WINC1500 Software

Release Notes

VERSION : **19.7.7**

DATE : **MAR, 2022**

Abstract

This document presents an overview of the WINC15x0 firmware release version 19.7.7, and corresponding driver.

1 Introduction

This document describes the WINC15x0 version 19.7.7 release package.

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

1.1 Firmware readiness

Microchip Technology Inc. considers version 19.7.7 firmware to be suitable for production release.

2 Release summary

2.1 Auditing information

Master Development Ticket	:	https://jira.microchip.com/projects/W1500/versions/69459
Release Repository	:	Wifi_M2M
Source Branch	:	/branches/rel_1500_19.7.7
Subversion Revision	:	r19772

2.2 Version information

WINC Firmware version	:	19.7.7
Host Driver version	:	19.7.7
Minimum driver version	:	19.3.0

Please note that the SVN revision advertised in the firmware serial trace will be **19759**.

```
(10)NMI M2M SW  VER 19.7.7 REV 19759
(10)NMI MIN DRV VER 19.3.0
(10)FW URL branches/rel_1500_19.7.7
(10)Built Mar 30 2022   13:32:43
```

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:

WINC1500_IoT_SW_APIs.chm

2.3.2 Binaries and programming scripts

The main WINC15x0 firmware binary is located in the `firmware` directory and named `m2m_aio_3a0.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_3a0.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 19.7.6	Changes in 19.7.7
Wi-Fi STA	
• IEEE 802.11 b/g/n. • OPEN security. • WPA Personal Security (WPA1/WPA2). • WPA Enterprise Security (WPA1/WPA2) supporting: EAP-TTLSv0/MS-Chapv2.0 EAP-PEAPv0/MS-Chapv2.0 EAP-PEAPv1/MS-Chapv2.0 EAP-TLS EAP-PEAPv0/TLS EAP-PEAPv1/TLS 'Fragattack' countermeasures	• Fix to ignore unknown OUI in message 3 of 3-way handshake
Wi-Fi Hotspot	
• Only ONE associated station is supported. After a connection is established with a station, further connections are rejected. • OPEN and WPA/WPA2 security modes. • The device cannot work as a station in this mode (STA/AP Concurrency is not supported). • 'Fragattack' countermeasures.	• Fixed handling of source address when forwarding ARP packets out from the host.
WPS	
The WINC1500 supports the WPS protocol v2.0 for PBC (Push button configuration) and PIN methods.	No change
TCP/IP Stack	
The WINC1500 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 11 divided as: • 7 TCP sockets (client or server). • 4 UDP sockets (client or server).	No change
TLS	
• Support TLS v1.2. • Client and server modes. • Mutual authentication in client mode.	• TLS client mode works with Subject Alternative Names in server certificate

Features in 19.7.6	Changes in 19.7.7
• X509 certificate revocation scheme. • SHA384 and SHA512 support in X509 certificates processing. • Integration with ATECC508 (ECDSA and ECDHE support). • Supported cipher suites are: TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (requires ECC508) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (requires ATECC508)	
Networking Protocols	
DHCPv4 (client/server) DNS Resolver IGMPv1, v2. SNTP	No change
Power saving Modes	
• M2M_PS_MANUAL • M2M_PS_DEEP_AUTOMATIC	No change
Device Over-The-Air (OTA) upgrade	
• Built-in OTA upgrade available. • Backwards compatible as far as 19.4.4, with the exception of: - Wi-Fi Direct (removed in 19.5.3) - Monitor mode (removed in 19.5.2)	• Allow OTA to use SSL options such as SNI and server name verification
Wi-Fi credentials provisioning via built-in HTTP server	
Built-in HTTP/HTTPS (TLS server mode) provisioning using AP mode (Open, or WPA/WPA2 secured).	No change

Features in 19.7.6	Changes in 19.7.7
Ethernet Mode (TCP/IP Bypass)	
Allow WINC1500 to operate in WLAN MAC only mode and let the host send/receive Ethernet frames.	No change
ATE Test Mode	
Embedded ATE test mode for production line testing driven from the host MCU.	No change
Miscellaneous features	
	• Removal of obsolete python scripts in release package, as image_tool now natively supports the functionality.

3 Test Information

Please refer to ticket W1500-837 for full details.

Testing was performed against the release candidate 19.7.7 against the following configuration(s):

H/W Version : WINC1510 Xplained module

Host MCU : ATSAMD21-Xplained

The following testing was performed in both open air and shielded environments;

1. General functionality including:
    1. HTTP Provisioning
    2. Station Mode
    3. AP Mode
    4. IP (TCP and UDP client and server)
    5. HTTP POST/GET
    6. WPS (PIN and PushButton methods)
    7. Over-The-Air (OTA) update functionality and robustness (with and without TLS)
2. TLS functionality including:
    1. RSA cipher-suites:
        i. TLS_RSA_WITH_AES_128_CBC_SHA
        ii. TLS_RSA_WITH_AES_128_CBC_SHA256
        iii. TLS_RSA_WITH_AES_128_GCM_SHA256
        iv. TLS_DHE_RSA_WITH_AES_128_CBC_SHA
        v. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
        vi. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

        Testing uses 1024-bit, 2048-bit and 4096-bit server certificates, 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
        ii. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

        Testing uses a NIST standard ECC P256 prime curve server certificate with two chains, one leading back to an ECDSA root certificate and the other leading to an RSA root certificate.
    3. Client authentication
3. Performance under interference
4. TCP/IP stack robustness testing
    1. Using an internal implementation of IPerf.
    2. Verification of multi socket functionality

4 Known issues

ID	Description
W1500-63	Occasionally WINC15x0 fails to receive an individual UDP broadcast frame when in M2M_PS_DEEP_AUTOMATIC powersave mode. Recommended workaround: **Use M2M_NO_PS powersave mode if reliability is preferred for UDP broadcast frames. Otherwise ensure the overlying protocol can handle the odd missing frame.**
W1500-108	The WINC15x0 cannot handle two simultaneous TLS handshakes, due to memory constraints. Recommended workaround: **When attempting to open two secure sockets in STA mode, the application should wait to be notified of the first one completing (succeeding or failing) before attempting the second one.**
W1500-325	1% of Enterprise conversations fail due to the WINC15x0 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.**
W1500-369	When connected to certain access points, the WINC15x0 sometimes fails to roam when the access point changes channel. The issue is seen with these access points: Linksys E2500, Linksys E4200, Linksys 6500. The failures to roam are due to two issues: 1. Sometimes the access point takes a long time to start sending beacons or probe responses on the new channel, so it is not discoverable. 2. Sometimes the access point does not initiate the 4-way handshake (for WPA/WPA2 PSK reconnections). Recommended workaround: **On reception of M2M_WIFI_DISCONNECTED event, the application should attempt to discover the access point using `m2m_wifi_request_scan()` API.**

W1500-387	If an AP uses an 802.11 ACK policy of “No Ack”, then the WINC15x0 sometimes fails to receive 802.11b frames. Recommended workaround: **Avoid using an ACK policy of “No Ack”. If “No Ack” is used, ensure frames are sent at 802.11g or higher rates.**
W1500-397	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.**
W1500-402	Occasionally during AP provisioning, after entering the credentials of the AP to connect to and pressing “connect”, an error will be returned even though provisioning was successful and the connection proceeds. Recommended workaround: **Add a delay in the application between receiving the provisioning info and connecting to the AP. Ignore the “Request Failed” message.**
W1500-699	When using a driver pre – 19.6.0 with this firmware, upon failure to obtain a DHCP address the WINC will not trigger a WiFi Disconnection and notify the driver of the failure. Recommended workaround: **In this case of an older driver running with later firmware, the application should monitor the time taken to obtain a DHCP address, if it takes too long then it can decide whether to disconnect and try again.**
W1500-854	When sustaining a maximum throughput TLS RX stream using a TLS record size of 16K, if another maximum throughput TLS RX stream with the same record size is established, the streams sometimes become stuck and fail to transfer any more data. Recommended workaround: **Try to avoid using 2 concurrent TLS streams receiving 16K records. If this scenario is used and the streams become stuck, close and reopen the sockets and try again.**
W1500-836	TLS Server mode does not work with ECDHE ciphersuites. Recommended workaround: **Disable ECDHE ciphersuites before starting TLS Server.**
W1500-865	When running a sustained high throughput TCP RX stream over a long period (>1 hour), the WINC can sometimes miss enough beacons from the AP to trigger link loss detection and will disconnect from the AP. Recommended workaround: **If the WiFi connection is closed by the WINC during a TCP RX stream, reconnect and re-open the socket and try again.**

5 New Features

New SSL options for OTA from an https server.

It is now possible to configure SSL related options for use by the WINC when it conducts an OTA from a server using TLS (via https).

The configuration is performed using the new API:

```
sint8 m2m_ota_set_ssl_option(tenuOTASSLOption enuOptionName, const void *pOptionValue, size_t OptionLen);
```

The configurable options defined in `tenuOTASSLOption` are:

`WIFI_OTA_SSL_OPT_BYPASS_SERVER_AUTH`

Bypass the authentication of the remote server.

Type is `int`, value 1=bypass server authentication, 0=authenticate the server.

`WIFI_OTA_SSL_OPT_SNI_VALIDATION`

Check the server name in the received subject name against the server name specified with `WIFI_OTA_SSL_OPT_SNI_SERVERNAME`.

Type is `int`, value 1=perform the check, 0=do not perform the check.

`WIFI_OTA_SSL_OPT_SNI_SERVERNAME`

Server name to send in the TLS SNI extension.

Type is null terminated string.

The options set via `m2m_ota_set_ssl_option` will be used for every subsequent OTA, and will be reset when the board restarts.

It is possible to get the currently configured options using the function `m2m_ota_get_ssl_option()`.

Further details can be found in the API documentation provided with the release.

6 Fixes and enhancements

These are the major fixes and enhancements since the previous released version (19.7.6).

6.1 Issues fixed

ID	Description
W1500-510	**Crash in TLS server mode if a DHE-RSA ciphersuite is selected and the WINC's server certificate is signed using ECDSA.** The software allocates insufficient memory for the DHE server key exchange message, resulting in buffer overflow and crash. Fixed: Memory is allocated correctly.
W1500-824	**Cross-signed TLS certificate chains are rejected once the original root certificate expires.** If the WINC has an expired entry in its root certificate store, it rejects any certificate chains which lead to it, even if it also has a non-expired entry which could be used to verify the chain. Fixed: When an expired root certificate is encountered, continue to search for a different, valid root certificate.
W1500-745	**Handling of source address in outgoing ARP packets when acting as AP in bypass mode** When running as an AP in bypass mode (i.e. the network stack is on the host MCU), the WINC overwrites the source address of ARP packets sent from the stack with it's own MAC address. This is incorrect behavior and results in connected stations sending unicast traffic to the WINC instead of through it to the stack. Fixed: Do not modify the source address of ARP packets in this configuration.
W1500-756	**WINC1500 responds to TCP SYN on port 0** If a TCP SYN is sent to the WINC on port 0, it responds with SYN/ACK. There should be no response sent to a SYN on port 0. Fixed: Send no response if a SYN is received on port 0.
W1500-800	**Some host MCUs lose SPI communication with the WINC1500 as it wakes up from sleep** When running the SPI bus at high speeds (around 40MHz), some hosts fail to read WINC registers over SPI around the WINC wakeup procedure. This results in loss of communication with the WINC. This has only been internally observed on a SAME54 host. Fixed: Decreasing the bus speed to around 10MHz during WINC wakeup fixes the problem. A framework has been added to the driver to allow the bus wrapper to lower the bus speed around WINC wakeup via calls to nm_bus_speed() which should be implemented on a per host basis.

	This function expects a single parameter – LOW or HIGH. When called with LOW, the bus speed should be decreased, and when called again with HIGH it should be reverted.
W1500-808	**AP mode connection instability** In AP mode, an authentication attempt by a STA when there is already an ongoing authentication attempt can cause the WINC1500 to crash Fixed: The state machine has been adjusted to handle this scenario gracefully.
W1500-811	**Unknown OUI causes 4-way handshake to fail** If an AP includes an unknown OUI (such as an AKM suite) in its RSNE, the WINC1500 fails to complete the 4-way handshake. Fixed: Ignore unknown OUIs, allowing the handshake to complete.
W1500-820	**Firmware crash when using defragmentation.** A race condition when defragmenting a frame at the 802.11 layer can result in a firmware crash. Fixed: Rework the code to remove the race condition.
W1500-830	**Race condition between timer delete and timer start can cause the timer not to start** A rare race condition can occur in WINC firmware when internal timers are used, which can result in the internal timer failing to start. If this occurs the effect can be wide ranging, depending on where the timer is being used. Fixed: Checks in the code are now improved, and the race condition is closed.

6.2 Enhancements

W1500-35	**Consider Subject Alternative Names when verifying TLS server name** If server name verification is enabled (via SO_SSL_ENABLE_SNI_VALIDATION or WIFI_OTA_SSL_OPT_SNI_VALIDATION), the verification succeeds if the server name matches the Common Name or any of the Subject Alternative Names in the server certificate.
W1500-827	**Removal of python helper scripts** The image_tool image creation utility has been enhanced which removes the need for two python helper scripts which are now removed from the release package: image_tool now reads the gain table directly without having to convert into a supported format, so gain_converter.py is obsolete and removed. image_tool now natively reads the xo offset from flash to compute the PLL table, which renders the extract_xo_offset.py and update_pll_table.bat scripts obsolete. These changes are mostly internal and do not affect the usage of the prepare_image.cmd script. For the remaining python scripts, checking of the correct version of python has been improved, with a relevant warning given if the wrong version is found.

7 Appendix A – TLS Root certificates

The WINC1500 19.7.7 module comes with a preselected selection of TLS root certificates that will allow a TLS connection to be established with a range of internet TLS servers out of the box.

These preselected certificates are described in 7.1

7.1 TLS root certificates

Issuer	Filename	Expiry	Public Key	Signature Alg.	Notes
Amazon Root CA 1	AmazonRootCA1.cer	17 January 2038 01:00:00	RSA (2048 bits)	SHA256RSA	AWS Cloud
Baltimore CyberTrust Root	BaltimoreCyber-TrustRoot.cer	13 May 2025 00:59:00	RSA (2048 bits)	SHA1RSA	Azure Cloud
DigiCert High Assurance EV Root CA	DigiCert.cer	10 November 2031 01:00:00	RSA (2048 bits)	SHA1RSA	
DigiCert High Assurance EV Root CA	DigiCertSHA2.cer	22 October 2028 13:00:00	RSA (2048 bits)	SHA256RSA	
Entrust Root Certification Authority	EnTrust.cer	27 November 2026 21:53:42	RSA (2048 bits)	SHA1RSA	
GlobalSign Root CA	GlobalSignRoot.cer	28 January 2028 13:00:00	RSA (2048 bits)	SHA1RSA	
Internet Security Research Group Root X1	isrgrootx1.cer	04 June 2035 12:04:38	RSA (4096 bits)	SHA256RSA	LetsEncrypt
QuoVadis Root CA 2	QuoVadis_Root.cer	24 November 2031 19:23:33	RSA (4096 bits)	SHA1RSA	
VeriSign Class 3 Primary Certification Authority	VeriSign.cer	17 July 2036 00:59:59	RSA (2048 bits)	SHA1RSA	

8 Terms and Definitions

Term	Definition
AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript and XML
AKM	Authentication and Key Management
ARP	Address Resolution Protocol
ATE	Automated Test Equipment
BSS	Basic Service Set
CBC	Cyclic Block Chaining
DHCP	Dynamic Host Control Protocol
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name Server
DTIM	Directed Traffic Indication Map
EAP	Extensible Authentication Protocol
EAPOL	EAP 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
EVM	Error Vector Magnitude
HIF	Host Interface
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electronic and Electrical Engineers
MAC	Media Access Control
OTA	Over The Air update
PEAP	Protected Extensible Authentication Protocol
PLL	Phase Locked Loop
PMK	Pair-wise Master Key
PSK	Pre-shared Key
QAM	Quadrature Amplitude Modulation
RSA	Rivest-Shamir-Adleman (public key cryptosystem)
RSN	Robust Security Network
RSSI	Receive Strength Signal Indicator
SHA	Secure Hash Algorithm
SNTP	Simple Network Time Protocol
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIM	Traffic Indication Map
TLS	Transport Layer Security

Term	Definition
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)