S

NINA-W1 series

Stand-alone Wi-Fi, Bluetooth and multiradio modules

User Manual

Abstract

This document describes the system integration of NINA-W1 series stand-alone modules, which includes the NINA-W13 (Wi-Fi) and NINA-W10 and NINA-W15 series (multiradio) and NINA-B2 (Bluetooth) modules. These modules feature a number of useful embedded security features, including secure boot that ensures that only authenticated software can run on the module. NINA-W1 modules are ideal for critical IoT applications where security is important. The modules connect to a host system using UART, high-speed RMII, or GPIO interfaces.

Document Information

Title	NINA-W1 series
Subtitle	Stand-alone Wi-Fi, Bluetooth and multiradio modules
Document type	User Manual

This document applies to the following products:

Product name
NINA-W106
NINA-W136
NINA-W156
NINA-B226

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.

The information contained herein is provided “as is” and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

Contents

1 System description

1.1 Overview

The NINA-W1 series of wireless and multiradio MCU IoT is suitable for industrial markets where security is important. NINA-W1 includes the following stand-alone modules:

Model	Description
NINA-W13 series	Wireless MCU modules integrate a powerful microcontroller (MCU) and a Wi-Fi radio for wireless communication. NINA-W13x modules come with pre-flashed application software and support 802.11b/g/n in the 2.4 GHz ISM band. Host systems set up and control the modules through an AT command interface to reduce the time and complexity of including Wi-Fi connectivity into your application designs. NINA-W13x modules offer top-grade security with secure boot functionality that ensures that applications start only with the original u-blox software, u-connectXpress.
NINA-W10 series	Multiradio MCU modules integrate a powerful microcontroller (MCU) and radio for wireless communication. With open CPU architecture, NINA-W10 series modules are ideal for advanced applications that run on dual core 32-bit MCUs. The radio provides support for Wi-Fi 802.11b/g/n in the 2.4 GHz ISM band, Bluetooth BR/EDR, and Bluetooth low energy communication. Leveraging integrated cryptographic hardware accelerators, NINA-W10 series modules offer top-grade security with secure boot functionality that ensures that applications start only with the original u-blox software, u-connectXpress.
NINA-W15 series	NINA-W15x modules have similar performance as NINA-W10x modules, but come with pre-flashed application software. Serving as a multiradio gateway, these modules provide support for Wi-Fi 802.11b/g/n and dual-mode Bluetooth (Bluetooth BR/EDR low energy v4.2). Host systems set up and control the modules through an AT command interface to reduce the time and complexity of including Wi-Fi connectivity into your application designs. NINA-W15x series modules offer top-grade security with secure boot functionality that ensures that applications start only with the original u-blox software, u-connectXpress.
NINA-B2 series	The NINA-B2 series are small stand-alone dual-mode Bluetooth modules designed for ease-of-use and integration in professional applications where security is important. The modules are delivered with u-blox connectivity software, which provides support for both peripheral and central roles, Serial Port Profile, GATT client and server, beacons, u-blox Bluetooth low energy Serial Port Service – all configurable from a host using AT commands.

⚠ NINA-W106, NINA-W136, NINA-W156 and NINA-B226 approvals are currently pending.

1.2 CPU

NINA-W1 series modules use a dual-core system that includes two Harvard Architecture Xtensa LX6 CPUs with maximum 240 MHz internal clock frequency. The internal memory of NINA-W1 supports:

- 448 kB ROM for booting and core functions
- 520 kB SRAM for data and instruction
- 16 or 32 Mbit FLASH memory for code storage, including hardware encryption to protect programs and data.
- 1 kbit EFUSE (non- erasable memory) for MAC addresses, module configuration, flash-encryption, and Chip-ID.

Open CPU variants (NINA-W10) also support external FLASH and SRAM memory through a Quad SPI interface.

1.3 Operating modes

1.3.1 Power modes

NINA-W1 series modules are power efficient devices capable of operating in different power saving modes and configurations. Different sections of the modules can be powered off when they are not needed, and complex wake up events can be generated from different external and internal inputs. For the lowest current consumption modes an external LPO clock is required (interface available for NINA-W10 series modules and NINA-W156).

1.4 Supply interfaces

1.4.1 Module supply design (VCC)

NINA-W1 series modules include an integrated Linear Voltage converter that transforms the supply voltage. The output of the converter, presented at the **VCC** pin, provides a stable system voltage.

1.4.2 Digital I/O interfaces reference voltage (VCC_IO)

NINA-W1 series modules include an additional voltage supply input for setting the I/O voltage level.

A separate **VCC_IO** pin enables module integration in many applications with different voltage supply levels (1.8 V or 3.3 V for example) without level converters. NINA-W1 series modules currently support 3.3 V IO levels only.

1.4.3 VCC application circuits

The power for NINA-W1 series modules is applied through the VCC pins. These supplies are taken from either of the following sources:

- Switching Mode Power Supply (SMPS)
- Low Drop Out (LDO) regulator

An SMPS is the ideal design choice when the available primary supply source is of a higher value than the operating supply voltage of the module. This offers the best power efficiency for the application design and minimizes the amount of current drawn from the main supply source.

⚠ When taking VCC supplies from an SMPS make sure that the AC ripple voltage is kept as low as possible at the switching frequency. Design layouts should focus on minimizing the impact of any high-frequency ringing.

Use an LDO linear regulator for primary VCC supplies that have a relatively low voltage. As LDO linear regulators dissipate a considerable amount of energy, LDOs are not recommended for the step down of high voltages.

DC/DC efficiency should be regarded as a trade-off between the active and idle duty cycles of an application. Although some DC/DC devices achieve high efficiency at light loads, these efficiencies typically degrade as soon as the idle current drops below a few milliamps. This can have a negative impact on the life of the battery.

If decoupling capacitors are needed on the supply rails, it is best practice to position these as close as possible to the NINA-W1 series module. The power routing of some host system designs makes decoupling capacitance unnecessary.

For electrical specifications, see the NINA-W1 series Data Sheets.

1.5 System function interfaces

1.5.1 Boot strapping pins

For normal operation, several boot configuration pins must be in their default state during the module boot. Left unconnected in the application design, the default state of these pins is automatically chosen with internal pull-up or pull-down resistors in the module. See data sheet for more information.

Care must be taken if an RMII interface is to be included in the application design. **Pin 25** and **pin 27** connect to the RMII. It is important that both of these pins are in the correct state during the module boot and before the RMII interface turns on. For connection information, see section 1.6.2.1.

☞ On NINA-W13/W15 modules, **pin 25** and **pin 27** must be in default state during the boot.

☞ On NINA-W10 modules, **pin 27** is used to enter the ESP bootloader. Consequently, this pin must be exposed on a pin header (or similar) to flash the module.

Pin 32 is used to control the Universal Asynchronous Serial Interface (UART) debug printout. With this pin left unconnected, NINA-W1 prints some short debug information when it boots before the module software starts up. For u-connectXpress software, this means before +STARTUP. To disable the printout, this pin must be pulled low during start-up.

1.6 Data interfaces

1.6.1 Universal asynchronous serial interface (UART)

For data communication and firmware upgrade, NINA-W1 series modules support an interface comprised of three UARTs. Each UART supports the following signals:

- Data lines (**RXD** as input, **TXD** as output)
- Hardware flow control lines (**CTS** as input, **RTS** as output)
- **DSR** and **DTS** set and indicate the system modes

You can use the UARTs in 4-wire mode with hardware flow control, or in 2-wire mode with **TXD** and **RXD** only. In 2-wire mode, **CTS** must be connected to the GND on the NINA-W1 module.

For further information, see chapter 2.

The u-connectXpress software adds the **DSR** and **DTR** pins to the UART interface. Not used as they were originally intended, these pins are used to control the state of NINA modules.

Depending on the configuration, **DSR** can be used to:

- Enter command mode
- Disconnect and/or toggle connectable status
- Enable/disable the rest of the UART interface
- Enter/wake up from sleep mode

The functionality of the **DSR** and **DTR** pins are configured by AT commands. For further information about these commands, see the u-blox Short Range Modules AT commands manual.

Table 1 shows the default settings of UART ports when using u-connectXpress software.

Interface	**Default configuration**
UART interface	115200 baud, 8 data bits, no parity, 1 stop bit, hardware flow control

Table 1: UART port default settings

It is advisable to make UART0 available either for test points or connect it to a header for firmware upgrade. The IO level of the UART follows **VCC_IO**.

For information about UART interface characteristics, see the NINA-W1 series Data Sheets.

1.6.2 Ethernet (RMII+SMI)

⚠ On NINA-W13, RMII is supported from software version 2.0.0 onwards.

⚠ On NINA-W15, direct MAC to MAC connection will be supported when the module reaches Engineering Sample status.

NINA-W1 series modules include a full RMII for Ethernet MAC to PHY communication using the included Station Management Interface (SMI). The RMII and SMI uses nine signals in total. The interface requires an external 50 MHz clock source either from a compatible PHY chip or from an external oscillator.

The two-wire SMI is used to configure the PHY chip. It uses a clock line and a data line to setup the internal registers on PHY chip.

The pin multiplexing of the RMII interface imposes limitations in the functionality of NINA-W13/W15 series module when using the interface. The following functions are turned off when RMII communication is initiated:

- Red, Green and Blue LEDs are disabled
- UART is run without flow control as CTS and RTS functionality is disabled
- DSR and DTR functionality is disabled

A 1.5 kΩ pull up resistor must be added to MDIO pin.

1.6.2.1 Startup precautions

To ensure that the boot mode is not entered inadvertently, the RMII_CLK input (GPIO27) is multiplexed with the ESP boot pin and must be held high 1.2 ms after the reset signal is released.

EVK-NINA-W1 uses two buffers and a low pass filter to delay the reset signal going to the PHY circuit, as shown in Figure .

This delays the clock so that it starts a short time after the module is released from reset.

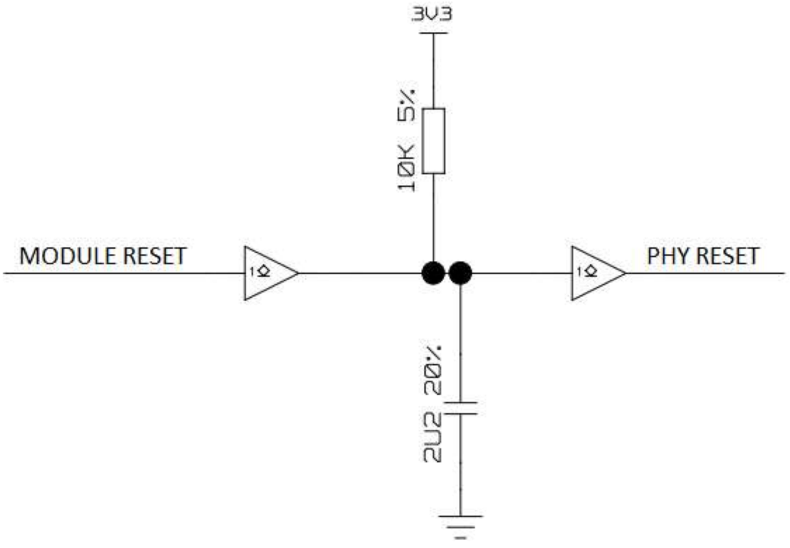


Figure 1: RMII clock delay circuit

u-connectXpress software senses the RMII_CLK input (GPIO27) at startup. If an RMII clock is discovered, then Ethernet communication is initiated.

During startup of NINA-W1 series modules the RMII clock must be started within 100 us, but not before an initial delay of 1.2 ms.

1.6.2.2 MAC to PHY connection

When connecting NINA-W1 series modules to an external PHY circuit, both the RMII and SMI interfaces must be connected. The default PHY address (0x1) must be configured on the PHY side. Follow the recommendations of your chosen PHY chip supplier for implementation details.

An example of a PHY implementation is shown in Figure . PHY KSZ8081 is recommended and used on the EVK-NINA-W1.

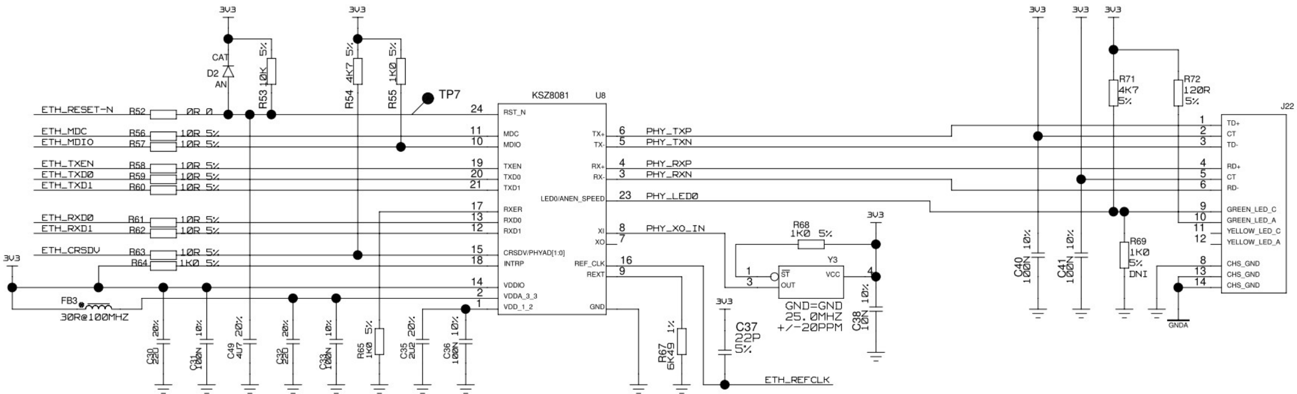


Figure 2: EVK-NINA-W1 Ethernet PHY implementation

1.6.2.3 MAC to MAC connection

When connecting NINA-W1 series modules using a direct MAC to MAC connection, the SMI interface can be left unconnected. Depending on the routing of the RMII interface on the host PCB, termination resistors can also be needed.

An external 50 MHz oscillator is needed while running an MAC to MAC connection.

1.7 W1x6 integrated antennas

To simplify integration, W1x6 modules are equipped with an integrated antenna. An integrated antenna design means there is no need for an RF trace design on the host PCB. This means less effort is required in the test lab.

The NINA-W1x6 modules are equipped with a PCB trace antenna that is based on technology licensed from Proant AB.

1.8 Reserved pins (RSVD)

Do not connect the reserved (**RSVD**) pin. Reserved pins are allocated for future interfaces and functionality.

1.9 GND pins

Good electrical connection of module GND pins, using solid ground layer of the host application board, is required for correct RF performance. Firm connections provide a thermal heat sink for the module and significantly reduce EMC issues.

2 Software

2.1 NINA-W13 and NINA-W15 u-connectXpress software

NINA-W13/W15 stand-alone modules are delivered with embedded u-connectXpress software.

Using industry-standard AT commands, this is the software that manages the combination of Bluetooth, Bluetooth low energy and Wi-Fi connectivity supported in NINA-W13 and NINA-W15 standalone modules, specifically:

- Wi-Fi (NINA-W13 and NINA-W15)
- Bluetooth (NINA-W15)
- Bluetooth low energy (NINA-W15)

For information about the features, capabilities and use of u-connectXpress software, see the u-blox Short range modules AT commands manual.

Typical examples of the applications and use cases supported by NINA-W13 and NINA-W15 series modules include:

- Gateway connection of Bluetooth low energy sensors to the cloud over Wi-Fi or Ethernet
- Bridge communication over serial, Wi-Fi, PPP, or Ethernet interfaces
- Wi-Fi hotspot connection using Local Area Network or Tethering
- Device configuration using Bluetooth or Wi-Fi connected smartphones
- Secure cloud connection using TLS and MQTT protocols

Figure shows the structure of the embedded u-connectXpress software delivered in NINA-W13 and NINA-W15 standalone modules.

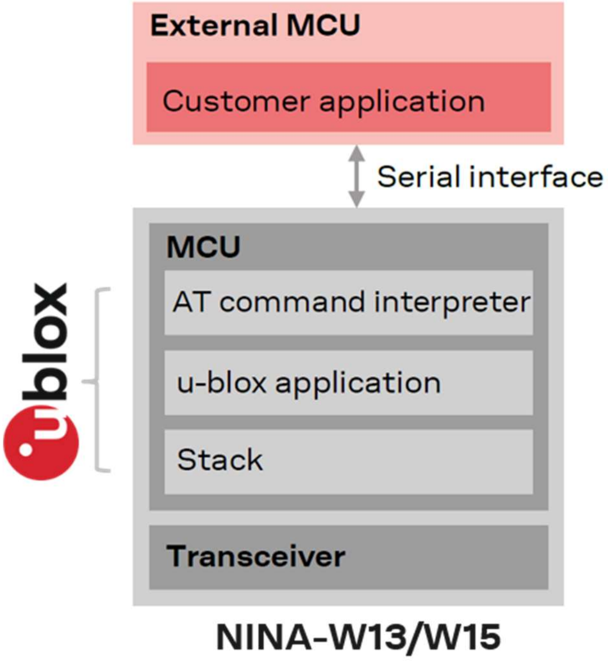


Figure 3: NINA-W13/W15u-connectXpress software structure

2.2 SDK for open-CPU modules

As NINA-W10 open-CPU modules are delivered without flashed software, you develop your application design using the utilities and device-level APIs supported by the module chipset supplier. The ESP-IDF Software Development Kit is available from the Expressif website. It bundles the Wi-Fi stack and the broad range of drivers and libraries necessary for building your development environment. See also section 2.5.

Figure shows the architecture of NINA-W10 open-CPU software in relation to the MCU, transceiver and ESP-IDF SDK.

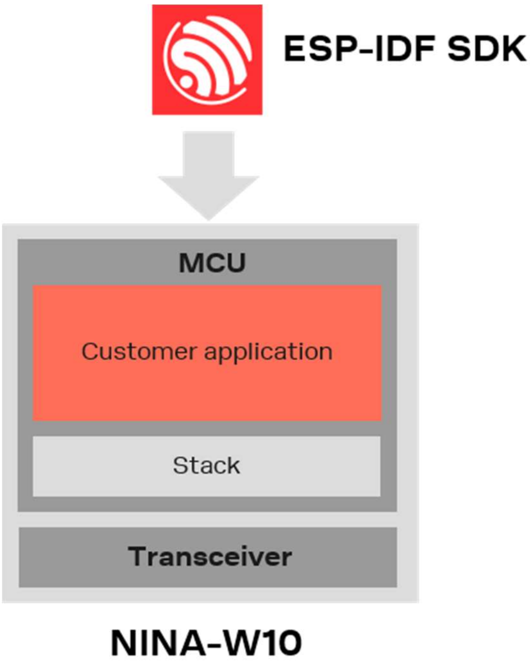


Figure 4: NINA-W10 open CPU software

2.3 Flashing stand-alone modules with s-center

ublox s-center client software provides a convenient tool with which to configure u-blox standalone modules. It runs on PCs running Windows XP onwards (x86 and x64) with Net Framework 4.5 or later and is available for download from www.u-blox.com. For further information about how to use this tool, see the s-center user guide.

You use s-center to upgrade the embedded u-connectXpress software in NINA-W13 and NINA-W15 stand-alone modules. The s-center client connects to ublox standalone modules using the XMODEM protocol using the s-center tool.

For the purpose of upgrading the connectivity software, the following pins are exposed as either headers or test points:

NINA-W13 and NINA-W15

- Mandatory:
  - UART (RXD, TXD, CTS, RTS)
- Additionally recommended:
  - RESET_N
  - SWITCH_1 and SWITCH_2

NINA-W10 header

- Mandatory:
  - SWD
  - ESP_BOOT (GPIO27)
- Additionally recommended:
  - RESET_N

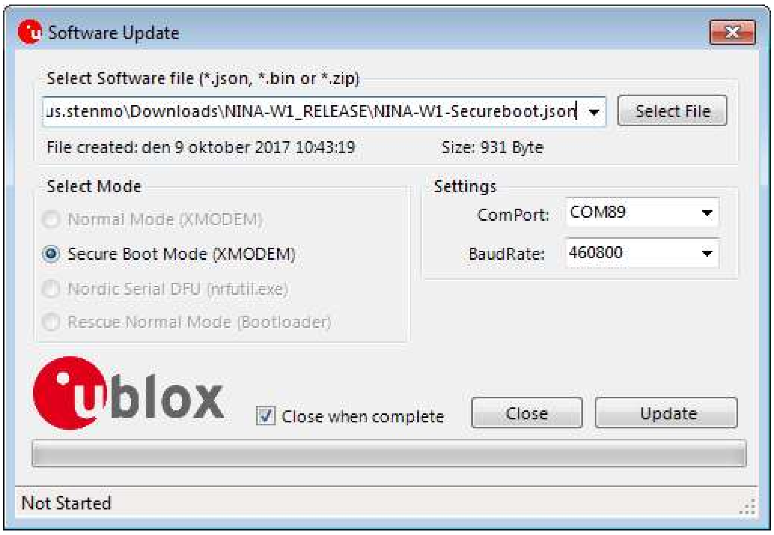
2.4 Updating u-connectXpress software with s-center

The u-connectXpress software, flashed into NINA-W13/W15 modules prior to delivery, is used to validate the hardware, bootloader, and the binary image. The u-connectXpress software runs only on validated hardware.

Updates of the u-connectXpress software is available for download from www.u-blox.com. The software is delivered in a zip container file, for example, `NINA-W1xX_SW1.0.0.zip`.

To upload the latest u-connectXpress software to the module:

1. Download and unpack the zip container, `NINA-W1xX_SW1.0.0.zip,` to your Windows workstation.
2. Open the s-center client software.
3. From the client, navigate to the `.json` file in the unpacked u-connect archive and select **Update**. The s-center handles the download using information contained in the *.json file without any further interaction is needed from the user. See also Figure 5.



Software Update

Select Software file (*.json, *.bin or *.zip)

us.stenmo\Downloads\NINA-W1_RELEASE\NINA-W1-Secureboot.json

File created: den 9 oktober 2017 10:43:19

Size: 931 Byte

Select Mode

○ Normal Mode (XMODEM)

● Secure Boot Mode (XMODEM)

○ Nordic Serial DFU (nrfutil.exe)

○ Rescue Normal Mode (Bootloader)

Settings

ComPort: COM89

BaudRate: 460800

☑ Close when complete

Not Started

Figure 5: Software Update using s-center

☞ Secure boot functionality is supported in u-connect v4.7 and above.

Updating To manually start the download using a software without using s-center, enter the following AT commands to start updating the NINA-W13/W15 u-connectXpress:

```
AT+UFWUPD=<mode>,<baud>,<image id>,<image size>,<base64 encoded signature>,<image
name>,<flags>
```

Sample parameters that can be used while doing the flash update is provided below:

```
AT+UFWUPD=0,115200,0,651840,jzlRIkg37ir/pVpDKVrPot2ZdsaNvUtSYP2pDAUVJc7iQI9yzIo8V
Fv8C1olP/9I4UJ4WmgC5oRay4AC0V8jRJSFFX/wop6x/sBJGOeDEu7yC/s0+Oj7CLs4TzNbiRqK0zLwKR
iHohgVyzWqhwKFpmcxcDXphjkCTIvpffY8TwDLzkowuuD59R+sQCueJtBHBg9KDB3TOs8bsXLaVtT2x1r
LfMg8/pb+BPQEK9NcNB4hbp693ATivYE3cmxzWykIjEje819SIRGhHFt0wAsqh7WFgSJYNgDi5cLdOYz+
r1+j7+l4RqrMl/A/QYyWS9z0Q15QcJ3GlAJlXYa5v/ISjA==,nina-w1-debug,rwx
```

When a “C” character is received from NINA-W13/W15, the XMODEM download is ready to begin from the host.

For more information about the parameters, see the Software update +UFWUPD command in u-blox Short Range Modules AT Commands Manual.

2.5 Developing and flashing NINA-W10 open-CPU software

As the u-connectXpress software embedded in NINA-W13/15 series modules is not available for use with the NINA-W10 open CPU series, you use Expressif SDK utilities and device-level APIs to develop your application hardware.

For the latest Espressif SDK documentation, see http://esp-idf.readthedocs.io/en/latest/get-started/index.html.

This URL provides information on how to set up the software environment using the hardware based on the Espressif ESP32 such as NINA-W10 and also how to use the latest ESP-IDF (Espressif IoT Development Framework), which might have been changed since the publication of this document.

The following must be setup in order to compile, flash, and execute a program on NINA-W10:

- Setup Toolchain
  - Windows, Mac, and Linux is supported
- Get ESP-IDF
  - Download the GIT repository provided by Espressif
- Setup Path to ESP-IDF
  - The toolchain program can access the ESP-IDF using IDF_PATH environment variable
- Build and Flash
  - Start a Project, Connect, Configure, Build and Flash a program

More information about this is available at - http://esp-idf.readthedocs.io/en/latest/index.html

2.5.1 Setup the ESP-IDF v3 toolchain

☞ ESP-IDF v3 toolchain can be used on NINA-W101/NINA-W102, but has not been verified on NINA-W106. On NINA-W106, use the ESP-IDF v4 toolchain.

To start development with ESP32, it is recommended to use a prebuilt toolchain. Currently, Windows, Mac, and Linux is supported. The example in this document will use a Toolchain for running Windows, that is, a bash shell window. The toolchain contains all programs and compiler to build an application.

The toolchain for Windows can be downloaded from
https://dl.espressif.com/dl/esp32_win32_msys2_environment_and_toolchain-20170918.zip

Unzip the file to `c:\ msys32`. This path is assumed in the following examples, but it can be located in another folder as well. The file size is around 500 MB.

Start the bash shell using the “`mingw32.exe`” (“`mingw64.exe`” is currently not supported).

☞ If you encounter any issues, use the “`autorebase.bat`” and the “`msys2_shell.cmd`” shortcuts. This will reset the path variable with a Cygwin installation on some computers, which can have problems with the path to the compiler or the python tool.

2.5.2 Get ESP-IDF v3

☞ ESP-IDF v3 can be used on NINA-W101/NINA-W102, but has not been verified on NINA-W106. On NINA-W106, use the ESP-IDF v4.

The source files for Espressif ESP-IDF repository is located on github at https://github.com/espressif/esp-idf.

To download the files, open the "mingw32.exe", navigate to the directory where you want to have the ESP-IDF (like c:\git), and clone it using "git clone" command.

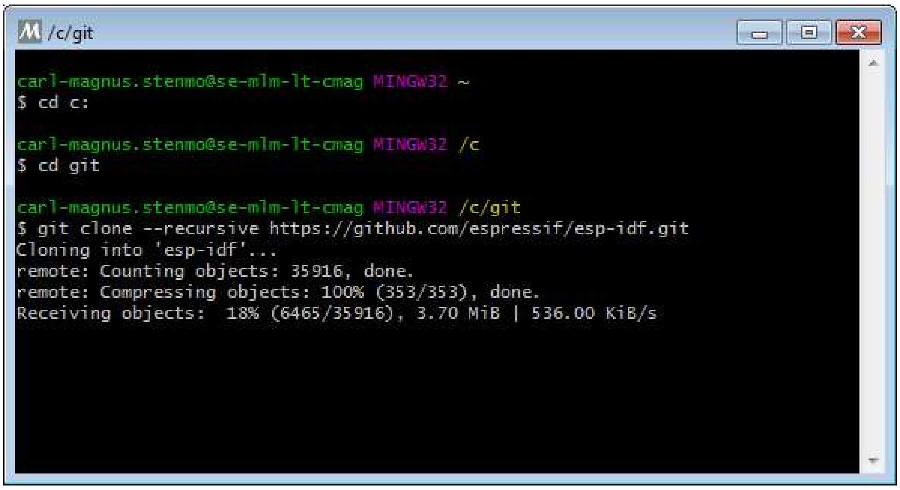
☞ Use the "--recursive" parameter

In this example, the esp-idf repository will be created in the c:\git folder.

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

To checkout a specific tag such as v3.1, use the following command as shown in the example below:

```
git clone https://github.com/espressif/esp-idf.git esp-idf-v3.1
cd esp-idf-v3.1/
git checkout v3.1
git submodule update --init --recursive
```

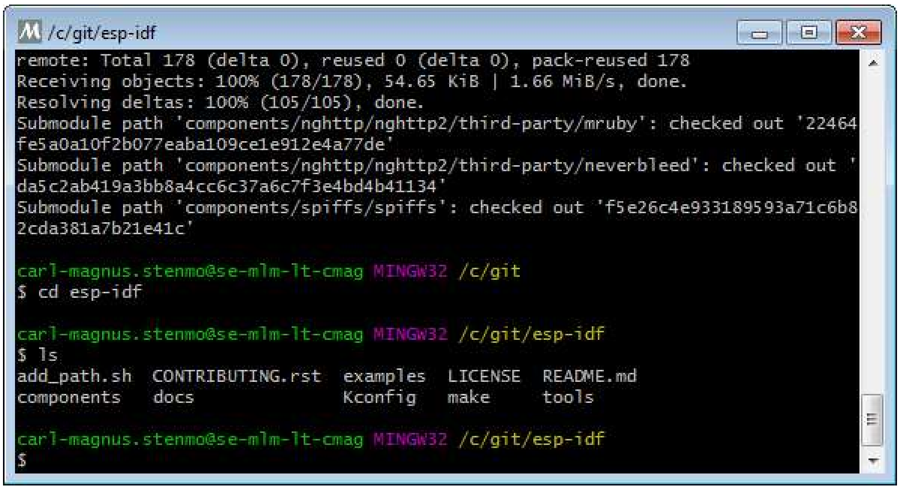


```
carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 ~
$ cd c:

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c
$ cd git

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git
$ git clone --recursive https://github.com/espressif/esp-idf.git
Cloning into 'esp-idf'...
remote: Counting objects: 35916, done.
remote: Compressing objects: 100% (353/353), done.
Receiving objects:  18% (6465/35916), 3.70 MiB | 536.00 KiB/s
```

Figure 6: Example of the git clone of ESP-IDF

Go to the new folder by typing "cd esp-idf" and then type "ls" to show the folder content.

```
cd esp-idf
ls
```



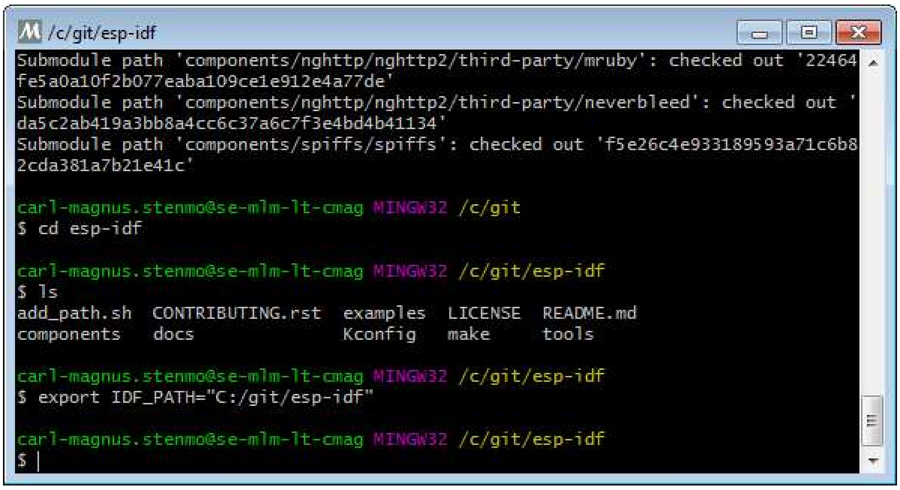
```
remote: Total 178 (delta 0), reused 0 (delta 0), pack-reused 178
Receiving objects: 100% (178/178), 54.65 KiB | 1.66 MiB/s, done.
Resolving deltas: 100% (105/105), done.
Submodule path 'components/nghttp/nghttp2/third-party/mruby': checked out '22464
fe5a0a10f2b077eaba109ce1e912e4a77de'
Submodule path 'components/nghttp/nghttp2/third-party/neverbleed': checked out '
da5c2ab419a3bb8a4cc6c37a6c7f3e4bd4b41134'
Submodule path 'components/spiffs/spiffs': checked out 'f5e26c4e933189593a71c6b8
2cda381a7b21e41c'

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git
$ cd esp-idf

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git/esp-idf
$ ls
add_path.sh  CONTRIBUTING.rst  examples  LICENSE  README.md
components   docs              Kconfig   make     tools

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git/esp-idf
$
```

Figure 7: Verification of all the downloaded files

2.5.3 Setup path to ESP-IDF

The toolchain for the ESP-IDF uses the IDF_PATH environment variable. This variable must be set up for building the projects.

```
export IDF_PATH="C:/git/esp-idf"
```



```
Submodule path 'components/nghttp/nghttp2/third-party/mruby': checked out '22464
fe5a0a10f2b077eaba109ce1e912e4a77de'
Submodule path 'components/nghttp/nghttp2/third-party/neverbleed': checked out '
da5c2ab419a3bb8a4cc6c37a6c7f3e4bd4b41134'
Submodule path 'components/spiffs/spiffs': checked out 'f5e26c4e933189593a71c6b8
2cda381a7b21e41c'

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git
$ cd esp-idf

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git/esp-idf
$ ls
add_path.sh  CONTRIBUTING.rst  examples  LICENSE  README.md
components   docs              Kconfig   make     tools

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git/esp-idf
$ export IDF_PATH="C:/git/esp-idf"

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/git/esp-idf
$ |
```

Figure 8: Setting up the PATH variable

2.5.4 Building and flashing ESP-IDF v3

The environment is now ready to build and flash a project. In this case, we use "hello world" as a sample project.

This project will print out "Hello World" ten times on the UART and then reboot.

To build this sample project, go to the "hello world" folder using the following command:

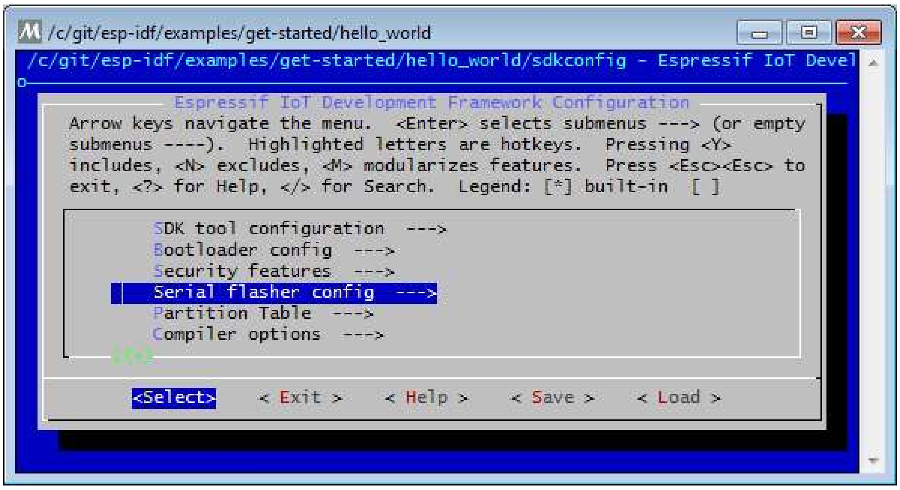
```
cd examples/get-started/hello_world
```

Plug in NINA-W10 to the PC and note down the com port number with which it is connected. In this example, the com port number is assumed to be "COM10".

Now enter "`make menuconfig`" to open the ESP-IDF configuration window. You can select and modify a lot of configuration options about the environment using this tool; in this example, only the com port that is used to flash NINA-W10 is modified.

```
make menuconfig
```

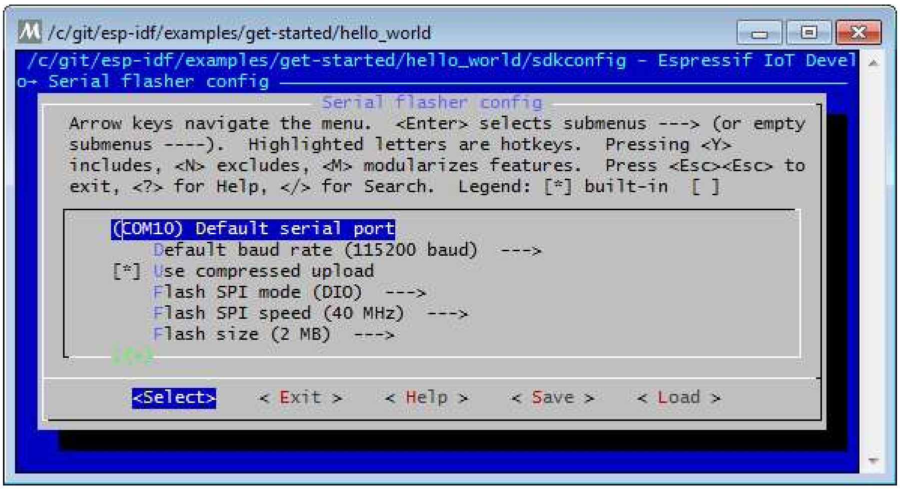
Use the arrow keys to navigate and select the "Serial flasher config" as shown in Figure



```
/c/git/esp-idf/examples/get-started/hello_world/sdkconfig - Espressif IoT Devel
o
              Espressif IoT Development Framework Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

          SDK tool configuration  --->
          Bootloader config  --->
          Security features  --->
          Serial flasher config  --->
          Partition Table  --->
          Compiler options  --->
      v(+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 9: Screenshot that shows selection of “Serial flasher config”

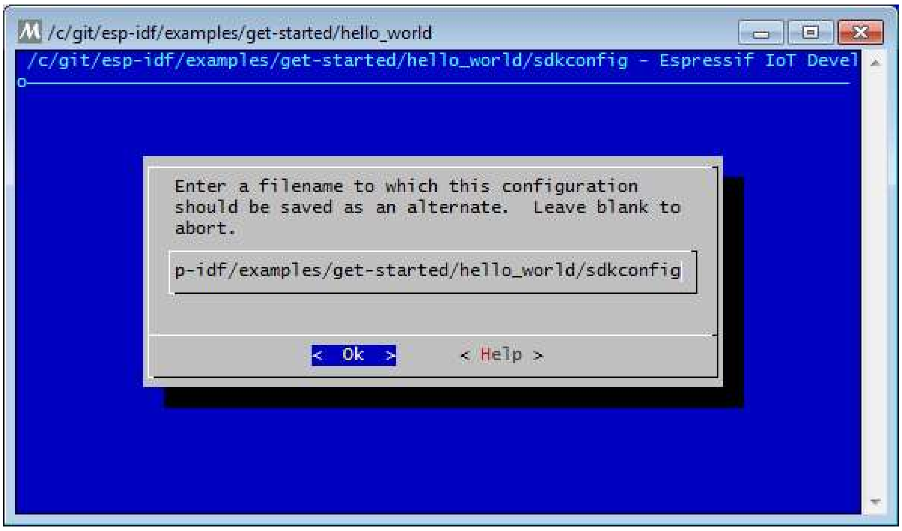
Enter the com port name; in this case, enter “COM10”, and click OK.



```
/c/git/esp-idf/examples/get-started/hello_world/sdkconfig - Espressif IoT Devel
o→ Serial flasher config
                       Serial flasher config
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

      (COM10) Default serial port
          Default baud rate (115200 baud)  --->
      [*] Use compressed upload
          Flash SPI mode (DIO)  --->
          Flash SPI speed (40 MHz)  --->
          Flash size (2 MB)  --->
      v(+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 10: Screenshot that shows selection of the sample com port number ("COM10")

Save the sdkconfig by entering a filename to which this configuration should be saved as shown in Figure .



```
/c/git/esp-idf/examples/get-started/hello_world/sdkconfig - Espressif IoT Devel
o

    Enter a filename to which this configuration
    should be saved as an alternate.  Leave blank to
    abort.

    p-idf/examples/get-started/hello_world/sdkconfig

                 <  Ok  >      < Help >
```

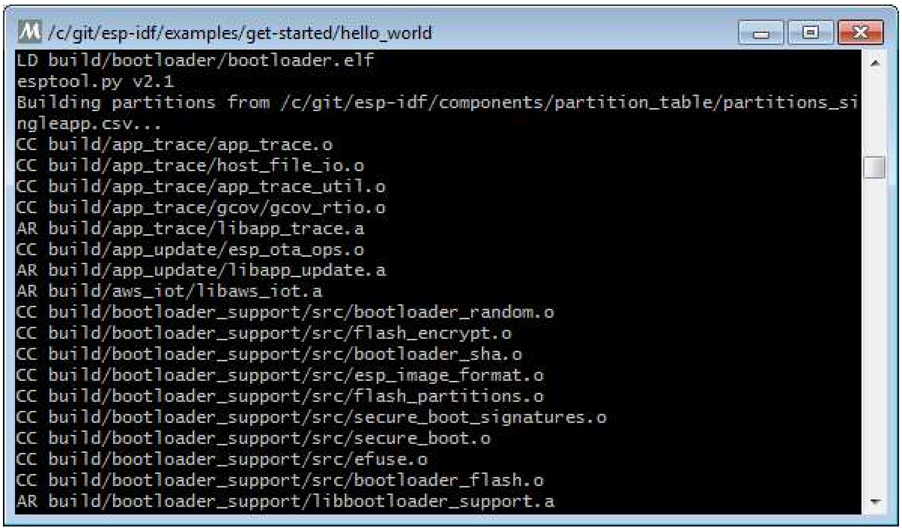
Figure 11: Screenshot after entering the filename for the sdkconfig

☞ Make sure your configuration is saved first and then enter Exit to exit the console.

Now the project is ready to build, but before building and flashing, NINA-W10 should be prepared to accept the downloaded file. This is done by holding the BOOT button while resetting or powering on the board.

Then, enter "`make flash`" to build and flash the NINA-W10 as shown below:

```
make flash
```



```
LD build/bootloader/bootloader.elf
esptool.py v2.1
Building partitions from /c/git/esp-idf/components/partition_table/partitions_si
ngleapp.csv...
CC build/app_trace/app_trace.o
CC build/app_trace/host_file_io.o
CC build/app_trace/app_trace_util.o
CC build/app_trace/gcov/gcov_rtio.o
AR build/app_trace/libapp_trace.a
CC build/app_update/esp_ota_ops.o
AR build/app_update/libapp_update.a
AR build/aws_iot/libaws_iot.a
CC build/bootloader_support/src/bootloader_random.o
CC build/bootloader_support/src/flash_encrypt.o
CC build/bootloader_support/src/bootloader_sha.o
CC build/bootloader_support/src/esp_image_format.o
CC build/bootloader_support/src/flash_partitions.o
CC build/bootloader_support/src/secure_boot_signatures.o
CC build/bootloader_support/src/secure_boot.o
CC build/bootloader_support/src/efuse.o
CC build/bootloader_support/src/bootloader_flash.o
AR build/bootloader_support/libbootloader_support.a
```

Figure 12: Compiling of the example application

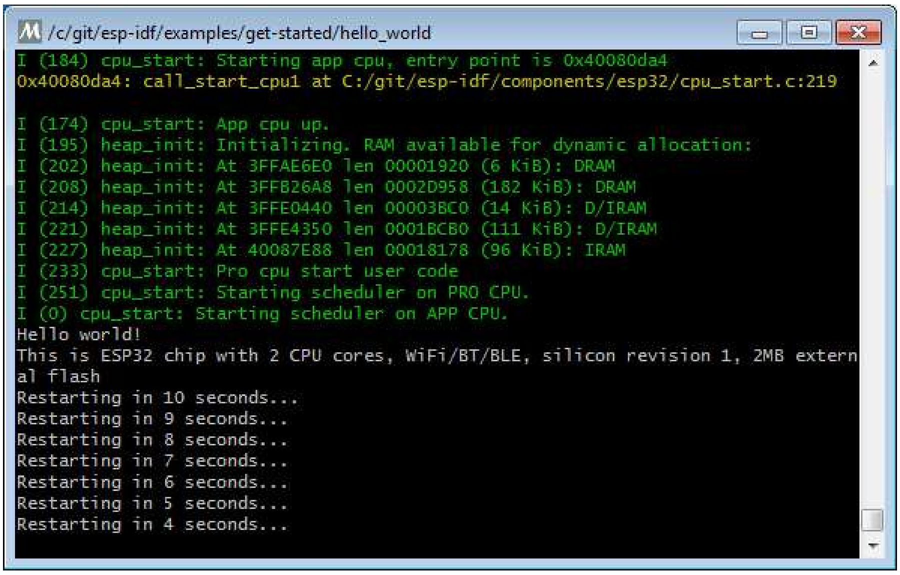
Now, reset the NINA-W10 by clicking the RESET button.

Then, enter "`make monitor`" to open a serial port monitor program to the NINA-W10.

```
make monitor
```

You could also enter "`make flash monitor`" to build and flash and then start the serial port monitor program using only one command.

```
make flash monitor
```



```
I (184) cpu_start: Starting app cpu, entry point is 0x40080da4
0x40080da4: call_start_cpu1 at C:/git/esp-idf/components/esp32/cpu_start.c:219

I (174) cpu_start: App cpu up.
I (195) heap_init: Initializing. RAM available for dynamic allocation:
I (202) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (208) heap_init: At 3FFB26A8 len 0002D958 (182 KiB): DRAM
I (214) heap_init: At 3FFE0440 len 00003BC0 (14 KiB): D/IRAM
I (221) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (227) heap_init: At 40087E88 len 00018178 (96 KiB): IRAM
I (233) cpu_start: Pro cpu start user code
I (251) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is ESP32 chip with 2 CPU cores, WiFi/BT/BLE, silicon revision 1, 2MB extern
al flash
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
Restarting in 6 seconds...
Restarting in 5 seconds...
Restarting in 4 seconds...
```

Figure 131: Hello world example as displayed on the monitor

2.5.5 Using ESP-IDF v4

☞ ESP-IDF v4 is mandatory for NINA-W106.

To use ESP-IDF v4, follow the applicable instructions for your development environment on https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/windows-setup.html

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-setup.html

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/macos-setup.html

Get the latest toolchain and installation instructions by using applicable path above.

After the toolchain has been installed, get and install the ESP-IDF by using the below link and following the given instructions.

The toolchain for the ESP-IDF uses the `IDF_PATH` environment variable. This variable must be set up for building the projects by following the given instructions.

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html#step-2-get-esp-idf

The source files for Espressif ESP-IDF repository is located on github at https://github.com/espressif/esp-idf.

Building and flashing the examples is basically done the same way as for ESP-IDF v3, but instead of `make`, the `idf.py` shall be invoked. The same command-line parameters applies. However, some differences exist:

☞ For NINA-W101/NINA-W102, during `idf.py make menuconfig` also set the configuration flag `CONFIG_SPI_FLASH_USE_LEGACY_IMPL` flag to `Y`, otherwise the application will fail to start.

☞ NINA-W106, during `idf.py make menuconfig` also set the `CONFIG_BOOTLOADER_VDDSDIO_BOOST_1_9V` and `CONFIG_SPI_FLASH_SUPPORT_ISSI_CHIP` configuration flags to `Y`.

☞ On NINA-W106, it may be required to add the `--no-stub` parameter to `esptool.py` when flashing the software and make a manual verification using a second invocation of `esptool.py` in case flashing and verification fails.

2.5.6 Automatic bootloader on NINA-W10 EVK

The “`esptool.py`” flash tool supports automatic entry to the bootloader on the NINA-W10 EVK without pressing the BOOT button and RESET the module. To use this functionality, you need to connect the following pins:

- RESET to IO19 (CTS)
- IO0 (IO zero) to IO26 (DSR)

The jumpers CTS (J14-8) and DSR (J14-7) should also be removed so that they do not interfere.

☞ It is not possible to use the Hardware Flow control or the DSR signals on the UART while using this setup.

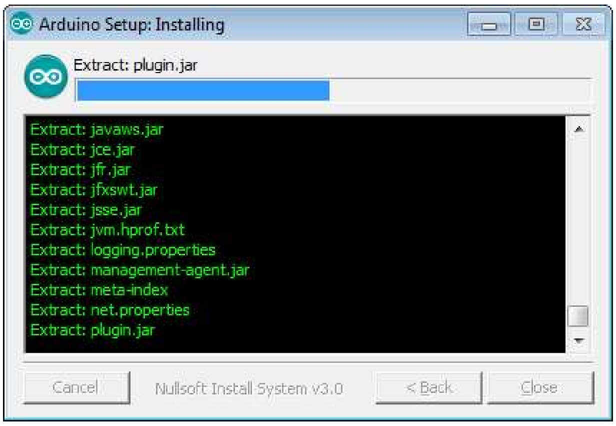
More information about the esptool is available at - https://github.com/espressif/esptool

2.6 Arduino support for NINA-W10

It is possible to use Arduino electronics platform on the NINA-W10. The Arduino platform and open source community provides the possibility to access a lot of third party hardware such as displays and sensors.

2.6.1 Downloading the Arduino IDE

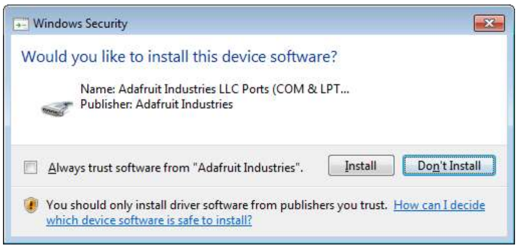
Windows, Mac, and Linux environment are supported. The example below uses the Windows environment. Download the Arduino IDE using the URL - https://www.arduino.cc/en/Main/Software.



```
Extract: javaws.jar
Extract: jce.jar
Extract: jfr.jar
Extract: jfxswt.jar
Extract: jsse.jar
Extract: jvm.hprof.txt
Extract: logging.properties
Extract: management-agent.jar
Extract: meta-index
Extract: net.properties
Extract: plugin.jar
```

Figure 14: Screenshot during installation of the Arduino IDE

Click **Install** button in the dialog box that pops up during installation as shown in the screenshots below:

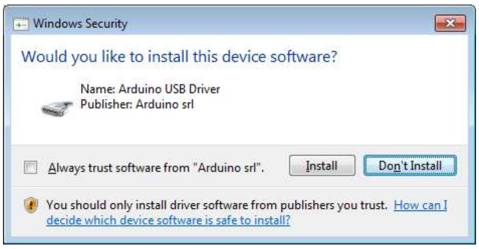


Would you like to install this device software?

Name: Adafruit Industries LLC Ports (COM & LPT...
Publisher: Adafruit Industries

□ Always trust software from "Adafruit Industries".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?



Would you like to install this device software?

Name: Arduino USB Driver
Publisher: Arduino srl

□ Always trust software from "Arduino srl".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Open the Arduino IDE - "`C:\Program Files (x86)\Arduino\arduino.exe`" and then close the program again. Do this to ensure that the folder is created correctly before downloading the Arduino files as mentioned in the next step.

The Arduino IDE user folder is typically located in "`C:\Users\user_name\Documents\Arduino`"

2.6.2 Downloading from the GIT repository

Download from the URL - https://github.com/espressif/arduino-esp32.git

The files should be placed in "`C:\Users\user_name\Documents\Arduino\hardware\espressif\esp32`"

Open the "`mingw32.exe`" located in `c:\msys32`.

The Arduino user folder is normally stored at "`C:\Users\user_name\Documents\Arduino`"

Check if the hardware folder exists. If not, create the same by entering the following command:

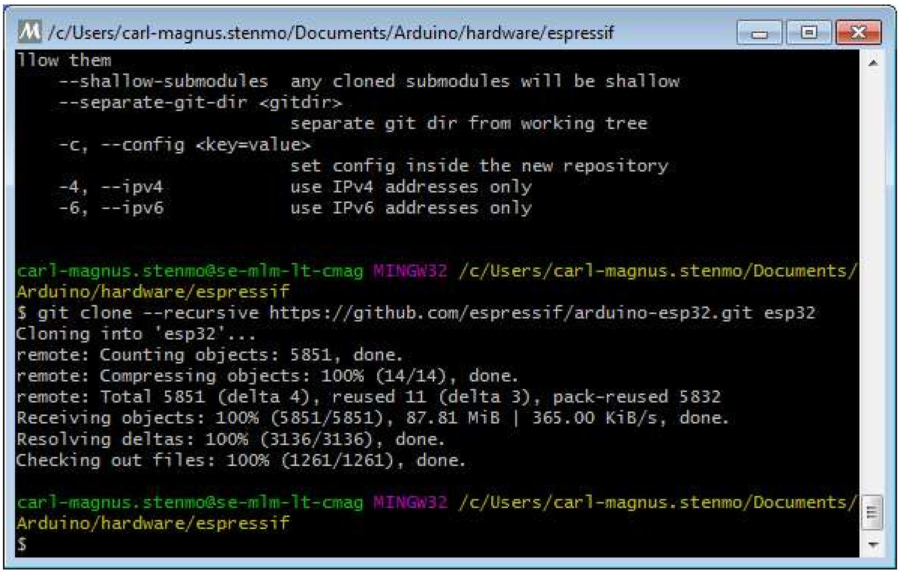
```
mkdir hardware
cd hardware
```

Check if the espressif folder exists; if not, create the same by entering the following command:

```
mkdir espressif
cd espressif
```

Now clone the repository to the folder esp32 folder.

```
git clone --recursive https://github.com/espressif/arduino-esp32.git esp32
```



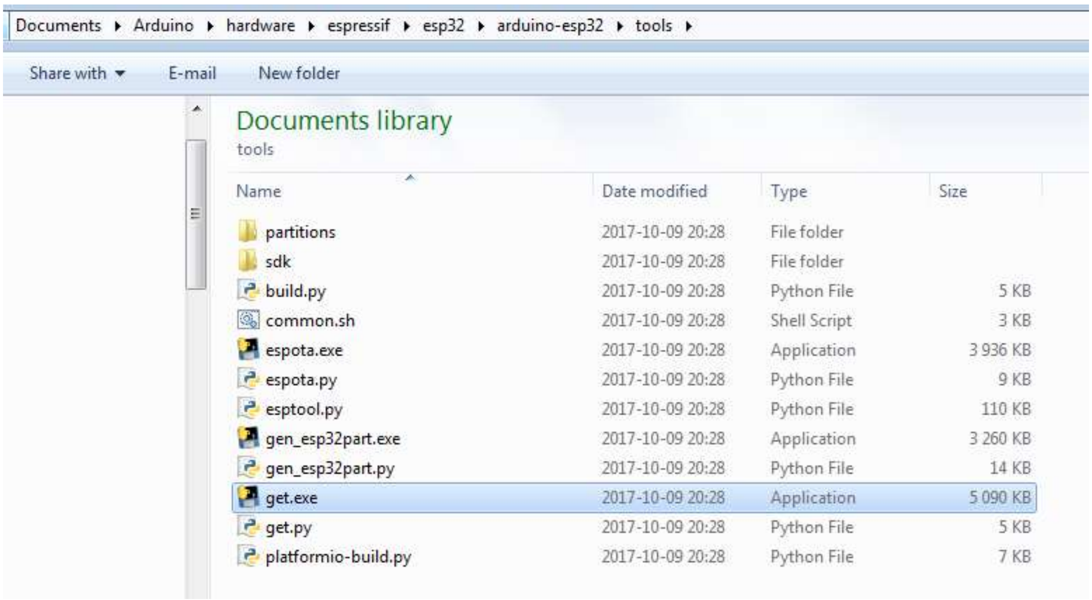
```
/c/Users/carl-magnus.stenmo/Documents/Arduino/hardware/espressif
llow them
    --shallow-submodules  any cloned submodules will be shallow
    --separate-git-dir <gitdir>
                          separate git dir from working tree
    -c, --config <key=value>
                          set config inside the new repository
    -4, --ipv4            use IPv4 addresses only
    -6, --ipv6            use IPv6 addresses only


carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/Users/carl-magnus.stenmo/Documents/
Arduino/hardware/espressif
$ git clone --recursive https://github.com/espressif/arduino-esp32.git esp32
Cloning into 'esp32'...
remote: Counting objects: 5851, done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 5851 (delta 4), reused 11 (delta 3), pack-reused 5832
Receiving objects: 100% (5851/5851), 87.81 MiB | 365.00 KiB/s, done.
Resolving deltas: 100% (3136/3136), done.
Checking out files: 100% (1261/1261), done.

carl-magnus.stenmo@se-mlm-lt-cmag MINGW32 /c/Users/carl-magnus.stenmo/Documents/
Arduino/hardware/espressif
$
```

Figure 2: Cloning the Arduino Esp32 repository

2.6.3 Downloading the toolchain

Go to the folder - "`C:\Users\user_name\Documents\Arduino\hardware\espressif\esp32\tools`" to execute the program – "`get.exe`".

Double click on the "`get.exe`" to start the download. This will download the toolchain that is needed to build and flash the project. All the files are extracted on successful download.

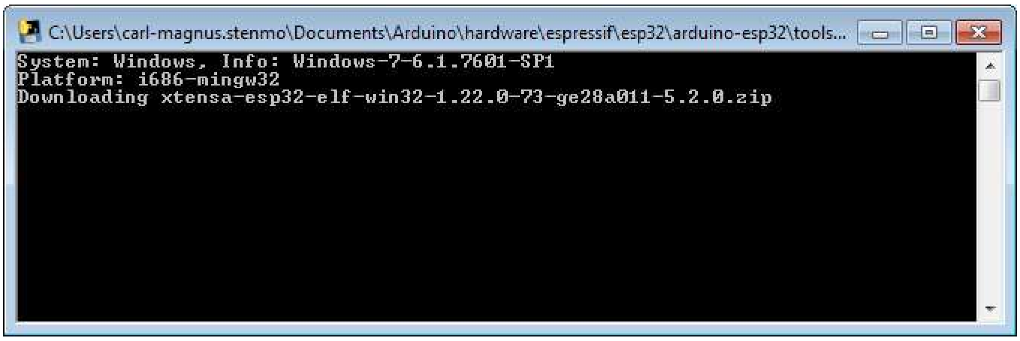


Documents ▸ Arduino ▸ hardware ▸ espressif ▸ esp32 ▸ arduino-esp32 ▸ tools ▸
Share with ▾
E-mail
New folder
Documents library
tools

Name	Date modified	Type	Size
partitions	2017-10-09 20:28	File folder	
sdk	2017-10-09 20:28	File folder	
build.py	2017-10-09 20:28	Python File	5 KB
common.sh	2017-10-09 20:28	Shell Script	3 KB
espota.exe	2017-10-09 20:28	Application	3 936 KB
espota.py	2017-10-09 20:28	Python File	9 KB
esptool.py	2017-10-09 20:28	Python File	110 KB
gen_esp32part.exe	2017-10-09 20:28	Application	3 260 KB
gen_esp32part.py	2017-10-09 20:28	Python File	14 KB
get.exe	2017-10-09 20:28	Application	5 090 KB
get.py	2017-10-09 20:28	Python File	5 KB
platformio-build.py	2017-10-09 20:28	Python File	7 KB

Figure 3: Screenshot after selecting "get.exe"

☞ Normally, it takes around 15-30 minutes to download this program.



C:\Users\carl-magnus.stenmo\Documents\Arduino\hardware\espressif\esp32\arduino-esp32\tools...

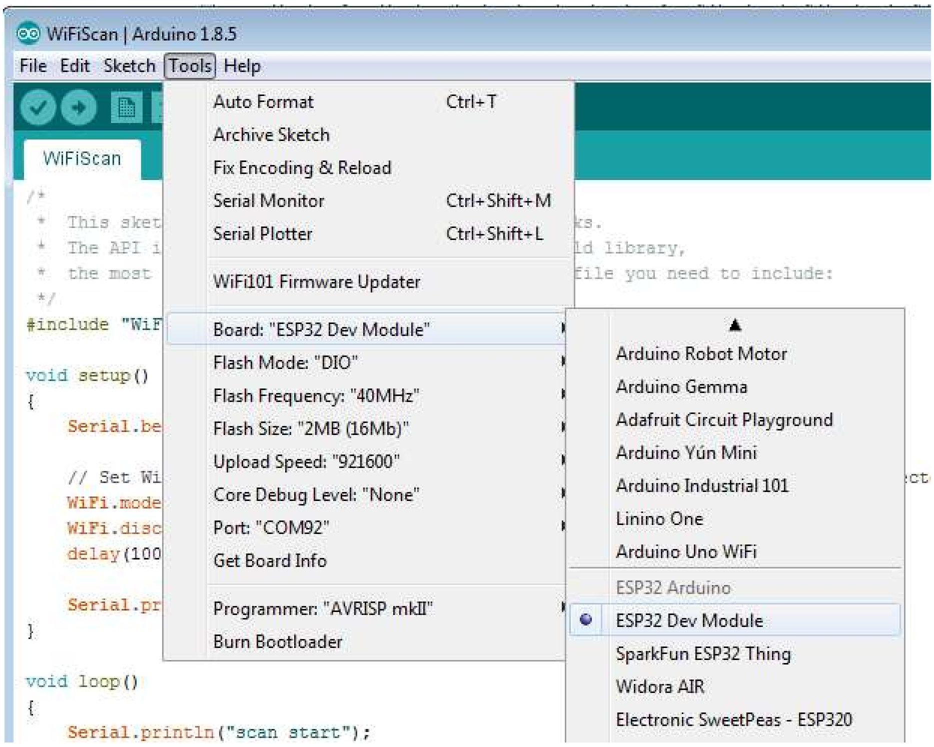
```
System: Windows, Info: Windows-7-6.1.7601-SP1
Platform: i686-mingw32
Downloading xtensa-esp32-elf-win32-1.22.0-73-ge28a011-5.2.0.zip
```

Figure 17: Sample screenshot during download

Open the Arduino application again from the following location - `"C:\Program Files (x86)\Arduino\arduino.exe"`

In the **Tools -> Board** menu, select "ESP32 Dev Module" and then select the following;

- Flash Mode: "DIO"
- Flash Frequency: "40 MHz"
- Flash Size: "2 MB (16 Mb)"
- Upload Speed "921600"
- Core Debug Level "Debug" (optional)



Board: "ESP32 Dev Module"
Flash Mode: "DIO"
Flash Frequency: "40MHz"
Flash Size: "2MB (16Mb)"
Upload Speed: "921600"
Core Debug Level: "None"
Port: "COM92"

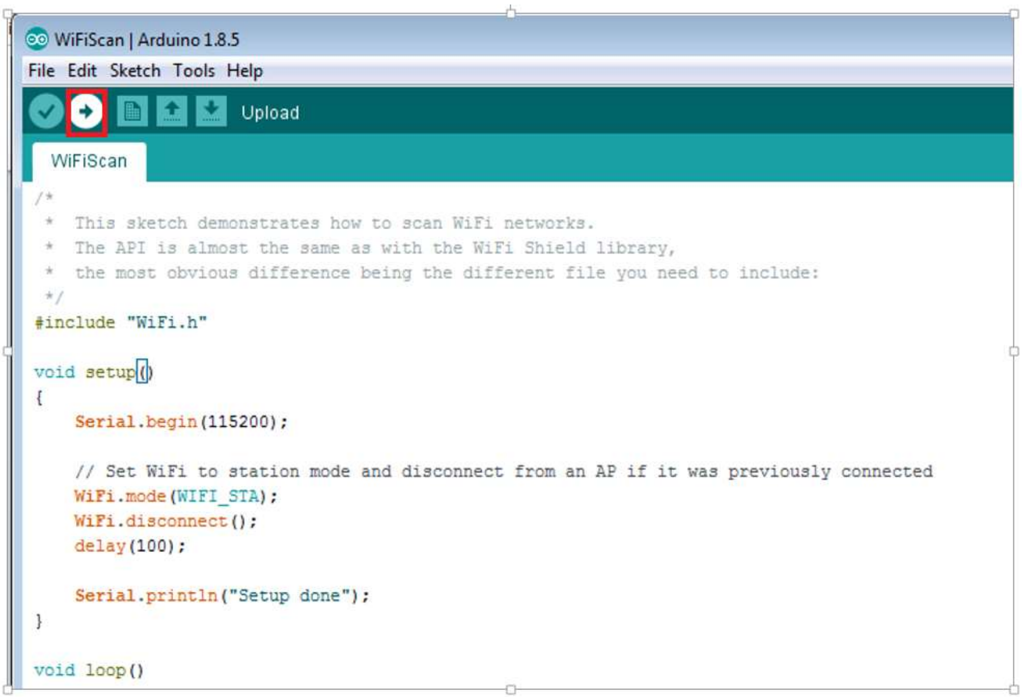
Figure 184: Screenshot that depicts selection of the ESP32 Dev Module

☞ u-blox NINA-W10 module will soon be added to the list of supported boards. Until then, use the ESP32 Dev Module.

Start the WiFiScan example, which is available at the following folder:

```
C:\Users\ user_name
\Documents\Arduino\hardware\espressif\esp32\libraries\WiFi\examples\WiFiScan
```

Press the "**->**" (arrow) button, as shown highlighted in red in the below screenshot (Figure), to start the upload to NINA-W10.



```
/*
 *  This sketch demonstrates how to scan WiFi networks.
 *  The API is almost the same as with the WiFi Shield library,
 *  the most obvious difference being the different file you need to include:
 */
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop()
```

Figure 19: Screenshot that depicts the arrow at the top

Select "Serial Monitor" from the Tools menu as shown in Figure to view the events.

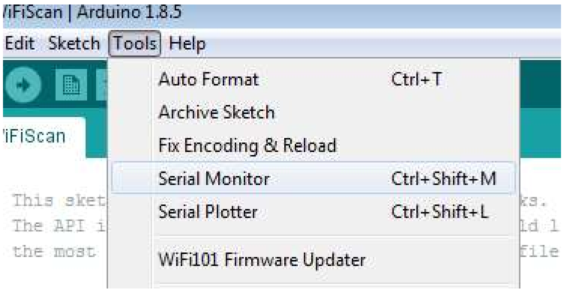
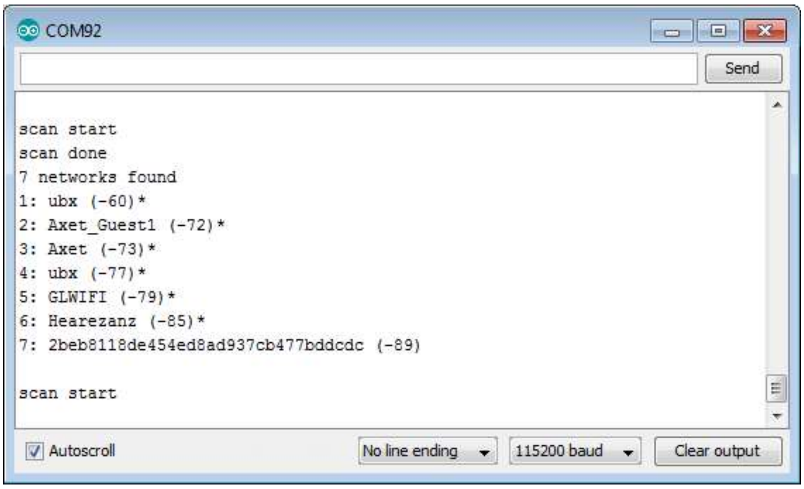


Figure 20: Screenshot that depicts selection of the "Serial Monitor"

The Wi-Fi scan starts and displays the results, as shown in Figure .



```
scan start
scan done
7 networks found
1: ubx (-60)*
2: Axet_Guest1 (-72)*
3: Axet (-73)*
4: ubx (-77)*
5: GLWIFI (-79)*
6: Hearezanz (-85)*
7: 2beb8118de454ed8ad937cb477bddcdc (-89)

scan start
```

Figure 21: Sample screenshot of the Wi-Fi scan

2.6.4 Output power configuration

2.6.5 NINA-W10 series

To operate within the regulatory output power limits, the integrator must configure the module as per the instructions in the following subsections.

The following power configurations for Wi-Fi, Bluetooth BR/EDR and Bluetooth low energy are only valid for the official esp-idf git repositories.

2.6.5.1 Wi-Fi output power configuration for version v2.1

The original file (`phy_init.c`) is located in the folder `...\esp-idf\components\esp32\` in the official v2.1 esp-idf git repository. Update this file with the values provided below:

```
const esp_phy_init_data_t* esp_phy_get_init_data()
{
    int8_t *init_data = malloc(sizeof(esp_phy_init_data_t));
    memcpy(init_data, &phy_init_data, sizeof(esp_phy_init_data_t));
    init_data[44] = 56;//target power 0
    init_data[45] = 58;//target power 1
    init_data[46] = 54;//target power 2
    init_data[47] = 47;//target power 3
    init_data[48] = 44;//target power 4
    init_data[49] = 37;//target power 5

    init_data[50] = 0; //msc0
    init_data[51] = 0; //msc1
    init_data[52] = 0; //msc2
    init_data[53] = 0; //msc3
    init_data[54] = 0; //msc4
    init_data[55] = 2; //msc5
    init_data[56] = 4; //msc6
    init_data[57] = 5; //msc7

    init_data[58] = 1; //11B special rate enable
    init_data[59] = 3; //11B 1m, 2m
    init_data[60] = 3; //11B 5.5, 11m

    init_data[61] = 1; //channel backoff enable

    init_data[62] = 18; //backoff channel 1
    init_data[63] = 4;//backoff channel 2
    init_data[64] = 2;//backoff channel 3
    init_data[65] = 2;//backoff channel 4
    init_data[66] = 2;//backoff channel 5
    init_data[67] = 0;//backoff channel 6
    init_data[68] = 0;//backoff channel 7
    init_data[69] = 0;//backoff channel 8
    init_data[70] = 0;//backoff channel 9
    init_data[71] = 0;//backoff channel 10
    init_data[72] = 14;//backoff channel 11
    init_data[73] = 26;//backoff channel 12
    init_data[74] = 255;//backoff channel 13
    init_data[75] = 255;//backoff channel 14

    init_data[76] = 15; //backoff rate on channel 1
    init_data[77] = 15; //backoff rate on channel 2
    init_data[78] = 8; //backoff rate on channel 3
    init_data[79] = 8; //backoff rate on channel 4
    init_data[80] = 8; //backoff rate on channel 5
    init_data[81] = 0; //backoff rate on channel 6
    init_data[82] = 0; //backoff rate on channel 7
    init_data[83] = 0; //backoff rate on channel 8
    init_data[84] = 0; //backoff rate on channel 9
    init_data[85] = 0; //backoff rate on channel 10
    init_data[86] = 7; //backoff rate on channel 11
    init_data[87] = 63; //backoff rate on channel 12
    init_data[88] = 63; //backoff rate on channel 13
    init_data[89] = 63; //backoff rate on channel 14
    apply_rf_frequency_calibration(init_data);

    ESP_LOGD(TAG, "loading PHY init data from application binary");
    return (esp_phy_init_data_t*)init_data;
}
```

2.6.5.2 Wi-Fi output power configuration for versions v3.1, v3.2 and v4.0

The original file (`phy_init.c`) is located in the folder `...\esp-idf\components\esp32\` in the official git repositories for the applicable esp-idf. Update the file with the values provided below:

```
const esp_phy_init_data_t* esp_phy_get_init_data()
{
    int8_t *init_data = malloc(sizeof(esp_phy_init_data_t));
    memcpy(init_data, &phy_init_data, sizeof(esp_phy_init_data_t));
    init_data[44] = 56;//target power 0
    init_data[45] = 54;//target power 1
    init_data[46] = 48;//target power 2
    init_data[47] = 46;//target power 3
    init_data[48] = 42;//target power 4
    init_data[49] = 36;//target power 5
    init_data[50] = 0; //msc0
    init_data[51] = 0; //msc1
    init_data[52] = 0; //msc2
    init_data[53] = 0; //msc3
    init_data[54] = 0; //msc4
    init_data[55] = 1; //msc5
    init_data[56] = 3; //msc6
    init_data[57] = 4; //msc7
    init_data[58] = 1; //11B special rate enable
    init_data[59] = 2; //11B 1m, 2m
    init_data[60] = 2; //11B 5.5, 11m
    init_data[61] = 2; //fcc enable 2: enable 62-80 bytes to set maximum power
    init_data[62] = 0x53; //channel 1
    init_data[63] = 0x52;//channel 2
    init_data[64] = 0x30;//channel 3
    init_data[65] = 0x20;//channel 4
    init_data[66] = 0x20;//channel 5
    init_data[67] = 0x20;//channel 6
    init_data[68] = 0x20;//channel 7
    init_data[69] = 0x20;//channel 8
    init_data[70] = 0x20;//channel 9
    init_data[71] = 0x20;//channel 10
    init_data[72] = 0x22;//channel 11
    init_data[73] = 0x10;//channel 12
    init_data[74] = 0x10;//channel 13
    init_data[75] = 0xAA;//channel 14
    init_data[76] = 0x44; //channel 3, 4
    init_data[77] = 0x44; //channel 5, 6
    init_data[78] = 0x44; //channel 7, 8
    init_data[79] = 0x44; //channel 9, 10
    init_data[80] = 0x44; //channel 11
    apply_rf_frequency_calibration(init_data);
    ESP_LOGD(TAG, "loading PHY init data from application binary");
    return (esp_phy_init_data_t*)init_data;
}
```

2.6.5.3 Bluetooth BR/EDR output power configuration

No output power configuration for Bluetooth BR/EDR is required. With default settings, the module will operate at ~6 dBm, which is within the regulatory limit for NINA-W1.

2.6.5.4 Bluetooth low energy output power configuration

No output power configuration for Bluetooth low energy is required. With default settings, the module will operate at ~6 dBm, which is within the regulatory limit for NINA-W1.

2.6.6 NINA-W13/W15 series

No output power configuration required by the integrator. Using the u-connectXpress guarantees operation within regulatory limits.

3 Design-in

3.1 Overview

For an optimal integration of NINA-W1 series modules in the final application board, it is recommended to follow the design guidelines stated in this chapter. Every application circuit must be properly designed to guarantee the correct functionality of the related interface, however a number of points require high attention during the design of the application device.

The following list provides important points sorted by rank of importance in the application design, starting from the highest relevance:

1. Module antenna connection: **Ant** pad

   Antenna circuit affects the RF compliance of the device integrating NINA-W1x1 modules with applicable certification schemes. Follow the recommendations provided in section 3.3 for schematic and layout design.

2. Module supply: **VCC**, **VCC_IO**, and **GND** pins

   The supply circuit affects the performance of the device integrating NINA-W1 series module. Follow the recommendations provided in section 3.2 for schematic and layout design.

3. High speed interfaces: **UART** pins

   High speed interfaces can be a source of radiated noise and can affect compliance with regulatory standards for radiated emissions. Follow the recommendations provided in section 3.4.1 for schematic and layout design.

4. System functions: RESET_N, GPIO and other System input and output pins

   Accurate design is required to guarantee that the voltage level is well defined during module boot.

5. Other pins:

   Accurate design is required to guarantee proper functionality.

3.2 Supply interfaces

3.2.1 Module supply (VCC) design

Good connection of the module's VCC pin with DC supply source is required for correct RF performance. The guidelines are summarized below:

- The VCC connection must be as wide and short as possible.
- The VCC connection must be routed through a PCB area separated from sensitive analog signals and sensitive functional units. It is a good practice to interpose at least one layer of the PCB ground between VCC track and other signal routing.

There is no strict requirement of adding bypass capacitance to the supply net close to the module. However, depending on the layout of the supply net and other consumers on the same net, bypass capacitors might still be beneficial. Though the GND pins are internally connected, connect all the available pins to solid ground on the application board, as a good (low impedance) connection to an external ground can minimize power loss and improve RF and thermal performance.

3.2.2 Digital I/O interfaces reference voltage (VCC_IO)

Good connection of the module's VCC_IO pin with DC supply source is required for correct performance. The guidelines are summarized below:

- The VCC_IO connection must be as wide and short as possible.
- The VCC_IO connection must be routed through a PCB area separated from sensitive analog signals and sensitive functional units. It is a good practice to interpose at least one layer of PCB ground between VCC_IO track and other signal routing.

There is no strict requirement of adding bypass capacitance to the supply net close to the module. However, depending on the layout of the supply net and other consumers on the same net, bypass capacitors might still be beneficial. Though the GND pins are internally connected, connect all the available pins to solid ground on the application board, as a good (low impedance) connection to an external ground can minimize power loss and improve RF and thermal performance.

3.3 Antenna interface

As the unit cannot be mounted arbitrarily, the placement should be chosen with consideration so that it does not interfere with radio communication. The W1x6 modules with a PCB trace antenna cannot be mounted in a metal enclosure. No metal casing or plastics using metal flakes should be used. Avoid metallic based paint or lacquer as well.

3.3.1 On-board antenna design

If a plastic enclosure is used, it is possible to use NINA-W1 with the embedded antenna. In order to reach an optimum operating performance, follow the instructions provided in the sections below.

3.3.1.1 NINA-W1x6 – PCB trace antenna

- The module shall be placed in the center of an edge of the host PCB.

A large ground plane on the host PCB is a prerequisite for good antenna performance. It is recommended to have the ground plane extending at least 10 mm on the three non-edge sides of the module. See

- Figure.
- The host PCB shall include a full GND plane underneath the entire module, with a ground cut out under the PCB trace antenna according to the description in Figure .
- The NINA-W1x6 has four extra GND pads under the antenna that need to be connected for a good antenna performance. Detailed measurements of the footprint including this extra GND pads can be found in the NINA-W1 series data sheets.
- High / large parts including metal shall not be placed closer than 10 mm to the module’s antenna.
- At least 10 mm clearance between the antenna and the casing is recommended. If the clearance is less than 10 mm, the antenna performance can be adversely affected.
- The module shall be placed such that the antenna faces outwards from the product and is not obstructed by any external items in close vicinity of the products intended use case.

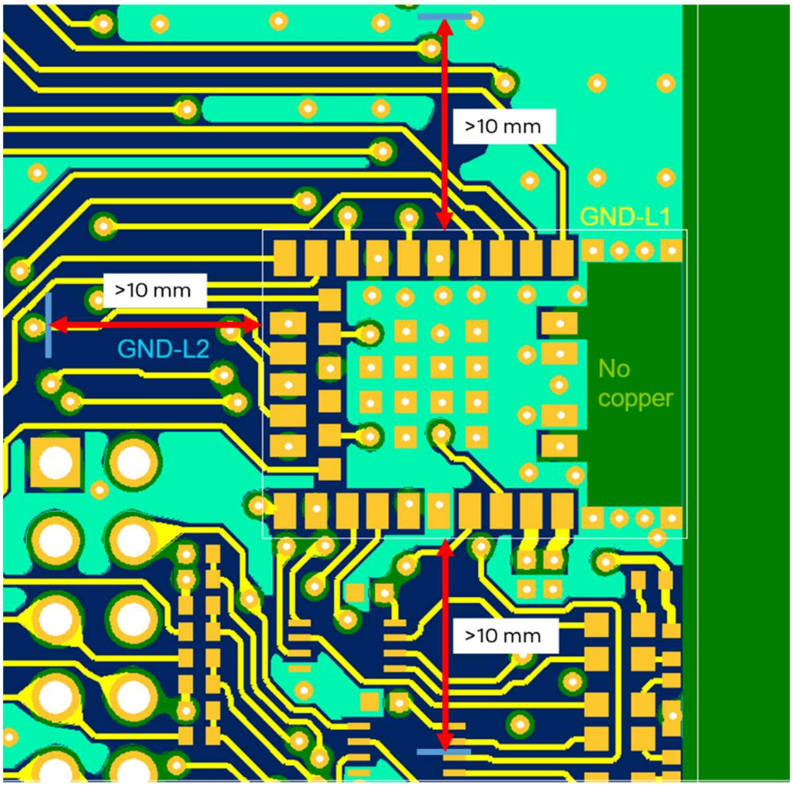


Figure 31: GND plane guard area enclosing the NINA-W1x6 module

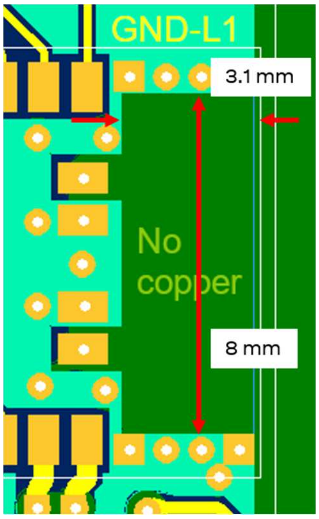


Figure 32: Size of the GND cut out for the NINA-W1x6 module's PCB trace antenna

3.4 Data communication interfaces

3.4.1 Asynchronous serial interface (UART) design

The layout of the UART bus should be done so that noise injection and cross talk are avoided. It is recommended to use the hardware flow control with RTS/CTS to prevent temporary UART buffer overrun.

The flow control signals RTS/CTS are active low thus a 0 (ON state =low level) will allow the UART to transmit.

- CTS is an input to the NINA-W1 module and if the host applies a 0 (ON state = low level), then the module is allowed to transmit.
- RTS is an output off the NINA-W1 module and the module will apply a 0 (ON state = low level) when it is ready to receive transmission.

3.4.2 Ethernet (RMII+SMI)

It is recommended to route all signals in the RMII bus with the same length and have appropriate grounding in the surrounding layers; total bus length should also be minimized. The layout of the RMII bus should be done so that crosstalk with other parts of the circuit is minimized providing adequate isolation between the signals, the clock and the surrounding busses/traces.

Termination resistors are recommended for the RX and TX lines on the RMII bus.

Pull-up resistor is required for MDIO.

The General High Speed layout guidelines in section 3.5 apply for the RMII and the SMI bus.

3.5 General High Speed layout guidelines

These general design guidelines are considered as best practices and are valid for any bus present in the NINA-W1 series modules; the designer should prioritize the layout of higher speed busses. Low frequency signals are generally not critical for layout.

☞ One exception is represented by High Impedance traces (such as signals driven by weak pull resistors) that may be affected by crosstalk. For those traces, a supplementary isolation of 4w from other busses is recommended.

3.5.1 General considerations for schematic design and PCB floor-planning

- Verify which signal bus requires termination and add series resistor terminations to the schematics.
- Carefully consider the placement of the module with respect to antenna position and host processor.
- Verify with PCB manufacturer allowable stack-ups and controlled impedance dimensioning.
- Verify that the power supply design and power sequence are compliant with the specification of NINA-W1 series module.

3.5.2 Module placement

- Accessory parts like bypass capacitors should be placed as close as possible to the module to improve filtering capability, prioritizing the placement of the smallest size capacitor close to module pads.

⚠ Particular care should be taken not to place components close to the antenna area. The designer should carefully follow the recommendations from the antenna manufacturer about the distance of the antenna vs. other parts of the system. The designer should also maximize the distance of the antenna to Hi-frequency busses like DDRs and related components or consider an optional metal shield to reduce interferences that could be picked up by the antenna thus reducing the module's sensitivity.

- An optimized module placement allows better RF performance. See section 3.3 for more information on antenna consideration during module placement.

3.5.3 Layout and manufacturing

- Avoid stubs on high speed signals. Even through-hole vias may have an impact on signal quality.

- Verify the recommended maximum signal skew for differential pairs and length matching of buses.
- Minimize the routing length; longer traces will degrade signal performance. Ensure that maximum allowable length for high speed busses is not exceeded.
- Ensure that you track your impedance matched traces. Consult with your PCB manufacturer early in the project for proper stack-up definition.
- RF and digital sections should be clearly separated on the board.
- Ground splitting is not allowed below the module.
- Minimize bus length to reduce potential EMI issues from digital busses.
- All traces (including low speed or DC traces) must couple with a reference plane (GND or power); Hi-speed busses should be referenced to the ground plane. In this case, if the designer needs to change the ground reference, an adequate number of GND vias must be added in the area of transition to provide a low impedance path between the two GND layers for the return current.
- Hi-Speed busses are not allowed to change reference plane. If a reference plane change is unavoidable, some capacitors should be added in the area to provide a low impedance return path through the different reference planes.
- Trace routing should keep a distance greater than 3W from the ground plane routing edge.
- Power planes should keep a distance from the PCB edge sufficient to route a ground ring around the PCB, the ground ring must then be connected to other layers through vias.

3.6 Module footprint and paste mask

The mechanical outline of the NINA-W1 series modules can be found in the NINA-W1 series Data Sheets.The proposed land pattern layout reflects the pads layout of the module.

The Non Solder Mask Defined (NSMD) pad type is recommended over the Solder Mask Defined (SMD) pad type, which implements the solder mask opening 50 μm larger per side than the corresponding copper pad.

The suggested paste mask layout for the NINA-W1 series modules is to follow the copper mask layout as described in the NINA-W1 series Data Sheets.

⚠ These are recommendations only and not specifications. The exact mask geometries, distances and stencil thicknesses must be adapted to the specific production processes of the customer.

3.7 Thermal guidelines

The NINA-W1 series modules have been successfully tested in -40 °C to +85 °C. A good grounding should be observed for temperature relief during high ambient temperature.

3.8 ESD guidelines

The immunity of devices integrating NINA-W1 modules to Electro-Static Discharge (ESD) is part of the Electro-Magnetic Compatibility (EMC) conformity, which is required for products bearing the CE marking, compliant with the R&TTE Directive (99/5/EC), the EMC Directive (89/336/EEC) and the Low Voltage Directive (73/23/EEC) issued by the Commission of the European Community.

Compliance with these directives implies conformity to the following European Norms for device ESD immunity: ESD testing standard *CENELEC EN 61000-4-2* and the radio equipment standards *ETSI EN 301 489-1*, *ETSI EN 301 489-7*, *ETSI EN 301 489-24*, the requirements of which are summarized in Table 2.

The ESD immunity test is performed at the enclosure port, defined by *ETSI EN 301 489-1* as the physical boundary through which the electromagnetic field radiates. If the device implements an integral antenna, the enclosure port is seen as all insulating and conductive surfaces housing the device. If the device implements a removable antenna, the antenna port can be separated from the enclosure port. The antenna port includes the antenna element and its interconnecting cable surfaces.

The applicability of ESD immunity test to the whole device depends on the device classification as defined by *ETSI EN 301 489-1*. Applicability of ESD immunity test to the related device ports or the related interconnecting cables to auxiliary equipment, depends on the device accessible interfaces and manufacturer requirements, as defined by the *ETSI EN 301 489-1*.

Contact discharges are performed at conductive surfaces, while air discharges are performed at insulating surfaces. Indirect contact discharges are performed on the measurement setup horizontal and vertical coupling planes as defined in the *CENELEC EN 61000-4-2*.

For the definition of integral antenna, removable antenna, antenna port, and device classification, refer to the ETSI EN 301 489-1. For the contact and air discharges definitions, refer to CENELEC EN 61000-4-2.

Parameter	Min.	Typical	Max.	Unit	Remarks
ESD immunity. All exposed surfaces of the radio equipment and ancillary equipment in a representative configuration			8*	kV	Indirect discharge according to IEC 61000-4-2
ESD sensitivity, tested for all pins except ANT and RSVD pins #11, #15, #33			2.5	kV	Human body model according to JEDEC JS001

* Tested on EVK-NINA-W1 evaluation board.

Table 2: Electro-Magnetic Compatibility ESD immunity requirements as defined by CENELEC EN 61000-4-2, ETSI EN 301 489-1, ETSI EN 301 489-7, ETSI EN 301 489-24

NINA-W1 is manufactured taking into account specific standards to minimize the occurrence of ESD events; the highly automated process complies with the IEC61340-5-1 (STM5.2-1999 Class M1 devices) standard. Thus, the designer should implement proper measures to protect from ESD events on any pin that may be exposed to the end user.

Compliance with standard protection level specified in the EN61000-4-2 can be achieved by including the ESD protections in parallel to the line, close to areas accessible by the end user.

4 Handling and soldering

No natural rubbers, hygroscopic materials or materials containing asbestos are employed.

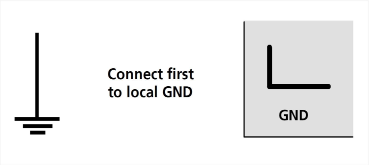
4.1 Packaging, shipping, storage and moisture preconditioning

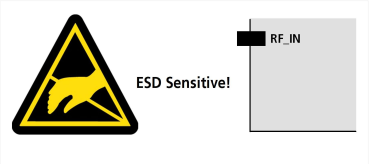
For information pertaining to reels, tapes or trays, moisture sensitivity levels (MSL), shipment and storage, as well as drying for preconditioning refer to NINA-W1 series Data Sheets, and u-blox Package Information Guide.

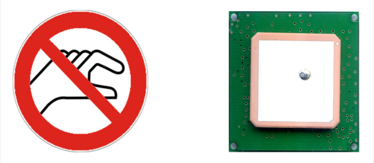
4.2 Handling

The NINA-W1 series modules are Electro-Static Discharge (ESD) sensitive devices and require special precautions during handling. Particular care must be exercised when handling patch antennas, due to the risk of electrostatic charges. In addition to standard ESD safety practices, the following measures should be taken into account whenever handling the receiver:

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, then the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device
- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10 pF, coax cable ~50-80 pF/m, soldering iron, …)
- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in non ESD protected work area, implement proper ESD protection measures in the design.
- When soldering RF connectors and patch antennas to the receiver’s RF pin, make sure to use an ESD safe soldering iron (tip).







4.3 Soldering

4.3.1 Reflow soldering process

The NINA-W1 series modules are surface mount modules supplied on a FR4-type PCB with gold plated connection pads and are produced in a lead-free process with a lead-free soldering paste. The bow and twist of the PCB is maximum 0.75% according to IPC-A-610E. The thickness of solder resist between the host PCB top side and the bottom side of the NINA-W1 series module must be considered for the soldering process.

The module is compatible with industrial reflow profile for RoHS solders. Use of "No Clean" soldering paste is strongly recommended.

The reflow profile used is dependent on the thermal mass of the entire populated PCB, heat transfer efficiency of the oven, and particular type of solder paste used. The optimal soldering profile used has to be trimmed for each case depending on the specific process and PCB layout.

Process parameter		Unit	Value
Pre-heat	Ramp up rate to T_{SMIN}	K/s	3
	T_{SMIN}	°C	150
	T_{SMAX}	°C	200
	t_S (from +25 °C)	s	150
	t_S (Pre-heat)	s	60 to 120
Peak	T_L	°C	217
	t_L (time above T_L)	s	40 to 60
	T_P (absolute max)	°C	245
Cooling	Ramp-down from T_L	K/s	4
	Allowed soldering cycles	-	1

Table 3: Recommended reflow profile

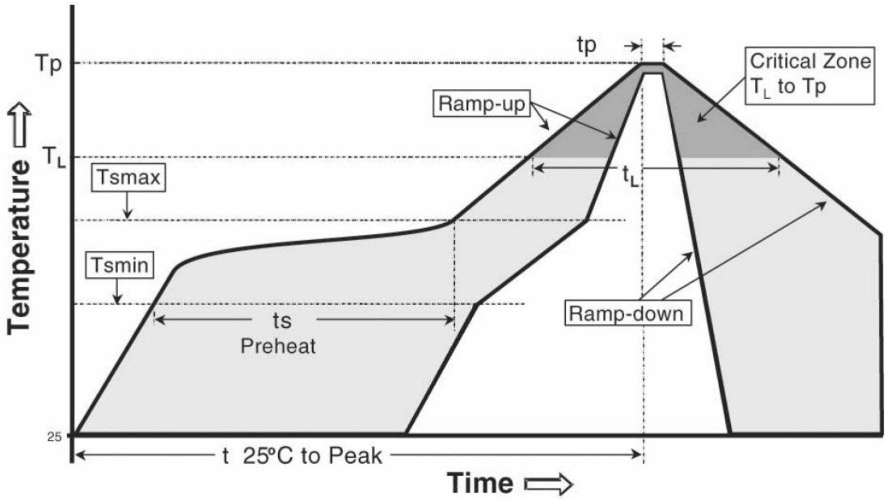


Figure 5: Reflow profile

Lower value of T_P and slower ramp down rate (2 – 3 °C/sec) is preferred.

After reflow soldering, optical inspection of the modules is recommended to verify proper alignment.

⚠ Target values in Table 3 should be taken as general guidelines for a Pb-free process. Refer to JEDEC J-STD-020C standard for further information.

4.3.2 Cleaning

Cleaning the modules is not recommended. Residues underneath the modules cannot be easily removed with a washing process.

- Cleaning with water will lead to capillary effects where water is absorbed in the gap between the baseboard and the module. The combination of residues of soldering flux and encapsulated water leads to short circuits or resistor-like interconnections between neighboring pads. Water will also damage the sticker and the ink-jet printed text.
- Cleaning with alcohol or other organic solvents can result in soldering flux residues flooding into the two housings, areas that are not accessible for post-wash inspections. The solvent will also damage the sticker and the ink-jet printed text.
- Ultrasonic cleaning will permanently damage the module, in particular the crystal oscillators.

For best results, use a "no clean" soldering paste and eliminate the cleaning step after the soldering process.

4.3.3 Other remarks

- Only a single reflow soldering process is allowed for boards with a module populated on it.
- Boards with combined through-hole technology (THT) components and surface-mount technology (SMT) devices may require wave soldering to solder the THT components. Only a single wave soldering process is allowed for boards populated with the modules. *Miniature Wave Selective Solder* process is preferred over traditional wave soldering process.
- Hand soldering is not recommended.
- Rework is not recommended.
- Conformal coating may affect the performance of the module, it is important to prevent the liquid from flowing into the module. The RF shields do not provide protection for the module from coating liquids with low viscosity, therefore care is required in applying the coating. Conformal coating of the module will void the warranty.
- Grounding metal covers: attempts to improve grounding by soldering ground cables, wick or other forms of metal strips directly onto the EMI covers is done at the customer's own risk and will void the warranty of the module. The numerous ground pins are adequate to provide optimal immunity to interferences.
- The module contains components that are sensitive to Ultrasonic Waves. Use of any ultrasonic processes such as cleaning, welding etc., may damage the module. Use of ultrasonic processes on an end product integrating this module will also void the warranty.

5 Approvals

⚠ Country approval for NINA-W106, NINA-W136, NINA-W156 and NINA-B226 is pending.

5.1 European Union regulatory compliance

⚠ Approval for NINA-W106, NINA-W136, NINA-W156 and NINA-B226 is pending.

Information about regulatory compliance of the European Union for NINA-W10 series modules is available in the NINA-W10 Declaration of Conformity..

5.1.1 Radio Equipment Directive (RED) 2014/53/EU

The NINA-W106, NINA-W136, NINA-W156 and NINA-B226 modules comply with the essential requirements and other relevant provisions of the Radio Equipment Directive (RED) 2014/53/EU.

5.1.2 Compliance with the RoHS directive

The NINA-W1 series modules complies with the Directive 2011/65/EU (EU RoHS 2) and its amendment Directive (EU) 2015/863 (EU RoHS 3).

5.2 United States (FCC)

The NINA-W106, NINA-W136, NINA-W156 and NINA-B226 modules have received Federal Communications Commission (FCC) CFR47 Telecommunications, Part 15 Subpart C “Intentional Radiators” modular approval in accordance with Part 15.247 Modular Transmitter approval. The modular approval allows the end user to integrate the module into a finished product without obtaining subsequent and separate FCC approvals for intentional radiation, provided no changes or modifications are made to the module circuitry. Changes or modifications could void the user’s authority to operate the equipment. The end user must comply with all of the instructions provided by the Grantee, which indicate installation and/or operating conditions necessary for compliance.

The finished product is required to comply with all applicable FCC equipment authorizations regulations, requirements and equipment functions not associated with the transmitter module portion. For example, compliance must be demonstrated to regulations for other transmitter components within the host product; to requirements for unintentional radiators (Part 15 Subpart B “Unintentional Radiators”), such as digital devices, computer peripherals, radio receivers, etc.; and to additional authorization requirements for the non-transmitter functions on the transmitter module (i.e., Verification, or Declaration of Conformity) (e.g., transmitter modules may also contain digital logic functions) as appropriate.

(OEM) Integrator has to assure compliance of the entire end-product incl. the integrated RF Module. For 15 B (§15.107 and if applicable §15.109) compliance, the host manufacturer is required to show compliance with 15 while the module is installed and operating.

Furthermore the module should be transmitting and the evaluation should confirm that the module's intentional emissions (15C) are compliant (fundamental / out-of-band). Finally the integrator has to apply the appropriate equipment authorization (e.g. Verification) for the new host device per definition in §15.101.

Integrator is reminded to assure that these installation instructions will not be made available to the end-user of the final host device.

5.2.1 Labeling and user information requirements

The NINA-W106, NINA-W136, NINA-W156 and NINA-B226 modules are assigned

FCC ID: XPYNINAW106

The final host device, into which this RF Module is integrated" has to be labeled with an auxiliary label stating the FCC ID of the RF Module, such as:

> “Contains FCC ID: **XPYNINAW106**
>
> This device is acting as slave and operating in the 2.4 GHz (2412 ~2462 MHz) band.
>
> "This device complies with part 15 of the FCC rules. Operation is subject to the following two conditions:
>
> 1. This device may not cause harmful interference, and
> 2. This device must accept any interference received, including interference that may cause undesired operation.
>
> "Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user’s authority to operate the equipment."

☞ This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Model	**FCC ID**
NINA-W106	FCC ID: XPYNINAW106
NINA-W136	FCC ID: XPYNINAW106
NINA-W156	FCC ID: XPYNINAW106
NINA-B226	FCC ID: XPYNINAW106

Table 04: FCC ID for NINA-W106, NINA-W136, NINA-W156 and NINA-B226

5.2.2 RF exposure

⚠ The Integrator will be responsible to satisfy SAR/ RF Exposure requirements, when the module integrated into the host device.

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment.
Having a separation distance of minimum 45 mm between the user and/or bystander and the antenna and /or radiating element ensures that the maximum output power of NINA-W106, NINA-W136, NINA-W156 and NINA-B226 is below the SAR test exclusion limits presented in KDB 447498 D01v06.

All transmitters regulated by FCC must comply with RF exposure requirements. KDB 447498 General RF Exposure Guidance provides guidance in determining whether proposed or existing transmitting facilities, operations or devices comply with limits for human exposure to Radio Frequency (RF) fields adopted by the Federal Communications Commission (FCC).

This module is approved for installation into mobile and/or portable host platforms and must not be co-located or operating in conjunction with any other antenna or transmitter except in accordance with FCC multi-transmitter guidelines. End users must be provided with transmitter operating conditions for satisfying RF Exposure compliance.

KDB996369 D03 section 2.4 (limited module procedures) and 2.5 (trace antenna designs) are not applicable to the NINA-W106, NINA-W136, NINA-W156 and NINA-B226 modules.

5.2.3 Module statement

The single-modular transmitter is a self-contained, physically delineated, component for which compliance can be demonstrated independent of the host operating conditions, and which complies with all eight requirements of § 15.212(a)(1) as summarized below.

1) The radio elements have the radio frequency circuitry shielded.

2) The module has buffered modulation/data inputs to ensure that the device will comply with Part 15 requirements with any type of input signal.

3) The module contains power supply regulation on the module.

4) The module contains a permanently attached antenna.

5) The module demonstrates compliance in a stand-alone configuration.

6) The module is labeled with its permanently affixed FCC ID label.

7) The module complies with all specific rules applicable to the transmitter, including all the conditions provided in the integration instructions by the grantee.

8) The module complies with RF exposure requirements.

5.2.4 End-product compliance

⚠ The modular transmitter approval of NINA-W106, NINA-W136, NINA-W156 and NINA-B226, or any other radio module, does not exempt the end product from being evaluated against applicable regulatory demands.

The evaluation of the end product shall be performed with the NINA-W106, NINA-W136, NINA-W156 or NINA-B226 INA-B3 module installed and operating in a way that reflects the intended end product use case. The upper frequency measurement range of the end product evaluation is the 5th harmonic of 2.4 GHz as declared in 47 CFR Part 15.33 (b)(1).

The following requirements apply to all products that integrate a radio module:

- Subpart B - UNINTENTIONAL RADIATORS
  To verify that the composite device of host and module comply with the requirements of FCC part 15B, the integrator shall perform sufficient measurements using ANSI 63.4-2014.
- Subpart C - INTENTIONAL RADIATORS
  It is required that the integrator carries out sufficient verification measurements using ANSI 63.10-2013 to validate that the fundamental and out of band emissions of the transmitter part of the composite device complies with the requirements of FCC part 15C.

When the items listed above are fulfilled, the end product manufacturer can use the authorization procedures as mentioned in Table 1 of 47 CFR Part 15.101, before marketing the end product. This means the customer has to either market the end product under a Suppliers Declaration of Conformity (SDoC) or to certify the product using an accredited test lab.

5.3 Canada (ISED)

The NINA-W106, NINA-W136, NINA-W156 and NINA-B226 modules are certified for use in Canada under Innovation, Science and Economic Development Canada (ISED) Radio Standards Specification (RSS) RSS-247 Issue 2 and RSSGen.

5.3.1 Labeling and user information requirements

The final host device, into which this RF Module is integrated has to be labeled with an auxiliary label stating the IC of the RF Module, such as:

“Contains transmitter module IC: 8595A-NINAW106

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada’s licence-exempt RSS(s). Operation is subject to the following two conditions:

1. This device may not cause interference.
2. This device must accept any interference, including interference that may cause undesired operation of the device.

Le périphérique hôte final, dans lequel ce module RF est intégré "doit être étiqueté avec une étiquette auxiliaire indiquant le CI du module RF, tel que"

“Contient le module émetteur IC: 8595A-NINAW106

L’émetteur/récepteur exempt de licence contenu dans le présent appareil est conforme aux CNR d’Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L’exploitation est autorisée aux deux conditions suivantes :

1. L’appareil ne doit pas produire de brouillage;

a. L’appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d’en compromettre le fonctionnement.

Model	**IC**
NINA-W106	IC: 8595A-NINAW106
NINA-W136	IC: 8595A-NINAW106
NINA-W156	IC: 8595A-NINAW106
NINA-B226	IC: 8595A-NINAW106

Table 05: IC for NINA-W106, NINA-W136, NINA-W156 and NINA-B226

5.3.2 RF exposure

This equipment complies with the requirements of IC RSS-102 issue 5 radiation exposure limits set forth for an uncontrolled environment.
Having a separation distance of minimum 40 mm between the user and/or bystander and the antenna and /or radiating element ensures that the output power (e.i.r.p.) of NINA-W106, NINA-W136, NINA-W156 and NINA-B226 is below the SAR evaluation Exemption limits defined in RSS-102 issue 5.

All transmitters regulated by ISED must comply with RF exposure requirements listed in RSS-102 - Radio Frequency (RF) Exposure Compliance of Radiocommunication Apparatus (All Frequency Bands). This module is approved for installation into mobile and/or portable host platforms and must not be co-located or operating in conjunction with any other antenna or transmitter except in accordance with Industry Canada's multi-transmitter guidelines. End users must be provided with transmitter operating conditions for satisfying RF Exposure compliance.

5.4 Japan radio equipment compliance

⚠ Approval for NINA-W106, NINA-W136, NINA-W156 and NINA-B226 is pending.

When a product integrating a NINA-W106, NINA-W136, NINA-W156 or NINA-B226 module is placed on the Japanese market the product must be affixed with a label with the “Giteki” marking as shown in Figure 6. The marking must be visible for inspection.

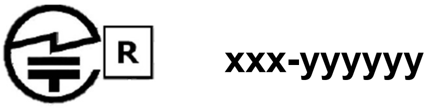


Figure 6: Giteki mark, R and the MIC certification number

6 Product testing

6.1 u-blox In-Series production test

u-blox focuses on high quality for its products. All units produced are fully tested automatically in production line. Stringent quality control process has been implemented in the production line. Defective units are analyzed in detail to improve the production quality.

This is achieved with automatic test equipment (ATE) in production line, which logs all production and measurement data. A detailed test report for each unit can be generated from the system. Figure 7 shows typical automatic test equipment (ATE) in a production line.

The following tests are performed as part of the production tests:

- Digital self-test (firmware download, MAC address programming)
- Measurement of currents
- Functional tests
- Digital I/O tests
- Measurement and verification of RF characteristics in all supported bands. For example, measurements of receiver RSSI and transmitter power levels and frequency tuning of the reference clock.

Figure 7: Automatic test equipment for module test

6.2 OEM manufacturer production test

As the testing is already done by u-blox, an OEM manufacturer does not need to repeat firmware tests or measurement of the module’s RF performance or tests over analog and digital interfaces in their production test.

However, an OEM manufacturer should focus on:

- Module assembly on the device: It should be verified that:
  - Soldering and handling process did not damage the module components
  - All module pins are well soldered on device board
  - There are no short circuits between the pins
- Component assembly on the device: It should be verified that:
  - Communication with the host controller can be established
  - The interfaces between the module and device are working
  - Overall RF performance test of the device including antenna

Dedicated tests can be implemented to check the device. For example, the measurement of module current consumption when set in a specified state can detect a short circuit if compared with a “Golden Device” result.

The standard operational module firmware and test software on the host can be used to perform functional tests (communication with the host controller, check interfaces) and to perform basic RF performance tests.

6.2.1 “Go/No go” tests for integrated devices

A “Go/No go” test compares the signal quality with a “Golden Device” in a location with known signal quality. This test can be performed after establishing a connection with an external device.

A very simple test can be performed by just scanning for a known Bluetooth low energy device and checking the signal level.

These kinds of test may be useful as a “go/no go” test but not for RF performance measurements.

This test is suitable to check the functionality of the communication with the host controller and the power supply. It is also a means to verify if the components are well soldered.

A basic RF functional test of the device including the antenna can be performed with standard Bluetooth low energy devices as remote stations. The device containing the NINA-W1 series module and the antennas should be arranged in a fixed position inside an RF shield box to prevent interferences from other possible radio devices to get stable test results.

Appendix

A Glossary

Abbreviation	Definition
AEC	Automotive Electronics Council
AP	Access Point
ARM	Arm (Advanced RISC Machines) Holdings
ASCII	American Standard Code for Information Interchange
ATE	Automatic Test Equipment
BBR	Battery Backed RAM
BER	Bit Error Rate
BP	Band Pass
BPF	Band Pass Filter
BT	Bluetooth
CAN	Controller Area Network
CPU	Central Processing Unit
CPU	Central Processing Unit
CTS	Clear To Send
DC	Direct Current
DC	Direct Current
DCE	Data Circuit-terminating Equipment* / Data Communication Equipment*
DDC	Display Data Channel
DDR	Double Data Rate
DL	Down Link (Reception)
DRX	Discontinuous Reception
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
EIRP	Equivalent Isotropically Radiated Power
EMC	Electro-magnetic Compatibility
EMI	Electromagnetic Interference
ESD	Electro-Static Discharge
GND	Ground
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical radio
LED	Light-Emitting Diode
LPO	Low Power Oscillator
LTE	Long-Term Evolution
MAC	Media Access Control
MCS	Modulation Coding Scheme
MCU	Microcontroller
MSL	Moisture Sensitivity Level
NSMD	Non Solder Mask Defined
OEM	Original Equipment Manufacturer
PCB	Printed Circuit Board

Abbreviation	Definition
RF	Radio Frequency
RMII	Reduced Media-independent Interface
ROM	Read-only Memory
RSSI	Received Signal Strength Indicator
RTC	Real-Time Clock
SDK	Software Development Kit
SMA	SubMiniature version A
SRAM	Static random-access memory
TBD	To be Defined
THT	Through-hole Technology
UART	Universal Asynchronous Receiver-Transmitter
UTC	Coordinated Universal Time

Table 6: Explanation of the abbreviations and terms used