WINC3400 Software

Release Notes

VERSION : 1.2.2

DATE : 30 OCT, 2017

Abstract

This document presents an overview of the WINC3400 firmware release version 1.2.2, and corresponding driver.

1 Introduction

This document describes the WINC3400 version 1.2.2 firmware RTP 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

- 802.11 power save support (refer to section 6.2)
- Wi-Fi passive scan support (refer to section 5.1)
- Stability and interoperability improvements (refer to section 6.1)

1.2 Firmware readiness

Microchip Technology Inc. considers version 1.2.2 firmware to be suitable for production release.

2 Release summary

2.1 Auditing information

Master Development Ticket : `Sprint:548`

Wi-Fi:

Release Repository Branch : svn/Wifi_M2M/branches/rel_3400_1.2.2
Subversion Revision : **15675**

BLE:

Release Repository Branch : `/svn/Bluetooth/branches/ATWILC3400_BT_BLE_API`
Subversion Revision : **r7149**

BLE API:

Release Repository Branch : `/svn/Bluetooth/branches/ATWILC3400_BLE_API`
D21 Subversion Revision : **r7136**
SAM4 Subversion Revision : **r7136**

2.2 Version information

WINC Firmware version : 1.2.2
Host Driver version : 1.0.8
Host Interface Level : 1.3

The firmware reports revision : **15572**

```
(10)NMI M2M SW  VERSION 1.2.2
(10)NMI HIF LEVEL (2) 1.3
(10)Built at Oct  5 2017        13:22:37
(10)SVN: 75:15572M
```

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_aio_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 (new for 1.2.x releases), can be found under the `src/Tools` directory.

3 Test Information

This section summarizes the tests conducted for this release

3.1 Internal testing

Please refer to ticket Jira:W3400-92 for full details.

Testing was performed against the release candidate 1.2.2 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)
    9. MCU access to WINC flash (firmware images and TLS root certificate store) via m2m_flash APIs (tested using a ATSAM4S-XPRO MCU for internal storage of large files)

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. HIF backwards compatibility of current firmware (HIF level 1.3) with older driver (HIF level 1.x)
4. Performance under interference
5. Manual Wi-Fi browser interoperability testing
6. Wi-Fi AP interoperability testing
7. Regression and longevity tests
8. TCP/IP stack robustness testing
    a. Using an internal implementation of IPerf.
    b. Verification of multi socket functionality
9. Wi-Fi and BLE coexistence testing under extreme throughput conditions (tested using a ATSAM4S-XPRO MCU to achieve high throughput). Throughput rates of around 18Mbps were achieved using piperf, and when BLE advertising/connection was brought in the rate only dropped by around 0.5Mbps.
   No adverse effects were seen, and spurious emissions were not present.
10. V1.2.1 is a small patch to address problems when the NTP servers encoded in the FW, so v1.2.1 had extra testing only in the areas affected in addition to standard automated testing of major functional blocks.

Known issues are declared in Section 0

4 Known Issues

TRAC ID	Severity	Description
9292	**Medium**	**Stuck transmitter leading to out of memory** On rare occasions, the radio transmitter locks up and is unable to transmit. Normally the situation is recovered internally within a period of approximately 1 second. However, very occasional instances have been seen where this recovery fails. **Workaround** 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()).
9299	**Medium**	**Applications attempting to access internet during HTTP provisioning can cause WINC3400 to crash** If applications are running on the device being used to provision the WINC3400, they will not be able to access the internet during provisioning. However, if they attempt to do so, then the WINC3400 can become flooded with DNS requests and crash. This applies to HTTP provisioning only. **Workaround** 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()).
8085	**Low**	**WINC3400 sometimes fails to receive broadcast ARP for prolonged periods** Sometimes the WINC3400 fails to see ARP responses sent from certain APs at 11Mbps. **Workaround** None. The ARP exchange will be retried several times and the response will eventually get through to the WINC3400.
8212	**Low**	**BLE Provisioning AP list is not cleaned up on re-scan request** If phone app or host application request multiple Wi-Fi scans as part of BLE provisioning, the AP scan list can sometimes display duplicate or old scan entries. **Workaround** None
8436	**Low**	**Gains returned by at_ble_tx_power_get() and at_ble_max_PA_gain_get() are incorrect** These two APIs return default values which do not correspond to the actual gain settings. **Workaround** None
8858	**Low**	**TLS handshake can fail if rekey occurs during server authentication** If the TLS server certificate chain contains RSA certificates with key 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. **Workaround** Retry opening the secure connection.
9290	**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. **Workaround** Process the return value with care, according to the API documentation.

5 New Features

There is one new feature added since the previous released version (1.1.5).

5.1 Wi-Fi Passive Scan

Wi-Fi passive scan is now implemented, for a device in STA mode. The application can call API m2m_wifi_request_scan_passive() to initiate passive scanning. The WINC then listens for Wi-Fi beacons then calls back to the application with information about the peer devices it heard.

5.1.1 Options

The application can set the following options relating to passive scan:

- Wi-Fi channel to scan (or all channels, according to region)
- Number of scans per channel (default 2)
- Length of time per scan (default 300ms)

5.1.2 Callback

The application will be notified of passive scan completion via event M2M_WIFI_RESP_SCAN_DONE to the callback previously registered via m2m_wifi_init().

On receipt of this event, the application can call m2m_wifi_req_scan_result(). The application will then be notified of passive scan completion via event M2M_WIFI_RESP_SCAN_RESULT.

Note that this notification sequence is the same as is used for active scan.

5.1.3 API details

Full details of the relevant APIs are available in WINC3400_IoT_SW_APIs.chm

6 Fixes and Enhancements

These are the fixes and enhancements since the previous released version (1.1.5).

6.1 Issues fixed.

TRAC ID	Description
8273	**BLE API at_ble_adv_set_tx_power() not working** Calling at_ble_adv_set_tx_power(-5) doesn't return and the D21 remains locked as the call doesn't return. **Resolution:** The API is marked as not implemented. If it is called, it will return AT_BLE_FAILURE.
8754	**Default connection attempt occasionally fails** After connecting to different access points several times, the API m2m_wifi_default_connect() occasionally fails to trigger connection to most recent AP. **Fixed:** Fixed algorithm for retrieving parameters for connecting to most recent AP.
8970	**TLS session remote closure not handled by WINC** 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. **Fixed:** WINC closes the session, closes the socket and reports socket closure to the application.
9050	**Auto-rate algorithm can get stuck at low rates** 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. **Fixed:** Disabled short preamble operation, to allow rate to recover through 2 and 5.5 Mbps. Tx success threshold reduced from 90% to 80%, to allow rate to recover through 11Mbps.
9079	**Occasional failure to connect to D-Link AP in 11n-WPA2 and mixed mode** This is an interoperability issue where the access point fails to increment the key replay counter, causing the connection to fail. **Fixed:** WINC discards incorrect frames and connection succeeds.
9142	**Incorrect parsing of DNS reply** Inter-op issue. A DNS server may (unusually) send DNS answer records in uncompressed format. The WINC DNS resolver does not parse these correctly, resulting in reporting of failed lookup. **Fixed:** Fixed parsing of DNS answer records.
9148	**Driver and firmware can get out of sync when scanning** Driver attempts to keep state of whether WINC is Wi-Fi scanning or not. However it is possible for the state to get out of sync, preventing future scan requests. **Fixed:** Remove scanning state in driver.

TRAC ID	Description
9203 **WSGA-887**	**Internal OTA fails with some HTTP servers** Inter-op issue. The WINC HTTP client fails if the HTTP header does not include a “content-length” field. However that field is optional. **Fixed:** HTTP client proceeds even if content length is not included in HTTP header.
9232 **WSGA-1123**	**Does not connect to TPLink Archer D2 router** Inter-op issue. The WINC fails to handle the particular order of supported data rates in the probe response. **Fixed:** WINC processes the list of supported data rates correctly.
9260 **WSGA-1174**	**TLS certificate server name check incorrect in wildcard case** Wildcard certificate name eg *.google.com incorrectly matches abcdgoogle.com. **Fixed:** Parsing of wildcard names corrected.
9261	**Unable to receive broadcast packets when configured with static IP** Broadcast address not set as part of static IP configuration. **Fixed:** Broadcast address is set.
9264 **WSGA-1194**	**RSSI measurement for CCA is too low** RSSI measurements are typically 6-8 dB below actual values. This is thought to be impacting RED certification. **Changed:** RSSI calibration register updated with new values per channel. Affect on performance seems to be neutral or positive. Affect on certification not yet known.
9303	**NTP server list not traversed in some error conditions.** V1.2.0 and earlier had a fixed list of NTP servers, when one of these failed the next server was not selected. **Fixed:** Now the FW uses the NIST NTP server pool and if that fails falls back to the ntp.org pool.
W3400-71	**Fix security issue** Address “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2” issue.

6.2 Enhancements.

TRAC ID	Description
9097	**Host MCU access to WINC flash** APIs to allow host application to access WINC flash areas for managing TLS root certificates and WINC firmware. **Enhanced:** Streamlined APIs for more user-friendly interface. Refer to WINC3400_IoT_SW_APIs.chm for details of APIs. Note this enhancement involves changes to API names that were introduced in the previous driver version.
9162	**Allow application to set the GAP role via the BLE API** In previous versions of the BLE API it was not possible for the application to explicitly specify the GAP role required for the BLE device. All GAP role changes were implicitly performed from within the BLE API itself, which was inflexible and sometimes lead to problems when switching between roles. **Changed:** The BLE API has now been modified to allow the application to specify the GAP role. To achieve this, the following BLE API function has been modified: **`at_ble_status_t at_ble_set_gap_deviceinfo(at_ble_gap_deviceinfo_t* gap_deviceinfo)`** The information contained in the **`gap_deviceinfo`** struct is now stored within the BLE API but not passed down to the WINC3400 at the point that the function is called. A new BLE API function has been added which will allow the application to both set the BLE role and write the information that is stored when calling **`at_ble_set_gap_deviceinfo()`** in one atomic action: **`at_ble_status_t at_ble_set_dev_config(at_ble_gap_role role)`** This allows the application to set the GAP device info once on initialization, but then switch the role as necessary. For example, to configure the device for BLE advertising, at some point after calling **`at_ble_set_gap_deviceinfo()`**, the new BLE API function should be called in the following manner: **`at_ble_set_dev_config(AT_BLE_GAP_PERIPHERAL_SLV)`** To configure the device for BLE scanning and connecting, the BLE API function show be called with the relevant parameter: **`at_ble_set_dev_config (AT_BLE_GAP_CENTRAL_MST)`**

TRAC ID	Description
9103	**802.11 legacy powersave enhancements** 802.11 legacy powersave was performing sub-optimally since the introduction of an improved hardware coexistence mechanism. **Enhancement:** Powersave has been improved to work more efficiently. Applications can set the powersave mode using the following API function: **`sint8 m2m_wifi_set_sleep_mode(uint8 PsTyp, uint8 BcastEn)`** The currently supported powersave modes on the WINC3400 are: **`M2M_NO_PS`** Disables all powersave (default) **`M2M_PS_DEEP_AUTOMATIC`** Activates legacy 802.11 powersave. When associated to a WiFi Access Point, the WINC3400 will enter low power mode between WiFi beacons, waking in accordance with the listen interval to check the TIM element in the beacon for buffered traffic from the AP. If **`BcastEn`** is set, the WINC will also wake every DTIM interval to check for buffered broadcast traffic.

7 Terms and Definitions

Term	Definition
AES	Advanced Encryption Standard
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
MQTT	Message Queuing Telemetry Transport
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)