduped
| | | | | +- inherits@2.0.3 deduped
| | | | | +- minimatch@3.0.4
```

```
|||||`-- brace-expansion@1.1.8
|||||  +-- balanced-match@1.0.0
|||||  `-- concat-map@0.0.1
|||||  +- once@1.4.0 deduped
|||||  `-- path-is-absolute@1.0.1
|||||  +- glob-parent@3.1.0
|||||  +- is-glob@3.1.0
|||||  `-- is-extglob@2.1.1
|||||  `-- path dirname@1.0.2
|||||  +- is-negated-glob@1.0.0
|||||  +- ordered-read-streams@1.0.1
|||||  `-- readable-stream@2.3.3 deduped
|||||  +- pumpify@1.3.5
|||||  +- duplexify@3.5.1 deduped
|||||  +- inherits@2.0.3 deduped
|||||  `-- pump@1.0.2 deduped
|||||  +- readable-stream@2.3.3 deduped
|||||  +- remove-trailing-separator@1.1.0
|||||  +- to-absolute-glob@2.0.1
|||||  +- extend-shallow@2.0.1
|||||  `-- is-extendable@0.1.1
|||||  +- is-absolute@0.2.6
|||||  +- is-relative@0.2.1
|||||  `-- is-unc-path@0.1.2
|||||  `-- unc-path-regex@0.1.2
|||||  `-- is-windows@0.2.0
|||||  `-- is-negated-glob@1.0.0 deduped
||||`-- unique-stream@2.2.1
||||  +- json-stable-stringify@1.0.1
||||  |`-- jsonify@0.0.0
||||  `-- through2-filter@2.0.0
||||  +- through2@2.0.3 deduped
||||  `-- xtend@4.0.1 deduped
||||  +- through2@2.0.3
||||  +- readable-stream@2.3.3 deduped
||||  `-- xtend@4.0.1 deduped
||||  `-- xtend@4.0.1 deduped
|||+- inherits@2.0.3
|||+- minimist@1.2.0
|||+- mqtt-packet@5.4.0
|||+- bl@1.2.1
|||`-- readable-stream@2.3.3 deduped
|||+- inherits@2.0.3 deduped
|||+- process-nextick-args@1.0.7
|||`-- safe-buffer@5.1.1
|||+- pump@1.0.2
|||+- end-of-stream@1.4.0 deduped
|||`-- once@1.4.0 deduped
|||+- readable-stream@2.3.3
|||+- core-util-is@1.0.2
|||+- inherits@2.0.3 deduped
|||+- isarray@1.0.0
|||+- process-nextick-args@1.0.7 deduped
|||+- safe-buffer@5.1.1 deduped
|||+- string_decoder@1.0.3
|||`-- safe-buffer@5.1.1 deduped
|||`-- util-deprecate@1.0.2
|||+- reinterval@1.1.0
|||+- split2@2.2.0
|||`-- through2@2.0.3 deduped
|||+- websocket-stream@5.0.1
|||+- duplexify@3.5.1
|||+- end-of-stream@1.4.0 deduped
|||+- inherits@2.0.3 deduped
|||+- readable-stream@2.3.3 deduped
|||`-- stream-shift@1.0.0
```

```
||||-- inherits@2.0.3 deduped
||||-- readable-stream@2.3.3 deduped
||||-- safe-buffer@5.1.1 deduped
|||+-- ws@3.2.0
|||+-- async-limiter@1.0.0
|||+-- safe-buffer@5.1.1 deduped
|||`- ultron@1.1.0
||`- xtend@4.0.1 deduped
||`- xtend@4.0.1
|+-- multer@1.3.0
||+-- append-field@0.1.0
||+-- busboy@0.2.14
||+-- dicer@0.2.5
|||+-- readable-stream@1.1.14
|||+-- core-util-is@1.0.2 deduped
|||+-- inherits@2.0.3 deduped
|||+-- isarray@0.0.1
|||`- string_decoder@0.10.31
|||- streamsearch@0.1.2
||`- readable-stream@1.1.14
||+-- core-util-is@1.0.2 deduped
||+-- inherits@2.0.3 deduped
||+-- isarray@0.0.1
||`- string_decoder@0.10.31
||+-- concat-stream@1.6.0 deduped
||+-- mkdirp@0.5.1
||`- minimist@0.0.8
||+-- object-assign@3.0.0
||+-- on-finished@2.3.0 deduped
||+-- type-is@1.6.15 deduped
||`- xtend@4.0.1 deduped
|+-- mustache@2.3.0
|+-- node-red-node-email@0.1.24
||+-- imap@0.8.19
||+-- readable-stream@1.1.14
|||+-- core-util-is@1.0.2 deduped
|||+-- inherits@2.0.3 deduped
|||+-- isarray@0.0.1
|||`- string_decoder@0.10.31
|||- utf7@1.0.2
||`- semver@5.3.0 deduped
||+-- mailparser@0.6.2
||+-- encoding@0.1.12
|||- iconv-lite@0.4.15 deduped
||+-- mime@1.3.4 deduped
||+-- mimelib@0.3.1
|||+-- addressparser@1.0.1
|||`- encoding@0.1.12 deduped
||`- uue@3.1.0
||`- extend@3.0.1 deduped
||+-- nodemailer@1.11.0
||+-- libmime@1.2.0
|||+-- iconv-lite@0.4.15 deduped
|||+-- libbase64@0.1.0
|||`- libqp@1.1.0
||+-- mailcomposer@2.1.0
||+-- buildmail@2.0.0
|||+-- addressparser@0.3.2
|||+-- libbase64@0.1.0 deduped
|||+-- libmime@1.2.0 deduped
|||+-- libqp@1.1.0 deduped
|||`- needle@0.10.0
|||+-- debug@2.6.9 deduped
|||`- iconv-lite@0.4.15 deduped
||`- libmime@1.2.0 deduped
||+-- needle@0.11.0
```

```
||||+- debug@2.6.9 deduped
||||`- iconv-lite@0.4.15 deduped
|||+- nodemailer-direct-transport@1.1.0
|||`- smtp-connection@1.3.8
|||`- nodemailer-smtp-transport@1.1.0
|||  +- clone@1.0.2
|||  +- nodemailer-wellknown@0.1.10
|||  `- smtp-connection@1.3.8 deduped
||`- poplib@0.1.7
||  +- optimist@0.6.1
||  +- minimist@0.0.8 deduped
||  `- wordwrap@0.0.3
|+- node-red-node-feedparser@0.1.8
|+- feedparser@1.1.3
||+- addressparser@0.1.3
||+- array-indexofobject@0.0.1
||+- readable-stream@1.0.34
|||+- core-util-is@1.0.2 deduped
|||+- inherits@2.0.3 deduped
|||+- isarray@0.0.1
|||`- string_decoder@0.10.31
||`- sax@0.6.1 deduped
|- request@2.74.0
|  +- aws-sign2@0.6.0
|  +- aws4@1.6.0
|  +- bl@1.1.2
|  |`- readable-stream@2.0.6
|  |  +- core-util-is@1.0.2 deduped
|  |  +- inherits@2.0.3 deduped
|  |  +- isarray@1.0.0 deduped
|  |  +- process-nextick-args@1.0.7 deduped
|  |  +- string_decoder@0.10.31
|  |  `-- util-deprecate@1.0.2 deduped
|  +- caseless@0.11.0
|  +- combined-stream@1.0.5
|  |`- delayed-stream@1.0.0
|  |  +- extend@3.0.1
|  |  +- forever-agent@0.6.1
|  |  +- form-data@1.0.1
|  |  +- async@2.5.0
|  |  ||`- lodash@4.17.4 deduped
|  |  +- combined-stream@1.0.5 deduped
|  |  |`- mime-types@2.1.17 deduped
|  |  +- har-validator@2.0.6
|  |  +- chalk@1.1.3
|  |  |`- ansi-styles@2.2.1
|  |  |  +- escape-string-regexp@1.0.5
|  |  |  +- has-ansi@2.0.0
|  |  |  ||`- ansi-regex@2.1.1
|  |  |  ||  +- strip-ansi@3.0.1
|  |  |  ||  |`- ansi-regex@2.1.1 deduped
|  |  |  |`- supports-color@2.0.0
|  |  |  +- commander@2.11.0
|  |  |  +- is-my-json-valid@2.16.1
|  |  |  +- generate-function@2.0.0
|  |  |  +- generate-object-property@1.2.0
|  |  |  ||`- is-property@1.0.2
|  |  |  +- jsonpointer@4.0.1
|  |  |  ||`- xtend@4.0.1 deduped
|  |  |  |`- pinkie-promise@2.0.1
|  |  |  |`- pinkie@2.0.4
|  |  |  +- hawk@3.1.3
|  |  |  +- boom@2.10.1
|  |  |  |`- hoek@2.16.3 deduped
|  |  |  +- cryptiles@2.0.5
|  |  |  |`- boom@2.10.1 deduped
```

```
|| | +- hoek@2.16.3
|| | `-- sntp@1.0.9
|| |   '-- hoek@2.16.3 deduped
|| |   +-- http-signature@1.1.1
|| |   +-- assert-plus@0.2.0
|| |   +-- jsprim@1.4.1
|| |   || +-- assert-plus@1.0.0 deduped
|| |   || +-- extsprintf@1.3.0
|| |   || +-- json-schema@0.2.3
|| |   || `-- verror@1.10.0
|| |   ||   '-- assert-plus@1.0.0 deduped
|| |   ||   '-- core-util-is@1.0.2 deduped
|| |   ||   '-- extsprintf@1.3.0 deduped
|| |   `-- sshpk@1.13.1
|| |     +- asn1@0.2.3
|| |     +-- assert-plus@1.0.0 deduped
|| |     +-- bcrypt-pbkdf@1.0.1
|| |     | '-- tweetnacl@0.14.5 deduped
|| |     +-- dashdash@1.14.1
|| |     | '-- assert-plus@1.0.0 deduped
|| |     +-- ecc-jisbn@0.1.1
|| |     | '-- isbn@0.1.1 deduped
|| |     +-- getpass@0.1.7
|| |     | '-- assert-plus@1.0.0 deduped
|| |     +-- jisbn@0.1.1
|| |     | '-- tweetnacl@0.14.5
|| |   +-- is-typedarray@1.0.0
|| |   +-- isstream@0.1.2
|| |   +-- json-stringify-safe@5.0.1 deduped
|| |   +-- mime-types@2.1.17 deduped
|| |   +-- node-uuid@1.4.8
|| |   +-- oauth-sign@0.8.2
|| |   +-- qs@6.2.3
|| |   +-- stringstream@0.0.5
|| |   +-- tough-cookie@2.3.3
|| |   | '-- punycode@1.4.1
|| |   | '-- tunnel-agent@0.4.3
|| |   +-- node-red-node-rbe@0.1.13
|| |   +-- node-red-node-twitter@0.1.11
|| |   || +-- oauth@0.9.14
|| |   || +-- request@2.83.0
|| |   ||   +-- aws-sign2@0.7.0
|| |   ||   +-- aws4@1.6.0 deduped
|| |   ||   +-- caseless@0.12.0
|| |   ||   +-- combined-stream@1.0.5 deduped
|| |   ||   +-- extend@3.0.1 deduped
|| |   ||   +-- forever-agent@0.6.1 deduped
|| |   ||   +-- form-data@2.3.1
|| |   ||   +-- asynckit@0.4.0
|| |   ||   || +-- combined-stream@1.0.5 deduped
|| |   ||   ||   '-- mime-types@2.1.17 deduped
|| |   ||   +-- har-validator@5.0.3
|| |   ||   +-- ajv@5.2.3
|| |   ||   || +-- co@4.6.0
|| |   ||   || +-- fast-deep-equal@1.0.0
|| |   ||   || +-- json-schema-traverse@0.3.1
|| |   ||   ||   '-- json-stable-stringify@1.0.1 deduped
|| |   ||   ||   '-- har-schema@2.0.0
|| |   ||   +-- hawk@6.0.2
|| |   ||   +-- boom@4.3.1
|| |   ||   | '-- hoek@4.2.0 deduped
|| |   ||   || +-- cryptiles@3.1.2
|| |   ||   ||   '-- boom@5.2.0
|| |   ||   ||   '-- hoek@4.2.0 deduped
|| |   ||   +-- hoek@4.2.0
|| |   ||   '-- sntp@2.0.2
```

```
|||| `-- hoek@4.2.0 deduped
|||+-- http-signature@1.2.0
|||+-- assert-plus@1.0.0
|||+- jsprim@1.4.1 deduped
|||`-- sshpk@1.13.1 deduped
|||+-- is-typedarray@1.0.0 deduped
|||+-- isstream@0.1.2 deduped
|||+-- json-stringify-safe@5.0.1 deduped
|||+-- mime-types@2.1.17 deduped
|||+-- oauth-sign@0.8.2 deduped
|||+-- performance-now@2.1.0
|||+-- qs@6.5.1
|||+-- safe-buffer@5.1.1 deduped
|||+-- stringstream@0.0.5 deduped
|||+-- tough-cookie@2.3.3 deduped
|||+-- tunnel-agent@0.6.0
|||`-- safe-buffer@5.1.1 deduped
||`-- uuid@3.1.0
||`-- twitter-ng@0.6.2
||`-- oauth@0.9.14 deduped
|+-- nopt@3.0.6
|`-- abbrev@1.1.1
|+-- oauth2orize@1.8.0
|+-- debug@2.6.9 deduped
|+-- uid2@0.0.3
|`-- utils-merge@1.0.0 deduped
|+-- on-headers@1.0.1
|+-- passport@0.3.2
|+-- passport-strategy@1.0.0
|`-- pause@0.0.1
|+-- passport-http-bearer@1.0.1
|`-- passport-strategy@1.0.0 deduped
|+-- passport-oauth2-client-password@0.1.2
|`-- passport-strategy@1.0.0 deduped
|+-- raw-body@2.2.0
|+-- bytes@2.4.0 deduped
|+-- iconv-lite@0.4.15 deduped
|`-- unpipe@1.0.0
|+-- semver@5.3.0
|+-- sentiment@2.1.0
|+-- uglify-js@3.0.20
|+-- commander@2.9.0
|||`-- graceful-readlink@1.0.1
|`-- source-map@0.5.7
|+-- when@3.7.8
|+-- ws@1.1.1
|+-- options@0.0.6
|`-- ultron@1.0.2
|`-- xml2js@0.4.17
|+-- sax@0.6.1
|`-- xmlbuilder@4.2.1
|`-- lodash@4.17.4
`-- npm@5.4.2
+-- abbrev@1.1.0
+-- ansi-regex@3.0.0
+-- ansicolors@0.3.2
+-- ansistyles@0.1.3
+-- aproba@1.1.2
+-- archy@1.0.0
+-- bluebird@3.5.0
+-- cacache@9.2.9
|+-- bluebird@3.5.0 deduped
|+-- chownr@1.0.1 deduped
|+-- glob@7.1.2 deduped
|+-- graceful-fs@4.1.11 deduped
|+-- lru-cache@4.1.1
```

```
||-- pseudomap@1.0.2
||`-- yallist@2.1.2
|+- mississippi@1.3.0 deduped
|+- mkdirp@0.5.1 deduped
|+- move-concurrently@1.0.1 deduped
|+- promise-inflight@1.0.1 deduped
|+- rimraf@2.6.1 deduped
|+- ssri@4.1.6 deduped
|+- unique-filename@1.1.0 deduped
|`-- y18n@3.2.1
+-- call-limit@1.1.0
+-- chownr@1.0.1
+-- cmd-shim@2.0.2
|+- graceful-fs@4.1.11 deduped
|`-- mkdirp@0.5.1 deduped
+-- columnify@1.5.4
|+- strip-ansi@3.0.1
||`-- ansi-regex@2.1.1
|`-- wcwidth@1.0.1
| `-- defaults@1.0.3
| `-- clone@1.0.2
+-- config-chain@1.1.11
|+- ini@1.3.4 deduped
|`-- proto-list@1.2.4
+-- debuglog@1.0.1
+-- detect-indent@5.0.0
+-- dezalgo@1.0.3
|+- asap@2.0.5
|`-- wrappy@1.0.2 deduped
+-- editor@1.0.0
+-- fs-vacuum@1.2.10
|+- graceful-fs@4.1.11 deduped
|+- path-is-inside@1.0.2 deduped
|`-- rimraf@2.6.1 deduped
+-- fs-write-stream-atomic@1.0.10
|+- graceful-fs@4.1.11 deduped
|+- iferr@0.1.5 deduped
|+- imurmurhash@0.1.4 deduped
|`-- readable-stream@2.3.3 deduped
+-- glob@7.1.2
|+- fs.realpath@1.0.0
|+- inflight@1.0.6 deduped
|+- inherits@2.0.3 deduped
|+- minimatch@3.0.4
||`-- brace-expansion@1.1.8
|| `-- balanced-match@1.0.0
|| `-- concat-map@0.0.1
|+- once@1.4.0 deduped
|`-- path-is-absolute@1.0.1
+- graceful-fs@4.1.11
+- has-unicode@2.0.1
+- hosted-git-info@2.5.0
+- iferr@0.1.5
+- imurmurhash@0.1.4
+- inflight@1.0.6
|+- once@1.4.0 deduped
|`-- wrappy@1.0.2 deduped
+- inherits@2.0.3
+- ini@1.3.4
+- init-package-json@1.10.1
|+- glob@7.1.2 deduped
|+- npm-package-arg@5.1.2 deduped
|+- promzard@0.3.0
||`-- read@1.0.7 deduped
|+- read@1.0.7 deduped
|+- read-package-json@2.0.12 deduped
```

```
| +- semver@5.4.1 deduped
| +- validate-npm-package-license@3.0.1 deduped
| `-- validate-npm-package-name@3.0.0 deduped
+-- JSONStream@1.3.1
| +- jsonparse@1.3.1
| `-- through@2.3.8
+- lazy-property@1.0.0
+- libnpn@9.6.0
| +- dotenv@4.0.0
| +- npm-package-arg@5.1.2 deduped
| +- rimraf@2.6.1 deduped
| +- safe-buffer@5.1.1 deduped
| +- update-notifier@2.2.0 deduped
| +- which@1.3.0 deduped
| +- y18n@3.2.1
| `-- yargs@8.0.2
|   +- camelcase@4.1.0
|   +- cliui@3.2.0
|     +- string-width@1.0.2
|       +- code-point-at@1.1.0
|       +- is-fullwidth-code-point@1.0.0
|         +- number-is-nan@1.0.1
|         `-- strip-ansi@3.0.1 deduped
|       +- strip-ansi@3.0.1
|     `-- ansi-regex@2.1.1
|   `-- wrap-ansi@2.1.0
|     +- string-width@1.0.2 deduped
|     `-- strip-ansi@3.0.1 deduped
|   +- decamelize@1.2.0
|   +- get-caller-file@1.0.2
|   +- os-locale@2.1.0
|   +- execa@0.7.0
|     +- cross-spawn@5.1.0
|       +- lru-cache@4.1.1 deduped
|       +- shebang-command@1.2.0
|       | `-- shebang-regex@1.0.0
|       `-- which@1.3.0 deduped
|     +- get-stream@3.0.0
|     +- is-stream@1.1.0
|     +- npm-run-path@2.0.2
|     | `-- path-key@2.0.1
|     | +- p-finally@1.0.0
|     | +- signal-exit@3.0.2
|     | `-- strip-eof@1.0.0
|     +- lcid@1.0.0
|     `-- invert-kv@1.0.0
|   `-- mem@1.1.0
|     `-- mimic-fn@1.1.0
|   +- read-pkg-up@2.0.0
|   +- find-up@2.1.0
|     `-- locate-path@2.0.0
|       +- p-locate@2.0.0
|       | `-- p-limit@1.1.0
|       `-- path-exists@3.0.0
|   `-- read-pkg@2.0.0
|     +- load-json-file@2.0.0
|     | +- graceful-fs@4.1.11 deduped
|     |   +- parse-json@2.2.0
|     |   | `-- error-ex@1.3.1
|     |   |   +- is-arrayish@0.2.1
|     |   |   +- pify@2.3.0
|     |   |     `-- strip-bom@3.0.0
|     |   +- normalize-package-data@2.4.0 deduped
|     |     `-- path-type@2.0.0
|     |       `-- pify@2.3.0
|   +- require-directory@2.1.1
```

```
| +- require-main-filename@1.0.1
| +- set-blocking@2.0.0
| +- string-width@2.1.1
| | +- is-fullwidth-code-point@2.0.0
| | `-- strip-ansi@4.0.0 deduped
| +- which-module@2.0.0
| +- y18n@3.2.1 deduped
| `-- yargs-parser@7.0.0
|   `-- camelcase@4.1.0 deduped
+- lockfile@1.0.3
+- lodash._baseindexof@3.1.0
+- lodash._baseuniq@4.6.0
| +- lodash._createset@4.0.3
| `-- lodash._root@3.0.1
+- lodash._bindcallback@3.0.1
+- lodash._cacheindexof@3.0.2
+- lodash._createcache@3.1.2
| `-- lodash._getnative@3.9.1 deduped
+- lodash._getnative@3.9.1
+- lodash.clonedeep@4.5.0
+- lodash.restparam@3.6.1
+- lodash.union@4.6.0
+- lodash.uniq@4.5.0
+- lodash.without@4.4.0
+- lru-cache@4.1.1
| +- pseudomap@1.0.2
| `-- yallist@2.1.2
+- meant@1.0.0
+- mississippi@1.3.0
| +- concat-stream@1.6.0
| | +- inherits@2.0.3 deduped
| | +- readable-stream@2.3.3 deduped
| | `-- typedarray@0.0.6
| +- duplexify@3.5.0
| | +- end-of-stream@1.0.0
| |   `-- once@1.3.3
| |     `-- wrappy@1.0.2 deduped
| | +- inherits@2.0.3 deduped
| | +- readable-stream@2.3.3 deduped
| | `-- stream-shift@1.0.0
| +- end-of-stream@1.4.0
| | -- once@1.4.0 deduped
| | +- flush-write-stream@1.0.2
| | | +- inherits@2.0.3 deduped
| | | -- readable-stream@2.3.3 deduped
| | | +- from2@2.3.0
| | | | +- inherits@2.0.3 deduped
| | | | -- readable-stream@2.3.3 deduped
| | | +- parallel-transform@1.1.0
| | | | +- cyclist@0.2.2
| | | | | +- inherits@2.0.3 deduped
| | | | | -- readable-stream@2.3.3 deduped
| | | | +- pump@1.0.2
| | | | | +- end-of-stream@1.4.0 deduped
| | | | | -- once@1.4.0 deduped
| | | | | +- pumpify@1.3.5
| | | | | | +- duplexify@3.5.0 deduped
| | | | | | +- inherits@2.0.3 deduped
| | | | | | -- pump@1.0.2 deduped
| | | | | +- stream-each@1.2.0
| | | | | | +- end-of-stream@1.4.0 deduped
| | | | | | -- stream-shift@1.0.0
| | | | | | `-- through2@2.0.3
| | | | | | | +- readable-stream@2.3.3 deduped
| | | | | | | `-- xtend@4.0.1
| | | | | | | `-- mkdirp@0.5.1
```

```
|`-- minimalist@0.0.8
+-- move-concurrently@1.0.1
|`-- aproba@1.1.2 deduped
|`-- copy-concurrently@1.0.3
|`-- aproba@1.1.2 deduped
||`-- fs-write-stream-atomic@1.0.10 deduped
||`-- iferr@0.1.5 deduped
||`-- mkdirp@0.5.1 deduped
||`-- rimraf@2.6.1 deduped
||`-- run-queue@1.0.3 deduped
|`-- fs-write-stream-atomic@1.0.10 deduped
|`-- mkdirp@0.5.1 deduped
|`-- rimraf@2.6.1 deduped
|`-- run-queue@1.0.3
|`-- aproba@1.1.2 deduped
+-- node-gyp@3.6.2
|`-- fstream@1.0.11
||`-- graceful-fs@4.1.11 deduped
||`-- inherits@2.0.3 deduped
||`-- mkdirp@0.5.1 deduped
||`-- rimraf@2.6.1 deduped
|`-- glob@7.1.2 deduped
|`-- graceful-fs@4.1.11 deduped
|`-- minimatch@3.0.4
||`-- brace-expansion@1.1.8
||`-- balanced-match@1.0.0
||`-- concat-map@0.0.1
|`-- mkdirp@0.5.1 deduped
|`-- nopt@3.0.6
|`-- abbrev@1.1.0 deduped
|`-- npmllog@4.1.2 deduped
|`-- osenv@0.1.4 deduped
|`-- request@2.81.0 deduped
|`-- rimraf@2.6.1 deduped
|`-- semver@5.3.0
|`-- tar@2.2.1
||`-- block-stream@0.0.9
|||`-- inherits@2.0.3 deduped
||`-- fstream@1.0.11 deduped
|||`-- inherits@2.0.3 deduped
|`-- which@1.3.0 deduped
+-- nopt@4.0.1
|`-- abbrev@1.1.0 deduped
|`-- osenv@0.1.4 deduped
+-- normalize-package-data@2.4.0
|`-- hosted-git-info@2.5.0 deduped
|`-- is-builtin-module@1.0.0
||`-- builtin-modules@1.1.1
|`-- semver@5.4.1 deduped
|`-- validate-npm-package-license@3.0.1 deduped
+-- npm-cache-filename@1.0.2
+-- npm-install-checks@3.0.0
|`-- semver@5.4.1 deduped
+-- npm-lifecycle@1.0.2
|`-- graceful-fs@4.1.11 deduped
|`-- slide@1.1.6 deduped
|`-- uid-number@0.0.6 deduped
|`-- umask@1.1.0 deduped
|`-- which@1.3.0 deduped
+-- npm-package-arg@5.1.2
|`-- hosted-git-info@2.5.0 deduped
|`-- osenv@0.1.4 deduped
|`-- semver@5.4.1 deduped
|`-- validate-npm-package-name@3.0.0 deduped
+-- npm-packlist@1.1.8
|`-- ignore-walk@3.0.0
```

```
||`-- minimatch@3.0.4
||  `-- brace-expansion@1.1.8
||    +- balanced-match@1.0.0
||    `-- concat-map@0.0.1
|`-- npm-bundled@1.0.3
+-- npm-registry-client@8.4.0
|++-- concat-stream@1.6.0
||`-- inherits@2.0.3 deduped
||`-- readable-stream@2.3.3 deduped
||`-- typedarray@0.0.6
|++-- graceful-fs@4.1.11 deduped
|++-- normalize-package-data@2.4.0 deduped
|++-- npm-package-arg@5.1.2 deduped
|++-- npmlog@4.1.2 deduped
|++-- once@1.4.0 deduped
|++-- request@2.81.0 deduped
|++-- retry@0.10.1 deduped
|++-- semver@5.4.1 deduped
|++-- slide@1.1.6 deduped
|`-- ssri@4.1.6 deduped
+- npm-user-validate@1.0.0
+- npmlog@4.1.2
|++-- are-we-there-yet@1.1.4
||`-- delegates@1.0.0
||`-- readable-stream@2.3.3 deduped
|++-- console-control-strings@1.1.0
|++-- gauge@2.7.4
||`-- aproba@1.1.2 deduped
||`-- console-control-strings@1.1.0 deduped
||`-- has-unicode@2.0.1 deduped
||`-- object-assign@4.1.1
||`-- signal-exit@3.0.2
||`-- string-width@1.0.2
||`-- code-point-at@1.1.0
||`-- is-fullwidth-code-point@1.0.0
|||`-- number-is-nan@1.0.1
||`-- strip-ansi@3.0.1 deduped
||`-- strip-ansi@3.0.1
||`-- ansi-regex@2.1.1
||`-- wide-align@1.1.2
||`-- string-width@1.0.2 deduped
|`-- set-blocking@2.0.0
+- once@1.4.0
|`-- wrappy@1.0.2 deduped
+- opener@1.4.3
+- osenv@0.1.4
|++-- os-homedir@1.0.2
|`-- os-tmpdir@1.0.2
+- pacote@6.0.2
|++-- bluebird@3.5.0 deduped
|++-- cacache@9.2.9 deduped
|++-- glob@7.1.2 deduped
|++-- lru-cache@4.1.1 deduped
|++-- make-fetch-happen@2.5.0
||`-- agentkeepalive@3.3.0
||`-- humanize-ms@1.2.1
||`-- ms@2.0.0
||`-- cacache@9.2.9 deduped
||`-- http-cache-semantics@3.7.3
||`-- http-proxy-agent@2.0.0
||`-- agent-base@4.1.1
|||`-- es6-promisify@5.0.0
|||`-- es6-promise@4.1.1
||`-- debug@2.6.8
||`-- ms@2.0.0
||`-- https-proxy-agent@2.1.0
```

```
||| +- agent-base@4.1.1
|||| `-- es6-promisify@5.0.0
||||   '-- es6-promise@4.1.1
||| `-- debug@2.6.8
|||   '-- ms@2.0.0
||+- lru-cache@4.1.1 deduped
||+- mississippi@1.3.0 deduped
||+- node-fetch-npm@2.0.2
||+- encoding@0.1.12
||| `-- iconv-lite@0.4.18
||| +- json-parse-better-errors@1.0.1
|||   '-- safe-buffer@5.1.1 deduped
||| +- promise-retry@1.1.1 deduped
||| +- socks-proxy-agent@3.0.0
||| +- agent-base@4.1.1
||| | `-- es6-promisify@5.0.0
||| |   '-- es6-promise@4.1.1
||| |   '-- socks@1.1.10
||| |   +- ip@1.1.5
||| |   '-- smart-buffer@1.1.15
||| |   '-- ssri@4.1.6 deduped
||| |   +- minimatch@3.0.4
||| |   '-- brace-expansion@1.1.8
||| |   +- balanced-match@1.0.0
||| |   '-- concat-map@0.0.1
||| |   +- mississippi@1.3.0 deduped
||| |   +- normalize-package-data@2.4.0 deduped
||| |   +- npm-package-arg@5.1.2 deduped
||| |   +- npm-packlist@1.1.8 deduped
||| |   +- npm-pick-manifest@1.0.4
||| |   +- npm-package-arg@5.1.2 deduped
||| |   '-- semver@5.4.1 deduped
||| |   +- osenv@0.1.4 deduped
||| |   +- promise-inflight@1.0.1 deduped
||| |   +- promise-retry@1.1.1
||| |   +- err-code@1.1.2
||| |   '-- retry@0.10.1 deduped
||| |   +- protoduck@4.0.0
||| |   '-- genfun@4.0.1
||| |   +- safe-buffer@5.1.1 deduped
||| |   +- semver@5.4.1 deduped
||| |   +- ssri@4.1.6 deduped
||| |   +- tar@4.0.1 deduped
||| |   +- unique-filename@1.1.0 deduped
||| |   '-- which@1.3.0 deduped
||| +- path-is-inside@1.0.2
||| +- promise-inflight@1.0.1
||| +- read@1.0.7
|||   '-- mute-stream@0.0.7
||| +- read-cmd-shim@1.0.1
|||   '-- graceful-fs@4.1.11 deduped
||| +- read-installed@4.0.3
|||   +- debuglog@1.0.1 deduped
|||   +- graceful-fs@4.1.11 deduped
|||   +- read-package-json@2.0.12 deduped
|||   +- readdir-scoped-modules@1.0.2 deduped
|||   +- semver@5.4.1 deduped
|||   +- slide@1.1.6 deduped
|||   '-- util-extend@1.0.3
||| +- read-package-json@2.0.12
|||   +- glob@7.1.2 deduped
|||   +- graceful-fs@4.1.11 deduped
|||   +- json-parse-better-errors@1.0.1
|||   +- normalize-package-data@2.4.0 deduped
|||   '-- slash@1.0.0
||| +- read-package-tree@5.1.6
```

```
| +- debuglog@1.0.1 deduped
| +- dezalgo@1.0.3 deduped
| +- once@1.4.0 deduped
| +- read-package-json@2.0.12 deduped
| `-- readdir-scoped-modules@1.0.2 deduped
+-- readable-stream@2.3.3
| +- core-util-is@1.0.2
| +- inherits@2.0.3 deduped
| +- isarray@1.0.0
| +- process-nextick-args@1.0.7
| +- safe-buffer@5.1.1 deduped
| +- string_decoder@1.0.3
| | `-- safe-buffer@5.1.1 deduped
| `-- util-deprecate@1.0.2
+-- readdir-scoped-modules@1.0.2
| +- debuglog@1.0.1 deduped
| +- dezalgo@1.0.3 deduped
| +- graceful-fs@4.1.11 deduped
| `-- once@1.4.0 deduped
+-- request@2.81.0
| +- aws-sign2@0.6.0
| +- aws4@1.6.0
| +- caseless@0.12.0
| +- combined-stream@1.0.5
| | `-- delayed-stream@1.0.0
| +- extend@3.0.1
| +- forever-agent@0.6.1
| +- form-data@2.1.4
| | +- asynckit@0.4.0
| | +- combined-stream@1.0.5 deduped
| | `-- mime-types@2.1.15 deduped
| +- har-validator@4.2.1
| | +- ajv@4.11.8
| | | +- co@4.6.0
| | | `-- json-stable-stringify@1.0.1
| | | | `-- jsonify@0.0.0
| | | `-- har-schema@1.0.5
| | +- hawk@3.1.3
| | +- boom@2.10.1
| | | `-- hoek@2.16.3 deduped
| | +- cryptiles@2.0.5
| | | `-- boom@2.10.1 deduped
| | +- hoek@2.16.3
| | `-- sntp@1.0.9
| | | `-- hoek@2.16.3 deduped
| +- http-signature@1.1.1
| | +- assert-plus@0.2.0
| | +- jsprim@1.4.0
| | | +- assert-plus@1.0.0
| | | +- extsprintf@1.0.2
| | | +- json-schema@0.2.3
| | | | `-- verror@1.3.6
| | | | `-- extsprintf@1.0.2 deduped
| | | `-- sshpk@1.13.1
| | | +- asn1@0.2.3
| | | +- assert-plus@1.0.0
| | | +- bcrypt-pbkdf@1.0.1
| | | | `-- tweetnacl@0.14.5 deduped
| | | +- dashdash@1.14.1
| | | | `-- assert-plus@1.0.0 deduped
| | | +- ecc-jisbn@0.1.1
| | | | `-- isbn@0.1.1 deduped
| | | +- getpass@0.1.7
| | | | `-- assert-plus@1.0.0 deduped
| | | +- isbn@0.1.1
| | | | `-- tweetnacl@0.14.5
```

```
| +- is-typedarray@1.0.0
| +- isstream@0.1.2
| +- json-stringify-safe@5.0.1
| +- mime-types@2.1.15
| | `-- mime-db@1.27.0
| +- oauth-sign@0.8.2
| +- performance-now@0.2.0
| +- qs@6.4.0
| +- safe-buffer@5.1.1 deduped
| +- stringstream@0.0.5
| +- tough-cookie@2.3.2
| | `-- punycode@1.4.1
| +- tunnel-agent@0.6.0
| | `-- safe-buffer@5.1.1 deduped
| `-- uuid@3.1.0 deduped
+- retry@0.10.1
+- rimraf@2.6.1
| `-- glob@7.1.2 deduped
+- safe-buffer@5.1.1
+- semver@5.4.1
+- sha@2.0.1
| +- graceful-fs@4.1.11 deduped
| `-- readable-stream@2.3.3 deduped
+- slide@1.1.6
+- sorted-object@2.0.1
+- sorted-union-stream@2.1.3
| +- from2@1.3.0
| | `-- inherits@2.0.3 deduped
| | `-- readable-stream@1.1.14
| |   +- core-util-is@1.0.2
| |   +- inherits@2.0.3 deduped
| |   +- isarray@0.0.1
| |   `-- string_decoder@0.10.31
| `-- stream-iterate@1.2.0
|   +- readable-stream@2.3.3 deduped
|   `-- stream-shift@1.0.0
+- ssri@4.1.6
| `-- safe-buffer@5.1.1 deduped
+- strip-ansi@4.0.0
| `-- ansi-regex@3.0.0
+- tar@4.0.1
| +- chownr@1.0.1 deduped
| +- minipass@2.2.1
| | `-- yallist@3.0.2 deduped
| +- minizlib@1.0.3
| | `-- minipass@2.2.1 deduped
| +- mkdirp@0.5.1 deduped
| `-- yallist@3.0.2
+- text-table@0.2.0
+- uid-number@0.0.6
+- umask@1.1.0
+- unique-filename@1.1.0
| `-- unique-slug@2.0.0
|   `-- imurmurhash@0.1.4 deduped
+- unpipe@1.0.0
+- update-notifier@2.2.0
| +- boxen@1.1.0
| | `-- ansi-align@2.0.0
| | `-- string-width@2.1.0 deduped
| | +- camelcase@4.1.0
| | +- chalk@1.1.3 deduped
| | +- cli-boxes@1.0.0
| | +- string-width@2.1.0
| | | +- is-fullwidth-code-point@2.0.0
| | | `-- strip-ansi@4.0.0
| | `-- ansi-regex@3.0.0 deduped
```

```
||+- term-size@0.1.1
||`- execa@0.4.0
||  +- cross-spawn-async@2.2.5
||  |+- lru-cache@4.1.1 deduped
||  |`- which@1.3.0 deduped
||  +- is-stream@1.1.0
||  +- npm-run-path@1.0.0
||  |`- path-key@1.0.0 deduped
||  +- object-assign@4.1.1
||  +- path-key@1.0.0
||  `-- strip-eof@1.0.0
||`- widest-line@1.0.0
||  `-- string-width@1.0.2
||  +- code-point-at@1.1.0
||  +- is-fullwidth-code-point@1.0.0
||  |`- number-is-nan@1.0.1
||  |  `-- strip-ansi@3.0.1
||  |    `-- ansi-regex@2.1.1
||  +- chalk@1.1.3
||  +- ansi-styles@2.2.1
||  +- escape-string-regexp@1.0.5
||  +- has-ansi@2.0.0
||  |`- ansi-regex@2.1.1
||  +- strip-ansi@3.0.1
||  |`- ansi-regex@2.1.1
||  `-- supports-color@2.0.0
|+- configstore@3.1.0
|+- dot-prop@4.1.1
|`- is-obj@1.0.1
|+- graceful-fs@4.1.11 deduped
|+- make-dir@1.0.0
|`- pify@2.3.0
|+- unique-string@1.0.0
|`- crypto-random-string@1.0.0
|+- write-file-atomic@2.1.0 deduped
|`- xdg-basedir@3.0.0 deduped
|+- import-lazy@2.1.0
|+- is-npm@1.0.0
|+- latest-version@3.1.0
|`- package-json@4.0.1
|+- got@6.7.1
|  |+- create-error-class@3.0.2
|  |`- capture-stack-trace@1.0.0
|  +- duplexer3@0.1.4
|  +- get-stream@3.0.0
|  +- is-redirect@1.0.0
|  +- is-retry-allowed@1.1.0
|  +- is-stream@1.1.0
|  +- lowercase-keys@1.0.0
|  +- safe-buffer@5.1.1 deduped
|  +- timed-out@4.0.1
|  +- unzip-response@2.0.1
|  `-- url-parse-lax@1.0.0
|    `-- prepend-http@1.0.4
|+- registry-auth-token@3.3.1
|+- rc@1.2.1
|  ||+- deep-extend@0.4.2
|  ||`- ini@1.3.4 deduped
|  |`- minimist@1.2.0
|  |`- strip-json-comments@2.0.1
|  |`- safe-buffer@5.1.1 deduped
|  +- registry-url@3.1.0
|  |`- rc@1.2.1
|  |  +- deep-extend@0.4.2
|  |  +- ini@1.3.4 deduped
|  |  +- minimist@1.2.0
```

```
|| | `-- strip-json-comments@2.0.1
|| `-- semver@5.4.1 deduped
| +- semver-diff@2.1.0
| | `-- semver@5.4.1 deduped
| `-- xdg-basedir@3.0.0
+-- uuid@3.1.0
+- validate-npm-package-license@3.0.1
| +- spdx-correct@1.0.2
| | `-- spdx-license-ids@1.2.2
| `-- spdx-expression-parse@1.0.4
+- validate-npm-package-name@3.0.0
| `-- builtins@1.0.3
+- which@1.3.0
| `-- isexe@2.0.0
+- worker-farm@1.5.0
| +- errno@0.1.4
| `-- p
```

8 Appendix

The software tools that are useful to work with SE59XX-SDK Node-RED are the following:

- **PuTTY:** to use command line interface - <https://www.putty.org/>
- **Tftpd64:** tftp64 is a free, opensource IPv6 ready application which includes DHCP, TFTP, DNS, SNTP and Syslog servers as well as a TFTP client. - <https://tftpd64.codeplex.com/releases/view/630491>
- **Node.js:** an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js has become one of the foundational elements of the "JavaScript everywhere" paradigm, allowing web application development to unify around a single programming language, rather than rely on a different language for writing server side scripts. <https://nodejs.org/en/download/>
- **Node-RED package (original):** Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. - <https://nodered.org/>



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