SGWireless™

APPA6.03-V1.0 Enabling SGW2828-EVK Evaluation Kit on Arduino

October 2020 V1.0

Introduction

The SGW2828-EVK Evaluation Kit is designed for the development and PoC testing of applications based on the SGW2828-01A LoRa Module. Supporting USB2.0, UART, I2C and J-Link SWD debug interfaces, the SGW2828-EVK is controlled via AT commands and can be plugged directly to Arduino, enabling IoT edge device development with various sensors.

In this document, the SGW2828-EVK is used to enable data collection via LoRa, on an Arduino-based remote sensor device with DHT11 temperature and humidity sensors (Figure 1).

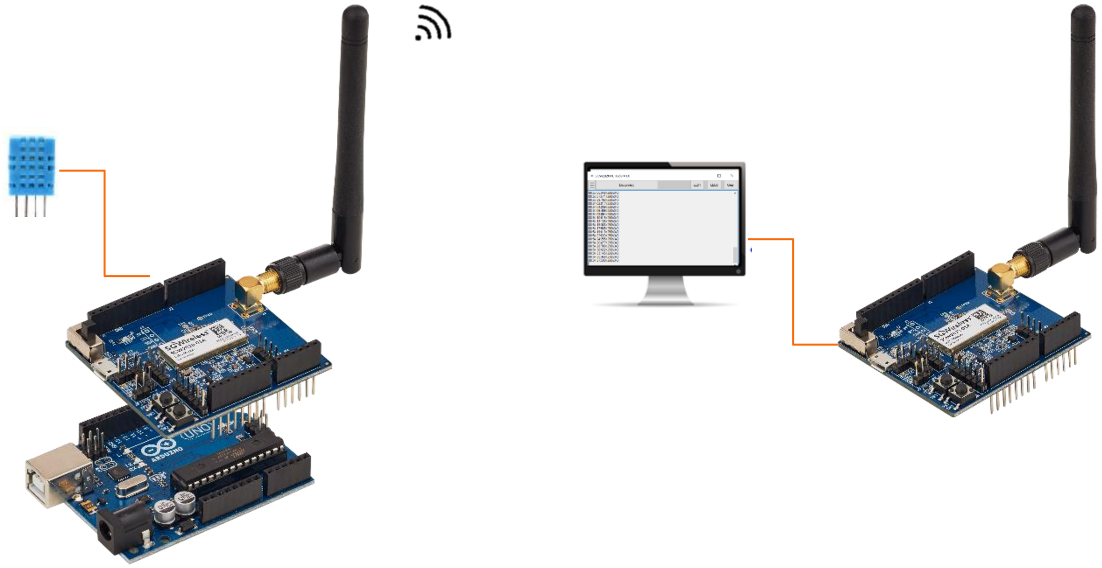


Figure 1: *Data collected on LoRa-enabled remote sensor (left) received by SGW2828-EVK host device (right)*

Required Hardware Tools

1. SGW2828-EVK Evaluation Kit
2. Arduino UNO R3
   Arduino UNO R3 is used for demonstration in this document. You can conduct an optional Blink test at https://www.arduino.cc/en/Tutorial/Blink to ensure there are no hardware issues.
3. DHT11 temperature and humidity sensor
4. 10k ohm resistor

Required Software Tools

1. SGW2828-EVK Evaluation Kit PC Tool program: https://sgwireless.com/static/tools/SGW2828-PC-Tool.7z.
2. Arduino IDE (1.8.10) with Arduino DHT Sensor Library: https://www.arduino.cc/en/main/software.

Set-up Procedures

1. SGW2828-EVK Connection to Arduino UNO (Figure 3)
   Plug the SGW2828-EVK's external headers into the Arduino UNO directly, per Table 1. Connect the device to a PC.

Table 1: *Arduino Pin Configuration with SGW2828-EVK*

Arduino UNO	SGW2828-EVK
D2	UART_TX
D3	UART_RX
D6	PB0
D7	PA0
D9	LoRa_RESET

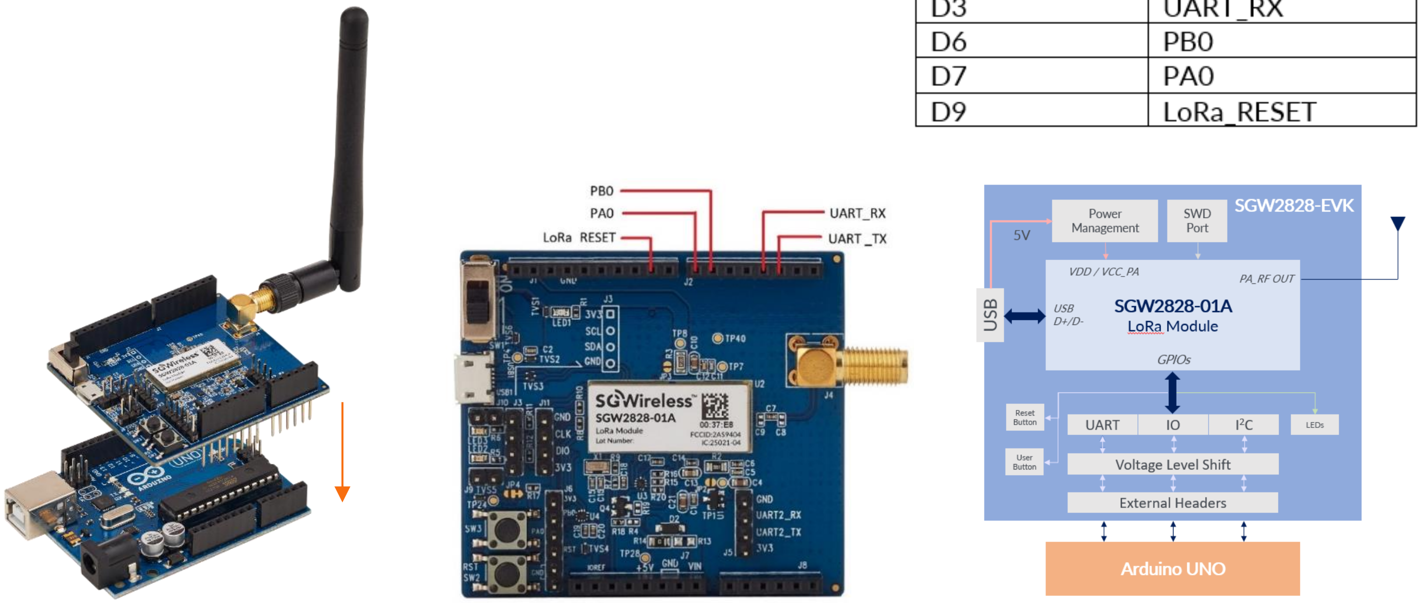


Figure 3: *SGW2828-EVK to Arduino UNO connection pins and box diagram*

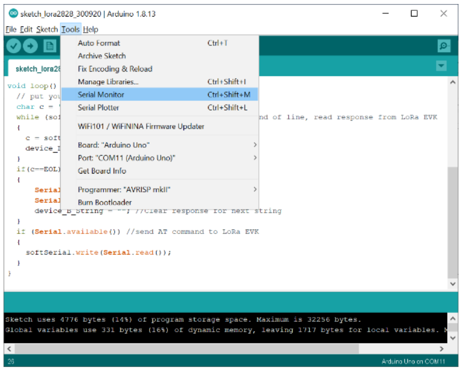
2. Configuration of SGW2828-EVK with Arduino
   Toggle the power switch on the SGW2828-EVK to ON and configure the Arduino UNO serial ports:
   - Configure D2 and D3 as serial ports RX and TX respectively.
   - Set the Arduino UNO baud to 4,800bps.
   - Compile below reference code for AT command test.

```
//uno code
#include<SoftwareSerial.h>
SoftwareSerial softSerial(2,3);//rx,tx
String device_B_String=""; // create string once, outside loop()
const char EOL = '\n'; //end of line

void setup() {
  // put your setup code here, to run once:
  softSerial.begin(4800);//
  softSerial.listen();
  Serial.begin(4800);


}

void loop() {
  // put your main code here, to run repeatedly:
  char c = '\0'; //initialise to NULL
  while (softSerial.available() && c!=EOL) //wait for end of line, read response from LoRa EVK
  {
    c = softSerial.read();
    device_B_String += c;
  }
  if(c==EOL)
  {
      Serial.print("Response: ");
      Serial.println(device_B_String);
      device_B_String = ""; //Clear response for next string
  }
  if (Serial.available()) //send AT command to LoRa EVK
  {
    softSerial.write(Serial.read());
  }
}
```

3. Testing AT Commands with Arduino IDE

Use ‘Serial Monitor’ (Figure 4) to test AT commands. Type AT commands into the text input field, and the system will respond with ‘OK’ if they are successfully received (Figure 5).

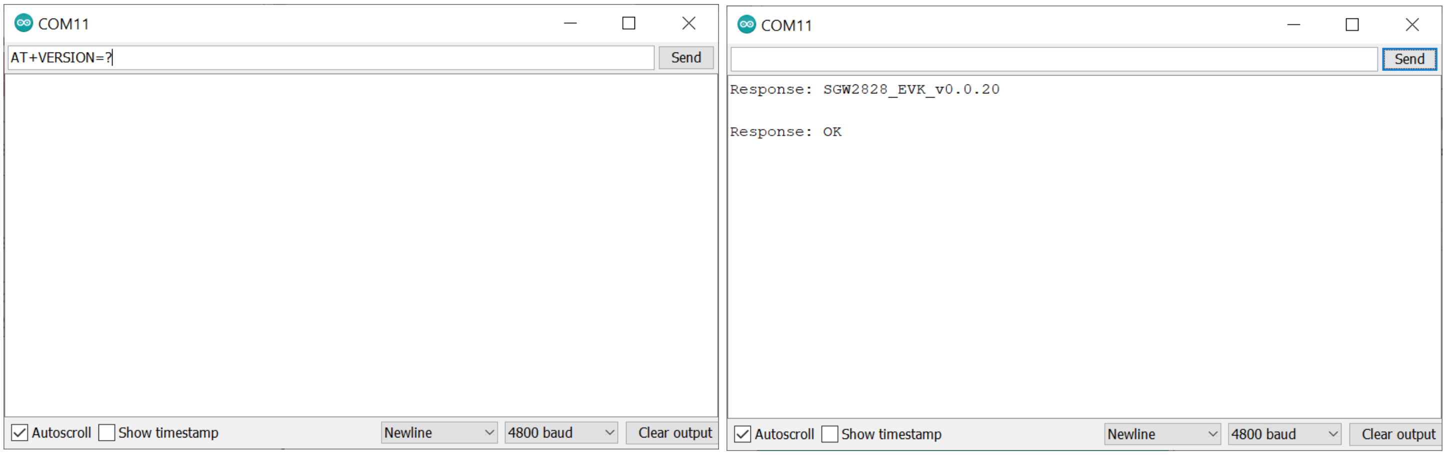
Remarks: The full AT command list can be found in the USGA5.03 SGW2828 AT Command User Manual.


sketch_lora2828_300920 | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_lora28
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM11 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
26
Arduino Uno on COM11

```
void loop()
  // put you
  char c = '
  while (sof                                        nd of line, read response from LoRa EVK
  {
    c = soft
    device_I
  }
  if(c==EOL)
  {
     Serial
     Serial
     device_B_String = ""; //Clear response for next string
  }
  if (Serial.available()) //send AT command to LoRa EVK
  {
    softSerial.write(Serial.read());
  }
}
```

```
Sketch uses 4776 bytes (14%) of program storage space. Maximum is 32256 bytes.
Global variables use 331 bytes (16%) of dynamic memory, leaving 1717 bytes for local variables. M
```

Figure 4: *Arduino Serial Monitor*



```
AT+VERSION=?
```

```
Response: SGW2828_EVK_v0.0.20

Response: OK
```

Figure 5: *Testing AT commands on Arduino Serial Monitor*

4. SGW2828-EVK Connection to DHT11 Sensor for Remote Sensor Device (Figure 6)

Connect the DHT11 sensor to Arduino Uno at Pin 4 and add the 10k ohm resistor between the data and 5V input to act as pullup on the data line. Connect the compiled device to the SGW2828-EVK.

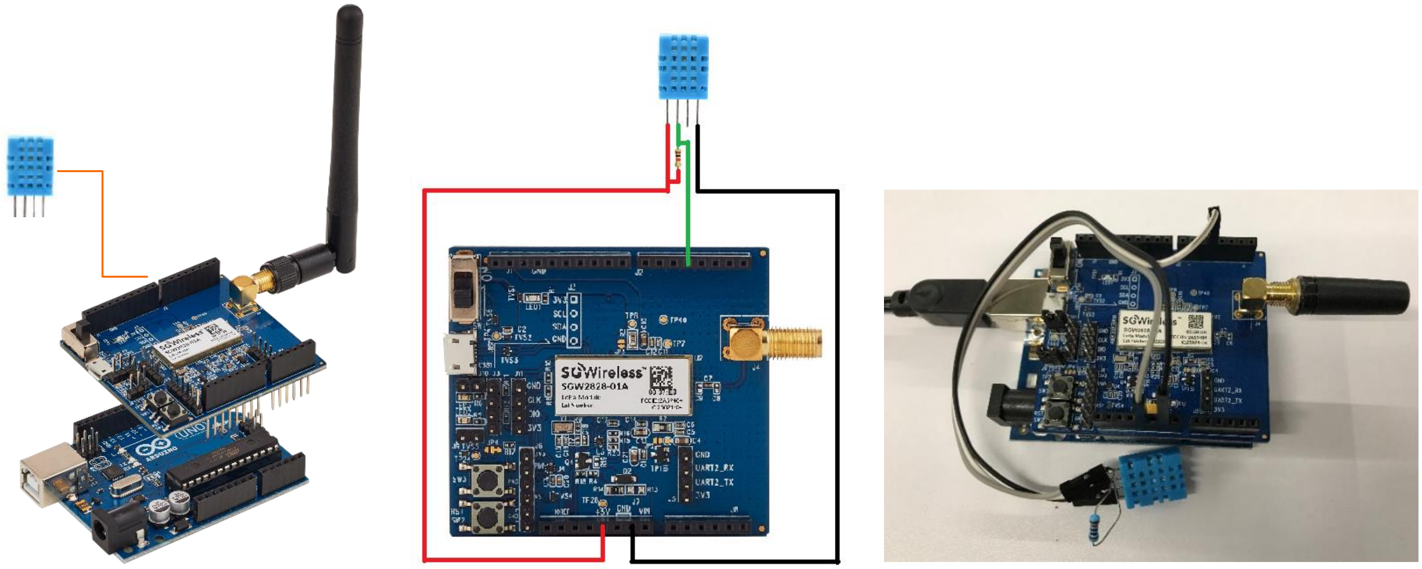
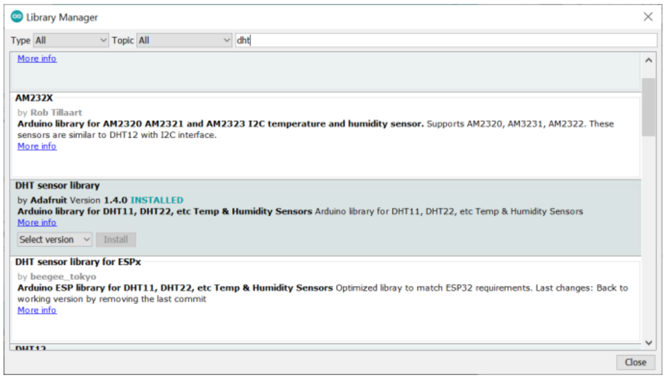


Figure 6: *DHT11 Sensor connection to Arduino UNO and SGW2828-EVK*

5. Configuration of Arduino DHT Sensor Library

Launch the Arduino DHT Sensor Library (Figure 7) and compile the below reference code.



Type All Topic All dht

More info

AM232X
by Rob Tillaart
Arduino library for AM2320 AM2321 and AM2323 I2C temperature and humidity sensor. Supports AM2320, AM3231, AM2322. These sensors are similar to DHT12 with I2C interface.
More info

DHT sensor library
by **Adafruit** Version **1.4.0** **INSTALLED**
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
More info

DHT sensor library for ESPx
by beegee_tokyo
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Back to working version by removing the last commit
More info

Figure 7: DHT Sensor Library by Library Manager

```
//uno code
#include<SoftwareSerial.h>

//for DHT11 sensor
#include "DHT.h"
#define DHTPIN 4 //connect data pin 4 to DHT11
#define DHTTYPE DHT11 //DHT 11
DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial softSerial(2,3);//rx,tx
String device_B_String=""; // create string once, outside loop()
const char EOL = '\n'; //end of line

String temperature;
String humidity;
String sbuff;
char datapacket[9]="";

void setup() {
  // put your setup code here, to run once:
  softSerial.begin(4800);//4800 baud rate
  softSerial.listen();
  Serial.begin(4800);
  dht.begin();

  //set RF parameters Preamble:16,BW:125kHz,CR:4,SF:12,Hop:0,chan:0,Pow:4dB, refer to AT command manual
  softSerial.write("AT+rf_config=16,0,4,12,0,0,4");

}

void loop() {
  // put your main code here, to run repeatedly:

  delay(2000); //wait 2 seconds between measurements

  // Read humidity as %
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.println(F("°C "));

  t=t*10;
  temperature = String(t,0);
  h=h*10;
  humidity = String(h,0);
  sbuff = "t"+temperature+"h"+humidity;
  sbuff.toCharArray(datapacket,9);    //convert String into char array
  Serial.println(datapacket);
```

```
          char c = '\0'; //initialise to NULL
          while (softSerial.available() && c!=EOL) //wait for end of line, read response from LoRa EVK
          {
            c = softSerial.read();
            device_B_String += c;
          }
          if(c==EOL)
          {
              Serial.print("Response: ");
              Serial.println(device_B_String);
              device_B_String = ""; //Clear response for next string
          }
          if (Serial.available()) //send AT command to LoRa EVK
          {
            softSerial.write(Serial.read());
          }
          softSerial.write("AT+rf_send=1,0,8");
          softSerial.write(10); //send newline
          delay(10); //wait 10ms to send
          softSerial.write(datapacket); //send data to LoRa EVK
          softSerial.write(10);
}
```

Data Collection with Host Device

1. SGW2828-EVK Connection to PC for Host Device
   Connect the SGW2828-EVK to the PC with the micro USB cable (Figure 8). Toggle the power switch to ON and run the SGW2828-PC-Tool program.

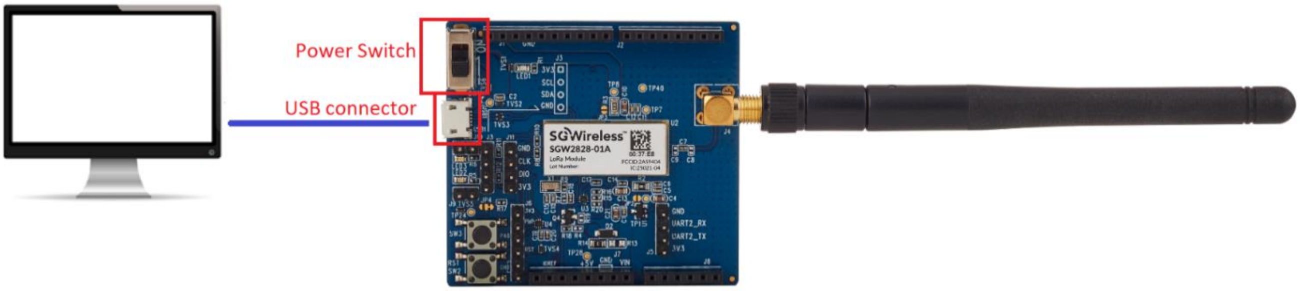
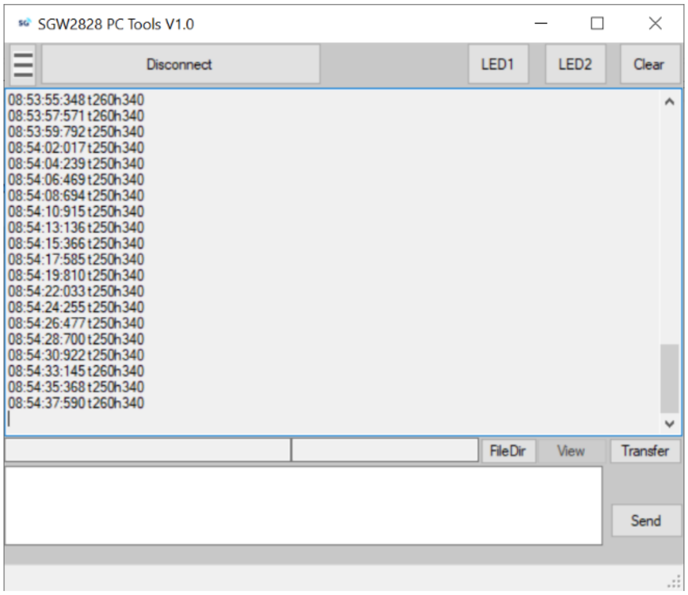


Figure 8: SGW2828-EVK to PC connection

2. Data Collection with SGW2828-EVK PC Software
   Temperature and humidity records are received every 2 seconds in “txxxhyyy” format (Figure 9):
   - “xxx”/10: Temperature value in Celsius
   - “yyy”/10: Humidity value in %

   For example, data value “t250h340” is translated to 25.0°C; 34.0%.

   Remarks: The full operation manual of the SGW2828-EVK PC Software can be found in the [USGA6.02 SGW2828-EVK Evaluation Kit PC Software User Manual](#).



```
08:53:55:348 t260h340
08:53:57:571 t260h340
08:53:59:792 t250h340
08:54:02:017 t250h340
08:54:04:239 t250h340
08:54:06:469 t250h340
08:54:08:694 t250h340
08:54:10:915 t250h340
08:54:13:136 t250h340
08:54:15:366 t250h340
08:54:17:585 t250h340
08:54:19:810 t250h340
08:54:22:033 t250h340
08:54:24:255 t250h340
08:54:26:477 t250h340
08:54:28:700 t250h340
08:54:30:922 t250h340
08:54:33:145 t260h340
08:54:35:368 t250h340
08:54:37:590 t260h340
```

Figure 9: Data records received on host device

Useful Links

1. SG Wireless SGW2828-01A LoRa Module: https://sgwireless.com/product/SGW2828.
2. SGW2828 AT Command User Manual: https://sgwireless.com/uploads/ueditor/upload/file/20201013/USGA5.03-V1.0%20SGW2828%20LoRa%20Module%20AT%20Command%20User%20Manual.pdf.
3. SGW2828-EVK Evaluation Kit PC Application User Manual: https://sgwireless.com/uploads/ueditor/upload/file/20201013/USGA5.03-V1.0%20SGW2828%20LoRa%20Module%20AT%20Command%20User%20Manual.pdf.
4. DHT Sensor reference guide: https://learn.adafruit.com/dht/connecting-to-a-dhtxx-sensor.
5. Semtech SX1276 documentation: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276.

Revision History

Revised	Version	Description
20-Oct-2020	1.0	Initial document release

Contact us at cs@sgwireless.com for any queries, or find us at any channel below:

Website: https://sgwireless.com/
LinkedIn: https://www.linkedin.com/company/sgwireless/
Facebook: https://www.facebook.com/sgwirelessIoT
Twitter: @sgwirelessIoT

Information in this document is provided solely to enable authorized users or licensees of SG Wireless products. Do not make printed or electronic copies of this document, or parts of it, without written authority from SG Wireless.

SG Wireless reserves the right to make changes to products and information herein without further notice. SG Wireless makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SG Wireless assume any liability arising out of the application of any product and specifically disclaims any and all liability, including without limitation consequential or incidental damages. SG Wireless does not convey any license under its patent rights nor the rights of others. SG Wireless products may not be used in life critical equipment, systems or applications where failure of such equipment, system or application would cause bodily injury or death. SG Wireless sells products pursuant to standard Terms and Conditions of Sale which may be found at https://www.sgwireless.com/page/terms.

SG Wireless may refer to other SG Wireless documents or third-party products in this document and users are requested to contact SG Wireless or those third parties for appropriate documentation.