MICROCHIP

MCW1001A
Data Sheet

TCP/IP Socket Communications

Interface with GPIO

Note the following details of the code protection feature on Microchip devices:

• Microchip products meet the specification contained in their particular Microchip Data Sheet.

• Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.

• There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

• Microchip is willing to work with the customer who is concerned about the integrity of their code.

• 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.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock 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.

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

© 2011, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

ISBN: 978-1-61341-681-5

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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
=== ISO/TS 16949:2009 ===

MCW1001A

TCP/IP Socket Communications Interface with GPIO

Key Features:

- Supports BSD like socket based connections
- Built-in Wi-Fi® connection management messages
  - Simplifies finding, connecting and maintaining a Wi-Fi connection
- Easy-to-use messaging interface
- No external memory required
- Connects through Universal Asynchronous Receiver/Transmitter (UART)
- Operates seamlessly with the MRF24WB0MA/MB 802.11 modules
- Easy integration into a final product
  - Accelerates product development, provides quicker time to market
- Designed for use with any microprocessor hosting a UART port
- 8 general-purpose digital I/O
- Small size: 28 SSOP

Operational:

- Single operating voltage: 2.7V–3.6V(3.3V typical)
- Temperature range: -40°C to +85°C industrial
- UART interface, up to 230 Kbaud
- Low-current consumption:
  - 10 mA operational current

Applications:

- Using MRF24WB0M module with standard UART interface
- Using custom stacks with Microchip microcontrollers

Markets:

- Utility and Smart Energy
  - Thermostats
  - Smart Meters
  - White Goods
  - HVAC
- Consumer Electronics
  - Home security
- Industrial Controls
  - Chemical sensors
  - HVAC
  - Security systems
  - M2M communication
- Remote Device Management
  - Automotive
  - Code update
- Retail
  - POS Terminals
- Medical, Fitness, and Health care
  - Glucose meters
  - Fitness equipment
  - Patient asset tracking

MCW1001A

Pin Diagram

	MCW1001A	
	SSOP	

Pin	Left		Right	Pin
1	$\overline{\text{RESET}}$		NC	28
2	GPIO0		GPIO7	27
3	GPIO1		GPIO6	26
4	GPIO2		GPIO5	25
5	GPIO3		$\overline{\text{URTS}}$	24
6	VCAP		$\overline{\text{UCTS}}$	23
7	GPIO4		$\overline{\text{WWP}}$	22
8	V_{SS}		$\overline{\text{WINT}}$	21
9	OSC1		V_{DD}	20
10	OSC2		V_{SS}	19
11	$\overline{\text{WCS}}$		URX	18
12	$\overline{\text{WRST}}$		UTX	17
13	WHIB		WSD0	16
14	WSCK		WSDI	15

Preliminary

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

   http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

• Microchip's Worldwide Web site; **http://www.microchip.com**
• Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

MCW1001A

NOTES:

Preliminary

1.0 DEVICE OVERVIEW

The MCW1001A is a companion chip to the MRF24WB0 802.11 module. It provides simple socket based method of sending and receiving data from the MRF24WB0 802.11 module. The MCW1001A has an on-board TCP/IP stack and 802.11 connection manager to simplify the connection between a wireless network and the TCP/IP stack management. After the initial configuration is set, the MCW1001A can access the MRF24WB0 802.11 module to connect to a network and send/receive serial data over a simple UART interface.

Figure 1-1 illustrates a general block diagram of the MCW1001A device.
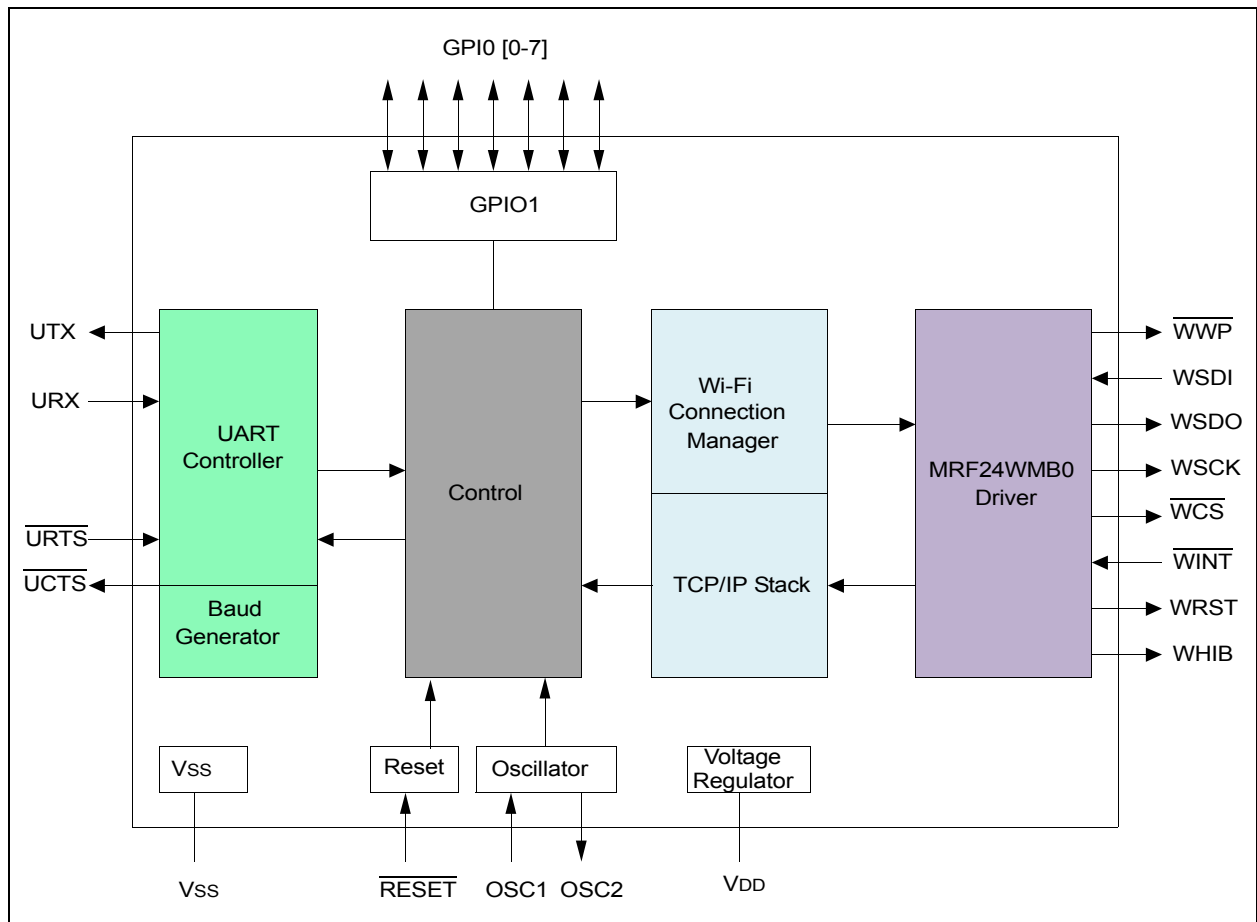
Figure 1-2 illustrates application schematic.

Figure 1-3 illustrates application block diagram.

Table 1-1 lists the functions of the various pins shown in the pin diagram.

Table 1-2 lists the UART API summary

FIGURE 1-1: BLOCK DIAGRAM

MCW1001A

TABLE 1-1: **PINOUT DESCRIPTION**

Pin Number	Pin Name	Type	Description
1	RESET	I	Active-low Reset to the device.
2	GPIO0	I/O	General-purpose I/O
3	GPIO1	I/O	General-purpose I/O
4	GPIO2	I/O	General-purpose I/O
5	GPIO3	I/O	General-purpose I/O
6	VCAP	I	External filter capacitor connection for internal regulator
7	GPIO4	I/O	General-purpose I/O
8	Vss	P	Ground
9	OSC1	I	Oscillator input
10	OSC2	O	Oscillator output
11	WCS	O	SPI Slave select output to MRF24WB0M
12	WRST	O	External Reset output control to MRF24WB0M
13	WHIB	O	HIBERNATE mode enable output to MRF24WB0M
14	WSCK	O	SPI clock output to MRF24WB0M
15	WSDI	I	SPI data input from MRF24WB0M
16	WSDO	O	SPI data output to MRF24WB0M
17	UTX	O	USART asynchronous transmit
18	URX	I	USART asynchronous receive
19	Vss	P	Ground
20	VDD	P	Power
21	WINT	I	External interrupt from MRF24WMB0
22	WWP	O	External write protect control to MRF24WB0M
23	UCTS	I	USART asynchronous clear to send (optional)
24	URTS	O	USART asynchronous request to send (optional)
25	GPIO5	I/O	General-purpose I/O
26	GPIO6	I/O	General-purpose I/O
27	GPIO7	I/O	General-purpose I/O
28	NC	NC	No Connect

Legend: Pin type abbreviation: P = Power input   I = Input
O = Output   NC = No Connect

Note: Signals of Type "I: Constant" must either be constantly driven by the host or have a pull-up or pull-down (in case the host is likely to tri-state the signal during power down modes). The constant drive is used to ensure defined operation of the part and to minimize leakage current during low-power operation.
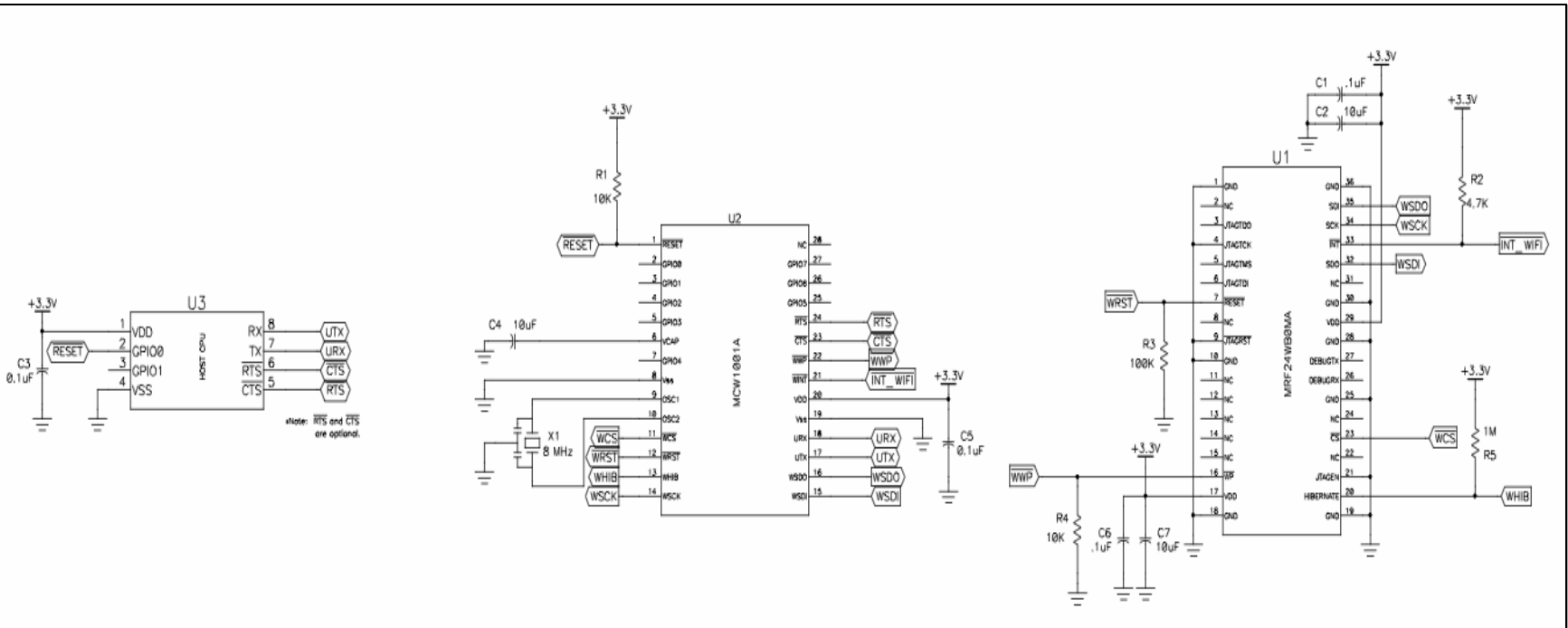
FIGURE 1-2: EXAMPLE APPLICATION SCHEMATIC
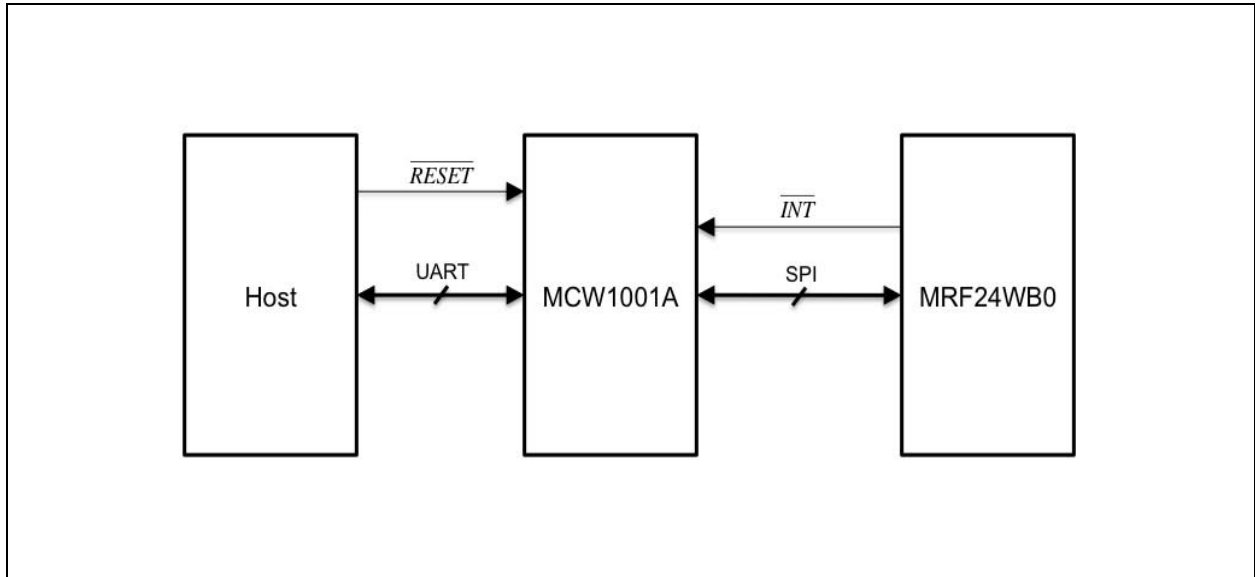
FIGURE 1-3: **APPLICATION BLOCK DIAGRAM**

TABLE 1-2:   **UART API SUMMARY**

Message ID (Dec)	Message ID (Hex)	Message Type	Destination	API Message Name	Description
Control Messages					
170	0xAA	Command	MCW1001A	RESET_MSG	Power-on reset
23	0x17	Command	MCW1001A	GET_VERSION_MSG	Retrieve Version Info
172	0xAC	Command	MCW1001A	GPIO_MSG	GPIO Set/Read
50	0x32	Response	Host CPU	GPIO_RESPONSE_MSG	GPIO Set or Read Result
Network Configuration Messages					
41	0x29	Command	MCW1001A	SET_IP_ADDRESS_MSG	Set IP address and DHCP state
42	0x2A	Command	MCW1001A	SET_NETWORK_MASK_MSG	Set Network Mask
44	0x2C	Command	MCW1001A	SET_GATEWAY_IP_ADDRESS _MSG	Set Gateway Address
48	0x30	Command	MCW1001A	GET_NETWORK_STATUS	Retrieve Network Status Info
48	0x30	Response	Host CPU	NETWORK_STATUS_RESPON SE_MSG	Network Status Information
49	0x31	Command	MCW1001A	SET_MACADDRESS_MSG	Set MAC Address
173	0xAD	Command	MCW1001A	SET_ARP_TIME_MSG	Set ARP Repeat Time
Wi-Fi General Configuration Messages					
55	0x37	Command	MCW1001A	SET_CP_NETWORK_MODE_ MSG	Set mode for Connection Profile
57	0x39	Command	MCW1001A	SET_CP_SSID_MSG	Set SSID for Connection Profile
56	0x38	Command	MCW1001A	SET_REGIONAL_DOMAIN_MSG	Set Geographical Region for Channel List
58	0x3A	Command	MCW1001A	SET_CHANNEL_LIST_MSG	Set 802.11 channel list
59	0x3B	Command	MCW1001A	SET_LIST_RETRY_COUNT_M SG	Set connection retry count
Wi-Fi Power Management Messages					
102	0x66	Command	MCW1001A	SET_POWER_MODE_MSG	Set mode for MRF24 Power Consumption
Wi-Fi Security Configuration Messages					
65	0x41	Command	MCW1001A	SET_CP_SECURITY_OPEN_ MSG	Set Open Security for Connection Profile
66	0x42	Command	MCW1001A	SET_CP_SECURITY_WEP40_ MSG	Set WEP40 Security for Connection Profile
67	0x43	Command	MCW1001A	SET_CP_SECURITY_WEP104 _MSG	Set WEP104 Security for Connection Profile
68	0x44	Command	MCW1001A	SET_CP_SECURITY_WPA_ MSG	Set WPA Security for Connection Profile
71	0x47	Command	MCW1001A	GET_CP_WPAKEY_MSG	Retrieve Calculated WPA Key
49	0x31	Response	Host CPU	WPAKEY_RESPONSE_MSG	Calculated WPA Key Response
Wi-Fi Scanning Messages					
80	0x50	Command	MCW1001A	SCAN_START_MSG	Initiate Scan for 802.11 networks
81	0x51	Command	MCW1001A	SCAN_GET_RESULTS_MSG	Retrieve Scan results
22	0x16	Response	Host CPU	SCAN_RESULT_MSG	Scan Result Information

MCW1001A

TABLE 1-2:   UART API SUMMARY

Message ID (Dec)	Message ID (Hex)	Message Type	Destination	API Message Name	Description
Wi-Fi Connection Messages					
90	0x5A	Command	MCW1001A	WIFI_CONNECT_MSG	Connect to specified Connection Profile network
91	0x5B	Command	MCW1001A	WIFI_DISCONNECT_MSG	Disconnect from 802.11 network
ICMP (Ping) Messages					
121	0x79	Command	MCW1001A	PING_SEND_MSG	Send Ping to ICMP Host
Socket Messages					
110	0x6E	Command	MCW1001A	SOCKET_CREATE_MSG	Create BSD Socket for TCP or UDP
23	0x17	Response	Host CPU	SOCKET_CREATE_RESPONSE _MSG	Socket Creation result
111	0x6F	Command	MCW1001A	SOCKET_CLOSE_MSG	Close BSD Socket
112	0x70	Command	MCW1001A	SOCKET_BIND_MSG	Bind Socket Handle to specified port
24	0x18	Response	Host CPU	SOCKET_BIND_RESPONSE_ MSG	Bind result
113	0x71	Command	MCW1001A	SOCKET_CONNECT_MSG	Connect to remote BSD socket
25	0x19	Response	Host CPU	SOCKET_CONNECT_RESPON SE_MSG	Connection result
114	0x72	Command	MCW1001A	SOCKET_LISTEN_MSG	Create child sockets to listen for clients
26	0x1A	Response	Host CPU	SOCKET_LISTEN_RESPONSE _MSG	Listen result
115	0x73	Command	MCW1001A	SOCKET_ACCEPT_MSG	Accept client request for socket connection
27	0x1B	Response	Host CPU	SOCKET_ACCEPT_RESPONS E_MSG	Accept result
116	0x74	Command	MCW1001A	SOCKET_SEND_MSG	Send Data using specified socket
28	0x1C	Response	Host CPU	SOCKET_SEND_RESPONSE_ MSG	Send result
117	0x75	Command	MCW1001A	SOCKET_RECV_MSG	Receive Data from specified socket
29	0x1D	Response	Host CPU	SOCKET_RECV_RESPONSE_ MSG	Receive result
118	0x76	Command	MCW1001A	SOCKET_SEND_TO_MSG	Send Data using specified socket and destination
30	0x1E	Response	Host CPU	SOCKET_SEND_TO_RESPON SE_MSG	Send To result
119	0x77	Command	MCW1001A	SOCKET_RECV_FROM_MSG	Receive Data from specified socket and origin
31	0x1F	Response	Host CPU	SOCKET_RECV_FROM_RESP ONSE_MSG	Receive From result
122	0x7A	Command	MCW1001A	SOCKET_ALLOCATE_MSG	Allocate Socket Buffers
32	0x20	Response	Host CPU	SOCKET_ALLOCATE_RESPO NSE_MSG	Socket Allocation result

Preliminary

TABLE 1-2: UART API SUMMARY

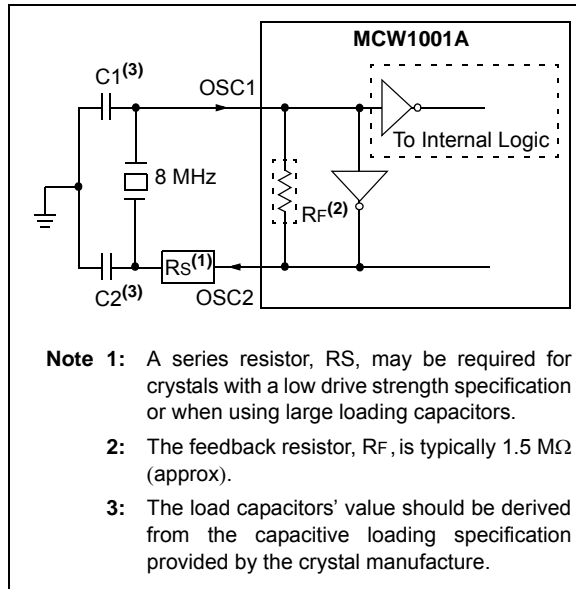| Message ID (Dec)	Message ID (Hex)	Message Type	Destination	API Message Name	Description
Asynchronous Event Messages Sent From MCW1001A to Host					
1	0x1	Event 16	Host CPU	EVENT_MSG – IP Address Assigned	New IP address assigned indication
1	0x1	Event 8	Host CPU	EVENT_MSG – Wi-Fi Connection Status Changed	Connection Status changed indication
1	0x1	Event 9	Host CPU	EVENT_MSG – Wi-Fi Scan Results Ready	Scan complete indication
1	0x1	Event 26	Host CPU	EVENT_MSG – Ping Response Event	Ping response from ICMP server result indication
1	0x1	Event 255	Host CPU	EVENT_MSG – Error Event	Error indication containing error information
1	0x1	Event 27	Host CPU	EVENT_MSG – Startup Event	Version Information and Reset Reason

NOTES:

Preliminary

2.0 EXTERNAL CONNECTIONS

2.1 Oscillator

MCW1001A devices are designed to operate from a fixed 8 MHz crystal clock input, as shown in Figure 2-1.

FIGURE 2-1: CRYSTAL OSCILLATOR OPERATION

Note 1: A series resistor, RS, may be required for crystals with a low drive strength specification or when using large loading capacitors.

2: The feedback resistor, RF, is typically 1.5 MΩ (approx).

3: The load capacitors' value should be derived from the capacitive loading specification provided by the crystal manufacture.

2.2 VDD AND VSS PINS

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on all power supply pins, such as VDD, VSS, are required.

Consider the following criteria when using decoupling capacitors:

• **Value and type of capacitor**: A 0.1 µF (100 nF) and 10–20V capacitor is recommended. The capacitor should be a low-ESR device with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are also recommended.

• **Placement on the printed circuit board**: The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer of the PCB using a via; however, ensure that the trace length from the pin to the capacitor is not greater than 0.25 inch (6 mm).

• **Handling high frequency noise**: If the board is experiencing high frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 µF in parallel with 0.001 µF).

• **Maximizing performance**: On the board layout from the power supply circuit, run the power and return traces to the device pins. This ensures that the decoupling capacitors are first in the power chain. It is equally important to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On PCB boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

2.2.3 VOLTAGE REGULATOR PINS (VCAP)

A low-ESR (<5Ω) capacitor is required on the VCAP pin to stabilize the voltage regulator output voltage. The VCAP pin must not be connected to VDD and must use a capacitor of 10 µF connected to ground. The type can be ceramic or tantalum. For example, Murata GRM21BF50J106ZE01 (10 µF, 6.3V) or equivalent.

2.3 RESET/PoR

2.3.1 RESET PIN

The RESET pin provides a method for triggering an external Reset of the device. Holding the pin low generates a Reset. These devices have a noise filter in the Reset path that detects and ignores small pulses.

2.3.2 PoR

A power-on Reset (PoR) pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the PoR circuitry, tie the RESET pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a PoR delay.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure the operation. If these conditions are not achieved, the device must be held in Reset until the operating conditions are met.

MCW1001A

2.4 MCW1001A to MRF24WB0 802.11 Module Interface

2.4.1 WSPI PINS

The MCW1001A Serial Peripheral Interface (WSPI) is used to communicate with the MRF24WB0 transceiver module.

2.4.2 $\overline{\text{WINT}}$ PIN

The WSPI interface works with the Interrupt line ($\overline{\text{WINT}}$) and must be pulled "High" through 4.7 kΩ resistor to VDD. When data is available for the MCW1001A during operation, the $\overline{\text{WINT}}$ line will be asserted (logic low) by the MRF24WB0 module. After the data is transferred to the MCW1001A, the $\overline{\text{WINT}}$ line will then be de-asserted (logic high) by the MRF24WB0 module.

2.4.3 $\overline{\text{WCS}}$ PIN

The $\overline{\text{WCS}}$ pin must be connected to the $\overline{\text{CS}}$ pin of the MRF24WB0 and cannot be held low permanently. The falling edge of $\overline{\text{WCS}}$ is used to indicate the start of a transfer. The rising edge of $\overline{\text{WCS}}$ is used to indicate the completion of a single transfer.

2.4.4 WHIB PIN

The WHIB pin is used to control the HIBERNATE feature of the MRF24WB0 and must be pulled "High" via 1 MΩ resistor to VDD. When WHIB is driven high, it puts the MRF24WB0M into Hibernate mode. The module contains about 70 uF of internal bulk capacitance so the system power supply should be provisioned to supply sufficient charge on release of hibernate to ensure proper startup. For more information on controlling the low-power modes of the MCW1001A, refer to Section 6.0, Messaging API.

2.4.5 $\overline{\text{WRST}}$ PIN

The $\overline{\text{WRST}}$ pin is used to control $\overline{\text{RESET}}$ to the MRF24WB0.

Preliminary

2.5 GPIO PINS

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

2.5.1 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of the GPIO pins varies across the bank of pins. Some of the pins when used as digital only inputs can handle DC voltages up to 5.5V, a level typical for digital logic circuits. Table summarizes the input voltage capabilities of the I/O pins. For more information on electrical characteristic, refer to Section 7.0, Electrical Characteristics. Voltage excursions beyond VDD on these pins should be avoided.

TABLE 2-1: INPUT VOLTAGE LEVELS

Pin	Tolerated Input	Description
GPIO<0:4>	VDD	Only VDD input levels are tolerated.
GPIO<5:7>	5.5V	Tolerates input levels above VDD and useful for most standard logic.

2.5.2 PIN OUTPUT DRIVE

The GPIO output pin drive strength varies across the GPIO bank of pins to meet the needs of a variety of applications. There are two classes of output pins in terms of drive capability.

- GPIO<5:7> – Designed to drive higher current loads, such as LED's.
- GPIO<0:4> – Designed for small loads, typically indication only. Table 2-2 summarizes the output capabilities. For more information, refer to the Section 7.0, Electrical Characteristics.

TABLE 2-2: OUTPUT DRIVE LEVELS

Pin	Drive	Description
GPIO<0:4>	Medium	Intended for indication.
GPIO<5:7>	High	Suitable for direct LED drive levels.

:

2.5.3 INTERFACING TO A +5V SYSTEM

Though the VDDMAX of the MCW1001A is 3.6V, this device is still capable of interfacing with 5V systems, even if the VIH of the target system is above 3.6V. This is accomplished by adding a pull-up resistor to the GPIO pin, see Figure 2-2. To produce a high output, the GPIO pin must be configured as an input, and to produce a low output, the GPIO pin must be configured as an output and set low. Only GPIO pins that are tolerant of voltages up to 5.5V can be used for this type of interface.

FIGURE 2-2: +5V SYSTEM HARDWARE INTERFACE

NOTES:

Preliminary   © 2011 Microchip Technology Inc.
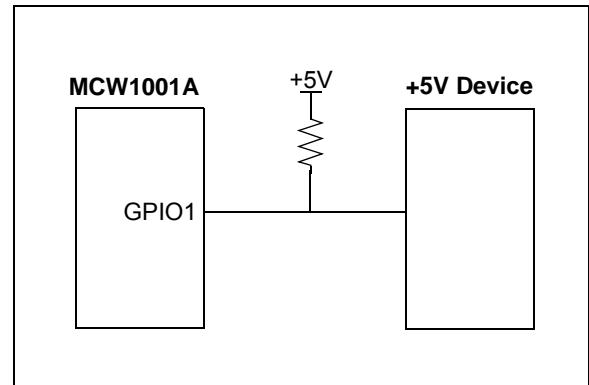
3.0 UNIVERSAL ASYNCHRONOUS RECEIVER (UART)

The MCW1001A UART interface consists of the UTX and URX data signals and the URTS/UCTS flow control pins. The UART is configurable for several baud rates. The available data formats and baud rates are listed in Table 3-1.

3.1 Initial Configuration

From Reset, the UART is configured for 115200 baud, 8 bits, no parity, and 2 stop bits. The receiver does not require 2 stop bits, but the MCW1001A will always transmit serial data with 2 stop bits. It is recommended that the host use 2 stop bits to improve synchronization with long packets.

3.2 Autobaud Rate

The autobaud feature of the MCW1001A requires that the initial byte sent from the host be 0x55. From Reset, autobaud is enabled, and the first byte sent will set the rate. This typically requires no special consideration as the first character of every well-formed packet contains 0x55. Because the receipt of this character cannot be guaranteed, there are two means of resetting back to the initialization point. The first is an active Reset (pull URTS low and return high), and the second is an auto-retry. If a valid fully-formed message is not received (indicating a synchronized baud rate) within 1s after the first byte that set the baud rate, WiComm-Socket will reinitialize the autobaud procedure and wait for the next 0x55 input to set the baud rate.

3.3 Rounding Errors

The primary baud rate setting (with the rounding errors) is shown in Table 3-1. If baud rates other than the ones shown in the table are used, the error percentage can be calculated using Equation 3-1 to find the actual baud rate.

3.4 Hardware Flow Control

Hardware flow control uses the URTS and UCTS pins as a handshake between two devices. The URTS pin of one device is typically connected to the UCTS of the other device.

URTS is an active-low output that notifies the other device when it is ready to receive data by driving the pin low. The MCW1001A trip point for de-asserting URTS (high) is 499 characters. This is one character short of "buffer full".

UCTS is an active-low input that notifies the MCW1001A when it is ready to send data. The MCW1001A will check UCTS just before loading and sending UART data. If the pin is asserted during a transfer, the transfer will continue as shown in Figure 3-1.

3.5 Flow Control Disabled

Hardware handshaking may be circumvented by physically tying the UCTS input low (active) instead of allowing it to be driven by the URTS signal from the host. In this case, the URTS output should be left unconnected, and allowed to freely change state.

FIGURE 3-1: URTS/UCTS CONNECTIONS EXAMPLE
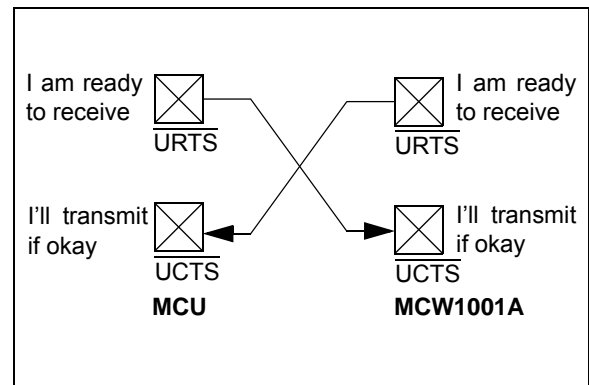
EQUATION 3-1: SOLVING FOR ACTUAL BAUD RATE

$$ActualRate = \frac{12MHz}{int(x)}$$

Where:

$$x = \frac{12MHz}{DesiredBaud}$$

MCW1001A

TABLE 3-1: UART PRIMARY BAUD RATES

Desired Rate	Actual Rate	AutoBaud Calculated Divisor(x)	% Error
300	300	40000	0.00%
1200	1200	10000	0.00%
2400	2400	5000	0.00%
4800	4800	2500	0.00%
9600	9600	1250	0.00%
19200	19200	625	0.00%
38400	38339	313	-0.16%
57600	57692	208	0.16%
115200	115385	104	0.16%
230400	230769	52	0.16%

Preliminary

4.0 NETWORK CONFIGURATION

Configuring the MCW1001A to make a connection at a minimum requires setting the network type, choosing the network name and setting up the security parameters. It is possible to store connection information for two different Connection Profiles (CP) in the MCW1001A. After this information is provided, the network connection process can be initiated by referencing only the profile number (1 or 2).

4.1 Configuring 802.11 Link

There are two types of networks. The most common network is infrastructure in which an access point (AP) is the common point linking all 802.11 devices. The access point keeps track of who's on the local network and directs IP packets. In many cases, the AP is also a router and will forward packets from the local network to other networks and to the Internet. It is also common for the AP to be running a DHCP server which tracks and assigns IP addresses. The second type is ad hoc, which is considered a point-to-point network in that each 802.11 device is linked directly to every other 802.11 device on the ad hoc network. There is no access point. All 802.11 devices on the ad hoc network participate in keeping the network alive and each keeps track of the other active devices on the network by sending and receiving beacon and probe packets. In most cases, IP addresses are assigned through Auto IP, although one of the 802.11 devices can be configured as a DHCP server. By default, the MCW1001A will operate as a DHCP server for one ad hoc client that may join a network created by the MCW1001A, but the IP address of the MCW1001A can be set to a static value through the API. The MCW1001A handles connection management internally under the direction of the host. The following example illustrates a typical connection process:

EXAMPLE 4-1: CREATING AN 802.11 LINK

The following is the procedure to create an 802.11 link:

1. Power up system and de-assert the $\overline{\text{RESET}}$ signal to the MCW1001A.
2. Host will receive an asynchronous Reset message from MCW1001A using the default UART configuration.
3. Host sets desired network configuration through the API (may include scanning for the available network connections and choosing based on the scan results).
4. Host Issues Connect command.
5. The MCW1001A then handles connection to the network and sends asynchronous Event message when connection status changes.

Note: An asynchronous event will be sent to the host if the status of the connection maintained by MCW1001A changes.

4.2 Verifying the Network Link

To verify a network link is established, the MCW1001A can operate as an ICMP host and/or client. To test the operation as a host, issue a Ping from another device on the network to the IP address obtained or set for the MCW1001A. The MCW1001A will respond to the Ping if the network is functional. To test the operation as a client, use the MCW1001A API to issue a Ping to a device on the network that has been previously verified to operate as an ICMP server. In both cases, verify that the IP subnet mask and gateway are configured through the MCW1001A API.

4.3 Network Stack Configuration

The MCW1001A has limited resources in terms of RAM available to buffer socket data and maintain information about socket connections. Support exists for TCP and UDP sockets, but a maximum of 8 instances of each are allowed.

UDP data is not buffered by MCW1001A. When the host MCU initiates an API UDP Send, the data sent over the serial port is immediately sent to the network. Similarly, whenever the MCW1001A receives UDP socket data from the network, that data is passed asynchronously to the host if the host has issued a RecvFrom or RECV Socket command for the socket receiving the data. By limiting the UDP socket instances to 8, the RAM required to maintain information about the UDP sockets is limited.

TCP data must be buffered by MCW1001A in both the directions. A TCP socket may be either a Server (responds to remote client requests) or a Client (requests data from remote Servers). 8192 bytes of RAM have been allocated for TCP socket data buffers. As it is impractical to decide in advance what the allocation of this RAM in terms of how many sockets are available or how large the buffers for the sockets are. The MCW1001A uses a socket allocation technique that allows the available RAM to be configured by the host prior to the creating of sockets. By default, 8 TCP sockets exist, each with a 512 byte transmit and receive buffer. Two sockets are client sockets, and remaining six are server sockets. The number of TCP sockets can be reduced to less than or equal to 8, and the amount of RAM for the send and receive buffers can be adjusted for the client and server socket types allowing the 8192 bytes to be allocated to TCP sockets in as large as 1500 byte blocks. All server sockets must be the same size, and all client sockets must be the same size.

NOTES:

5.0 TRANSMITTING AND RECEIVING PACKETS

5.1 TCP

TCP is a standard transport layer protocol described in Request for Comments (RFC) 793. It provides reliable stream-based connections over unreliable networks, and forms the foundation for HTTP, SMTP, and many other protocol standards. Connections made over TCP guarantee data transfer at the expense of throughput. Connections are made through a three-way handshake process, ensuring a one-to-one connection. Remote nodes advertise how much data they are ready to receive, and all data transmitted must be acknowledged. If a remote node fails to acknowledge the receipt of data, it is automatically retransmitted. This ensures that network errors such as lost, corrupted, or out-of-order packets are automatically corrected. To accomplish this, TCP must operate in a buffer. Once the transmit buffer is full, no more data can be sent until the remote node has acknowledged receipt. Likewise, the remote node cannot transmit more data until the local device has acknowledged receipt and that space is available in the buffer. The following example illustrates how to use TCP packets:

EXAMPLE 5-1:   USING TCP PACKETS

The MCW1001A TCP sockets can be configured as servers or clients. The procedure for transferring TCP packet data in each configuration differs slightly.

5.1.1   MCW1001A TCP SERVER EXAMPLE:

1. Create a TCP socket with call to socket(). This will return an 8-bit socket handle to be used when later referencing this socket.
2. Bind the socket to the listen port with call to bind(). The bind API specifies the 16-bit local port number that the client will attempt a connection on.
3. Prepare the socket to listen for a connection with call to listen(). The listen process opens the specified number of listening sockets that become the children sockets of the bound parent. They will all listen to the port specified in the bind process, but they may accept connections from different clients (different IP addresses, different remote port numbers). If the number of sockets specified is not available, the listen process will return the number of sockets it was not able to allocate.

4. Accept incoming connection through a call to accept(). Accept is non-blocking for the MCW1001A and accepting a connection from a client must be done through polling the socket with the Accept API.
5. Communicate with the remote host using send() and recv().
6. Close the socket with call to close().

5.1.2   MCW1001A TCP CLIENT EXAMPLE:

1. Create a TCP socket with call to socket().
2. Optionally apply a local port number with a call to bind(). Otherwise, a pseudo-random local port number will be assigned.
3. Connect to the server with call to connect(), passing sockaddr, IP address.
4. Communicate with the remote host using send() and recv().
5. Close the socket with call to close().

5.2 UDP

UDP is a standard transport layer protocol described in Request for Comments (RFC) 768. It provides fast but unreliable data-gram based transfers over networks, and forms the foundation SNTP, SNMP, DNS, and many other protocol standards. Connections over UDP should be thought of as data-gram based transfers. Each packet is a separate entity, the application should expect some packets to arrive out-of-order or even fail to reach the destination node. This is in contrast to TCP, in which the connection is thought of as a stream and network errors are automatically corrected. These tradeoffs in reliability are made for an increase in throughput. In general, UDP transfers operate two to three times faster than those made over TCP. As UDP is packet-oriented, each packet must be dealt with in its entirety by your application before returning to the main stack loop. When a packet is received, your application will be called to handle it. This packet will no longer be available the next time your application is called. Therefor, you must either perform all necessary processing or copy the data elsewhere before returning. When transmitting a packet, your application must build and transmit the complete packet in one cycle. The following example illustrates how to use the TCP packets.

EXAMPLE 5-2:   USING UDP PACKETS

The MCW1001A UDP sockets require less configuration than the TCP counterparts. Sending packets using UDP sockets can be accomplished by creating the socket and issuing a Sendto() command specifying the destination address and port number. Receiving packets on a UDP socket requires an extra bind step to set the local port number.

5.2.1 MCW1001A UDP SERVER EXAMPLE:

1. Create a UDP socket with a call to socket specifying the UDP socket type. If successful, an 8-bit socket handle will be returned.
2. Bind the socket to a local port number with a call to bind()
3. Receive data using this socket with a call to recvfrom()
4. Optionally reply to the client with a call to Sendto() specifying the returned address and port number returned in the recvfrom() call.

5.2.2 MCW1001A UDP CLIENT EXAMPLE:

1. Create the UDP socket with a call to socket specifying the UDP socket type. If successful, an 8 bit socket handle will be returned.
2. Transmit data to a remote UDP server with a call to Sendto() specifying the remote IP address and port number.

3. Optionally receive data from the server using this socket with a call to recvfrom().

6.0   MESSAGING API

The API between the host processor and the MCW1001A can be divided into two logical groups of commands. The first set of commands deal with 802.11 connection management, while the second set deals with the BSD Socket Interface. In addition to commands sent from the host to the MCW1001A, several asynchronous messages can be transmitted directly from the MCW1001A to the host like:

• IP address assigned

• 802.11 connection status changed

• 802.11 scan results ready

• Ping response event

• Error Event (including Reset condition)

The size of the transmit and receive buffers for the serial API are limited for MCW1001A. While, it is possible for the host to receive a message from the MCW1001A as large as 1536 bytes, it is possible only to transmit messages to the MCW1001A in 500 byte packets.

6.1   Overview

The WiComm-Socket Serial interface supports UART communication with a host system using no parity, 8 data bits, and 2 stop bits (N,8,2). By default, the baud rate is 115200 BPS, but autobaud may be used to adjust this rate. The rate at which the host sends the first character to WiComm-Socket (0x55) determines the data rate. The allowed values are between 300 baud and 230400 baud.

6.2   WiComm-Socket Packet Format

Figure 6-1 shows the packet format for all serial messages exchanged between the host and WiComm-Socket.

FIGURE 6-1:   HOST CPU PACKET FORMAT

Header0 (1)	Header1 (1)	Msg Type (2)	Msg Data Length (2)	Msg Data (0 thru N)	Trailer (1)

• Header0 – Always 0x55

• Header1– Always 0xAA

• MsgType – 16-bit field containing the message type

• Msg Data Length – 16-bit field containing the number of data bytes that follow (can be `0`)

• Msg Data – Message data bytes

• Trailer – Always one byte equal to 0x45

From the perspective of the host, WiComm-Socket has a little-endian interface. All 16 and 32-bit message fields are represented in an little-endian format.

Where an IP address is specified, 16 bytes are allocated for future compatibility with IPv6. However, for the current implementation of WiComm-Socket, IPv4 is used exclusively, and the first (left most) 4 bytes of the 16 byte field refer to the IPv4 address.

MCW1001A

6.3 WiComm-Socket Control Messages

Messages described in this section are used to manage Wicomm-Socket in areas unrelated to wireless networking.

6.3.1 RESET_MSG (170 COMMAND)

WiComm-Socket can be reset to the initialization vector by issuing the RESET_MSG. Becasue there is no internal non-volatile memory, all configuration parameters will be reset to default values. A single ACK_MSG will confirm receipt of this command prior to the reset. Upon a successful reset, WiComm-Socket will issue an EVENT_MSG - Startup Event.

Msg Byte Index	Description
0:5	Header; Message Type = 170, Data Length = 0
6	0x45 (Frame Trailer)

Response Message: ACK_MSG

6.3.2 GET_VERSION_MSG (23 COMMAND)

In response to this command, Wicomm-Socket will issue the EVENT-MSG – Startup Event which contains version information.

Msg Byte Index	Description
0:5	Header; Message Type = 23, Data Length = 0
6	0x45 (Frame Trailer)

Response Message: ACK_MSG followed by asynchronous EVENT_MSG – Startup Event

6.3.3 GPIO_MSG (172 COMMAND)

WiComm-Socket can be directed to write/read to/from GPIO pins by the host. Refer to the component diagram for GPIO index numbers that correspond to particular package pins. Using this index, the host may use this command to force the I/O pin to source current as an output in either a high or low logic state, or the host may read the pin state by using "Read Input" as the GPIO operation. A GPIO response will be issued to allow WiComm-Socket to report the current state of the selected pin.

Msg Byte Index	Description
0:5	Header; Message Type = 172, Data Length = 2
6	GPIO Index
7	GPIO Operation - 0 = Set Output Low, 1 = Set Output High, 2 = Read Input
8	0x45 (Frame Trailer)

Response Message: GPIO_RESPONSE_MSG

6.3.4 GPIO_RESPONSE_MSG (50 RESPONSE)

In response to a GPIO_MSG, WiComm-Socket will return the current state of the particular I/O pin. This message contains the index of the pin that has been operated on, and the result of the operation.

Msg Byte Index	Description
0:5	Header; Message Type = 50, Data Length = 2
6	GPIO Index
7	GPIO Result – 0 = Output Low
	1 = Output High
	2 = Input Low
	3 = Input High
	255 = Invalid Index
8	0x45 (Frame Trailer)

Response Message: GPIO_RESPONSE_MSG

Preliminary

6.4   Network Configuration Messages

The Network configuration messages are used to configure the basic network settings.

The default network configuration messages are:

- Use DHCP to assign IP address
- DHCP name 'W' + last 6 digits of MAC address
- No Gateway IP address needed
- No DNS IP address needed
- Netmask = `255.255.255.0`

The messages described in this section are used to change the default network configuration.

6.4.1   `SET_IP_ADDRESS_MSG` (41 COMMAND)

By default WiComm-Socket uses DHCP to get an IP address. This message can configure WiComm-Socket to use a static IP address or switch back to using DHCP. WiComm-Socket always responds to this message with an `ACK_MSG`. In addition, if using DHCP, an `EVENT_MSG` containing the IP address will be sent to the host CPU asynchronously when WiComm- Socket is assigned an IP address by an AP.

Msg Byte Index	Description
0:5	Header; Message Type = 41, Data Length = 18
6	Reserved
7	Static IP address flag: • 0 – Use DHCP to assign IP address (default) • 1 – Use static IP address If using static IP address then set the next data field to the static IP address. Otherwise, set the IP address bytes to 0.
8:23	Static IP address
24	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.4.2   `SET_NETWORK_MASK_MSG` (42 COMMAND)

By default WiComm-Socket will use a network mask of `255.255.255.0` to set the network mask configuration.

Msg Byte Index	Description
0:5	Header; Message Type = 42, Data Length = 16
6:21	Network Mask (e.g. `255.255.0.0`)
22	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.4.3   `SET_GATEWAY_IP_ADDRESS_MSG` (44COMMAND)

By default WiComm-Socket has a Gateway IP address of `0.0.0.0` (signifying that there is no Gateway) to set the network mass configuration.

Msg Byte Index	Description
0:5	Header; Message Type = 44, Data Length = 16
6:21	Gateway IP address. If the Gateway IP address is set to `0.0.0.0` then WiComm-Socket will not use a gateway.
22	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.4.4 GET_NETWORK_STATUS (48 COMMAND)

This function queries WiComm-Socket for the current network status (see network status response msg).

Msg Byte Index	Description
0:5	Header; Message Type = 48, Data Length = 0
6	0x45 (Frame Trailer)

Response Message: Network_Status_Response_Msg

6.4.5 NETWORK_STATUS_RESPONSE_MSG (48 RESPONSE)

This message sent by WiComm-Socket describes the specified network interface configuration and whether it is currently connected.

Msg Byte Index	Description
0:5	Header; Message Type = 48, Data Length = 56
6	Reserved
7:12	MAC Address (6 bytes)
13:28	IP Address (16 bytes)
29:44	Network Mask (16 bytes)
45:60	Gateway Address (16 bytes)
61	Network Status – 0 = Not Connected Static IP
	1 = Connected Static IP
	2 = Not Connected DHCP
	3 = Connected DHCP
62	0x45 (Frame Trailer)

6.4.6 SET_MACADDRESS_MSG (49 COMMAND)

Specifies the MAC Address. This command should only be used at initialization.

Msg Byte Index	Description
0:5	Header; Message Type = 48, Data Length = 6
6:11	MAC Address (6 bytes)
12	0x45 (Frame Trailer)

Response Message: ACK_MSG

6.4.7 SET_ARP_TIME_MSG (173 COMMAND)

This message allows the automatic or gratuitous ARP timing to be customized. By default, when connected, WiComm-Socket will issue an ARP at 5 second intervals. By using this API, the ARP period can be adjusted up to 65535 seconds. A special case of 0 turns off the automatic ARP function.

Msg Byte Index	Description
0:5	Header; Message Type = 173, Data Length = 2
6:7	ARP period in seconds
8	0x45 (frame trailer)

Response Message: ACK_MSG

6.5   Wi-Fi General Configuration Messages

Two CPs can be defined and stored on WiComm-Socket FLASH. A CP consists of the following elements:

- Network mode – (default: Infrastructure)
- SSID – (default: "WiCommSocketAP")
- Channel List – (default for infrastructure: channels 1,6,11) and (default for ad hoc: channel 6)
- Security mode – (default: Open)

The messages described in this section configure the first three elements of the CP. The Security mode configuration change is described in Section 6.6, Wi-Fi Security Configuration Messages.

6.5.1   SET_CP_NETWORK_MODE_MSG (55 COMMAND)

This message is used to identify the profile either as an infrastructure or as ad hoc network.

Msg Byte Index	Description
0:5	Header; Message Type = 55, Data Length = 2
6	Connection Profile ID Range: 1 or 2
7	Network mode: 1 – Infrastructure (default) 2 – ad hoc
8	0x45 (Frame Trailer)

Response Message: ACK_MSG

6.5.2   SET_CP_SSID_MSG (57 COMMAND)

The default SSID is 'WiCommSocketAP".

Msg Byte Index	Description
0:5	Header; Message Type = 57, Data Length = equal to SSID Length + 2
6	Connection Profile ID Range: 1 or 2
7	SSID Length (1 to 32 characters)
8	1st character of SSID
9	2nd character of SSID
10:N	Remaining characters of SSID
N + 1	0x45 (Frame Trailer)

Response Message: ACK_MSG

6.5.3 `SET_REGIONAL_DOMAIN_MSG` (56 COMMAND)

This message sets the regional domain and are applicable to all CPs.

Msg Byte Index	Description
0:5	Header; Message Type = 56, Data Length = 1
6	Country Code 0: FCC (default) 1: IC 2: ETSI 3: SPAIN 4: FRANCE 5: JAPANA 6: JAPANB
7	Reserved
8	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.5.4 `SET_CHANNEL_LIST_MSG` (58 COMMAND)

This message sets the channel list that will be scanned if the specified CP is activated. The default channel list is 1, 6, and 11 for infrastructure and channel 6 for ad hoc and are applicable to all CPs.

Msg Byte Index	Description
0:5	Header; Message Type = 58, Data Length = Number of channels in list + 2
6	Number of channels in list Number of channels to scan in the list that follows. Maximum of 11 channels are supported.
7	Reserved
8	1st channel number in list
9	2nd channel number in list
10:N	Remaining channel numbers in list
N + 1	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.5.5 `SET_LIST_RETRY_COUNT_MSG` (59 COMMAND)

This message sets the connection manager retry count. There are separate values for Infrastructure and ad hoc because ad hoc behavior may require terminating the connection retries after a certain number in order to establish a new network. Using a value of 0 represents no retries for the connection if the first attempt fails, and using a value of 255 represents retrying the network connection forever.

Msg Byte Index	Description
0:5	Header; Message Type = 59, Data Length = 2
6	Infrastructure Mode Retry Connection Count (default is 255 = forever)
7	ad hoc Mode Retry Connection Count (default is 5)
8	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.6   Wi-Fi Security Configuration Messages

This section includes messages required to set a CP to a specified security level. The default security level is open security. The messages in this section are mutually exclusive – every message in this group that is sent to WiComm-Socket overrides the previous security settings for that CP.

6.6.1   `SET_CP_SECURITY_OPEN_MSG` (65 COMMAND)

This is the WiComm-Socket default for a CP.

Msg Byte Index	Description
0:5	Header; Message Type = 65, Data Length = 2
6	Connection Profile ID: Range: 1 or 2
7	Reserved
8	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.6.2   `SET_CP_SECURITY_WEP40_MSG` (66 COMMAND)

This message is used to configure the WEP40 key.

Msg Byte Index	Description
0:5	Header; Message Type = 66, Data Length = 24
6	Connection Profile ID: - Range: 1 or 2
7	Key mode: - Open Key = `0` - Shared Key = `1`
8	Default WEP Key Index Specifies which of the 4 keys defined in security keys to use when connecting. Range: 0–3 (normally 0)
9	—
10:29	Security keys Four 5 byte keys are written to this block
30	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

MCW1001A

6.6.3 SET_CP_SECURITY_WEP104_MSG (67 COMMAND)

This message is used to configure the WEP104 key.

Msg Byte Index	Description
0:5	Header; Message Type = 67, Data Length = 56
6	Connection Profile ID: Range: 1 or 2
7	Key mode: - Open Key = 0 - Shared Key = 1
8	Default WEP Key Index Specifies which of the 4 keys defined in security keys to use when connecting. Range: 0–3 (normally 0)
9	Reserved
10-61	Security keys Four 13 byte keys are written to this block
62	0x45 (Frame Trailer)

Response Message: ACK_MSG

Preliminary

6.6.4 `SET_CP_SECURITY_WPA_MSG` (68 Command)

This message is used to configure WiComm-Socket to use either WPA or WPA2 wireless security.

Msg Byte Index	Description		
0:5	Header; Message Type = 68, Data Length = Security key Length + 4		
6	Connection Profile ID: Range: 1 or 2		
7	**Security Type**		
	Value	Description	
	3	Use WPA security. Binary Pre-shared Key (PSK) key will be provided in security key	
	4	Use WPA security. ASCII WPA pass phrase will be provided in security key and while connecting the MRF24WB0M will calculate the PSK key (which can take up to 30 seconds)	
	5	Use WPA-2 security. Binary WPA-2 key will be provided in security Key	
	6	Use WPA-2 security. ASCII WPA-2 pass phrase will be provided in security Key, and while connecting, the MRF24WB0M will calculate the PSK key (which can take up to 30 seconds).	
	7	Same as (3) and (5), except WiComm-Socket connects to the AP using highest level security that the AP supports (WPA or WPA2).	
	8	Same as (4) and (6), except connection manager connects to the AP using highest level security the AP supports (WPA or WPA2).	
	Note:	When using an ASCII pass phrase instead of a binary key, WiComm-Socket, when connecting, will take approximately 30 (TBD) seconds to calculate the binary key.	
8	Reserved		
9	Security key Length Number of bytes in security key field (1–64)		
10:N + 9	Security key Either the binary key bytes or the pass phrase bytes. The length of this field must be equal to security key length		
N + 10	0x45 (Frame Trailer)		

Response Message: `ACK_MSG`

6.6.5 `GET_CP_WPAKEY_MSG` (71 COMMAND)

Message sent from the Host to WiComm-Socket to determine the calculated PSK key. After a connection has been established with a WPA passphrase, this command can be used to read the 32 byte calculated key from WiComm-Socket. The Host can then store this key for use at a later time in security modes 3, 5 or 7 to avoid the 30 second key calculation step described above in future connections.

Msg Byte Index	Description
0:5	Header; Message Type = 71, Data Length = 2
6	Connection Profile ID:
	Range: 1 or 2
7	Reserved
8	0x45 (Frame Trailer)

Response Message: WPAKEY_RESPONSE_MSG

6.6.6 `WPAKEY_RESPONSE_MSG` (49 RESPONSE)

This response contains the 32 byte security key calculated from the SSID and passphrase by WiComm-Socket. This value is only valid if a connection based on the supplied passphrase has been made in one of the WPA security modes.

Msg Byte Index	Description
0:5	Header; Message Type = 49, Data Length = 32
6:37	Security Key (32 bytes)
38	0x45 (Frame Trailer)

Preliminary

6.7 Wi-Fi Scanning Messages

6.7.1 SCAN_START_MSG (80 COMMAND)

This message directs WiComm-Socket to scan for access points (APs) using the settings from the specified CP. The immediate response to this message is the ACK_MSG. After the FTP operation has completed the EVENT_MSG – Wi-Fi Scan Results Ready will be sent to the host CPU. The scan results can then be retrieved through the SCAN_GET_RESULTS_MSG. Note that Scan may only be performed in the disconnected state, and when a scan is in process, the only command WiComm-Socket will accept is the reboot command.

Msg Byte Index	Description
0:5	Header; Message Type = 80, Data Length = 2
6	Connection Profile ID: - Range: 1 or 2 [select defined CP] - 0xFF [scan everything] If using a defined CP, then the criteria within that CP will be used for the scan (for example: scan for that CP's SSID using the CP's channel list). If the CP is infrastructure then only scan results from infrastructure networks are reported. If the CP is ad hoc then only scan results from ad hoc networks are reported. If the field is set to 0xFF then WiComm-Socket will scan all channels within its regional domain and report all results in infrastructure and ad hoc networks.
7	Reserved
8	0x45 (Frame Trailer)

Response Message: ACK_MSG

6.7.2 SCAN_GET_RESULTS_MSG (81 COMMAND)

This message is used to retrieve the scan results from the most recent scan. It can be called after the EVENT_MSG – Wi-Fi Scan Results Ready is received.

Msg Byte Index	Description
0:5	Header; Message Type = 81, Data Length = 2
6	List Index The index within the WiComm-Socket scan list The valid range is: 0 Number of Scan results – 1 (from the EVENT_MSG – Wi-Fi Scan Results Ready)
7	Reserved
8	0x45 (Frame Trailer)

Response Message: SCAN_RESULT_MSG

MCW1001A

6.7.3   SCAN_RESULT_MSG (22 RESPONSE)

This message is sent in response to the SCAN_GET_RESULTS_MSG.

Msg Byte Index	Description
0:5	Header; Message Type = 22, Data Length = 57
6:11	BSSID
12	SSID Length Number of valid SSID characters that follow
13:44	SSID
45	AP configuration (8 bits) This byte is defined as follows: <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>WPA2</td><td>WPA</td><td>Preamble</td><td>Privacy</td><td>Res</td><td>Res</td><td>Res</td><td>IE</td></tr></table> <table><tr><td>IE</td><td>1 if AP broadcasting one or more information elements, otherwise 0</td></tr><tr><td>Res</td><td>Reserved</td></tr><tr><td>Privacy</td><td>0: AP is open (no security) 1: AP using security; if neither WPA or WPA2 is set then security is WEP.</td></tr><tr><td>Preamble</td><td>0: AP transmitting with short preamble 1: AP transmitting with long preamble</td></tr><tr><td>WPA</td><td>Only valid if Privacy is 1. 0: AP does not support WPA 1: AP supports WPA</td></tr><tr><td>WPA2</td><td>Only valid if Privacy is 1. 0: AP does not support WPA2 1: AP supports WPA2</td></tr></table>
46:47	Beacon Period Network beacon interval
48:49	ATIM Window Only valid if BSS Type is ad hoc
50:57	Basic Rates, each 1 byte, in the following format: Bit 7:0: Rate is not part of the "basic rates set" 1: Rate is part of the "basic rates set" Bits 6:0: Multiple of 500 kbps giving the supported rate. For example, a value of 2 (2x500 kbps) indicates that 1 Mbps is a supported rate. A value of 4 in this field indicates a 2 Mbps rate (4x500 kbps).
58	RSSI of received beacon or probe response
59	Basic Rate List Length Number of basic rates that are valid in above basic rate list
60	DTIM period (1 byte) Part of TIM element
61	BSS Type (1 byte) 1: Infrastructure 2: ad hoc
62	Channel number (1 byte)
63	0x45 (Frame Trailer)

Preliminary

6.8   Wi-Fi Connection Messages

The messages in this section are used to initiate the connection to a Wi-Fi network, or to disconnect an existing connection.

6.8.1   `Wi-Fi_CONNECT_MSG` (90 COMMAND)

This message directs WiComm-Socket to initiate a Wi-Fi connection process as defined in the specified CP. WiComm-Socket will send an `ACK_MSG` in response to this message. After the connection succeeds or fails, an `EVENT_MSG` will asynchronously be sent from WiComm-Socket to the CPU host.

Msg Byte Index	Description
0:5	Header; Message Type = 90, Data Length = 2
6	Connection Profile ID to connect to Range: 1 or 2
7	Reserved
8	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.8.2   `Wi-Fi_DISCONNECT_MSG` (91 COMMAND)

This message will cause WiComm-Socket to close its existing Wi-Fi connection. WiCom-Socket will send an `ACK_MSG` in response to this message. After the connection has been closed an `EVENT_MSG` will be sent to notify the host CPU.

Msg Byte Index	Description
0:5	Header; Message Type = 91, Data Length = 0
6	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.9 Wi-Fi Power Management Messages

The messages in this section are used to set the power mode for WiComm-Socket. Note that PSPOLL modes are only applicable to Infrastructure networks.

6.9.1 `SET_POWER_SAVE_MODE_MSG` (102 COMMAND)

This message sets Power Save mode for WiComm-Socket. Options include full power, an AP polling mode with self set wake up times, an AP polling method using the DTIM period set by the AP, and a Hibernate (radio module off) mode. By default, Power Save mode of the module is 4 (Power Save mode Off).

IMPORTANT:

1)Hibernate mode can only be exited through a WiComm Reset. This can be performed through the API, or by using the physical reset pin. After exiting reset, the device must be reinitialized including any connection profile information to reestablish a connection.

2)When using the Power Save mode of 3, the self set wake up period is used only after it is negotiated with the AP. In practice it is suggested that the DTIM listen interval be set while disconnected so that it is applied during the connection process. The AP may choose not to allow association with the radio if the DTIM listen interval is too long.

Msg Byte Index	Description
0:5	Header; Message Type = 102, Data Length = 4
6	Power Save mode:
	1 – Power Save mode Hibernate (radio module off)
	2 – Power Save mode PSPOLL with AP suggested DTIM period
	3 – Power Save mode PSPOLL with self set wake up period
	4 – Power Save mode Off - Module fully powered
7	Reserved
8:9	DTIM Listen Interval in PSPOLL mode (self-set wake up time in Power Save mode 3). Actual time is this interval multiplied by 100msec, e.g. 10 = 1second.
10	0x45 (Frame Trailer)

Response Message: `ACK_MSG`

6.10 ICMP (Ping) Messages

6.10.1   PING_SEND_MSG (121 COMMAND)

This message sends a ping to a remote host. The immediate response to this message is an ACK_MSG. When WiComm-Socket receives a ping response, or times out waiting for a ping response, it will send the EVENT_MSG – Ping Response Event (see Section 6.12.2.4, EVENT_MSG – Ping Response Event (1-26 Async) to the host. In the current version of WiComm-Socket, the time-out value is fixed at 4 seconds.

Msg Byte Index	Description
0:5	Header; Message Type = 121, Data Length = 16
6:21	Remote IP Address to ping
22	0x45 (Frame Trailer)

Response Message: ACK_MSG (followed later by EVENT_MSG – Ping Response Event)

MCW1001A

6.11 Socket Messages

6.11.1 SOCKET_CREATE_MSG (110 COMMAND)

This message attempts to create a TCP or UDP socket.

Msg Byte Index	Description
0:5	Header; Message Type = 110, Data Length = 2
6	Socket Type: • 0 – UDP • 1 – TCP
7	Reserved
8	0x45 (Frame Trailer)

Response Message: SOCKET_CREATE_RESPONSE_MSG

6.11.2 SOCKET_CREATE_RESPONSE_MSG (23 RESPONSE)

This message indicates the result of the attempted socket created by returning a "handle" to the socket. If a socket is not created, WiComm-Socket will return 0xFE as the handle or 0xFF if the socket type was unknown (not 0 or 1 in above message).

Msg Byte Index	Description
0:5	Header; Message Type = 23, Data Length = 2
6	Socket Handle: 0–253 – Valid Socket handle 254 – Invalid Socket 255 – Unknown Socket Type
7	Reserved
8	0x45 (Frame Trailer)

6.11.3 SOCKET_CLOSE_MSG (111 COMMAND)

This message attempts to close a previously created TCP or UDP socket.

Msg Byte Index	Description
0:5	Header; Message Type = 111, Data Length = 2
6	Socket Handle
7	Reserved
8	0x45 (Frame Trailer)

Response Message: ACK_MSG (may generate an asynchronous Error Event if socket close failed)

　Preliminary

6.11.4 SOCKET_BIND_MSG (112 Command)

The Bind command associates a local IP address and port with a socket. This information allows the socket to advertise its presence. As WiComm-Socket is always associated with only one local IP address, it is not required to supply the local IP address in the Bind command. If the object of the bind process is a UDP socket, WiComm-Socket will attempt to open the UDP port to remote connections. In this mode, the UDP socket can function as a server to accept client messages which can be subsequently read from the socket. If this open process for the UDP port fails, the bind response will return unsuccessful. It is not required to bind a UDP socket if its only purpose is to transmit as a client, or if it will transmit before receiving from a remote connection. If the local port number is specified as zero, WiComm-Socket will assign a non-zero value to the port beginning with 1024.

Msg Byte Index	Description
0:5	Header; Message Type = 112, Data Length = 4
6:7	Local Port Number to be associated with this socket
8	Socket Handle (previously created with SOCKET_CREATE_MSG)
9	Reserved
10	0x45 (Frame Trailer)

Response Message: SOCKET_BIND_RESPONSE_MSG

6.11.5 SOCKET_BIND_RESPONSE_MSG (24 Response)

This message indicates the result of the attempted socket bind as either successful (0) or not (0xFF).

Msg Byte Index	Description
0:5	Header; Message Type = 24, Data Length = 4
6:7	Local Port Number bound to
8	Bind Result: 0 – Success 1-255 – Unsuccessful Bind
9	Reserved
10	0x45 (Frame Trailer)

6.11.6 SOCKET_CONNECT_MSG (113 Command)

For TCP sockets, the Connect command attempts to establish a connection between two sockets. For UDP sockets, the Connect command specifies the peer or remote endpoint for the socket. The socket parameter specifies the local socket to be used, and the remote address and port specify the peer or remote socket to connect. The Connect command will perform an implicit bind for UDP sockets if not done previously. The Connect command is non-blocking and will return immediately with a result of either Success (0), Connection In Process (0xFE), or Error (0xFF).

Msg Byte Index	Description
0:5	Header; Message Type = 113, Data Length = 20
6	Socket Handle (previously created with SOCKET_CREATE_MSG)
7	Reserved
8:9	Remote Port Number to connect to
10:25	Remote IP address – for IPv4 the first 4 bytes are the IP address
26	0x45 (Frame Trailer)

Response Message: SOCKET_CONNECT_RESPONSE_MSG

MCW1001A

6.11.7 SOCKET_CONNECT_RESPONSE_MSG (25 Response)

This message indicates the result of the attempted socket connect.

Msg Byte Index	Description
0:5	Header; Message Type = 25, Data Length = 2
6	Socket connect result: 0 – Success 254 – Socket connect in process 255 – Socket connect failed
7	Reserved
8	0x45 (Frame Trailer)

6.11.8 SOCKET_LISTEN_MSG (114 Command)

For TCP sockets, the Listen command makes the socket passive. After the socket becomes passive, it cannot be used to initiate connection requests to remote servers (connect cannot be used on a listening socket). For UDP sockets, this function has no meaning. The socket handle parameter specifies the local socket to be used for listening that has been previously created and bound with the SOCKET_CREATE_MSG and the SOCKET_BIND_MSG. The backlog parameter specifies the number of connections to make available to remote clients. Each of these uses a limited number of WiComm-Socket server socket. The return value will indicate how many server sockets were actually allocated by returning the modified backlog count. For instance, if a socket is set into Listen mode with a backlog parameter of 5, and the backlog return value in the response is 3, there were 2 server sockets allocated. This command may be issued to a socket that is already listening, and if it is possible for more sockets to be opened based on the current socket backlog, the allocations will be performed.

Msg Byte Index	Description
0:5	Header; Message Type = 114, Data Length = 2
6	Socket Handle (previously created with SOCKET_CREATE_MSG)
7	Server socket count (backlog)
8	0x45 (Frame Trailer)

Response Message: SOCKET_LISTEN_RESPONSE_MSG

6.11.9 SOCKET_LISTEN_RESPONSE_MSG (26 RESPONSE)

This message indicates the result of the attempted socket Listen command.

Msg Byte Index	Description
0:5	Header; Message Type = 26, Data Length = 2
6	Socket connect result: 0 – Success 254 – Socket connect in process 255 – Socket connect failed
7	New Unassigned backlog count
8	0x45 (Frame Trailer)

MCW1001A

6.11.10 SOCKET_ACCEPT_MSG (115 COMMAND)

For TCP sockets, the Accept command will indicate a connected server socket by returning the local server socket number of a remote client connection.

Msg Byte Index	Description
0:5	Header; Message Type = 115, Data Length =2
6	Socket Handle (previously created with SOCKET_CREATE_MSG)
7	Reserved
8	0x45 (Frame Trailer)

Response Message: SOCKET_ACCEPT_RESPONSE_MSG

6.11.11 SOCKET_ACCEPT_RESPONSE_MSG (27 RESPONSE)

This message indicates the result of the attempted socket Accept command.

Msg Byte Index	Description
0:5	Header; Message Type = 27, Data Length = 4
6	Socket Handle value returned - If the value is 0xFF, the connection is not valid
7	Reserved
8:9	Remote Port Number
10:25	Remote IP address – for IPv4 the first 4 bytes are the IP address
26	0x45 (Frame Trailer)

6.11.12 SOCKET_SEND_MSG (116 COMMAND)

The Send command sends data on the target socket. The remote connection is not defined in the call to this function, but instead it relies on the socket having been connected prior to the send. In the case of a memory limited WiComm-Socket, the host may need to limit the size of the data sent in each command. A facility will be provided to allow messages to be fragmented by the host and reassembled in WiComm-Socket to create a complete packet.

Msg Byte Index	Description
0:5	Header; Message Type = 116, Data Length = 4 + N Data Bytes
6	Socket Handle (previously created with SOCKET_CREATE_MSG and connected)
7	Reserved
8:9	Data byte count (N)
10:N+9	N bytes of data to be sent
N+10	0x45 (Frame Trailer)

Response Message: SOCKET_SEND_RESPONSE_MSG

6.11.13 SOCKET_SEND_RESPONSE_MSG (28 RESPONSE)

This message indicates the result of the attempted socket Sendto command.

Msg Byte Index	Description
0:5	Header; Message Type = 28, Data Length = 2
6:7	Socket byte count sent
8	0x45 (Frame Trailer)

6.11.14 SOCKET_RECV_MSG (117 COMMAND)

The Recv command attempts to read data from an open socket. In this version of WiComm-Socket, the Recv command is always non-blocking, but UDP received data is not buffered in WiComm-Socket. If data is available to be read from the socket, the amount of data available will be returned (not to exceed the specified buffer size in the call). If no data is available, the RECV response will return 0 as the received count. As a result of non-buffered approach to UDP receive buffers, WiComm-Socket uses an asynchronous SOCKET_RECV_FROM_RESPONSE_MSG to report any data received on a bound UDP listener socket. In this way, the SOCKET_RECV_MSG is not typically used to poll for UDP socket received data and by design, a call to this function on a UDP socket will return 0 bytes read.

Msg Byte Index	Description
0:5	Header; Message Type = 117, Data Length = 4
6	Socket Handle (previously created with SOCKET_CREATE_MSG)
7	Reserved
8:9	Amount of data to be read
10	0x45 (Frame Trailer)

Response Message: SOCKET_RECV_RESPONSE_MSG

6.11.15 SOCKET_RECV_RESPONSE_MSG (29 RESPONSE)

This message indicates the result of the attempted socket receive message.

Msg Byte Index	Description
0:5	Header; Message Type = 29, Data Length = 4 + N Data Bytes
6	Socket Handle (used when data was read)
7	Reserved
8:9	Byte count received (N)
10:N + 9	N bytes of data read
N + 10	0x45 (Frame Trailer)

6.11.16 SOCKET_SEND_TO_MSG (118 COMMAND)

The Sendto command sends data on the target socket. Unlike the socket Send command, the remote connection is defined in the call to this function. By specifying the remote address and port, the connection can be created if it does not exist. In the case of a memory limited WiComm-Socket, the host may need to limit the size of the data sent in each command. A facility will be provided to allow messages to be fragmented by the host and reassembled in WiComm-Socket to create a complete packet.

Msg Byte Index	Description
0:5	Header; Message Type = 118, Data Length = 22 + N Data Bytes
6	Socket Handle (previously created with SOCKET_CREATE_MSG)
7	Reserved
8:9	Remote Port Number to send the data to
10:25	Remote IP address – for IPv4 the first 4 bytes are the IP address
26:27	Data byte count (N)
28:N + 27	N bytes of data to be sent
N + 28	0x45 (Frame Trailer)

Response Message: SOCKET_SEND_TO_RESPONSE_MSG

6.11.17 `SOCKET_SEND_TO_RESPONSE_MSG` (30 RESPONSE)

This message indicates the result of the attempted socket Sendto command. If successful, the number of bytes transferred will be returned. If unsuccessful, an `ACK_MSG` will be returned with a subsequent asynchronous Error event message.

Msg Byte Index	Description
0:5	Header; Message Type = 30, Data Length = 2
6:7	Byte count sent
8	0x45 (Frame Trailer)

6.11.18 `SOCKET_RECV_FROM_MSG` (119 COMMAND)

The RecvFrom command attempts to read data from an open socket. It is normally used with UDP sockets as the return values include the source address of the received data. In this version of WiComm-Socket, the RecvFrom command is always non-blocking, but UDP received data is not buffered in WiComm-Socket. If data is available to be read from the socket, the amount of data available will be returned (not to exceed the specified buffer size in the call). If no data is available, the RecvFrom response will return 0 as the received count. As a result of the non-buffered approach to UDP receive buffers, WiComm-Socket uses an asynchronous `SOCKET_RECV_FROM_RESPONSE_MSG` to report any data received on a bound UDP listener socket. In this way, the `SOCKET_RECV_FROM_MSG` is not typically used to poll for UDP socket received data and by design, a call to this function on a UDP socket will return 0 bytes read.

Msg Byte Index	Description
0:5	Header; Message Type = 119, Data Length = 4
6	Socket Handle (previously created with `SOCKET_CREATE_MSG`)
7	Reserved
8:9	Amount of data to be read
10	0x45 (Frame Trailer)

Response Message: `SOCKET_RECV_FROM_RESPONSE_MSG`

6.11.19 `SOCKET_RECV_FROM_RESPONSE_MSG` (31 RESPONSE)

This message indicates the result of the attempted socket RecvFrom command. The number of bytes transferred will be returned (can be zero).

Msg Byte Index	Description
0:5	Header; Message Type = 31, Data Length = 22 + N Data Bytes
6	Socket Handle (used when data was read)
7	Reserved
8:9	Port number of sender
10:25	IP Address of sender
26:27	Byte count received (N)
28:N + 27	N bytes of data read
N + 28	0x45 (Frame Trailer)

6.11.20 SOCKET_ALLOCATE_MSG (122 COMMAND)

WiComm-Socket has a limited amount of buffer space available for TCP sockets. The current total memory space available to all TCP sockets is 8192 bytes. Each TCP socket can be used in a client or server configuration, and every socket must have a receive buffer and a transmit buffer. By default, there are 8 TCP sockets defined. 6 of them are predefined as server sockets, and 2 of them are predefined as client sockets. The available 8K of memory is split evenly between the 8 sockets and split evenly again between the transmit and receive buffers so that every buffer is allocated a total of 512 bytes. The host may customize the allocation of socket memory, not to exceed 8 total TCP sockets. However, the allocation must be the same for all server sockets and all client sockets. The SOCKET_ALLOCATE_MSG allows the size of the transmit and receive buffers to be set for server sockets and client sockets, and the count for each socket type to be defined. The total socket count (server+client) must not exceed 8, and the total buffer allocation must not exceed 8192 bytes. This command should be executed shortly after initialization prior to establishing a Wi-Fi connection as it will reinitialize the TCP stack.

Msg Byte Index	Description
0:5	Header; Message Type = 122, Data Length = 10
6	TCP Server Socket Count
7	TCP Client Socket Count
8:9	TCP Server Socket Receive Buffer Size (bytes)
10:11	TCP Server Socket Transmit Buffer Size (bytes)
12:13	TCP Client Socket Receive Buffer Size (bytes)
14:15	TCP Client Socket Transmit Buffer Size (bytes)
16	0x45 (Frame Trailer)

Response Message: SOCKET_ALLOCATE_RESPONSE_MSG

6.11.21 SOCKET_ALLOCATE_RESPONSE_MSG (32 RESPONSE)

This message indicates the result of the attempted socket allocation command. If successful, the result field returned will be 0. If the result is 0xFF, the maximum number of sockets was exceeded, and if the result is 0xFE, the total buffer allocation size was exceeded.

Msg Byte Index	Description
0:5	Header; Message Type = 32, Data Length = 2
6	Result (see description)
7	Allocated Socket Count
8	0x45 (Frame Trailer)

6.12 Messages Sent From WiComm-Socket to Host

Messages are sent from the WiComm-Socket to the host to respond to messages from the host, or to indicate an asynchronous event. All messages from the host are acknowledged. This can be done either with a specific message (ACK_MSG) or with a PIGGYBACKACK if data is going to be sent by the WiComm-Socket in response to the host message.

All messages sent from WiComm-Socket to the host contain a PIGGYBACKACK bit. The most significant bit of the 16 bit message type field is reserved for the PIGGYBACKACK. If a reply is sent from WiComm-Socket in response to a message sent by the host, the PIGGYBACKACK bit will be set (i.e. message type 0x8000) in the message type to indicate that the previous message was received correctly.

6.12.1 ACK_MSG (0 RESPONSE)

This message is an acknowledge to the message sent from the host.

Msg Byte Index	Description
0:5	Header; Message Type = 0, Data Length = 0

6.12.2 EVENT_MSG

The EVENT_MSG is sent to the host CPU asynchronously whenever an event has occurred that the host CPU needs to be aware of.

6.12.2.1 EVENT_MSG – IP Address Assigned (1-16 Async)

This event message signifies that the DHCP client has been assigned an IP address.

Msg Byte Index	Description
0:5	Header; Message Type = 1, Data Length = 6
6	Event Type: IP Assigned Event = 16
7	Reserved
8	1st octet of IP address
9	2nd octet of IP address
10	3rd octet of IP address
11	4th octet of IP address
12	0x45 (frame trailer)

MCW1001A

6.12.2.2 `EVENT_MSG` – Wi-Fi Connection Status Changed (1-8 Async)

This event message signifies a change in the Wi-Fi connection status.

Msg Byte Index	Description		
0:5	Header; Message Type = 1, Data Length = 4		
6	Event Type: Wi-Fi Connection Status Event = 8		
7	Event Status: 1: Connection attempt succeeded 2: Connection attempt failed 3: Connection temporarily lost, WiComm-Socket is retrying 4: Connection reestablished, WiComm-Socket reestablished the link 5: Connection permanently lost, retries exhausted		
8	**Event Data (not always used)**		
	If Status equals	**The Optional Data signifies**	
	1	Optional data field not used	
	2	Reason that connection failed	
		Join failure	2
		Authentication failure	3
		Association failure	4
		WEP handshake failure	5
		PSK calculation failure	6
		PSK handshake failure	7
		ad hoc join failure	8
		Security mismatch failure	9
		No suitable AP found failure	10
		Retry forever not supported failure	11
	3	Reason that connection was temporarily lost	
		Beacon time out	1
		Deauth ReceivedDE	2
		Disassociate received	3
	4	Optional data field not used	
	5	Reason that connection was permanently lost: Same as when status equals 3	
9	Reserved		
10	0x45 (frame trailer)		

6.12.2.3 `EVENT_MSG` – Wi-Fi Scan Results Ready (1-9 Async)

This event message signifies that the scan results are ready.

Msg Byte Index	Description
0:5	Header; Message Type = 1, Data Length = 2
6	Event Type: Wi-Fi Scan Results Ready Event = 9
7	Number of Scan Results
	The number of Scan Results
8	0x45 (frame trailer)

6.12.2.4 `EVENT_MSG` – Ping Response Event (1-26 Async)

This event message is sent to the host when either a ping response is received, or WiComm-Socket has timed-out waiting for a ping response. See Section 6.10.1, PING_SEND_MSG (121 Command)

Msg Byte Index	Description
0:5	Header; Message Type = 1, Data Length = 4
6	Event Type: Ping Response Event = 26

7	Ping Response Status:
	    0 – valid ping response was received
	    1 – ping response time out

8:9	Response time in milliseconds
10	0x45 (frame trailer)

6.12.2.5 `EVENT_MSG` – Error Event (1-255 Async)

This event is sent to the host to report an error condition.

Msg Byte Index	Description
0:5	Header; Message Type = 1, Data Length = 8
6	Event Type: Error Event = 255
7	Reserved
8:9	Error Type: 60 – Baud rate generator error 61 – Invalid Connection Profile ID 62 – Wi-Fi already connected error 63 – Wi-Fi already disconnected error 64 – Close socket failed error 65 – Socket Sendto Time out Error (ARP request Timed Out) 66 – Scan Index Out of Range Error 67 – ICMP Ping Flood Error 68 – ICMP Ping In Use Error 69 – Socket RecvFrom Failed 71 – Serial Transmit Buffer Allocation Error 72 – General Assert Error 73 – Invalid Powersave Mode Error 74 – Busy (Hibernate Mode) Error 75 – Busy (Scan Mode) Error
10:13	Error Data: Error details if applicable
14	0x45 (frame trailer)

MCW1001A

6.12.3   `EVENT_MSG` – STARTUP EVENT (1-27 ASYNC)

This event is sent to the host to indicate that WiComm-Socket has started.

Msg Byte Index	Description
0:5	Header; Message Type = 1, Data Length = 6
6	Event Type: Startup Event = 27
7	Startup Condition BitMask:
	0x01 – Power On Reset
	0x02 – Brown Out Reset
	0x04 – Idle Reset
	0x08 – Sleep Reset
	0x10 – Watchdog Reset
	0x40 – Software Reset
	0x80 – External Reset
	e.g. 0x11 = Power on and Watchdog Reset
8:9	WiComm-Socket Version
10:11	Radio Version
12	0x45 (frame trailer)

7.0   ELECTRICAL CHARACTERISTICS

This section provides an overview of the MCW1001A electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the MCW1001A devices are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions, above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings[†]

Ambient temperature under bias...-40°C to +125°C
Storage temperature .... -65°C to +150°C
Voltage on $\overline{\text{RESET}}$ with respect to V_{SS} .... -0.3V to 6.0V
Voltage on all other pins with respect to V_{SS} .... -0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS}.. -0.3V to 4.0V
Total power dissipation **(Note 1)**...1.0W
Maximum current out of V_{SS} pin ....300 mA
Maximum current into V_{DD} pin ....250 mA
Maximum output current sunk by GPIO[5:7] pin ....25 mA
Maximum output current sourced by GPIO[5:7] pins ....25 mA
Maximum output current sunk by GPIO[4:0] pins .... 8 mA
Maximum output current sourced by all GPIO[0:4] pins... 8 mA

Note 1:   Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{(V_{DD} - V_{OH}) \times I_{OH}\} + \Sigma (V_{OL} \times I_{OL})$.

---

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

MCW1001A

7.1 DC Characteristics

TABLE 7-1: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Operating Conditions (unless otherwise indicated): 3.0V ≤ VDD ≤ 5.5V at -40°C ≤ TA ≤ +85°C (I-Temp)				
Param. No.	Characteristics	Symbol	Min.	Typical	Max.	Units	Conditions
D001	Supply voltage	VDD	2.15	—	3.6	V	—
	VDD start voltage to ensure internal Power-on reset signal	VPOR	—	—	0.7	V	—
	—	—	—	—	—	—	—
D003	VDD Rise rate to ensure Power-on reset	SVDD	0.05	—	—	V/ms	Design guidance only not tested
D004	Supply current	IDD	—	—	—	—	—
	VDD = 3.0V	—	—	10	12	mA	FOSC = 12 MHz (330 nF on VUSB)
	VDD = 5.0V	—	—	13	15	mA	
Input Low-Voltage							
D031	Schmitt Trigger (GPIO)	VIL	—	—	0.2 VDD	V	3.0V ≤ VDD ≤ 5.5V
	TTL (UCTS pin)		—	—	0.8		4.5V ≤ VDD ≤ 5.5V
Input High-Voltage							
D041	Schmitt Trigger (GPIO)	VIH	0.8 VDD	—	VDD	V	3.0V ≤ VDD ≤ 5.5V
	TTL (URTS pin)		2.0	—	VDD		4.5V ≤ VDD ≤ 5.5V
Input Leakage current							
D060	GPIO, UCTS	IIL	—	±50	±100	nA	VSS ≤ VPIN ≤ VDD, pin at Hi-Z
	URTS		—	±50	±200		
	OSC1		—	±50	±100		
Output Low-Voltage							
D080	GPIO	VOL	—	—	0.6	V	IOL = 8.0 mA, VDD = 0.5V
			—	—	0.6		IOL = -6.0 mA, VDD = 3.3V
Output High-Voltage							
D090	GPIO	VOH	VDD - 0.7	—	—	V	IOL = -3.5 mA, VDD = 5.0V
			VDD - 0.7	—	—		IOL = -3.0 mA, VDD = 3.3V
Capacitance Loading Specs on Output Pins							
D101	OSC2	COSC2	—	—	15	pF	**Note 1**
D102	GPIO	CIO	—	—	50	pF	**Note 1**

Note 1: This parameter is characterized, but not tested.

Preliminary

© 2011 Microchip Technology Inc.

TABLE 7-2: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

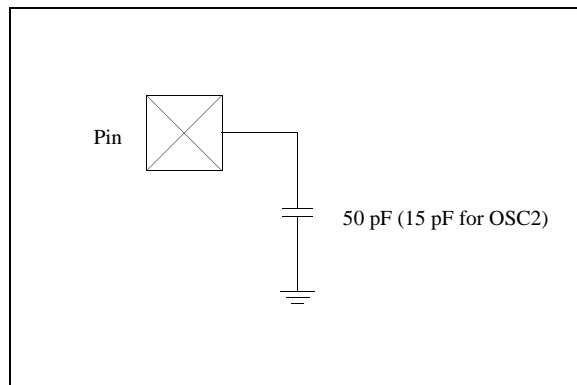| Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
	Operating Conditions: -40°C $<$ TA $<$ +85°C (unless otherwise stated)						
1	V$_{RGOUT}$	Regulator Output Voltage	2.35	2.5	2.7	V	Regulator enabled, V$_{DD}$ = 3.0V
2	C$_{EFC}$	External filter Capacitor	5.4	10	18	µF	ESR < 3Ω recommended ESR < 5Ω required

7.2   AC Characteristics

7.2.1   TIMING CONDITIONS

The operating temperature and voltage specified in Table 7-3 apply to all timing specifications unless otherwise noted. Figure 7-1 specifies the load conditions for the timing specifications.

TABLE 7-3: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC Characteristics	Standard Operating Conditions (unless otherwise stated) Operating temperature: -40°C $\leq$ TA $\leq$ +85°C (unless otherwise stated) Operating Voltage V$_{DD}$ range as described in DC spec section 3.1 DC.

FIGURE 7-1:   LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

Pin

50 pF (15 pF for OSC2)

MCW1001A

7.2.2 TIMING DIAGRAMS AND SPECIFICATIONS

TABLE 7-4: RESET, OSCILLATOR START-UP TIMER AND POWER UP TIMER PARAMETERS

| **Standard Operating Conditions (unless otherwise stated)** Operating Temperature -40°C < T_A < +85°C |

Param. No.	Symbol	Characteristics	Min.	Typical(†)	Max.	Units	Conditions
30	T_{RST}	$\overline{RESET}$ Pulse Width (low)	2	—	—	µs	—
31	T_{PWRT}	Power up timer	40	65	140	ms	—
32	T_{OST}	Oscillator startup time	—	1024	—	T_{OST}	—

* These parameters are characterized but not tested.
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
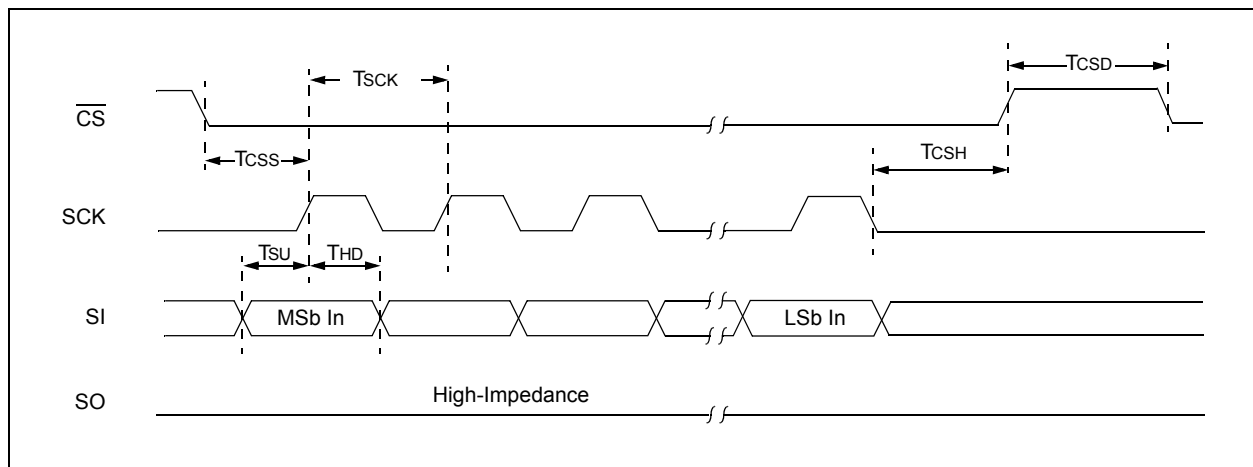
FIGURE 7-2: SPI INPUT TIMING
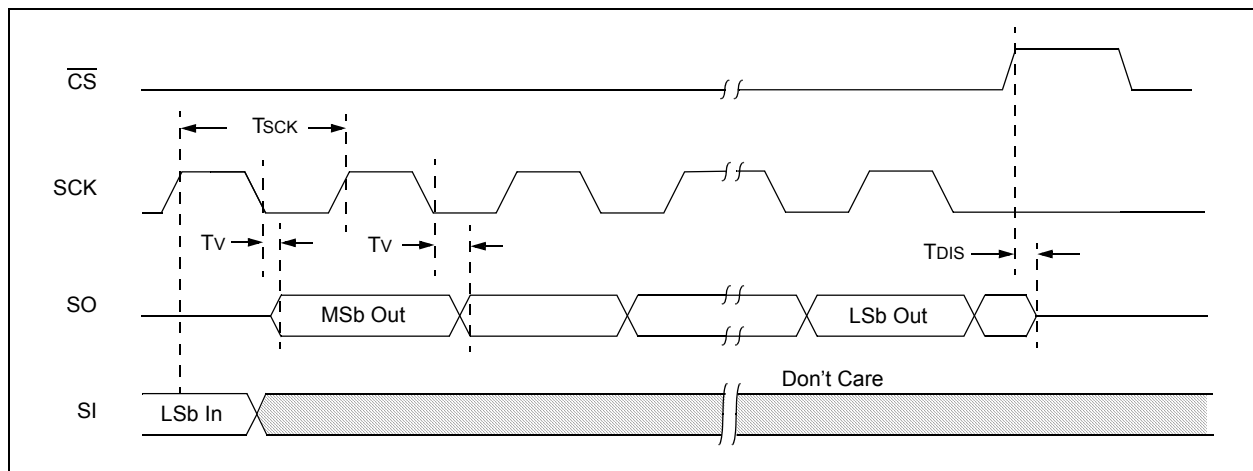

FIGURE 7-3: SPI OUTPUT TIMING

Preliminary

TABLE 7-5: **SPI INTERFACE AC CHARACTERISTICS**

Symbol	Parameters	Min.	Max.	Units
T_{SCK}	SCK Period	40	—	nS
T_{CSD}	$\overline{CS}$ High Time	50	—	nS
T_{CSS}	$\overline{CS}$ Setup Time	50	—	nS
T_{CSH}	$\overline{CS}$ Hold Time	50	—	nS
T_{SU}	SDI Setup Time	10	—	nS
T_{HD}	SDI Hold Time	10	—	nS
T_V	SDO Valid time	—	15	nS

NOTES:

8.0   PACKAGING INFORMATION

8.1   Package Marking Information

28-Lead SSOP

```
┌─────────────────────┐
│  XXXXXXXXXXXX       │
│  Ⓜ XXXXXXXX        │
│  ◯    YYWWNNN VVV  │
└─────────────────────┘
```

Example

```
┌─────────────────────┐
│  MCW1001A           │
│  Ⓜ I/SS            │
│  ◯    1010017 100  │
└─────────────────────┘
```

Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
		Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (ⓔ3) can be found on the outer packaging for this package.
	VVV	Revision
Note:		If the full Microchip part number cannot be marked on one line, it is carried over to the next line, thus limiting the number of available characters for customer-specific information.

8.2   Package Details

This section provides the technical details of the packages.

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

> **Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging
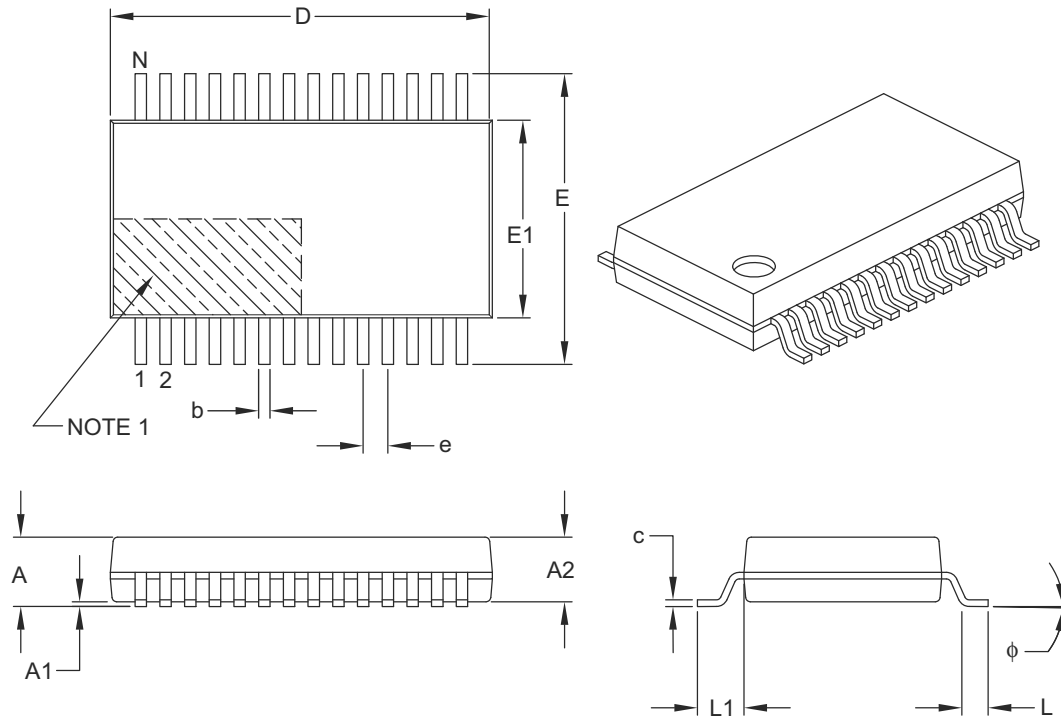
	Units		MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
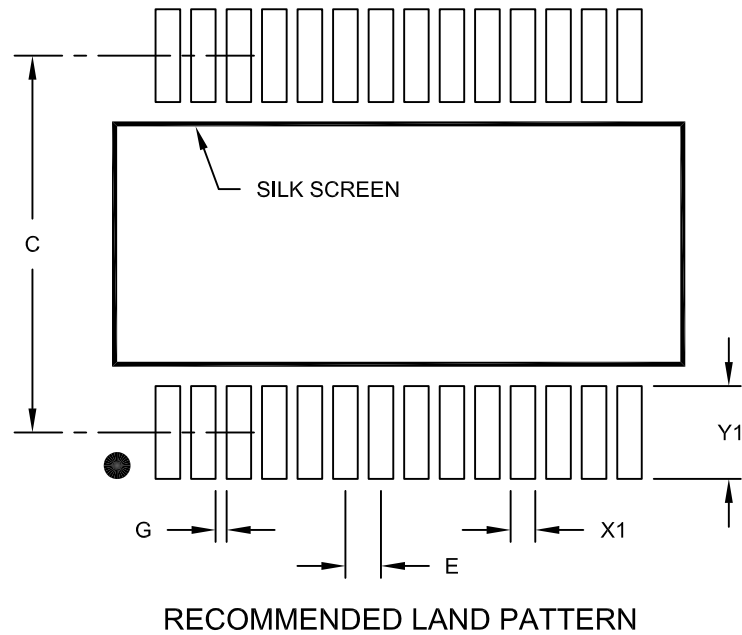3. Dimensioning and tolerancing per ASME Y14.5M.
    BSC: Basic Dimension. Theoretically exact value shown without tolerances.
    REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

Preliminary

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

RECOMMENDED LAND PATTERN

	Units		MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

NOTES:

Preliminary

APPENDIX A: REVISION HISTORY

Revision A (October 2011)

This is the initial release of the document.

NOTES:

Preliminary

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at 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 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
- Development Systems Information Line

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://microchip.com/support

MCW1001A

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager   Total Pages Sent _____

RE: Reader Response

From: Name _____

       Company _____

       Address _____

       City / State / ZIP / Country _____

       Telephone: (_____) _____ - _____   FAX: (_____) _____ - _____

Application (optional): _____

Would you like a reply?_____Y_____N

Device: MCW1001A   Literature Number: DS70671A

Questions:

1. What are the best features of this document?

    _____

    _____

2. How does this document meet your hardware and software development needs?

    _____

    _____

3. Do you find the organization of this document easy to follow? If not, why?

    _____

    _____

4. What additions to the document do you think would enhance the structure and subject?

    _____

    _____

5. What deletions from the document could be made without affecting the overall usefulness?

    _____

    _____

6. Is there any incorrect or misleading information (what and where)?

    _____

    _____

7. How would you improve this document?

    _____

    _____

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX
Device	Temperature Range	Package

Device	MCW1001A TCP/IP Socket to UART
Temperature Range	I = -40°C to +85°C (Industrial)
Package	SS = Plastic Shrink Small Outline - 5.30 mm body (SS) 28-Lead

Examples:

a) MCW1001A-I/SS100: = Industrial temperature, 28LD SSOP Package.

Preliminary

Worldwide Sales and Service

AMERICAS

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

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
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 - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
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: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
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

ASIA/PACIFIC

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: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

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: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/02/11

Preliminary

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Microchip](#):

[MCW1001A-I/SS](#)