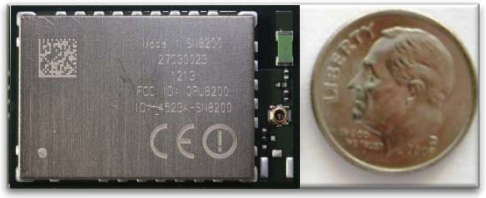
Murata Team Connectivity

SN8200 Module Software

May 2013

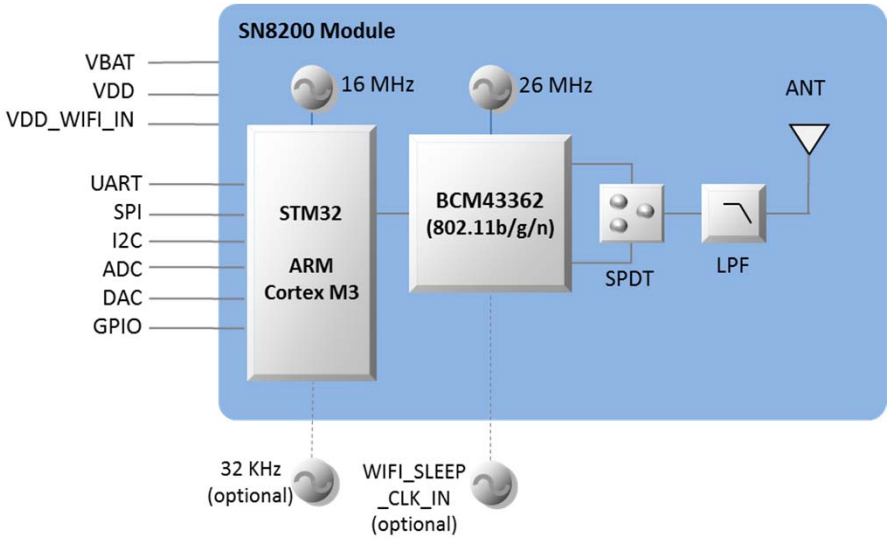
SN8200 Module



Features

- 2.4GHz IEEE 802.11b/g/n radio technology
- Wi-Fi Chip – Broadcom BCM43362
- MCU – ST Microelectronics STM32 ARM Cortex-M3
- Dimension – 30.5 x 19.4 x 2.8 mm
- Package – LGA
- Antenna configurations – on-board antenna
- Transmit power – +18 dBm
- Max Receive sensitivity – -96dbm @ b mode/11Mbps
- Host interfaces – UART, SPI
- Other interfaces – ADC, DAC, I2C, GPIO
- Operating temperature range – -30ºC to 85ºC
- ROHS compliant
- FCC/IC certified; CE compliant
- PN 88-00151-00
- EVK/SDK P/N 88-00151-85

SN8200 Module Block Diagram



Application

- Home automation
- Home energy management system
- Smart appliance
- Industrial control and monitoring

Availability

- EVK/SDK Now
- MP May 2012

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SN8200 – Access from Anywhere

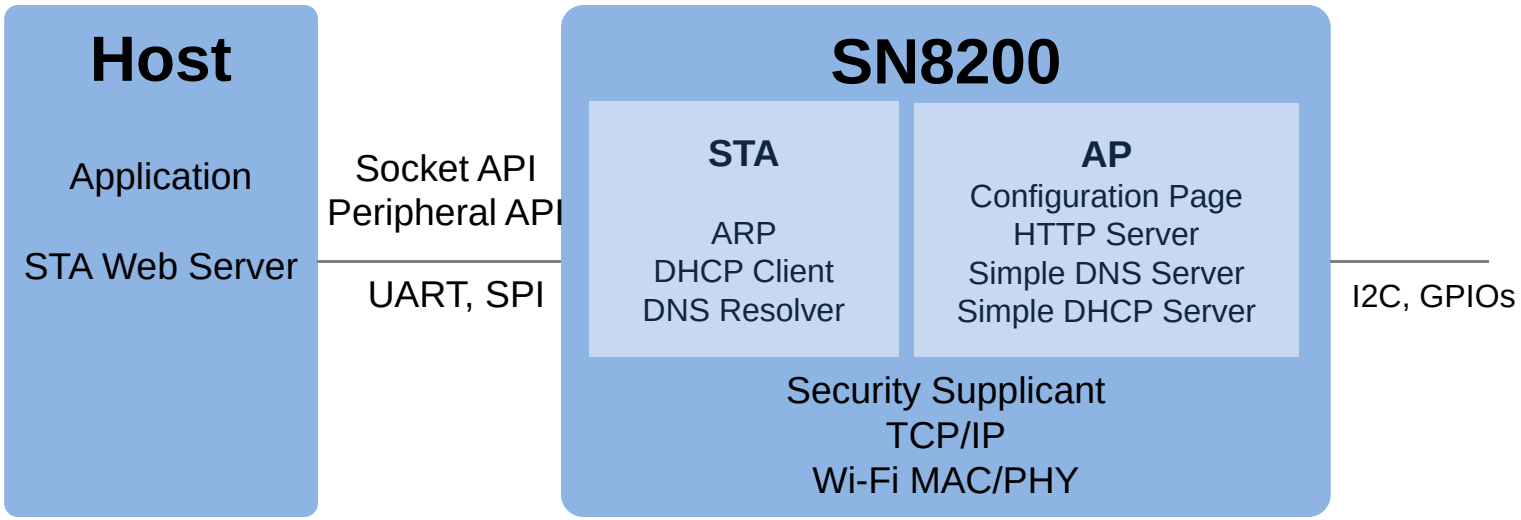
Access from internet

Access from the local network

Access directly

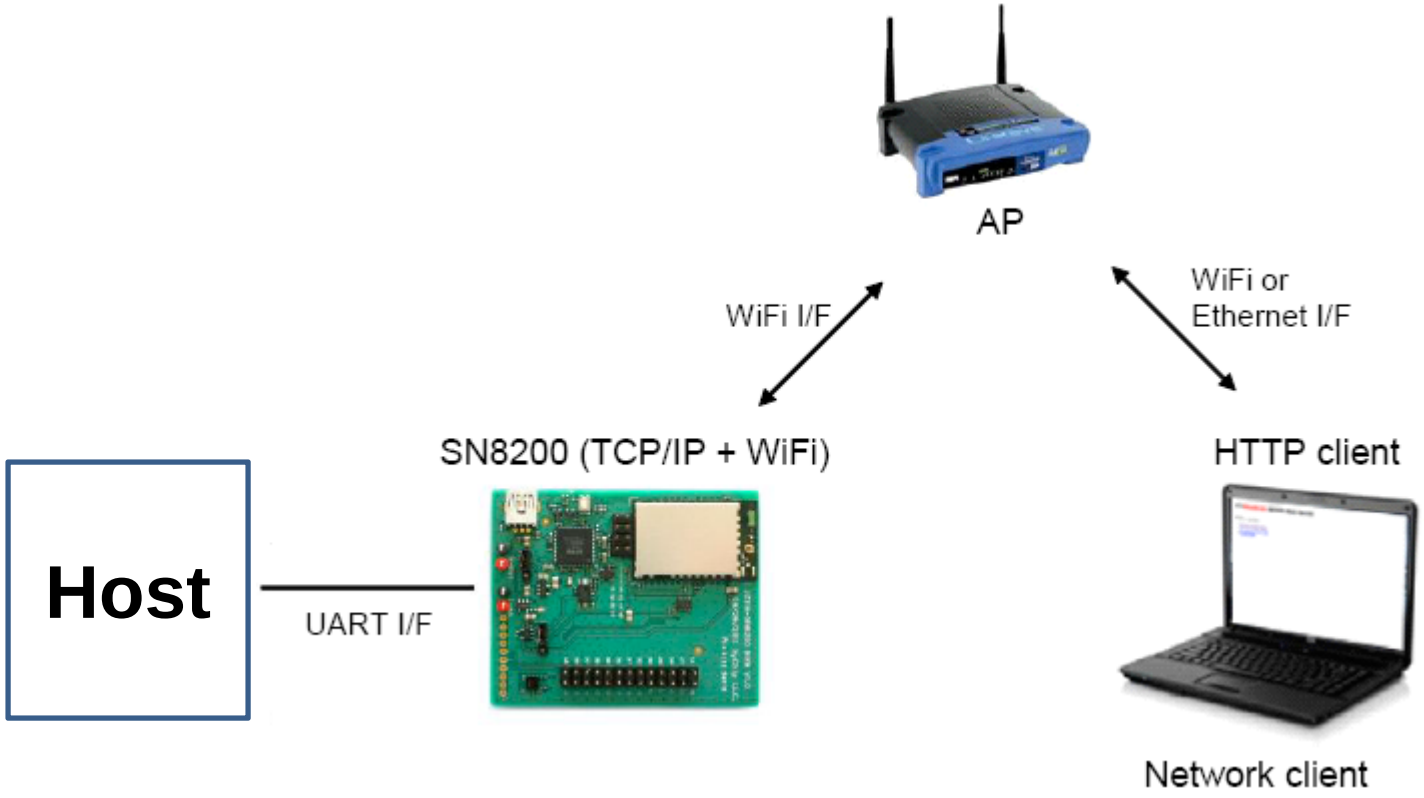
All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Simple Network Interface Controller (SNIC)



All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SNIC Use Case Example

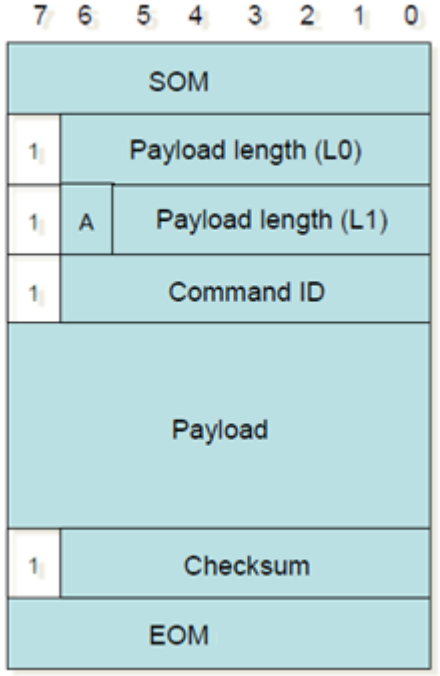


All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SNIC Serial Interface Protocol

The customer application communicates with the SN8200 module via the UART interface.

Message exchange between the customer application and SN8200 is accomplished through a simple frame structures shown below.



SN8200 structured frame format

Example of Command Frame

Command ID: CMD_ID_GEN

Payload Structure:

Byte	Descriptions
0	GEN_FW_VER_GET_RSP
1	Request Sequence
2	Status of operation: GEN_SUCCESS or GEN_FAILED
3	Version string length
4…	Version string []

← Sub-Command

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SNIC Protocol Command and Sub-command Example

Command Categories

Command	Description
CMD_ID_GEN	General Management
CMC_ID_SNIC	Simple Network Interface Card API for data exchange
CMD_ID_WIFI	Wi-Fi Configuration

Wi-Fi Configuration Related Sub-Command

Sub-command ID	Description
WIFI_ON_REQ	Turn on Wi-Fi
WIFI_OFF_REQ	Turn off Wi-Fi
WIFI_JOIN_REQ	Associate to a network
WIFI_DISCONNECT_REQ	Disconnect from a network
WIFI_GET_STATUS_REQ	Get Wi-Fi status
WIFI_SCAN_REQ	Scan Wi-Fi networks
WIFI_GET_STA_RSSI_REQ	Get STA signal strength (RSSI)
WIFI_NETWORK_STATUS_IND	Network status indication
WIFI_SCAN_RESULT_IND	Scan result indication

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

More Sub-command Example

SNIC Communication Sub-Command

Sub Command ID	Description
GEN_FW_VER_GET_REQ	Get firmware version string
SNIC_SEND_FROM_SOCKET_REQ	Send from socket
SNIC_CLOSE_SOCKET_REQ	Close socket
SNIC_SEND_ARP_REQ	Send ARP request
SNIC_GET_DHCP_INFO_REQ	Get DHCP info
SNIC_RESOLVE_NAME_REQ	Resolve a host name to IP address
SNIC_IP_CONFIG_REQ	Configure DHCP or static IP
SNIC_DATA_IND_ACK_CONFIG_REQ	ACK configuration for data indications
SNIC_TCP_CREATE_SOCKET_REQ	Create TCP socket
SNIC_TCP_CREATE_CONNECTION_REQ	Create TCP connection server
SNIC_TCP_CONNECT_TO_SERVER_REQ	Connect to TCP server
SNIC_UDP_CREATE_SOCKET_REQ	Create UDP socket
SNIC_UDP_START_RECV_REQ	Start UDP receive on socket
SNIC_UDP_SIMPLE_SEND_REQ	Send UDP packet
SNIC_UDP_SEND_FROM_SOCKET_REQ	Send UDP packet from socket

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Peripheral API

- **One I2C Host**
- **Up to 19 GPIOs**

Sub Command ID	Description
IO_I2C_INIT_REQ	Initialize I2C interface
IO_I2C_READ_REQ	Read data from I2C device
IO_I2C_WRITE_REQ	Write data to I2C device
IO_I2C_WRITE_TO_READ_REQ	Write data to I2C device followed by read command
IO_GPIO_CONFIG_REQ	Configure GPIO pins
IO_GPIO_WRITE_REQ	Set GPIO pin outputs
IO_GPIO_READ_REQ	Read GPIO pin inputs
IO_GPIO_INT_IND	GPIO interrupt indication

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Application Example

Example for TCP data communication:

Sequence of the command need to be executed from the host to the module:

1. *SNIC Init request*
2. *SNIC IP config request*
3. *TCP create socket request*
4. *TCP create connection request*
5. *Send from socket request*
6. *Close socket request*
7. *SNIC cleanup request*

The host application does not have to handle any Wi-Fi related control. The Wi-Fi control can be handled by the web sever hosted by the soft AP.

Optional Wi-Fi control API is also available.

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Sample Application

- Included sample application source code
- Included sample application user guide for customer programmer

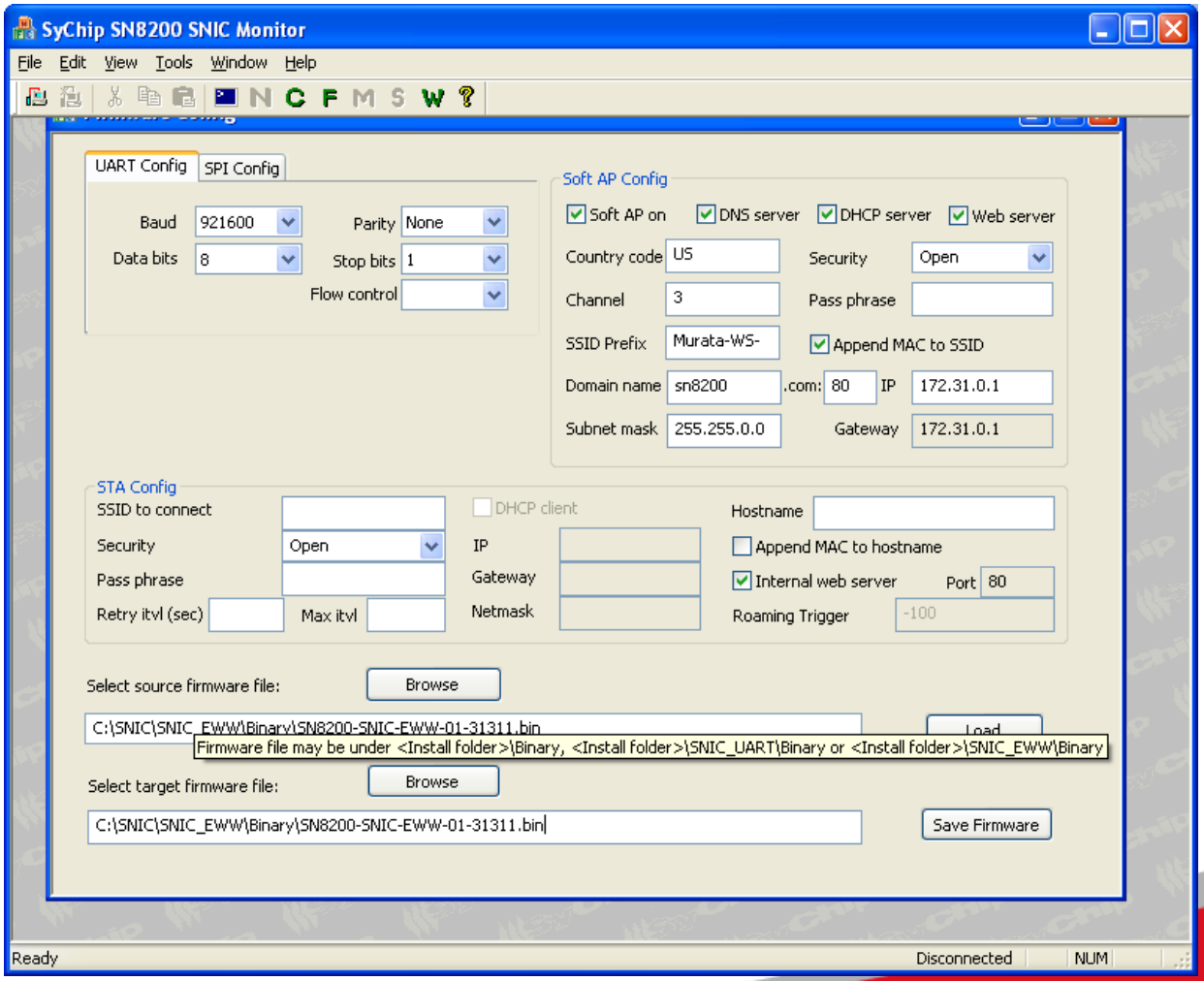
The codes implementing the protocol are files **'ser_comm.h', 'ser_rx.cpp' and 'ser_tx.cpp'**. They are written mainly in ANSI Cto make porting easy. Customer needs to port these three files into target system to support the Sychip serial protocol.

To send a UART message to the module, host application calls the **serial_transmit()** function. This function handles the protocol encoding according to reference [3], and calls the platform dependent **dummy_tx()** function to handle the actual transmission. The function **dummy_tx ()** is the function customer needs to implement to transmit one byte out.

To receive and decode a UART message from the module, host application should have a task or thread to receive data from UART port. Whenever the task or thread receives a byte, it should call the **sci_serial_redirect()** function. This function filters and decodes the bytes received. If the series of bytes conforms to the Sychip serial protocol (reference [3]), it should be passed to **sci_ser_cmd_proc()** function for packet processing. The sample client app running on PC is a multithreaded program and has circular buffers (**rx_frame**[NUM_RX_BUF]) to save messages. Customer implementation does not need to follow that. As long as the two above mentioned functions are hooked up, the receiving path should work. Host application then needs to implement **sci_ser_cmd_proc()** to handle the decoded complete UART packets.

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

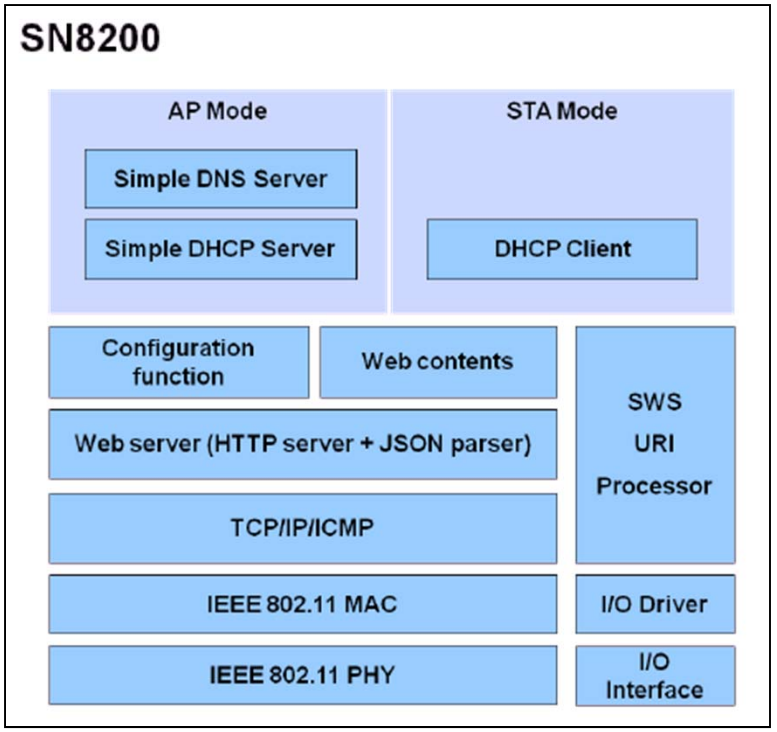
Firmware Configuration



All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

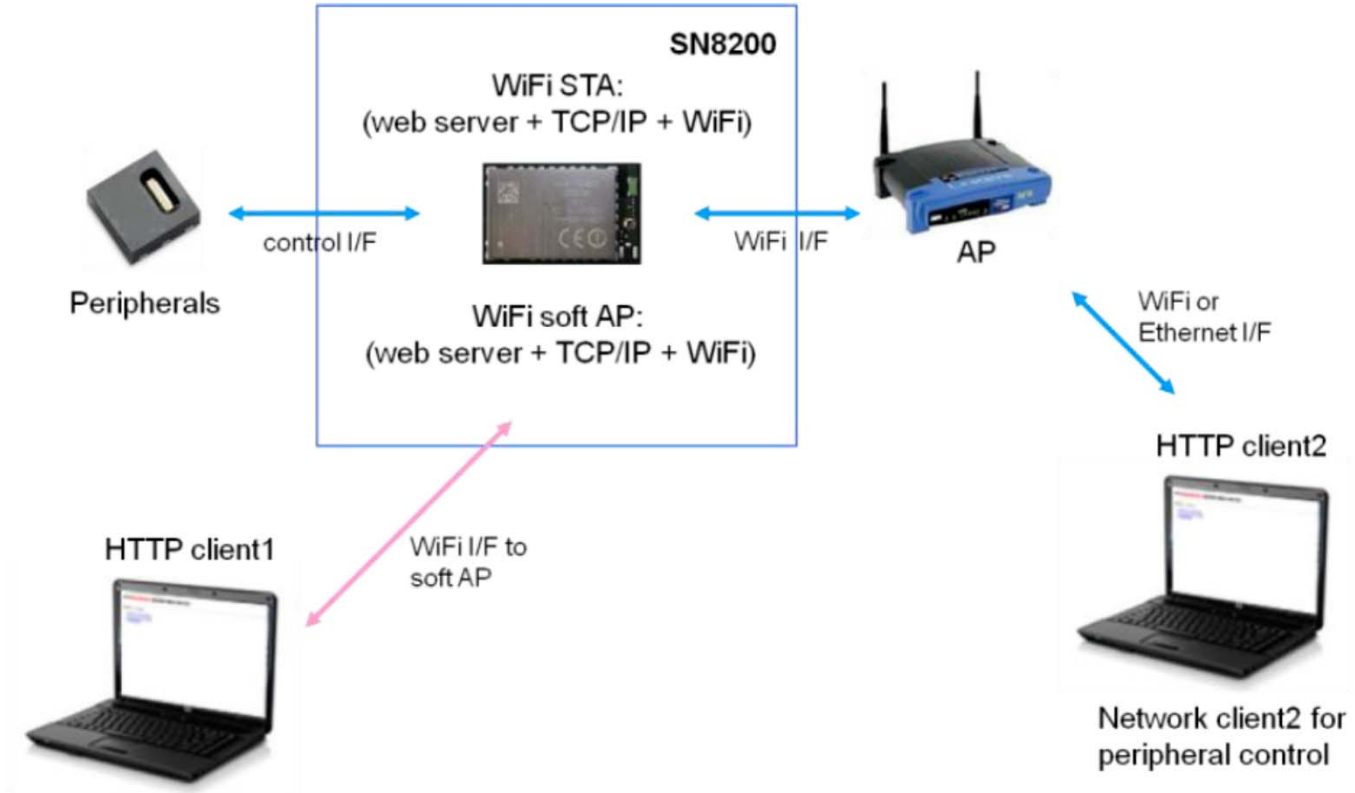
EZ Web Wizzard

- Murata **EZ Web Wizzard™ (EWW)** software supports easy custom web-based control to save the cost on additional host microcontroller
  - Control I/O interfaces through HTML, JavaScript and JSON
  - Support UART, I2C and GPIOs
  - Support wireless network configuration, JSON parser, HTTP server, TCP/IP network stack, and WiFi driver
  - Enabling the creation of wireless IP-capable nodes



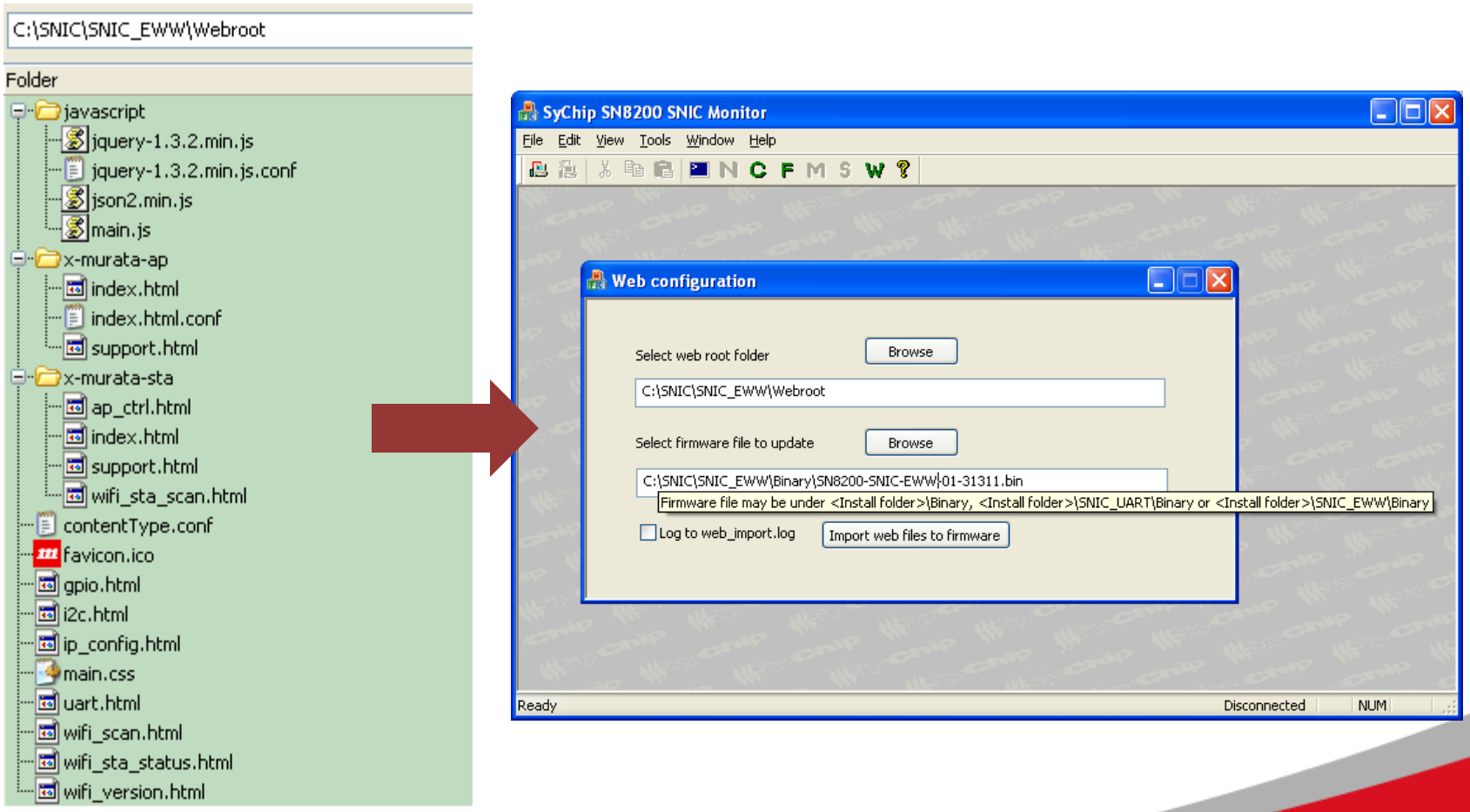
All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

EZ Web Wizaard Use Case Example



All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Web Content Customization



All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Custom Web Content

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

GPIO Control Sample

- Sample HTML source is included in the software release

```
function lamp(onoff)
{
		if (cfged == 0) {
			cfg();
			cfged = 1;
		}
		var params = {"m": "04", "v": "04"};
		if (onoff == 0)
			params = {"m": "04", "v": "00"};

		$.ajax({
		type: "POST",
		url: '/sws/gpio/wr',
		data: {Params: JSON.stringify(params)},
		success: function(data){
		},
		dataType: "jsonp"
		});
}
```


All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SN82xx EVB / SN8200EVK+

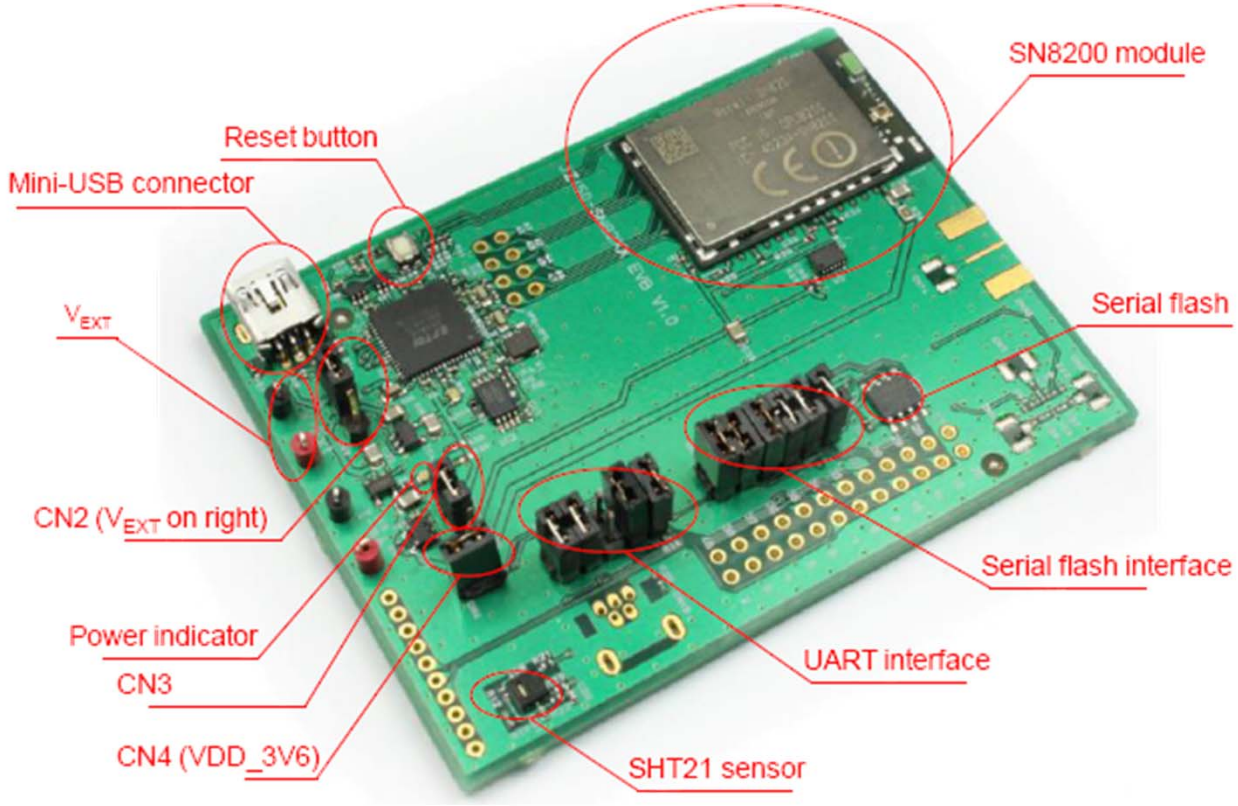


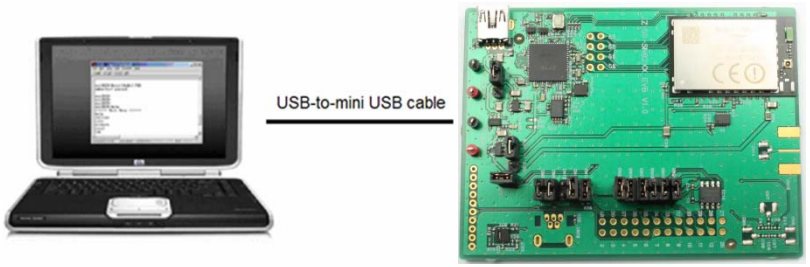
Figure 1 SN82XX EVB

All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

SN8200 Development Kit +

- **What are included in SDK**
  - 1 Evaluation board
  - 1 USB cable
  - SN8200 EVK/SDK User Guide
  - SN8200 Programmer's Guide
  - SN8200 Firmware Image
- **Benefit**
  - Serves as both evaluation and development toolkit
  - User-friendly development platform
  - No expertise needed for Wi-Fi protocol and network protocol

PN 88-00151-95



All Rights Reserved, Copyright© Murata Manufacturing Co., Ltd.

Thank You

For inquiries, please contact

modules@murata.com