WINC3400 Software

Release Notes

VERSION :	1.4.2
DATE :	21 DEC, 2020

Abstract

This document presents an overview of the WINC3400 firmware release version 1.4.2, and corresponding driver.

1 Introduction

This document describes the WINC3400 version 1.4.2 firmware release package and corresponding driver. 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

- Fixed:
  - “Sweyntooth” BLE vulnerabilities
  - “Amnesia” TCP/IP security vulnerabilities
  - TCP TX stream freezing
  - Many other fixes and improvements (see section 6.1 for more detail)
- MS Azure support
- Reading of IQ calibration values from efuse
- Allow application to implement HTTPS via a proxy tunnel
- Allow BLE reprovisioning without having to re-pair
- BLE provisioning support with iOS13.x
- SHA224, SHA384 and SHA512 now supported for certificate verification

1.2 Firmware readiness

Microchip Technology Inc. considers version 1.4.2 firmware to be suitable for production release.

2 Release summary

2.1 Auditing information

Master Development Ticket	:	`https://jira.microchip.com/projects/W3400/versions/65690`

Wi-Fi:

Release Repository Branch : /chn-vm-svnrepo01.microchip.com/repo/wsg/Wifi_M2M/branches/rel_3400_1.4.2
Subversion Revision : **19093**

BLE:

Release Repository Branch	:	`/svn/Bluetooth/branches/ATWILC3400_BT_BLE_API`
Subversion Revision	:	**r7634**

BLE API:

Release Repository Branch	:	`/svn/Bluetooth/branches/ATWILC3400_BLE_API`
D21 Subversion Revision	:	**r7604**
SAM4 Subversion Revision	:	**r7604**

2.2 Version information

WINC Firmware version	:	1.4.2
Host Driver version	:	1.2.0
Host Interface Level	:	1.5

2.3 Released components

The release contains documentation, sources and binaries.

2.3.1 Documentation overview

The Application manuals, Release notes and Software API guides can be found in the `doc/` folder of the release package.

Release Notes:

This document

Software APIs:

WINC3400_IoT_SW_APIs.chm

WINC3400_BLE_APIs.chm

2.3.2 Binaries and programming scripts

The main 3400 firmware binary is in the `firmware` directory and named `m2m_image_3400.bin`. This can be flashed to a WINC device using, for example, a serial bridge application available from ASF.

An OTA image is provided in the `ota_firmware` directory named `m2m_ota_3400.bin`.

2.3.3 Sources

Source code for the host driver can be found under the `src/host_drv` directory.

Source code for the tools, including crypto_lib can be found under the `src/Tools` directory.

2.4 Release Comparison

Features in 1.3.1	**Changes in 1.4.2**
Wi-Fi STA	

• IEEE 802.11 b/g/n.
• OPEN, WEP security.
• WPA Personal Security (WPA1/WPA2), including protection against key re-installation attacks (KRACK).
• WPA/WPA2 Enterprise support:
  o EAP-TTLSv0/MSCHAPv2
  o EAP-PEAPv0/MSCHAPv2
  o EAP-PEAPv1/MSCHAPv2
  o EAP-PEAPv0/TLS
  o EAP-PEAPv1/TLS
  o EAP-TLS
• Simple Roaming Support
	• Add option to stop scanning on first result

Wi-Fi Hotspot	
• Only ONE associated station is supported. After a connection is established with a station, further connections are rejected. • OPEN and WEP security modes. • The device cannot work as a station in this mode (STA/AP Concurrency is not supported).	• Fix to ensure DHCP offered address is consistent when STA disconnects/reconnects • Fix to close race condition when a STA disconnects and reconnects that could cause the WINC to disallow all further connection attempts.
WPS	
• The WINC3400 supports the WPS protocol v2.0 for PBC (Push button configuration) and PIN methods.	No changes
TCP/IP Stack	

• The WINC3400 has a TCP/IP Stack running in firmware side. It supports TCP and UDP full socket operations (client/server). The maximum number of supported sockets is currently configured to 12 divided as:
  o 7 TCP sockets (client or server).
  o 4 UDP sockets (client or server).
  o 1 RAW socket
	• Fix TCP RX window leak
	• Address “Amnesia” vulnerabilities

Transport Layer Security	
• The WINC 3400 supports TLS v1.2, 1.1 and 1.0. • Client mode only. • Mutual authentication. • Supported cipher suites are: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_128_CBC_SHA256	• Add feature to allow a TCP socket to connect (as client) without TLS, then upgrade to TLS later • Fix to make client certificate type independent of cipher suite

◦ TLS_RSA_WITH_AES_128_GCM_SHA256 ◦ TLS_DHE_RSA_WITH_AES_128_CBC_SHA ◦ TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 ◦ TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 ◦ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (requires host-side ECC support eg ATECCx08) ◦ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (requires host-side ECC support eg ATECCx08) ◦ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (requires host-side ECC support eg ATECCx08) • TLS ALPN support	• Fix verification of certificate chains which include ECDSA signatures • SHA224, SHA384 and SHA512 verification capability added
Networking Protocols	
• DHCPv4 (client/server) • DNS Resolver • IGMPv1, v2. • SNTP	No changes
Power saving Modes	
• The WINC3400 supports these powersave modes: ◦ M2M_NO_PS ◦ M2M_PS_DEEP_AUTOMATIC • BLE powersave is always active	No changes
Device Over-The-Air (OTA) upgrade	
• The WINC3400 has built-in OTA upgrade. • Firmware is backwards compatible with driver 1.0.8 and later. • Driver is backwards compatible with firmware 1.2.0 and later (though the functionality will be limited by the firmware version in use).	No changes
Wi-Fi credentials provisioning via built-in HTTP server	
The WINC3400 has built-in HTTP provisioning using AP mode (Open or WEP secured).	No changes
WLAN MAC only mode (TCP/IP Bypass, or Ethernet Mode)	
Allow WINC3400 to operate in WLAN MAC only mode and let the host send/receive Ethernet frames.	• Ensure broadcast frames contain correct destination MAC address

	• Ensure NULL frames are sent to keep the AP connection alive during periods of low activity
ATE Test Mode	
Embedded ATE test mode for production line testing driven from the host MCU.	• Ensure ATE image is included in compound image • Fix TX test in demo application
Miscellaneous features	
Host FLASH API – allows a host to store and retrieve data on the WINC stacked flash.	• I/Q calibration values read and applied from efuse
BLE functionality	
BLE 4.0 functional stack	• Allow capture of RSSI of received advertising frames • Improve BLE powersave • Fix BLE pairing with iOSv13.x • Allow a device to reprovision the WINC without having to re-pair

3 Test Information

This section summarizes the tests conducted for this release

3.1 Internal testing

Please refer to ticket Jira:W3400-650 for full details.

Testing was performed against the release candidate 1.4.2 against the following configuration(s):

H/W Version	:	WINC3400 XPRO module
Host MCU	:	ATSAMD21-XPRO

For Elliptic Curve cryptography support verification, a CRYPTOAUTH XPLAINED PRO board (containing an ECC508A chip) was inserted into the EXT2 socket on the ATSAMD21-XPRO board.

Testing was performed in both open air and shielded environments.

The following testing has been performed:

1. General functionality including:
    1. HTTP Provisioning
    2. BLE API verification
    3. Station Mode
    4. AP Mode
    5. IP (TCP and UDP client and server)
    6. HTTP POST/GET
    7. WPS (PIN and PushButton methods)
    8. Over-The-Air (OTA) update functionality and robustness (with and without TLS)

2. TLS functionality including:
    1. RSA ciphersuites:
        i. TLS_RSA_WITH_AES_128_CBC_SHA
        ii. TLS_RSA_WITH_AES_128_CBC_SHA256
        iii. TLS_DHE_RSA_WITH_AES_128_CBC_SHA
        iv. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
        v. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

       Testing uses a 1024 bit server certificate, with a chain of 7 certificates of varying key lengths (1024,2048 and 4096 bit) leading to a 2048 bit root certificate.
    2. ECDSA ciphersuites:
        i. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

       Testing uses a NIST standard ECC P256 prime curve server certificate with two chains, one leading back to an ECC root certificate and the other leading to an RSA root certificate.
    3. SNI Client Hello extension
    4. Server certificate name validation
    5. Client authentication
    6. Amazon AWS IoT environment with client authentication and ciphersuite TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256. MQTT connection, publishing and subscribing all succeed.
3. Performance under interference
4. Wi-Fi AP interoperability testing
5. Regression and longevity tests
6. TCP/IP stack robustness testing
    a. Using an internal implementation of IPerf.
    b. Verification of multi socket functionality

Known issues are declared in Section 4

4 Known Issues

Jira	Severity	Description
W3400-605	Medium	Prolonged heavy IP traffic load can result in the SPI becoming unusable between the WINC3400 and the host. Observed with SAMD21 host and WINC powersave disabled. Could potentially occur with other host platforms, but not yet observed. Recommended workaround: On SAMD21 host, the frequency of the issue can be minimized by using M2M_PS_DEEP_AUTOMATIC when transferring IP traffic. The issue could be detected by checking the return value of an API such as m2m_get_system_time(). A negative return value indicates that the SPI is unusable. If this occurs, reset the system via system_reset(). Alternatively, m2m_wifi_reinit() can be used to reset just the WINC. In this case, the different driver modules also need to be initialized (m2m_ota_init(), m2m_ssl_init(), socketInit()).
W3400-621	Medium	The AP initiated group rekey process sometimes fails when the WINC is processing a high volume of receive traffic. Recommended workaround: Reconnect the Wi-Fi connection to the AP if a disconnection occurs due to this issue.
W3400-102	Medium	During HTTP provisioning, if applications are running on the device being used to provision the WINC3400, they will not be able to access the internet during provisioning. Furthermore, if they attempt to do so, then the WINC3400 can become flooded with DNS requests and crash. This applies to HTTP provisioning only; BLE provisioning is unaffected. Also, this only applies if powersave is enabled. Recommended workarounds: (1) Use M2M_NO_PS when WINC3400 is in HTTP provisioning mode. (2) Close other internet applications (browsers, skype etc) before HTTP provisioning. If crash occurs, reset system via system_reset(). Alternatively, m2m_wifi_reinit() can be used to reset just the WINC. In this case, the different driver modules also need to be initialized (m2m_ota_init(), m2m_ssl_init(), socketInit()).
W3400-40	Medium	The WINC3400 occasionally fails to proceed with 4-way handshake in STA mode, when using 11N WPA2. It does not send M2 after receiving M1. Recommended workaround: Retry the Wi-Fi connection.
W3400-293	Medium	1% of Enterprise conversations fail due to the WINC3400 not sending an EAP response. The response is prepared and ready to send but does not appear on the air. After 10 seconds the firmware times-out the connection attempt and the application is notified of the failure to connect. Recommended workaround: Configure the authentication server to retry EAP requests (with interval < 10 seconds). The application should retry the connection request when it is notified of the failure.

W3400-298	Medium	70% of Enterprise connection requests fail with a TP Link Archer D2 access point (TPLink-AC750-D2). The access point does not forward the initial EAP Identity Response to the authentication server. The issue is bypassed by PMKSA caching (WPA2 only), so reconnection attempts will succeed. Recommended workaround: The application should retry the connection request when it is notified of the failure.
W3400-461	Low	Sometimes the WINC3400 fails to see ARP responses sent from certain APs at 11Mbps. Recommended workaround: None. The ARP exchange will be retried several times and the response will eventually get through to the WINC3400.
W3400-60	Low	During BLE provisioning, the AP list is not cleaned up at the start of each scan request. As a result, the AP scan list can sometimes display duplicate or old scan entries. Recommended workaround: Only use one scan request during BLE provisioning.
W3400-59	Low	APIs at_ble_tx_power_get() and at_ble_max_PA_gain_get() return default values which do not correspond to the actual gain settings. Recommended workaround: None. Do not use these APIs.
W3400-30	Low	If the TLS server certificate chain contains RSA certificates with keys longer than 2048 bits, the WINC takes several seconds to process it. A Wi-Fi group rekey occurring during this time can cause the TLS handshake to fail. Recommended workaround: Retry opening the secure connection.
W3400-64	Low	at_ble_tx_power_set() needs special handling. Return values 0 and 1 should both be interpreted as successful operation. Refer to WINC3400_BLE_APIs.chm for more detail. Recommended workaround: Process the return value with care, according to the API documentation.
W3400-240	Low	After writing new firmware to the WINC3400, the first Wi-Fi connect attempt in STA mode takes an extra 5 seconds. Recommended workaround: Allow longer for the Wi-Fi connection to complete.
W3400-451	Low	When running in AP mode, the WINC3400 DHCP Server sometimes takes 5 to 10seconds to assign an IP address. Recommended workaround: Allow longer for DHCP to complete.

5 New Features

New features added since the previous released version (1.3.1).

5.1 Read and apply I/Q calibration values from efuse

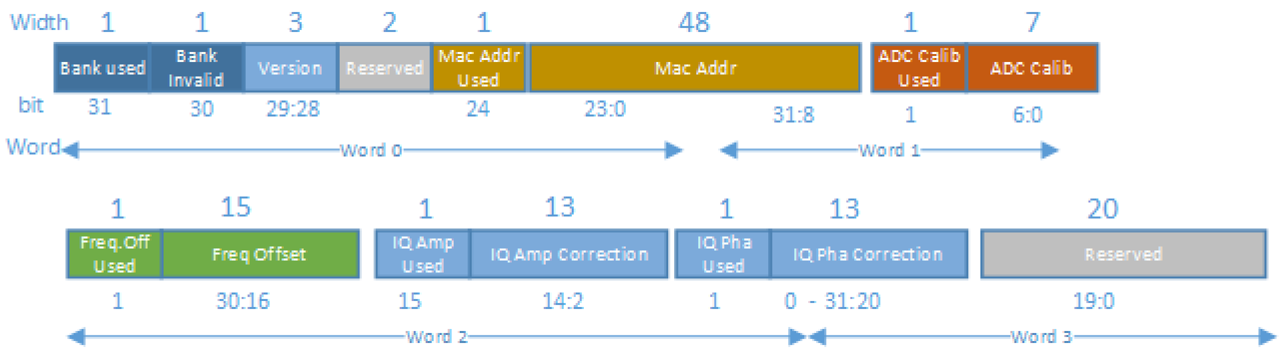
Transmitted RF is composed of a baseband signal modulated with an RF carrier signal.

The QAM modulated signal used in 802.11 Wi-Fi has two carrier components that have the same frequency but are phase shifted by 90 degrees - **I** and **Q** ("In-phase" and "Quadrature"). The transmitted signal is the sum of these two components.

In practice, there may be a slight imbalance between the amplitude and phase of the I and Q components. I/Q calibration compensates for this on a device basis, allowing for the best possible EVM.

5.1.1 Using the I/Q calibration feature

The WINC1500 has two registers, one for phase correction and one for amplitude correction that can be written to by firmware. The actual values programmed into these registers are obtained during production line testing and written to the WINC efuse as follows:



The scheme currently being used to store the WiFi MAC address and frequency offset value has been extended to store the I/Q Amplitude and Phase correction parameters. The existing method of invalidating a bank and writing a new bank to update any of the contents in the efuse can be used if the values need to be modified at any time.

Note – if I/Q calibration correction values are to be written to efuse, it is necessary to only use banks 1-5. Writing these values to bank 0 may have undesired consequences due to legacy bootrom code.

5.2 Add capability to allow application to obtain RSSI of received BLE advertising frames

5.2.1 Overview

The BLE API released with 1.4.2 has been extended to include the RSSI of the received advertising frame when passing details to the host in a `at_ble_scan_info_t` structure via the `AT_BLE_SCAN_INFO` event.

The struct now looks as follows:

```
typedef struct
{
                at_ble_adv_type_t type;
                at_ble_addr_t dev_addr;
                uint8_t adv_data[AT_BLE_ADV_MAX_SIZE];
                uint8_t adv_data_len;
                uint8_t rssi;
}at_ble_scan_info_t;
```

The new member is highlighted in bold.

5.3 Upgrade a TCP socket to TLS after connection as a client

5.3.1 Overview

This feature allows a TCP socket to be created in one of the following configurations:

i) Not secure

ii) Secure

iii) Potential to be secure

Configuration (iii) “potential to be secure” is new in this release.

A TCP socket created with the “potential to be secure” configuration can connect (as a client), but will remain insecure until the application calls the secure() API. In the meantime, the socket is a plain TCP socket and the application may send and receive data unencrypted.

This configuration allows the application to implement HTTPS via a proxy tunnel, as described in https://tools.ietf.org/html/rfc7230.

5.3.2 Usage

These steps describe how to make use of the feature:

1. Create a socket using the socket() API, with the u8Type parameter set to SOCK_STREAM and the u8Config parameter set to SOCKET_CONFIG_SSL_DELAY.
2. Open a TCP connection on the socket using the connect() API and await notification of successful connection via the SOCKET_MSG_CONNECT message.
3. Send and receive data unencrypted over the TCP connection, using the send() and recv() APIs. As an example, this data may comprise an HTTP CONNECT message, in order to set up a proxy tunnel to a secure server.
4. Make the socket secure by calling the secure() API. This initiates a TLS handshake. Await notification of successful securing via the SOCKET_MSG_SECURE message.

Notes:

- If either connecting or securing fails, the application must call the close() API to close the socket. Even if the failure is at the securing stage, the socket must be closed in this way.
- If a TCP connection is established via listening and accepting a connection, the connection cannot be made secure via the secure() API.

Please refer to *WINC3400_IoT_SW_APIs.chm* for full details of these APIs and their parameters.

6 Fixes and Enhancements

These are the fixes and enhancements since the previous released version (1.3.1).

6.1 Issues fixed

Jira ID	Description
W3400-483	**In AP mode, the WINC3400 offers 0.0.0.0 as the subnet to clients that connect and restart** When running the WINC as an AP, if a client connects then power cycles and reconnects, the WINC erroneously offers a 0.0.0.0 subnet to that client Resolution: Fix a bug in firmware so the WINC offers the correct subnet
W3400-484	**WINC expects TLS client certificate to use the same type of key as the server certificate** The WINC expects the client and server certificates to both be RSA or both be ECDSA. Resolution: Fixed so client and server certificates can use different key types.
W3400-485	**Destination MAC address of broadcast frames replaced by WINC MAC address** When running in bypass ("eth") mode, the MAC address of received broadcast frames is re-placed by the WINC MAC address. Resolution: Leave the destination MAC address untouched in received broadcast frames
W3400-488	**BLE powersave not working** The BLE processor isn't running in optimal powersave mode. Resolution: WINC now powersaves optimally when running as a BLE peripheral device.
W3400-496	**Scan does not display different bssid with same ssid in scan results** If two different APs exist configured with the same SSID and within range from the WINC, when instructing the WINC to perform a scan, the scan results only include one of the APs - typically the one with the strongest signal. Resolution: If multiple APs with same SSID are picked up by the scan, display each one as an individual result.
W3400-506	**Race condition causes "AP Full" error in AP mode** When running the WINC as an AP, a race condition can occur between one station disconnect-ing and a second station connecting, which could cause some frames to be discarded and cause the WINC to "lock" by not allowing any more stations to connect. Resolution: Protect the disconnection sequence so that it is not interrupted.
W3400-507	**WINC does not update the remote peer IP address when running TLS server** When running in TLS server mode, the WINC does not populate the IP address of the remote peer device before informing the host. Resolution: WINC now populates the IP address even when running in TLS server mode.

W3400-508	**BLE pairing fails with iOS v13.x** Phones running iOS v13.x are unable to pair with the WINC Resolution: Relaxing some checks in the SMP handshake allows pairing to succeed.
W3400-513	**Bypass/ETH mode fails to send NULL frames** When running in ETH mode (bypassing the WINC onboard network stack), the WINC fails to send NULL frames to keep the AP connection alive during periods of no traffic. Resolution: NULL frames are now transmitted
W3400-519	**Phones sometimes disconnect from WINC running in AP mode** Some phones connect to the WINC in AP mode and subsequently disconnect for no apparent reason. Resolution: If the WINC is sent many packets from a phone on connection it can temporarily become resource starved. This was being treated as a terminal error in firmware. When treated as a temporary error, the WINC recovers and the phone remains connected.
W3400-527	**The TCP window grows too large when receiving TCP RX traffic** During receipt of a TCP TX stream, the TCP RX window size increases and becomes much larger than expected. Resolution: Fix a leak in the TCP window management code.
W3400-534	**Fix Sweyntooth BLE vulnerability** The WINC3400 BLE stack is vulnerable to the attacks described in the Sweyntooth security vulnerability published by the Singapore University of Technology and Design. Resolution: The vulnerabilities have been fixed in the WINC3400 BLE code.
W3400-540	**WINC attempts to process TLS certificate chains which contravene its signature algorithms extension** If the host processor does not indicate support for ECC processing, the WINC excludes ECDSA from its signature algorithms hello extension. If the server sends an ECDSA signature (in contravention of this), the WINC attempts to process it, passing it up to the host processor. Resolution: WINC rejects certificate chains which contravene its signature algorithms extension.
W3400-544	**APIs for Enterprise connection return success when they fail due to memory allocation failure** Resolution: These APIs have now been fixed to return the relevant result code when encountering memory issues.
W3400-549	**Preparation for TLS ECDSA verification is broken.** When assembling the list of signatures for verifying the server certificate chain and server key exchange parameters, an ECDSA signature can cause the list to become corrupted, in which case some or all verifications may be bypassed. Resolution: the list does not get corrupted and all signatures are passed on for verification.

W3400-550	**Signature verification broken for ECDSA-SHA, ECDSA-SHA224, ECDSA-SHA384 and EC-DSA-SHA512** ECDSA signature verification only works if the hash digest is 32 bytes (i.e. SHA256). Resolution: Hashes with shorter/longer digests are handled correctly for ECDSA verification.
W3400-555	**Incorrect parsing of ECDSA signatures** If individual coordinates are shorter or longer than the curve's field size they are processed in-correctly. Resolution: short coordinates are prepended with 0's; long coordinates are rejected.
W3400-557	**ATE image not included in firmware compound image** Resolution: Ensure the ATE image is included in the firmware package.
W3400-573	**ATE TX test fails to send frames** The example application to load the ATE firmware and control it from the host application was failing to transmit frames. Resolution: Correctly configuring the ATE in the application fixes the issue.
W3400-577	**Certificate "Valid from" field ignored** A TLS server certificate chain is accepted even if the certificate is only valid from some future date. Resolution: Certificate verification now fails if a certificate in the chain is only valid from some future date. Note, however, that the "valid from" field of the root certificate is still ignored.
W3400-595	**Repeatedly roaming between APs results in the WINC running out of memory** Resolution: Fixed a memory leak during roaming.
W3400-598	**TLS handshake failure with some certificates** The TLS "finished" message is encrypted incorrectly when it spans two internal buffers. Resolution: Ensure encrypted TLS messages do not span two internal buffers.
W3400-611	**Fix CERT Amnesia vulnerabilities** CERT released a suite of vulnerabilities that affect the uIP stack used internally by the WINC3400. Resolution: Address all relevant vulnerabilities in WINC3400 firmware.
W3400-617	**Transmitted TCP stream from the WINC3400 can suddenly stop** When the WINC3400 sends a TCP packet that hits maximum transmit retries, it tries once more but with an incorrect 802.11 sequence number which means subsequent messages get dis-carded by the AP. Resolution: Stop the WINC3400 from re-sending the TCP packet with the erroneous 802.11 se-quence number

6.2 Enhancements

W3400-503	**Poor performance with some 802.11n APs** Certain APs, including the Ubiquity Unify AP-AC-LR exhibit poor 802.11n performance with the WINC3400. Resolution: Set the channel estimation scheme to HT-LTF which greatly improves performance with these APs.
W3400-538	**Add SHA224, SHA384, SHA512 capabilities for certificate verification** TLS server chains containing SHA224, SHA384 and SHA512 hashes can now be verified
W3400-541	**Add option to stop scan on first result** Previously, every time a scan was requested, the WINC would scan for a pre-determined amount of time and would never return earlier, even if it had found the SSID it was looking for, causing higher power consumption and longer waiting times. This version introduces a new scan option which lets the application configure the WINC to return the scan results upon finding the first SSID (useful for example for directed scan).
W3400-553	**New m2m_ssl APIs for ECDSA verification** New API m2m_ssl_retrieve_next_for_verifying() is an improvement over m2m_ssl_retrieve_cert(), allowing the application to indicate the lengths of the buffers that it is providing for receiving the verification signature and value. New API m2m_ssl_stop_retrieving() is an improved name for m2m_ssl_stop_processing_certs(). The old APIs are still present for legacy applications.
W3400-583	**The PLL lookup table is not being calculated on a per-board basis when writing the WINC firmware** Each WINC has a custom PLL lookup table, calculated from the XO offset which is present in efuse. This table is being calculated using a fixed XO offset so is not optimal for each board. Resolution: Modify flash scripts to ensure the PLL table is calculated using the board specific XO offset.
W3400-478	**Allow a BLE device to reprovision without having to re-pair.** A BLE central device cannot reprovision the WINC having already provisioned it without unpairing and re-pairing. The **wifiprov_configure_provisioning()** API is modified to accept an **at_ble_auth_t** parameter which allows the caller to specify the authentication type used in provisioning. An auth type of **AT_BLE_AUTH_MITM_NO_BOND** allows a device to reprovision without having to pair. Note: In the scenario where only firmware is updated and not host software, the old format API call will still be recognised and functional.

7 Terms and Definitions

Term	Definition
AES	Advanced Encryption Standard
ALPN	Application Layer Protocol Negotiation
ARP	Address Resolution Protocol
BLE	Bluetooth Low Energy
BSS	Basic Service Set
CBC	Cyclic Block Chaining
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name Server
DTIM	Directed Traffic Indication Map
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)
GAP	Generic Access Profile
HTTP	Hypertext Transfer Protocol
IBSS	Independent BSS (ad-hoc network)
IEEE	Institute of Electronic and Electrical Engineers
MIB	Management Information Base
NDIS	Network Driver Interface Specification
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
TIM	Traffic Indication Map
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
WPA™	Wi-Fi Protected Access
WPA2™	Wi-Fi Protected Access 2 (same as IEEE 802.11i)