
ATWINC15x0 in Ethernet Mode

Introduction

This application note describes the ATWINC15x0 host driver Station mode application in detail, and demonstrates basic Wi-Fi® connection between the device which acts as a Station (STA) and an Access Point (AP) in Ethernet mode.

Please Note:

WINC1500 Native mode provides an integrated and simplest solution - **Our recommended mode of operation**. Fully supported in ASF and Harmony.

- Main Features:
 - Wi-Fi and communication protocols (IP and TLS) are all fully integrated on WINC1500 (FW loaded from WINC flash)
 - Host MCU runs user application and WINC driver (limited resources needed)
 - No royalties / 3rd party code / support is required
 - Solution can scale to any MCU Core and size from 8-bit to 32-bit
- Main Use Cases:
 - Wi-Fi add on to any MCU
 - Cloud connected devices that support only Wi-Fi connectivity (no ETH)

WINC1500 Ethernet mode is used when Ethernet (ETH) is used in parallel to Wi-Fi (i.e., ETH on PIC32MZ) or a specific IP stack is required (e.g., IPv6, customer proprietary).

- Main Features:
 - Wi-Fi driver and stack runs on the WINC1500 (FW loaded from WINC flash)
 - Host runs TCP/IP and TLS stack (Bypass 1500 communication stacks)
 - Royalties / support contract might be needed for IP / TLS stack
 - Solutions focus mainly on 32-bit MCUs
- Main Use Cases:
 - MCUs with ETH and Wi-Fi interfaces (i.e., PIC32MZ)
 - IPv6 support is required
 - Specific TLS stack is required (i.e., TLS1.3)

Features

- ATWINC15x0 host MCU driver architecture
- ATWINC15x0 internal architecture
- Application description with code snippets
- Events handled in the Wi-Fi callback function with appropriate structure for each event

- Steps to execute the HTTP provision mode application demo using the SAM G55 Xplained Pro (XPro) board and the ATWINC15x0 XPro board

Table of Contents

Introduction.....	1
Features.....	1
1. Application Description.....	4
2. Host Driver Architecture.....	5
3. ATWINC15x0 System Architecture.....	6
4. Wi-Fi Host Driver Initialization.....	8
5. Handling Wi-Fi Host Driver Event and Callback.....	10
6. Running Application in Station Mode	11
7. Changes Needed for Running the Application in Ethernet Mode	14
8. TCP/ IP Networking Stack - lwIP.....	16
9. References.....	17
10. Document Revision History.....	18
The Microchip Web Site.....	19
Customer Change Notification Service.....	19
Customer Support.....	19
Product Identification System.....	20
Microchip Devices Code Protection Feature.....	20
Legal Notice.....	21
Trademarks.....	21
Quality Management System Certified by DNV.....	22
Worldwide Sales and Service.....	23

1. Application Description

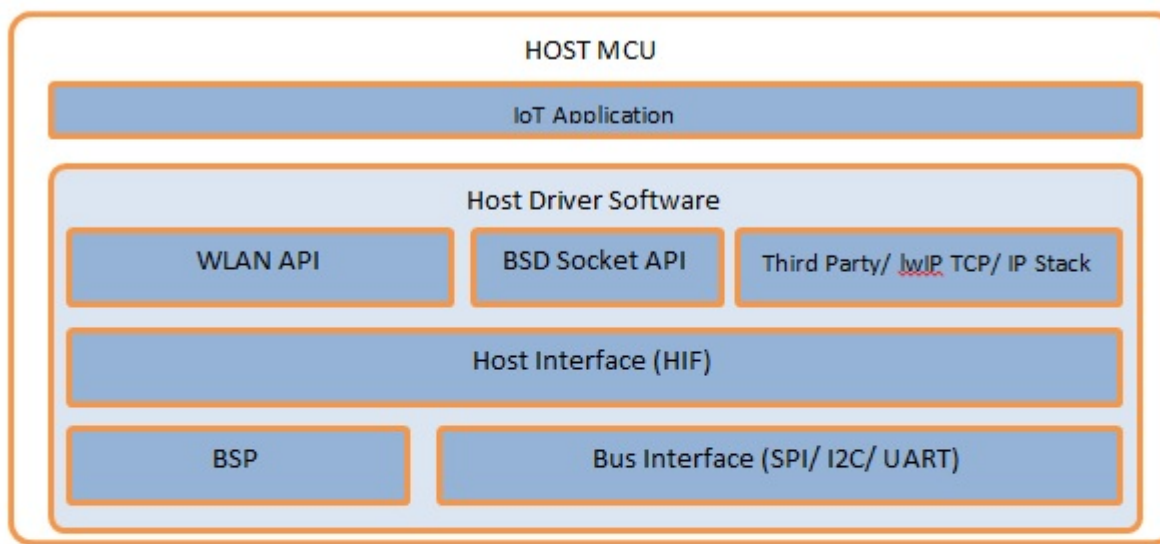
This Application demonstrates the execution of ATWINC15x0 as a Wi-Fi station mode using the SAM G55 XPro board as a host MCU. The device is running in Ethernet (also known as ByPass) Mode.

Note: The Ethernet Mode and Bypass Mode shall be used interchangeably in the document.

2. Host Driver Architecture

The ATWINC15x0 host driver software is a C library, which provides the host MCU application with the necessary APIs to perform WLAN and socket operations. It also has a third party TCP/IP stack if the ATWINC1500 is running in Ethernet mode and not using its own TCP/IP stack. The following figure shows the architecture of the ATWINC1500 host driver software, which runs on the host MCU. For more details on the components of the host driver, refer to [ATWINC1500 Wi-Fi Network Controller Software Design Guide \(DS00002389\)](#).

Figure 2-1. Host Architecture



3. ATWINC15x0 System Architecture

The following figures show the ATWINC15x0 system architecture. In addition to its built-in Wi-Fi IEEE-802.11 physical layer and RF front end, the ASIC contains an embedded APS3S-Cortus 32-bit CPU to run the ATWINC15x0 firmware. The firmware comprises the Wi-Fi IEEE-802.11 MAC layer and embedded protocol stacks. For more details on the components of the system, refer to [ATWINC1500 Wi-Fi Network Controller Software Design Guide \(DS00002389\)](#).

Figure 3-1. ATWINC15x0 Native Mode

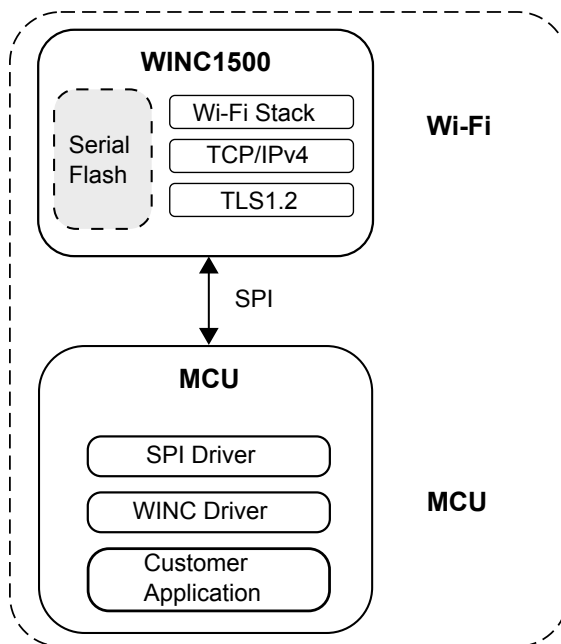
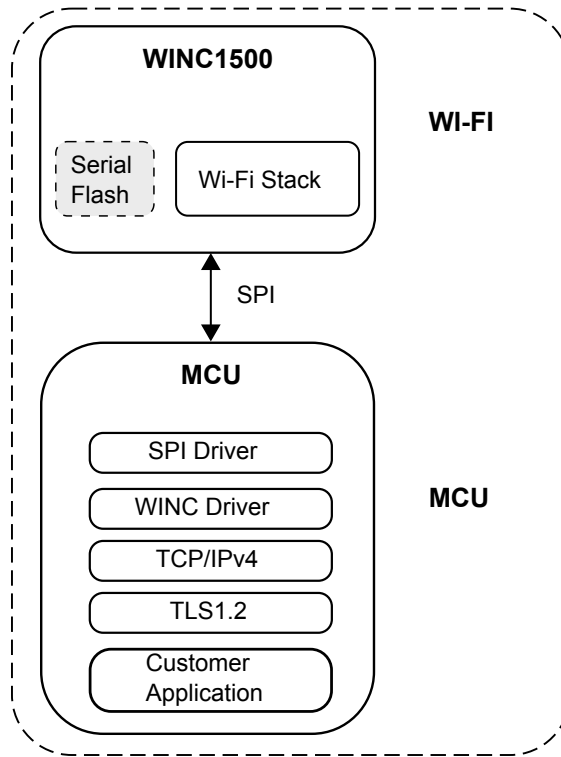


Figure 3-2. ATWINC15x0 Ethernet Mode



4. Wi-Fi Host Driver Initialization

This chapter describes a set of synchronous initialization sequences to execute for operation of the Wi-Fi functions.

1. Initialize the ATWINC15x0 XPro board, which includes clock, hardware, events, and external hardware (if any).

```
/* Initialize the board. */
sysclk_init();
board_init();
```

2. Configure the UART interface console to debug the output. The debug log level value can be set using the `M2M_LOG_LEVEL` macro in `nm_debug.h` file.

```
/* Initialize the UART console. */
configure_console();
puts(STRING_HEADER)
```

3. Initialize the Network stack using the following function.

```
/* Initialize the network stack. */
net_init();
```

4. To initialize the BSP (Board Support Package) driver, the ATWINC15x0 initialization sequence must be followed. Follow the Chip Enable and Reset sequence of the ATWINC15x0.

```
/* Initialize the BSP. */
nm_bsp_init();
```

5. Initialize the Wi-Fi host driver starting with the API `m2m_wifi_init()` and `tstrWifiInitParam` structure, completed with appropriate information. The Wi-Fi initialization sequence configures the SPI communication interface and external interrupt, with respect to the host MCU and the ATWINC15x0 pin connections. Apart from this, the Wi-Fi host application layer callback function `wifi_cb()` is registered with the initialization sequence.
6. After the successful initialization of the communication interface between the ATWINC15x0 and the host MCU, it is able to read the ATWINC15x0 chip-id indicating that the ATWINC15x0 is ready to accept the Wi-Fi initialization and configuration commands of the WLAN module. In this process, the ATWINC15x0 WLAN module is reset by the host MCU and waits for the module firmware to start confirmation.
7. The driver initialization needs to ensure that the user configures the Ethernet mode in the ATWINC15x0. Since the extra parameters are initialized for the `m2m_wifi_init()` function, the ATWINC15x0 bypasses its own Networking stack.

```
void winc_fill_callback_info(tstrEthInitParam *info)
{
    info->pfAppEthCb = winc_netif_rx_callback;
    info->au8ethRcvBuf = rx_buf;
    info->u16ethRcvBufSize = sizeof(rx_buf);
    info->u8EthernetEnable = 1; //For bypassing the TCPIP Stack of WINC
}
```

This code ensures that the the bit `rHAVE_ETHERNET_MODE_BIT` (0x80) is set in the register `rNMI_GP_REG_1` (0x14A0), when the driver is initialized via `m2m_wifi_init()`.

8. The `pfAppEthCb` and the above function are initialized to the Ethernet callback function which is a handle to the Networking stack for passing the Ethernet packet received from the ATWINC15x0.

```
static void winc_netif_rx_callback (uint8 u8MsgType, void * pvMsg, void * pvCtrlBuf)
{
    switch(u8MsgType)
    {
```



```
case M2M_WIFI_RESP_ETHERNET_RX_PACKET:
//.....insert code as per app Requirements break;

default:
break;
}
}
```

9. The purpose of the Wi-Fi application callback function is to indicate the events such as connect and disconnect status, and obtaining the IP address from the Dynamic Host Configuration Protocol (DHCP) server with respect to the DHCP request from the Networking stack running on the host.

```
/* Initialize Wi-Fi parameters structure. */
memset((uint8_t *)&param, 0, sizeof(tstrWifiInitParam));
/* Initialize Wi-Fi driver with data and status callbacks. */
param.pfAppWifiCb = wifi_cb;
winc_fill_callback_info(&param.strEthInitParam);
ret = m2m_wifi_init(&param);
if (M2M_SUCCESS != ret)
{
printf("main: m2m_wifi_init call error!(%d)\r\n", ret);
while (1) { }
}
```

10. For transmission, the following API can be used for sending the TCP/IP packets provided by the TCP/IP stack.

```
m2m_wifi_send_etherenet_pkt(tx_buf, num_bytes_transmitted);
```

5. Handling Wi-Fi Host Driver Event and Callback

All the Wi-Fi host driver events are handled in the `m2m_wifi_handle_events()` by running in the infinite loop. This function internally uses the HIF (Host Hardware Interface) layer API to monitor the ATWINC15x0 external interrupt. This external interrupt is registered using the following host MCU external interrupt configuration.

```
while (1) {  
    /* Handle pending events from network controller. */  
    while (m2m_wifi_handle_events(NULL) != M2M_SUCCESS)  
    {  
    }  
}
```

When the ATWINC15x0 external interrupt occurs, the host interface ISR layer reads the ATWINC15x0 control register to identify the type of event which triggered the external interrupt by the ATWINC15x0 and reads the data related to the event from the ATWINC15x0 buffer. After reading the complete data for the event, the corresponding callback function is triggered. The host MCU Wi-Fi driver handles two types of event categories, including `M2M_REQ_GRP_WIFI` and `M2M_REQ_GRP_OTA`.

```
m2m_wifi_cb() /*handles all the Wi-Fi configuration and connection events.*/
```

6. Running Application in Station Mode

This example demonstrates the execution of ATWINC15x0 as a Wi-Fi station mode using the SAM G55 XPro board as a host MCU.

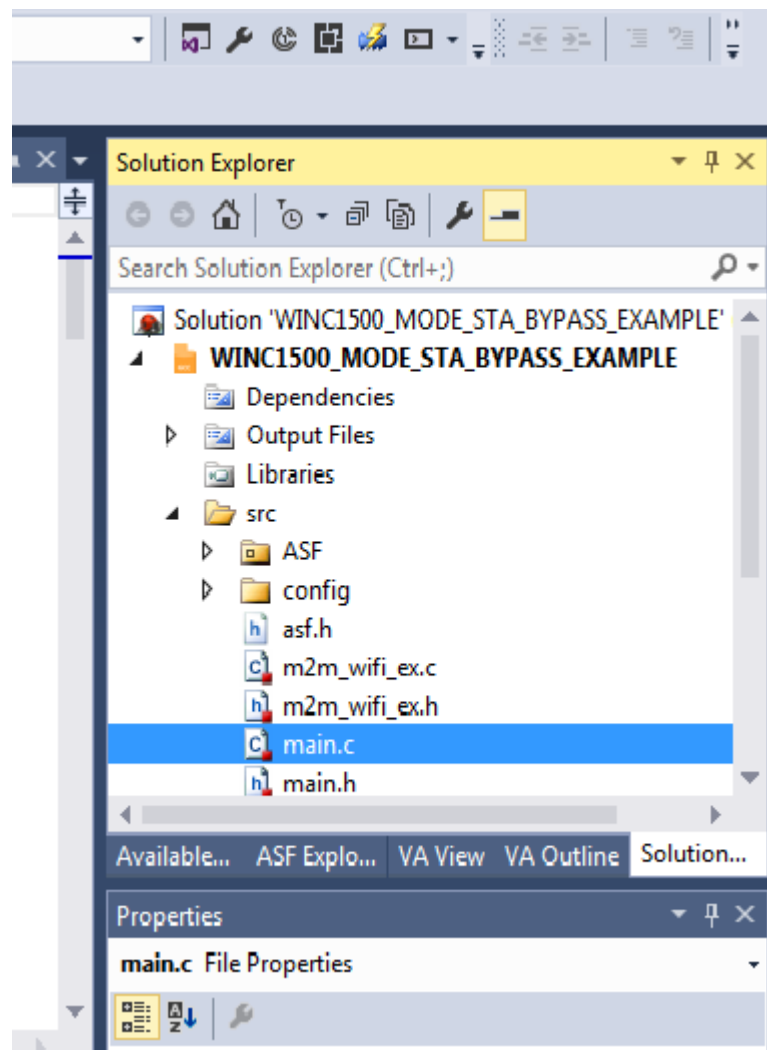
The demo uses the following hardware:

- SAM G55 Xplained Pro
- ATWINC15x0 on EXT1 header
- IEEE 802.11 b/g/n supported AP or Router

Perform the following steps to run the application in Station mode:

1. Open Atmel Studio 7 File > New > Example Projects
 - 1.1. In the search area, enter ATWINC1500 sample application for different MCU.
 - 1.2. Select the Station mode WINC1500_MODE_STA_BYPASS_EXAMPLE project for SAM G55 and open the project.
 - 1.3. Station Mode ByPass Application Directory Structure is shown in the following image.

Figure 6-1. Station Mode ByPass Directory Structure



2. Station mode demo requires Wi-Fi credentials such as SSID and Security type in order to setup the connection with the Access Point (AP). This demo is explained based on the authentication type set

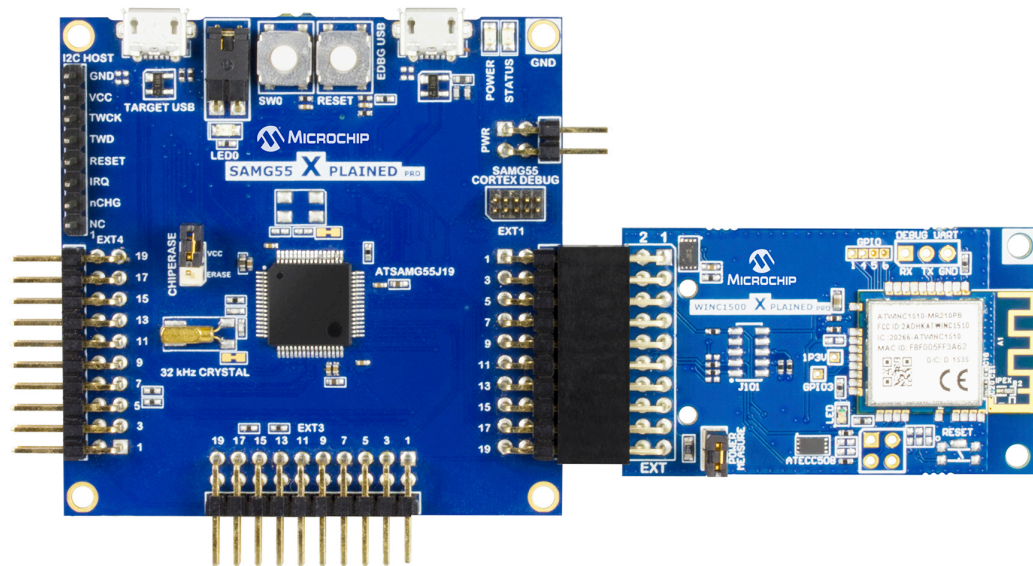
by the user on security method (for example: OPEN). To set an authentication type, the `STA_WLAN_AUTH` macro value needs to be configured as mentioned in the code snippet below. The corresponding AP's SSID must be filled for `STA_WLAN_SSID` macro. These macros are defined in the `sta.h` file.

- 2.1. Configure the AP credentials, SSID and password as mentioned above. Password is not required if OPEN authentication is used.
- 2.2. This application demo is based on the `M2M_WIFI_SEC_OPEN` method. Be sure to configure the security settings based on the AP settings used.

```
/* AP configuration parameters*/
#define STA_WLAN_SSID "MicrochipDemoApp" /* Access Point SSID */
#define STA_WLAN_AUTH M2M_WIFI_SEC_OPEN /* Security manner */
```

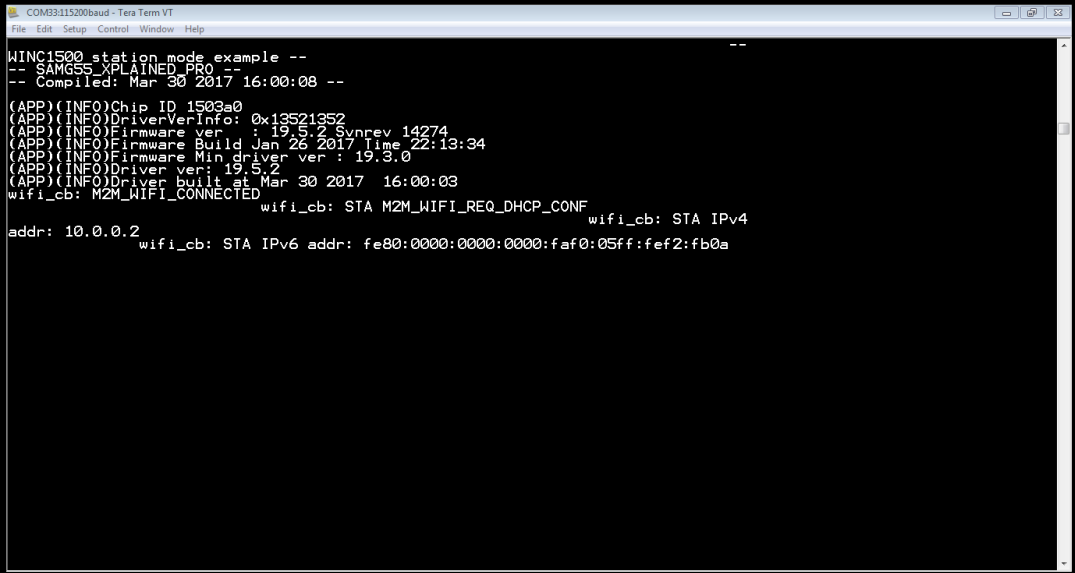
- 2.3. Open the serial port terminal application with the following COM port configuration:
 - Baudrate – 115200
 - Data – 8 bit
 - Parity – None
 - Stop – 1 bit
 - Flow control – None

Figure 6-2. SAM G55 Xplained Pro Board Setup



- 2.4. Compile and download the image into the SAM G55 XPro board.
- 2.5. Run the application to display success or error messages in the serial port terminal.

Figure 6-3. Application Output on Serial Terminal



```
COM3:115200baud - Tera Term VT
File Edit Setup Control Window Help
--
WINC1500 station mode example --
-- SAMG55XPLAINED PRO --
-- Compiled: Mar 30 2017 16:00:08 --
(APP)(INFO)Chip ID 1503a0
(APP)(INFO)DriverVerInfo: 0x13521352
(APP)(INFO)Firmware ver : 19.5.2 Sunrev 14274
(APP)(INFO)Firmware Build Jan 26 2017 Time 22:13:34
(APP)(INFO)Firmware Min_driver ver : 19.3.0
(APP)(INFO)Driver ver: 19.5.2
(APP)(INFO)Driver built at Mar 30 2017 16:00:03
wifi_cb: M2M_WIFI_CONNECTED
wifi_cb: STA M2M_WIFI_REQ_DHCP_CONF
wifi_cb: STA IPv4
addr: 10.0.0.2
wifi_cb: STA IPv6 addr: fe80:0000:0000:0000:faf0:05ff:fef2:fb0a
```

7. Changes Needed for Running the Application in Ethernet Mode

The user can configure the ATWINC15x0 behaviour from the host application. The following information (basic snippet of the code) is provided for better understanding of using or configuring the ATWINC15x0 module in Ethernet mode:

1. Compile the host driver with `ETH_MODE`. The user can define the compile time switch in `nm_common.h`.
2. Perform the Network stack initialization before calling the `m2m_wifi_init()`. Set the additional parameters for `m2m_wifi_init()`:

```
tstrWifiInitParam param;
param.pfAppWifiCb = wifi_cb;
param.strEthInitParam.au8ethRcvBuf = eth_buffer;
param.strEthInitParam.ul6ethRcvBufSize = sizeof(eth_buffer);
param.strEthInitParam.u8EthernetEnable = M2M_WIFI_MODE_ETHERNET;
param.strEthInitParam.pfAppEthCb = eth_cb;
ret = m2m_wifi_init(&param);
```

The above code ensures that the the bit `rHAVE_ETHERNET_MODE_BIT` (0x80) is set in the register `rNMI_GP_REG_1` (0x14A0). Also, the `eth_buffer` is the buffer given by the host from which the packet is received ATWINC15x0.

3. The following Ethernet callback function can be implemented as per the user requirements:

```
static void eth_cb(uint8 u8MsgType, void * pvMsg, void * pvCtrlBuf) {
    switch(u8MsgType) {
        case M2M_WIFI_RESP_ETHERNET_RX_PACKET:
            //.....insert code as per app requirements
            break;

        default:
            break;
    }
}
```

Table 7-1. Ethernet Callback

Parameter	Type	Description
<code>u8MsgType</code>	<code>UInt8</code>	Message Type - <code>M2M_WIFI_RESP_ETHERNET_RX_PACKET</code> . This is an event occurring when the Ethernet packet is received from ATWINC15x0.
<code>pvMsg</code>	<code>Void *</code>	Pointer to the Rx buffer. In case of an unfragmented packet, the packet received is in the 802.3 Ethernet format.
<code>pvCtrlBuf</code>	<code>Void *</code>	If the packet received is bigger than the Rx buffer, the packet can be fragmented and sent in multiple such messages to the host. The structure <code>tstrM2mIpCtrlBuf</code> is used for sending this information. <code>ul6DataSize</code> has the number of bytes in the current buffer, and <code>ul6RemainigDataSize</code> has the remaining bytes of the packet received which is sent to the host in subsequent calls.

4. For transmission, the user can use the following API to send the TCP/IP packets to the ATWINC15x0.

```
m2m_wifi_send_ethernet_pkt(tx_buf, num_bytes_transmitted)
```

Table 7-2. Ethernet Packet

Parameter	Type	Description
tx_buf	Void *	Valid pointer to the buffer having the data packet in the 802.3 Ethernet format
num_bytes_transmitted	Uint16	Size of the buffer

5. For reception, when the device receives any Ethernet packets it notifies the application through the callback `eth_cb()` which is called in case of `M2M_WIFI_RESP_ETHERNET_RX_PACKET` in `wifi_cb()`.

8. TCP/ IP Networking Stack - lwIP

The TCP/ IP stack used in this example is a third party Networking stack lwIP. The lwIP is a small independent implementation of the TCP/IP protocol suite, initially developed by Adam Dunkels. The focus of the lwIP TCP/IP implementation is to reduce resource usage while still having a full scale TCP.

The main features include:

- Protocols – IP, ICMP, UDP, TCP, IGMP, ARP, PPPoS, and PPPoE
- Clients – DHCP client, DNS client, AutoIP/APIPA (Zeroconf), and SNMP agent (private MIB support)
- APIs – specialized APIs to enhance performance and optional Berkeley-alike socket API
- Extended features – IP forwarding over multiple network interfaces, TCP congestion control, RTT estimation, and fast recovery/fast re-transmit
- Add on applications – HTTP server, SNTP client, SMTP client, ping, and NetBIOS nameserver

9. References

The following documents can be used for further study:

- [ATWINC1500 Wi-Fi Network Controller Software Design Guide](#)
- [ATWINC1500 Wi-Fi Network Controller - Station Mode](#)

10. Document Revision History

Rev A - 08/2017

Section	Changes
Document	Initial Release

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

PART NO. **[X]⁽¹⁾** **-** **X** **/XX** **XXX**

Device Tape and Reel Option Temperature Range Package Pattern

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2017, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-2172-6

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com	Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong Tel: 852-2943-5100 Fax: 852-2401-3431 Australia - Sydney Tel: 61-2-9868-6733 Fax: 61-2-9868-6755 China - Beijing Tel: 86-10-8569-7000 Fax: 86-10-8528-2104 China - Chengdu Tel: 86-28-8665-5511 Fax: 86-28-8665-7889 China - Chongqing Tel: 86-23-8980-9588 Fax: 86-23-8980-9500 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 Fax: 86-571-8792-8116 China - Hong Kong SAR Tel: 852-2943-5100 Fax: 852-2401-3431 China - Nanjing Tel: 86-25-8473-2460 Fax: 86-25-8473-2470 China - Qingdao Tel: 86-532-8502-7355 Fax: 86-532-8502-7205 China - Shanghai Tel: 86-21-3326-8000 Fax: 86-21-3326-8021 China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393 China - Shenzhen Tel: 86-755-8864-2200 Fax: 86-755-8203-1760 China - Wuhan Tel: 86-27-5980-5300 Fax: 86-27-5980-5118 China - Xian Tel: 86-29-8833-7252 Fax: 86-29-8833-7256	China - Xiamen Tel: 86-592-2388138 Fax: 86-592-2388130 China - Zhuhai Tel: 86-756-3210040 Fax: 86-756-3210049 India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123 India - New Delhi Tel: 91-11-4160-8631 Fax: 91-11-4160-8632 India - Pune Tel: 91-20-3019-1500 Japan - Osaka Tel: 81-6-6152-7160 Fax: 81-6-6152-9310 Japan - Tokyo Tel: 81-3-6880-3770 Fax: 81-3-6880-3771 Korea - Daegu Tel: 82-53-744-4301 Fax: 82-53-744-4302 Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934 Malaysia - Kuala Lumpur Tel: 60-3-6201-9857 Fax: 60-3-6201-9859 Malaysia - Penang Tel: 60-4-227-8870 Fax: 60-4-227-4068 Philippines - Manila Tel: 63-2-634-9065 Fax: 63-2-634-9069 Singapore Tel: 65-6334-8870 Fax: 65-6334-8850 Taiwan - Hsin Chu Tel: 886-3-5778-366 Fax: 886-3-5770-955 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Fax: 886-2-2508-0102 Thailand - Bangkok Tel: 66-2-694-1351 Fax: 66-2-694-1350	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 France - Saint Cloud Tel: 33-1-30-60-70-00 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-7289-7561 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820