BLUEGIGA WI-FI PRODUCTS

GETTING STARTED

Friday, 22 November 2013

Version 1.2

SILICON LABS

VERSION HISTORY

Version	Comment
1.0	First version
1.1	Minor changes
1.2	Added WFD documentation and comparison to AP mode

TABLE OF CONTENTS

1 Introduction

This application note discusses how to start developing Wi-Fi applications using Bluegiga WF121 Wi-Fi Module and Bluegiga Wi-Fi Software and SDK. The document gives a short introduction to the Wi-FI technology, describes the Bluegiga Wi-Fi products and software architecture and discusses the development option with the SDK to develop Wi-Fi applications.

2 What Is Wi-Fi® Technology?

Wi-Fi is the marketing name for a 2.4 and 5GHz wireless local area network technology based on a set of 802.11 standards developed IEEE organization. Wi-Fi brand and marketing is maintained by an organization called Wi-Fi Alliance.

The standard was originally developed to provide wireless connectivity to Ethernet networks and that is still the main use case today although new standards like Wi-Fi Direct provide also point-to-point connectivity.

The typical 802.11 architecture is displayed in the figure below also including the Internet protocol stack and short description of each layer and their functions are also given below:

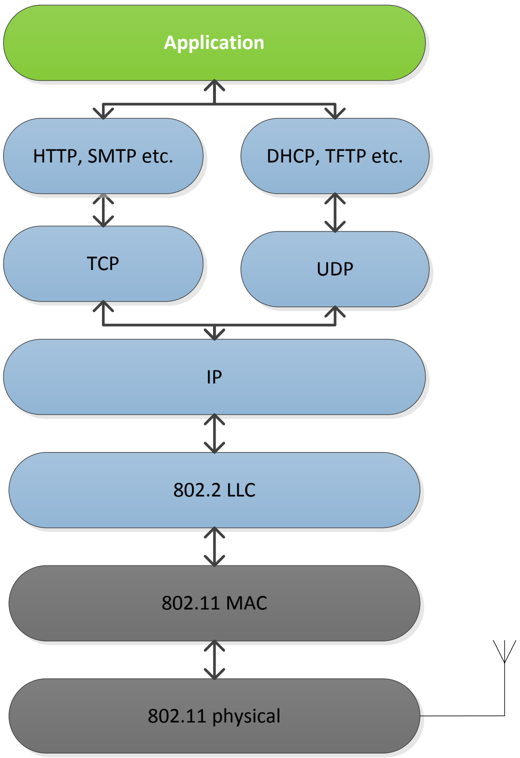


Figure 1: 802.11 and Internet protocol architecture

2.1 802.11 Physical Layer

The 802.11 physical layer implements the 2.4 GHz and 5GHz transceiver and modulations like DSSS, FHSS and OFDM, so it's responsible of transmitting and receiving raw bit stream over a wireless connection.

The physical layer divides the total frequency band to 14 channels from 2412Mhz to 2482MHz and each channel has a bandwidth of 20Mhz except with 802.11n and 802.11ac, where the bandwidth can be up to 160 Mhz. 802.11n with MIMO and 802.11ac are typically used for applications requiring high bandwidth or multiple connections, like access points, PCs, laptops, tablets and smart phones. 802.11 b/g and single stream 802.11n on the other hand are ideal for lower data rate, longer range and lower bandwidth applications like internet-of-things type of devices.

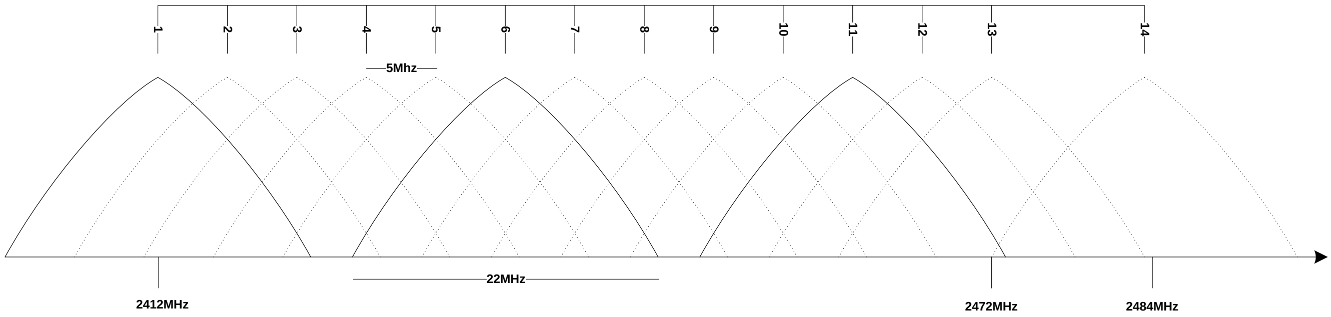


Figure 2: 802.11 2.4GHz frequency band and channels

The original 802.11 standard was developed 1997 several amendments to the standard have been released since then such as 802.11b, 802.11g and 802.11n. A short summary and comparison of the different 802.11 standards is shown below.

Standard	Frequency	Bandwidth	Symbol rate	MIMO streams	Modulation	Range (factor)
802.11	2.4GHz	20 MHz	1,2 Mbps	1	DSSS, FHSS	1
802.11b	2.4GHz	20 MHz	1, 2, 5.5 and 11 Mbps	1	OFDM	1.4
802.11g	2.4GHz	20 MHz	6 to 54 Mbps	1	OFDM, DSSS	1.4
802.11a	5 GHz	20 MHz	6 - 54 Mbps	1	OFDM	1.2
802.11n	2.4/5 GHz	20 / 40 MHz	7.2 to 150 Mbps	4	OFDM	2.5
802.11ac	2.4/5 GHz	20 / 40 / 80 / 160 MHz	up to 433.3 Mbps	8	OFDM	2.5

Table 1: 802.11 standards

2.2 802.11 Media Access Control

The 802.11 Media Access Control (MAC) maintains the communication between 802.11 stations (STAs) and Access Points (APs) and coordinates access to the shared radio channels using the Carrier Sense Multiple Access / Collision Avoidance (CSMA/CD) algorithm. The 802.11 MAC provides the following services:

- Active (probing) and passive (beacon) scanning of Access Points
- Authentication between the stations and Access Points
- Association between stations and Access Points
- Encryption of data payload
- RTS and CTS flow control that allows the AP to control the STAs accessing the medium
- Power saving that allows the stations to save power
- Fragmentation that enables the stations to divide large amount of data into smaller chinks

2.3 802.2 Logical Link Control

In short the 802.2 LLC provides end-to-end link control over the 802.11 based networks. The 802.2 LLC provides the following services:

- **Unacknowledged connectionless service:** Used mainly for peer-to-peer, multicast and broadcast communication and the higher layers must provide error correction, flow control and acknowledgement services.
- **Acknowledged connectionless service:** Used mainly for peer-to-peer, multicast and broadcast communication, but provides flow and error control
- **Connection oriented service:** Provides connection oriented means of communication including error and flow control.

2.4 Wi-Fi Security

The 802.11 standard provides the following security mechanism:

- **Authentication:** Authentication is done using either pre shared keys or dynamically exchanging the keys with Wireless Protected Setup (WPS). Also open networks, which do not require authentication, can be established.
- **Association: T**he 802.11 stations need to associate with the AP in order to be able to communicate
- **Access Control:** Access Points have the capability to decide which stations are able to join the network or not. The access control is usually doing based on the stations MAC address.
- **Encryption:** Data payload can be encrypted with WPA2, WPA, WEP or WAPI (China) encryption algorithms.
- **Data integrity:** Data integrity is guaranteed with the encryption algorithm.
- **Data confidentiality:** Data confidentiality (eavesdrop prevention) is guaranteed with the encryption algorithm.

The 802.11 technology has in the past suffered from multiple security flaws and attacks which have enabled the eavesdropping of data or unauthorized network access. At the moment only WPA2 and WPS algorithm can be considered secure and for example Wi-Fi alliance does not allow the certification of devices that implement WPA or WEP.

2.5 Wi-Fi Direct

Wi-Fi Direct (WFD) is a new emerging Wi-Fi Alliance standard that enables peer-to-peer (P2P) networking between Wi-Fi stations. The WFD technology uses a combination of Station, Software Access Point mode (SoftAP), WPA2 and WPS technologies to enable point-to-point connections without the use of Wi-Fi infrastructure.

The Wi-Fi direct specification provides the following methods:

- **Device discovery**: WFD devices, when put into discovery mode, will start to listen three RF channels and they will also actively probe for nearby WFD devices in order to start the communication with them.
- **Role negotiation:** After two WFD devices have discovered each other a negotiation process will start between them in order to decide which device start will be the Access Point and which will be the client (STA). The negotiation will be done using low level MAC frames and devices have the capability to advertise if they prefer to be in the client mode or the access point mode.
- **Authentication, association and encryption:** Once the role negotiation has been done and the devices have selected their operational modes the authentication and association process will take place. Typically WPS (push button or PIN code) method is used for authentication and WPA2 encryption algorithm for encrypting the data.
- **Service discovery:** At the moment the WFD specification does not contain standardized methods for service discovery so the service discovery implementation is up to the vendor.

At the moment WFD has limited support on devices and only some Android based devices and Windows 8 devices support WFD at the moment. The Wi-Fi Alliance is developing the technology further to include concepts like service discovery.

2.5.1 Wi-Fi Direct vs. Access Point Mode

Wi-Fi Direct and Access Point mode can both be used to setup point-to-point Wi-Fi connection and more or less accomplish the same use case, however each of the methods have some benefits and drawbacks.

Wi-Fi Direct:

- Benefit of Wi-Fi Direct mode is that smart phones and tablets that are able to run WFD are also able to maintain Internet connectivity either over the cellular radio or Wi-Fi when being connected to the other WFD devices in the P2P manner.
- With WFD typically concurrent Wi-Fi mode is supported so a device can act as a Wi-Fi client and Wi-Fi Access Point. This however might depend on the Wi-Fi radio supplier.
- Drawback of the WFD mode however is that at the time of writing this document it’s only supported on some specific Android and Windows 8.1 devices the mainstream support of WFD is lacking on devices like iOS devices, Windows7 or other platforms.

Access Point Mode:

- Access Point mode’s advantage is that it’s supported on more or less all Wi-Fi capable devices whether they are iOS, Android, Windows and Linux devices, so you are able to use it in much wider manner than WFD.
- Drawback when using the Access Point mode is that when a phone or tablet connects a Wi-Fi access point in a P2P manner, it will typically loose Internet connectivity over the regular cellular radio or Wi-Fi. This is because the phone or tablet expects to get Internet connectivity via Wi-Fi (unless WFD is used).
- With AP mode concurrent Wi-FI mode might not be supported so typically when the device is in the Access Point mode it will not be a Wi-Fi client. This however might depend on the Wi-Fi radio supplier.

3 Introduction to the Bluegiga Wi-Fi Software

The Bluegiga Wi-Fi Software enables developers to quickly and easily develop Wi-Fi applications without in-depth knowledge of the Wi-Fi technology. The Wi-Fi Software consists of two parts:

- The Bluegiga Wi-Fi Software
- The Bluegiga Wi-Fi Software Development Kit (SDK)

3.1 The Bluegiga Wi-Fi Stack

The Bluegiga Wi-Fi Stack is an embedded 802.11 MAC and IPv4 stack targeted for Bluegiga's WF121 Wi-Fi module. The Wi-Fi software implements full 802.11 functionality, Station and Access Point modes, WPA2, WPA and WEP and WPS security and various IP based protocols such as TCP, UDP, DHCP, DNS, ICMP and HTTP server.

The Bluegiga Wi-Fi Stack provides powerful, low overhead and easy-to-use Bluegiga BGAPI^{TM} binary API that can be used over UART, USB or SPI interfaces and it provides functions such as such as Access Point discovery, Access Point association, encryption, and connection establishment. The Wi-Fi Stack also supports the 802.11 access point mode and implements an embedded HTTP server for the easy configuration and offers direct connections to phones, tablets and PC's. To simplify the software development the Bluegiga Wi-Fi software also includes a Bluegiga BGLib^{TM} C-library that implements the BGAPI protocol parser for various embedded systems.

For standalone applications the Bluegiga Wi-Fi Software also provides a simple BGScript^{TM} scripting language and VM environment, which enables the users to write simple applications for the WF121 Wi-Fi module. This enables lower cost and smaller designs to be made as there is no need to use an external MCU.

3.2 The Bluegiga Wi-Fi SDK

The Bluegiga Wi-Fi SDK is a software development kit, which enables the device and software vendors to develop products on top of the Bluegiga’s Wi-Fi hardware and software.

The Wi-Fi SDK supports multiple development models and the software developers can decide whether the application software runs on a separate host (a low power MCU) or whether they want to make fully standalone devices and execute their code on the MCU embedded in the Bluegiga Wi-Fi modules. The SDK also contains documentation, tools for compiling the firmware, installing it into the hardware and lot of example applications speeding up the development process.

Fully standalone applications can be developed using a simple scripting language called BGScript^{TM}. Several examples are also offered as a part of the Wi-Fi SDK in order to easily develop Wi-Fi compatible end products. These examples contain for example embedded HTTP server, WPS and TCP to Serial applications.

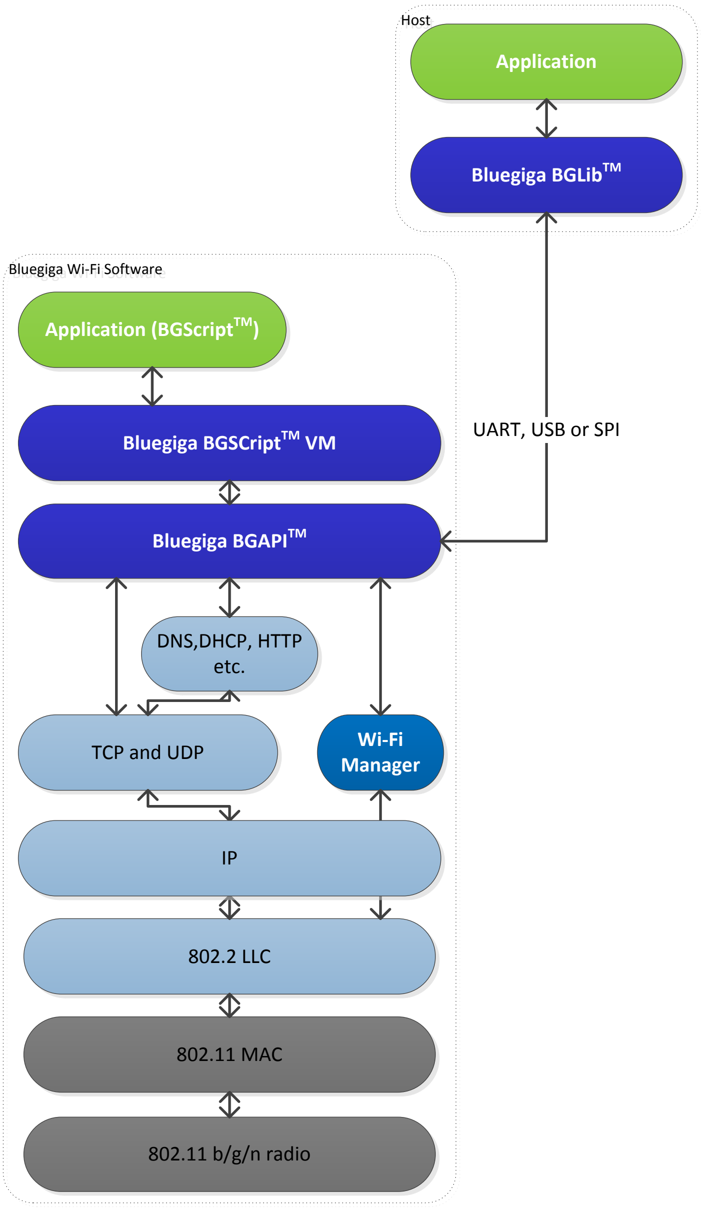


Figure 3: The Bluegiga Wi-Fi Software Architecture

The Bluegiga Wi-Fi Software architecture is illustrated above and it consists of the following components

- The Bluegiga Wi-Fi stack implementing the 802.11, 802.2, IPv4 and higher level protocols
- **BGAPI™** APIs that enable the software developers to interface to the Wi-Fi Stack
- **BGLib™** lightweight host library which implements the BGAPI binary protocol and parser and is target for applications where separate host processor is used to interface to the Wi-Fi modules over UART, USB or SPI.
- **BGScript™** Virtual Machine (VM) and scripting language which enable application code to be developed and executed directly on the Bluegiga Wi-Fi hardware

Each of these components are described in more detail in the following chapters.

3.3 The BGAPI^{TM} Protocol

For applications where a separate host is used to implement the end user application, a transport protocol is needed between the host and the Wi-Fi stack. The transport protocol is used to communicate with the Wi-stack as well to transmit and receive data packets. This protocol is called BGAPI and it's a lightweight binary based communication protocol designed specifically for ease of implementation within host devices with limited resources.

The BGAPI protocol is a simple command, response and event based protocol and it can be used over UART USB or SPI.

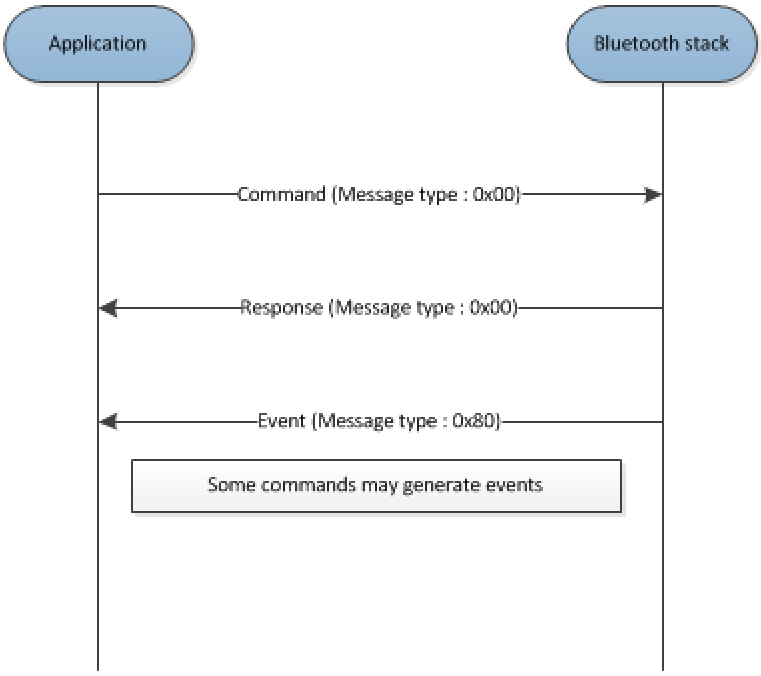

Application
Bluetooth stack
Command (Message type : 0x00)
Response (Message type : 0x00)
Event (Message type : 0x80)
Some commands may generate events

Figure 4: The BGAPI protocol

The BGAPI provides access for example to the following layers in the Wi-Fi Stack:

- **Wi-Fi** – This interface enables access for example to functions like Access Point discovery, authentication, association, security management and switching between station and access point modes.
- **TCP and IP** – These functions allow the creation of TCP and UDP client and server connections and DNS.
- **Hardware -** An interface to access the various hardware layers such as timers, ADC and other hardware interfaces
- **Persistent Store** - User to access the parameters of the radio hardware and read/write data to non-volatile memory
- **System** - Various system functions, such as querying the hardware status or reset it
- **HTTP** - Enables the access to the embedded HTTP server
- **Ethernet** – Enables access to the Ethernet interface and functions like Ethernet to Wi-Fi bridging.

3.4 The BGLib™ Host Library

For easy implementation of BGAPI protocol an ANSI C host library is available. The library is easily portable ANSI C code delivered within the Wi-Fi SDK. The purpose is to simplify the application development to various host environments.

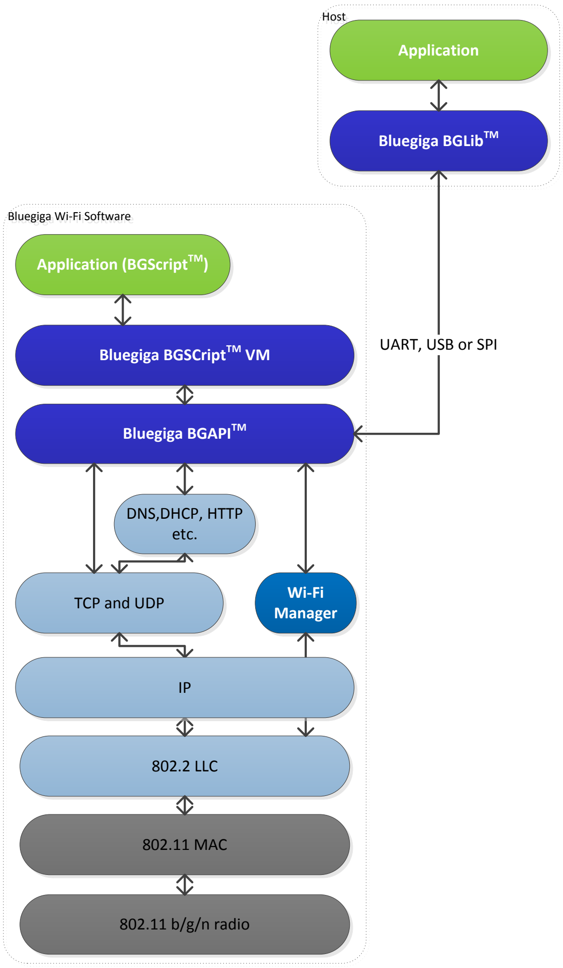
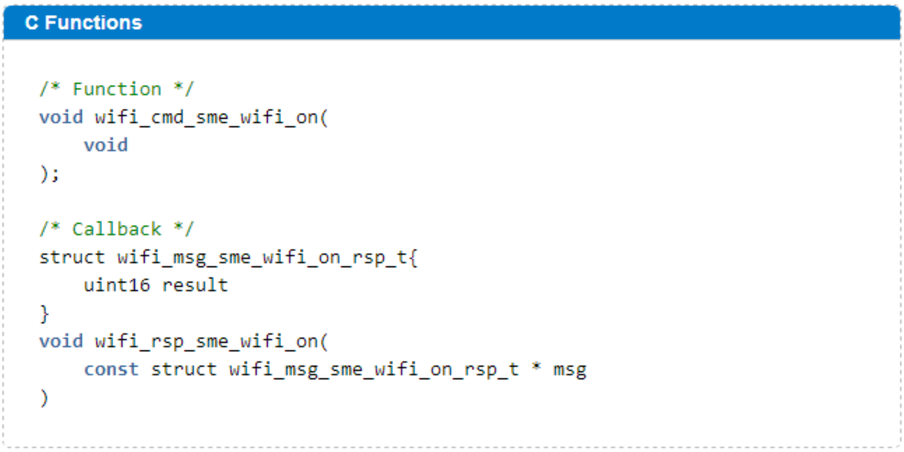


Figure 5: BGLib host library



C Functions

```
/* Function */
void wifi_cmd_sme_wifi_on(
    void
);

/* Callback */
struct wifi_msg_sme_wifi_on_rsp_t{
    uint16 result
}
void wifi_rsp_sme_wifi_on(
    const struct wifi_msg_sme_wifi_on_rsp_t * msg
)
```

Figure 6: BGLib C functions and call backs

3.5 BGScript^{TM} Scripting Language

The Wi-Fi SDK also allows the application developers to create fully standalone devices without a separate host MCU and run all the application code on the Bluegiga Wi-Fi hardware. The Wi-Fi modules can run simple applications along the Wi-Fi stack and this provides a benefit when one needs to minimize the end product's size, cost and current consumption. For developing standalone Wi-Fi applications the SDK includes the Script VM, compiler and other BGScript development tools. BGScript provides access to the same software and hardware interfaces as the BGAPI protocol and the BGScript code can be developed and compiled with the free-of-charge tools provided by Bluegiga.

Typical BGScript applications are only few tens to hundreds lines of code, so they are really quick and easy to develop and lots of readymade examples are provides with the SDK.

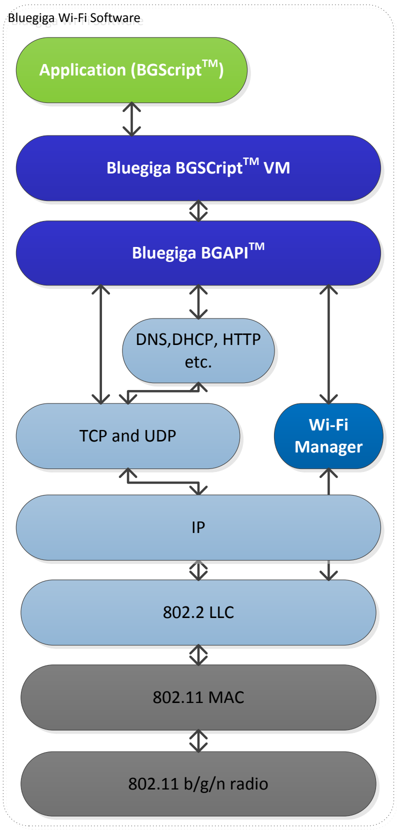


Figure 7: BGScript application model

BGScript code example:

```
#System boot event listener
event system_boot(major, minor, patch, build, bootloader_version, tcpip_version, hw)
  # Set operating mode to AP
  call sme_set_operating_mode(AP_MODE)


  # Enable Wi-Fi
  call sme_wifi_on()
end
```

4 Bluegiga Wi-Fi Modules

4.1 WF121 Wi-Fi Module

Bluegiga WF121 is a stand-alone Wi-Fi module providing fully integrated 2.4GHz 802.11 b/g/n radio, TCP/IP stack and a 32-bit micro controller (MCU) platform for embedded applications requiring simple, low-cost and low-power wireless IP connectivity. WF121 also provides flexible peripheral interfaces such as SPI, I2C, ADC, GPIO, *Bluetooth* co-existence and timers to connect various peripheral interfaces directly to the WF121 Wi-Fi module.

WF121 Wi-Fi module also allows end user applications to be embedded onto the on-board 32-bit MCU using a simple Bluegiga BGScript^{TM} scripting language and free-of-charge development tools. This cuts out the need of and additional MCU and enables end users to develop smaller and lower cost Wi-Fi devices. WF121 can also be used in modem-like mode in applications where the external MCU is needed. The 802.11 access point and HTTP server functionality is also included for the easy configurations and direct connections with phones, tablets and PC's.

With an integrated 802.11 radio, antenna, single power supply, and CE, FCC and IC regulatory certifications, WF121 provides a low-risk and fast time-to-market for applications requiring Wi-Fi connectivity.

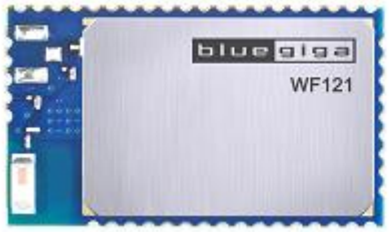


Figure 8: WF121 Wi-Fi Module

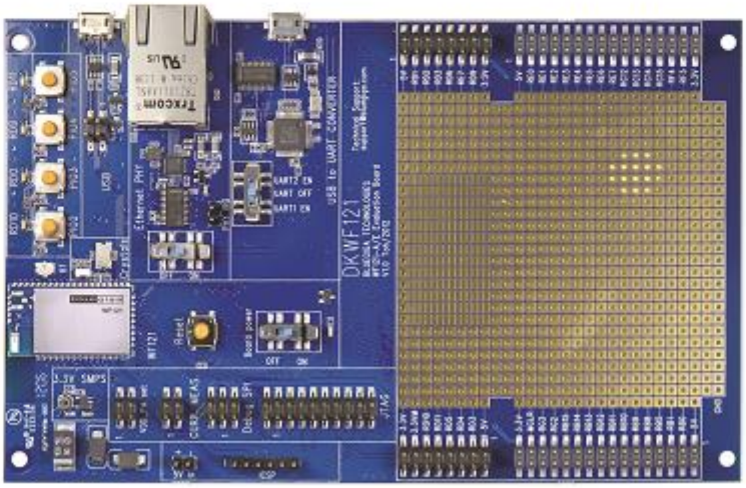


Figure 9: WF121 Development Kit

Key features

- IEEE 802.11 b/g/n radio
  - Single 2.4 GHz band
  - Symbol rate up to 72.2Mbps
  - Integrated antenna or U.FL connector
- Excellent radio performance:
  - TX power: +17 dBm
  - RX sensitivity: -97 dBm
- Host interfaces: UART, USB, SPI
- Peripheral interfaces:
  - GPIO, AIO and timers
  - I2C, SPI and UART
- Embedded TCP/IP stack on 802.11 MAC:
  - IP, TCP and UDP
  - DHCP and DNS protocols
  - HTTP server
- 802.11 features:
  - Client (STA) mode support
  - Access Point (AP) mode support up to five clients
  - WPA2, WPA, WEP and WPS security
- 32-bit embedded micro controller:
  - 80Mhz, 128kB RAM and 512kB Flash
  - MIPS architecture
- Bluegiga BGScript^{TM} scripting language for stand-alone applications
- Temperature range: -40°C - +85°C
- Small size: 15.4 x 26.2 x 2.1 mm
- Fully CE, FCC and IC, Japan and South-Korea qualified

4.2 WF111 Wi-Fi Module

WF111 is a fully integrated single 2.4GHz band 802.11 b/g/n module designed for portable and battery powered applications that need Wi-Fi connectivity. WF111 integrates an IEEE 802.11 b/g/n radio, antenna or U.FL antenna connector and SDIO host interface. WF111 provides a low cost and simple Wi-Fi solution for devices that run an operating system and a TCP/IP stack on-board, but still offers the benefits of a module – small form factor, certifications, and easy integration. Bluegiga also offers WF111 drivers for the Linux and Android operating systems. WF111 has hardware support for Wi-Fi encryption protocols and for many co-existence schemes that enable exceptional performance when using IEEE 802.11 and *Bluetooth* simultaneously.

Key features

- IEEE 802.11 b/g/n radio
    - Single 2.4 GHz band
    - Symbol rate up to 72.2Mbps
- Integrated antenna or U.FL connector
- Support for WEP, WPA and WPA2 encryption
- Software Access Point mode up to eight clients
- Advanced *Bluetooth* co-existence support
- Temperature range: -40°C - +85°C
- SDIO host interface
- Linux and Android operating system drivers for ARM, x86 and PowerPC processors
- Low power solution designed for mobile and battery power applications
- Dimensions: 12.0 x 19.0 x 2.1 mm
- CE, FCC, IC, South Korea and Japan qualified

Figure 10: WF111 Wi-Fi Module

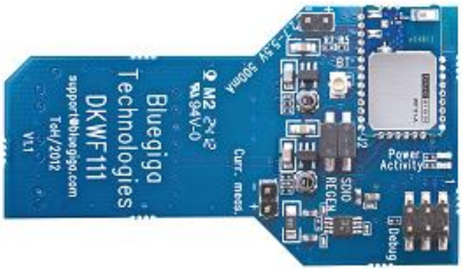
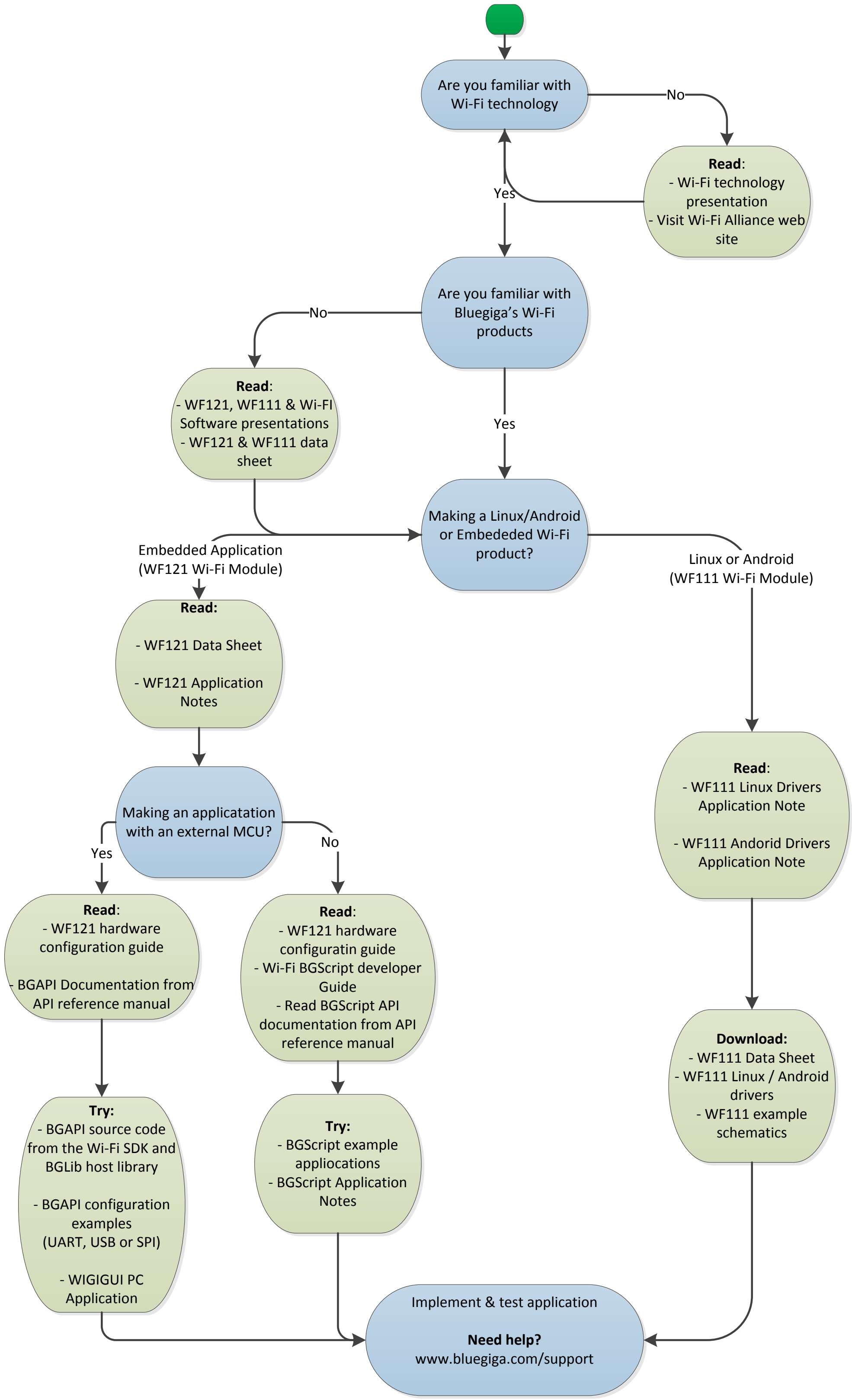


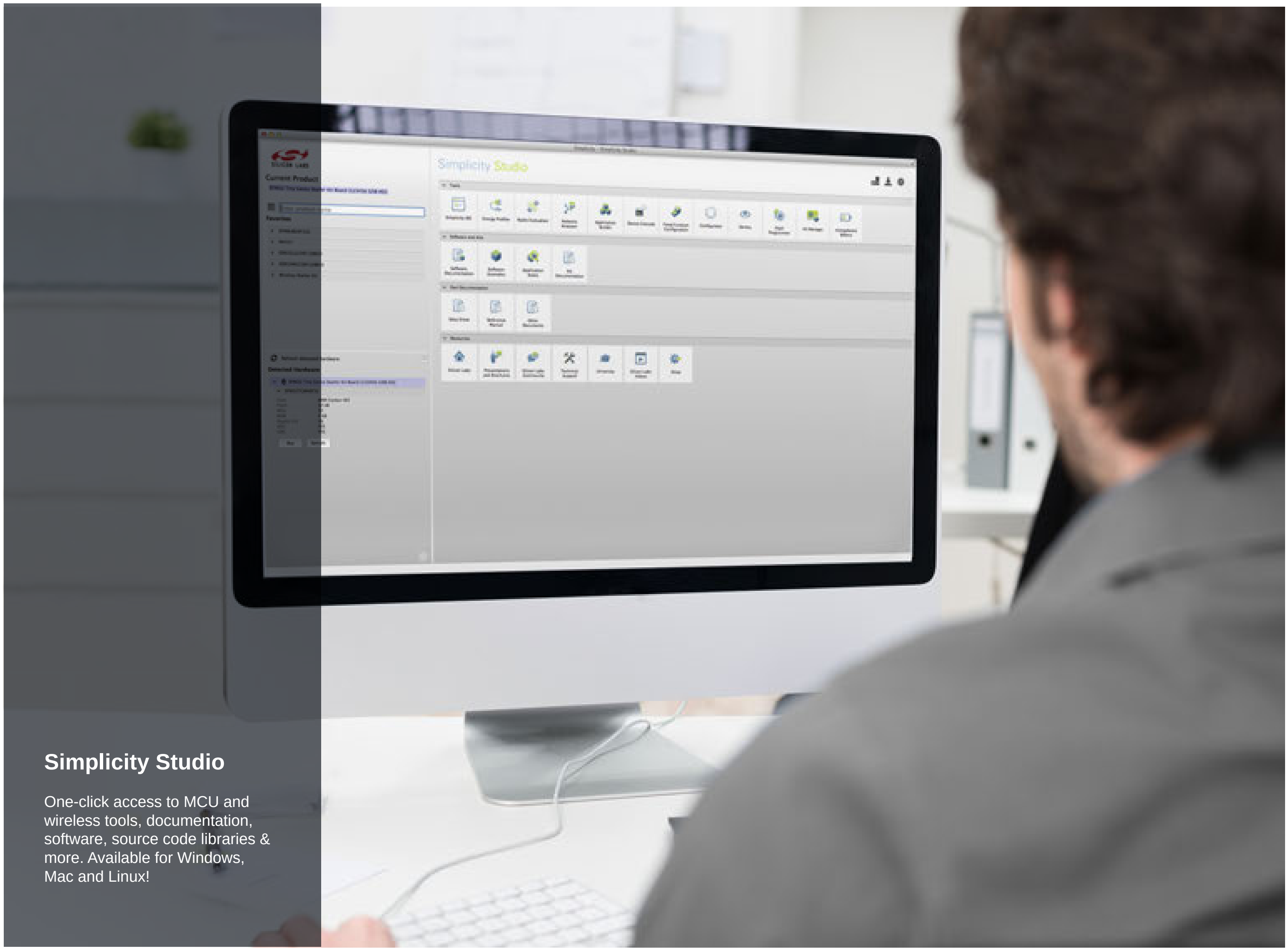
Figure 11: WF111 Development Kit

5 Developing with Bluegiga Wi-Fi Products



Are you familiar with Wi-Fi technology
No
Read: - Wi-Fi technology presentation - Visit Wi-Fi Alliance web site
Yes
Are you familiar with Bluegiga’s Wi-Fi products
No
Read: - WF121, WF111 & Wi-FI Software presentations - WF121 & WF111 data sheet
Yes
Making a Linux/Android or Embededed Wi-Fi product?
Embedded Application (WF121 Wi-Fi Module)
Linux or Android (WF111 Wi-Fi Module)
Read: - WF121 Data Sheet - WF121 Application Notes
Making an applicatation with an external MCU?
Yes
No
Read: - WF121 hardware configuration guide - BGAPI Documentation from API reference manual
Read: - WF121 hardware configuratin guide - Wi-Fi BGScript developer Guide - Read BGScript API documentation from API reference manual
Read: - WF111 Linux Drivers Application Note - WF111 Andorid Drivers Application Note
Try: - BGAPI source code from the Wi-Fi SDK and BGLib host library - BGAPI configuration examples (UART, USB or SPI) - WIGIGUI PC Application
Try: - BGScript example appliocations - BGScript Application Notes
Download: - WF111 Data Sheet - WF111 Linux / Android drivers - WF111 example schematics
Implement & test application
Need help? www.bluegiga.com/support

Figure 12: Developing with Bluegiga Wi-Fi Products



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are not designed or authorized for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world’s most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com