# WINC3400 Software

## Release Notes

**VERSION :** **1.1.5**

**DATE :** **MAY 02, 2017**


## Abstract

This document presents an overview of the WINC3400 firmware release version 1.1.5, and corresponding driver.

# 1 Introduction

This document describes the WINC3400 version 1.1.5 firmware release package. This is a release containing Wi-Fi functionality with basic BLE support including an on-chip provisioning profile and custom BLE profiles using the Atmel BLE API and BluSDK.

The release package contains all the necessary components (binaries and tools) required to make use of the latest features including tools, and firmware binaries.

## 1.1 Highlights of the release

- Addition of a WINC Flash access API, allowing controlled writing of the WINC OTA image and modification of TLS server certificates stored on the WINC by the host processor (see 3.1).
- Hardware controlled CoEx mechanism (see 3.2).
- Improvements to internal memory buffer handling which enhances multiple data-stream scenarios
- Driver optimizations to enhance throughput
- Many firmware fixes

## 1.2 Firmware readiness

Microchip Technology Inc. considers version 1.1.5 firmware to be suitable for production release.

# 2 Known Issues

| TRAC ID | Severity | Description |
|---|---|---|
| **9050** | **High** | **Auto-rate algorithm can get stuck at low rates**<br>After a period of 2-12 hours running high throughput traffic (>10Mbps) either under interference or with high levels of attenuation on the WINC3400 TX path, the PHY rate can go down to 1Mbps and sometimes never recovers back up to higher rates when conditions improve.<br><br>**Workaround**<br>Reconnect at the WiFi level to reset the auto-rate algorithm. |
| **8085** | **Low** | **WINC3400 sometimes fails to hear broadcast ARP for prolonged periods**<br>Sometimes the WINC3400 fails to see ARP responses sent from certain APs at 11Mbps.<br><br>**Workaround**<br>The ARP exchange will be retried several times and the response will get through to the WINC3400. |
| **8273** | **Low** | **BLE API at_ble_adv_set_tx_power() not working**<br>During BLE API testing it was noticed that calling at_ble_adv_set_tx_power(-5) doesn't return and the D21 remains locked as the call doesn’t return.<br><br>**Workaround**<br>None |
| **8212** | **Low** | **BLE Provisioning AP list clean up required on re-scan request**<br>When using BLE provisioning application, the AP scan list can sometimes display duplicate or old scan entries.<br><br>**Workaround**<br>None |
| **8581** | **Low** | **Out of Memory when closing the Windows login pop-up during HTTP provisioning**<br>When testing HTTP provisioning it was found that sometimes when closing the browser window that was created by clicking the “Additional Login Information may be Required” pop-up on Windows 7, the WINC3400 showed “out of memory” errors in the serial trace<br><br>**Workaround**<br>Don’t close the browser window until provisioning has completed. |
| **8970** | **Low** | **TLS session remote closure not handled by WINC**<br>If the TLS peer closes the TLS session (Close Notify) then the WINC does not terminate the session. This means that a subsequent data transfer will be rejected by the host.<br><br>**Workaround**<br>If data transfer fails (socket callback with type SOCK_ERR_CONN_ABORTED), open a new socket to continue. |

# 3 New Features

## 3.1 Host Flash Access API

### 3.1.1 Overview

The host MCU is capable of accessing WINC flash. This can be used for:

- Application-specific modification of areas of WINC flash, such as the device's TLS certificate stores.
- Increased application-involvement in WINC image update process (compared to in-build OTA).

Accessing the WINC flash contents can be done most effectively by a module with understanding of the WINC's flash use. Thus WINC flash access is now available to the MCU application via a new module m2m_flash.

### 3.1.2 Scope

The flash accesses available via this module are:

- Write, validate and switch-to a firmware image in the WINC inactive partition.
- Add, read or remove an entry from the WINC TLS root certificate entry store.

It is intended that additional flash accesses will be made available in future releases.

Full API details are available in WINC3400_IoT_SW_APIs.chm.

### 3.1.3 Usage

During the MCU application's code to initialize the WINC, the API m2m_flash_access_init should be called. The placement of this call is important. Refer to the documentation of this API in WINC3400_IoT_SW_APIs.chm.

After any flash access, the WINC firmware will be in a reset condition. In this state, the MCU application may request further flash accesses. Then, to recover normal WINC functionality, the MCU application must run its normal initialization sequence, including initialization of other WINC modules.

Depending on the relative complexity of different parts of the MCU application, it may be simplest to do a system reset here. In that case, details of the latest attempted flash access will be available via m2m_flash_access_init.

#### 3.1.3.1 Usage note on WINC firmware images

When writing a firmware image to the WINC, an image to write can be obtained from the OTA binary file which accompanies a release. The OTA binary file contains extra fields before the image. (Some or all of these fields could be used for image verification by the MCU application.) The format of a released OTA binary file is:

| Name | Offset | Length | Contents |
|---|---|---|---|
| Checksum | 0 | 32 | SHA256 digest of binary from offset 32 onwards |
| Signature | 32 | 4 | 0x1ABCDEF9 |
| Length | 36 | 4 | Length of binary from offset 32 onwards |

| | | | |
|---|---|---|---|
| Image | 40 | Image length | Image to write to WINC |

#### 3.1.3.2 Usage note on TLS root certificate entries

The format used for entries in the WINC TLS root certificate store is:

| Name | Offset | Length | Contents |
|---|---|---|---|
| Identifier | 0 | 20 | SHA1 digest of certificate issuer name |
| Expiry | 20 | 8 | Certificate expiry date in format tstrSystemTime (refer to WINC3400_IoT_SW_APIs.chm). |
| Key Info | 28 | 8 | Certificate public key info in format tstrRootCertPubKeyInfo (refer to WINC3400_IoT_SW_APIs.chm). |
| Key | 36 | Key length | Public key (format given below) |
| Padding | 36 + Key length | Pad to 4-byte alignment | 0xFF |

For RSA public keys, the format of Key is:

| Name | Offset | Length | Contents |
|---|---|---|---|
| Modulus | 36 | Modulus length | RSA modulus, with leading 0's stripped off |
| Exponent | 36 + Mod len | Exponent length | Public exponent, with leading 0's stripped off |

For ECDSA public keys, the format of Key is:

| Name | Offset | Length | Contents |
|---|---|---|---|
| X Coord | 36 | Coord length | Public key X-coordinate |
| Y Coord | 36 + Coord len | Coord length | Public key Y-coordinate |

When adding an entry, the buffer must be provided in this format.

One way to get a root certificate into the required format would be to use root_certificate_downloader.exe to load a board with the root certificate (and no others), then use FA_READIDX_ROOTCERT, with index of 0.

When reading an entry, the buffer must be large enough to accommodate the entry in this format. In addition, if using mode FA_READ_ROOTCERT, the Identifier field must be populated according to the entry to be read.

### 3.1.4 Limitations

- m2m_flash_access_image: No checking is done on the buffer that the application provides as the new WINC firmware. The application must ensure that the buffer contains a valid firmware image.
- m2m_flash_access_item: This does not work with mode FA_REMOVE_ROOTCERT. For removing a certificate entry, the dedicated API m2m_flash_access_remove must be used instead.

- m2m_flash_access_remove: This must not be used in the (unlikely) case that a device needs to store a root certificate whose 20-byte identifier (SHA1 digest of the certificate issuer name) is all-zero. All-zero is used to indicate a removed entry.

It is intended that these limitations will be removed in future releases.

## 3.2 Hardware Controlled Coexistence Mechanism

### 3.2.1 Overview

The WINC3400 shares its radio and antenna between both BLE and WiFi, therefore there needs to be a method of co-operation between the two protocols to allow clean and efficient sharing of the single radio resource, with no unnecessary RF emissions.

The WiFi/BLE coexistence mechanism in 1.1.5 has been redesigned to ensure optimized performance even during heavy data throughput of both WiFi and BLE concurrently.

### 3.2.2 Basic operation

The scheme is controlled entirely by hardware with very little firmware involvement, aside from register setup.

As BLE is a time critical protocol, it is prioritised over WiFi when both require use of the radio. Each time the radio is needed for a BLE transmission or listen period, a handshake is performed with the WiFi controller which will terminate any WiFi transmission or reception that is underway and cleanly handover the radio to the BLE controller. Once the BLE controller has finished, it will perform another handshake to hand back the radio to the WiFi controller where any transmission that was aborted will be retried.

# 4 Fixes and Enhancements

## 4.1 BLE API timer improvements

The BLE API library has been modified to use an external timer provided by the application.

If using provided BSP application code, this timer variable is defined in /bsp/source/nm_bsp_samd21_app.c:

```
uint32 gu32Jiffies1ms;
```

If provided BSP code is not being used, then this variable must be defined in the application and incremented approximately every 1ms.

## 4.2 General fixes and enhancements

Known issues mentioned in previous releases:

| TRAC ID | Description |
|---|---|
| **8771** | **WINC3400 fails to disconnect from AP**<br>After a high number of connections/disconnections the WINC fails to disconnect and may require a reset to regain functionality.<br><br>**Fixed:**<br>An incorrect setup of hardware by firmware code was found and fixed. |
| **8973** | **Receiving data over secure socket connection occasionally stalls**<br>When transferring large files (around 400KB) to the WINC3400 over secure TCP sockets, the transfer stalls on about 1% of transfers.<br><br>**Fixed:**<br>Fixes have been made in the area of buffer management that would improve this particular scenario. This problem was no longer seen in regression testing for this release. |

Other fixes and enhancements of note:

| TRAC ID | Description |
|---|---|
| **9011** | **TCP RX window shrinks and wraps**<br>Under some socket error conditions, the TCP window could shrink and eventually wrap to become negative. |
| **8850** | **Validate RX buffer management algorithm**<br>Improvements have been made to the buffer management algorithm and targets, allowing better sharing of resources under multi-socket traffic load, and also better handling of a buffer exhaustion condition. |
| **8894** | **Make timer function in BLE API platform agnostic**<br>Allow the application to provide the timing functions used within the BLE API library code via a gu32Jiffies1ms variable which is extern'd from the BLE API (see 4.1). |
| **9008** | **Driver throughput optimizations**<br>Some modifications have been made in driver code to ensure the datapath runs as efficiently as possible. This increases the socket throughput capability. |

| TRAC ID | Description |
|---|---|
| 8727 | **HTTP provisioning server enhancements**<br>Enhanced to handle HTTP requests over multiple packets for better interoperability with browsers such as Chrome. Improved the default.html template and compressed the css to save flash space. |

# 5 Terms and Definitions

| Term | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| AP | Access Point |
| ARP | Address Resolution Protocol |
| BLE | Bluetooth Low Energy |
| BSS | Basic Service Set |
| CBC | Cyclic Block Chaining |
| DHE | Diffie-Hellman Ephemeral |
| EAPOL | Extensible Authentication Protocol over LAN |
| ECC | Elliptic Curve Cryptography |
| ECDHE | Elliptic Curve Diffie-Hellman Ephemeral |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EEPROM | Electrically Erasable Programmable Read Only Memory |
| ESD | Electrostatic Discharge |
| ESS | Extended Service Set (infrastructure network) |
| HTTP | Hypertext Transfer Protocol |
| IBSS | Independent BSS (ad-hoc network) |
| IEEE | Institute of Electronic and Electrical Engineers |
| MIB | Management Information Base |
| MQTT | Message Queuing Telemetry Transport |
| NDIS | Network Driver Interface Specification |
| OS | Operating System |
| OTA | Over The Air update |
| PCI | Peripheral Component Interconnect |
| PMK | Pair-wise Master Key |
| PSK | Pre-shared Key |
| RSA | Rivest-Shamir-Adleman (public key cryptosystem) |
| RSN | Robust Security Network |
| SHA | Secure Hash Algorithm |
| SPI | Serial Peripheral Interface |
| SSID | Service Set Identifier |
| RSSI | Receive Strength Signal Indicator |
| TLS | Transport Layer Security |
| WEP | Wired Equivalent Privacy |
| WINC | Wireless Network Controller |
| WLAN | Wireless Local Area Network |
| WMM™ | Wi-Fi Multimedia |
| WMM-PS™ | Wi-Fi Multimedia Power Save |
| WoWLAN | Wake On WLAN |
| WPA™ | Wi-Fi Protected Access |
| WPA2™ | Wi-Fi Protected Access 2 (same as IEEE 802.11i) |