Microchip MiWi™ P2P Wireless Protocol

Authors: Yifeng Yang
Pradeep Shamanna
Derrick Lattibeaudiere
Vivek Anchalia
Microchip Technology Inc.

INTRODUCTION

The demand is growing for most applications to move to wireless communication. The benefits are reduced costs and easy implementation. Wireless communication does not require cabling and other hardware, and the associated installation costs. It can also be implemented in locations where installing cable is difficult.

Since the IEEE released the Wireless Personal Area Network (WPAN) specification (IEEE 802.15.4™) in 2003 and onwards, it has become the real industry standard for low-rate WPANs (LR-WPAN). The specification applies to low-data rate applications with low-power and low-cost requirements.

The Microchip MiWi™ P2P Wireless Protocol is supported in MiWi Development Environment (DE). It is a variation of IEEE 802.15.4, using Microchip's IEEE 802.15.4 compliant and other proprietary RF transceivers which are controlled by Microchip 8-, 16-, or 32-bit microcontroller with a Serial Peripheral Interface (SPI). The Microchip MiWi P2P protocol stacks are now expanded beyond the IEEE 802.15.4 specification to support Microchip proprietary transceivers while using IEEE 802.15.4 Media Access Control (MAC) layer design as the reference. The MiWi P2P Wireless Protocol Stack supports both P2P and Star topologies.

The MiWi P2P protocol provides reliable direct wireless communication through a user friendly programming interface. It has a rich feature set that can be compiled in and out of the stack to meet a wide range of customer needs while minimizing the stack footprint.

This application note describes all about MiWi P2P Protocol and also its differences from IEEE 802.15.4. The document details the supported features, implementations and usage for wireless application developers.

It is recommended for the readers to go through the IEEE 802.15.4 specification and Microchip MiMAC/MiApp interfaces before reading this application note or working with the MiWi P2P and Star wireless protocols. For more information on MiMAC and MiApp, refer to the Application Notes "*AN1283 Microchip Wireless MiWi™ Media Access Controller – MiMAC*" (DS00001283) and "*AN1284 Microchip Wireless MiWi™ Application Programming Interface – MiApp*" (DS00001284).

Protocol Overview

The MiWi P2P protocol stack modifies the IEEE 802.15.4 specification MAC layer by adding commands that simplify the handshaking process. It simplifies link disconnection and channel hopping by providing supplementary MAC commands. However, application-specific decisions, such as when to perform an energy scan or when to jump channels, are not defined in the protocol. These issues are left to the application developer.

Protocol Features

The MiWi P2P Wireless Protocol has the following features:

- Supports Microchip PIC16, PIC18, PIC24, dsPIC33 and PIC32 platforms through Microchip XC8, XC16 and XC32 compilers, respectively
- Supports MRF24J40 (IEEE 802.15.4 compliant radio transceiver) and MRF89XA (proprietary radio transceiver) through Microchip Application Libraries
- Functions as a state machine (not RTOS-dependent)
- Supports a sleeping device at the end as a communication node
- Enables Energy Detect (ED) scanning to operate on the least-noisy channel
- Provides active scan for detecting new and existing connections
- Supports frequency agility (channel hopping)

Protocol Considerations

The MiWi P2P protocol is a variation of IEEE 802.15.4 and supports both peer-to-peer (P2P) and star topologies. It has no routing mechanism, hence the wireless communication coverage is defined by the radio range. The Guaranteed Time Slot (GTS) and Beacon networks by option are not supported, hence both the sides of the communication cannot simultaneously go to Sleep mode. If the application requires wireless networking and routing instead of P2P and Star type communication, MiWi Mesh is a suitable communication platform for proprietary standards. For details on MiWi Mesh, refer to the Application Note "*Microchip MiWi™ Mesh Wireless Networking Protocol*" from the Microchip website. However, for interoperability type of requirements with wireless devices or nodes of other vendors, ZigBee® protocol based on IEEE802.15.4 is a good option.

IEEE 802.15.4™ SPECIFICATION AND MiWi™ P2P WIRELESS PROTOCOL

Most of the products in the market use the original IEEE 802.15.4a specification, also known as IEEE 802.15.4-2003 or Revision A. In 2006, a revised edition was published to clarify a few issues. Referred to as IEEE 802.15.4b or 802.15.4-2006, the revision added two PHY layer definitions in the sub-GHz spectrum and modified the security module.

In this document, references to IEEE 802.15.4 means Revision A of the specification. The MiWi P2P protocol takes IEEE 802.15.4 specification as the design reference and expands the support from IEEE 802.15.4 compliant transceiver to Microchip proprietary transceivers.

Device Types

The MiWi P2P protocol categorizes devices based on their IEEE definitions and their role in making the communication connections as shown in Table 1 and Table 2. The MiWi P2P protocol supports all of these device types.

TABLE 1: IEEE 802.15.4™ DEVICE TYPES BASED ON FUNCTIONALITY

Functional Type	Power Source	Node Idle Mode Configuration	Data Reception Method
Full Function Device (FFD)	Wall Power/ Mains	On	Direct
Reduced Function Device (RFD)	Battery	Off	Poll from the associated device

TABLE 2: IEEE 802.15.4™ DEVICE TYPES BASED ON ROLE

Role Type	Functional Type	Role Description
Personal Area Network (PAN) Coordinator	FFD	The device starts first and waits for a connection.
End Device	FFD or RFD	The device starts after the PAN Coordinator has started to establish a connection.

Supported Topologies

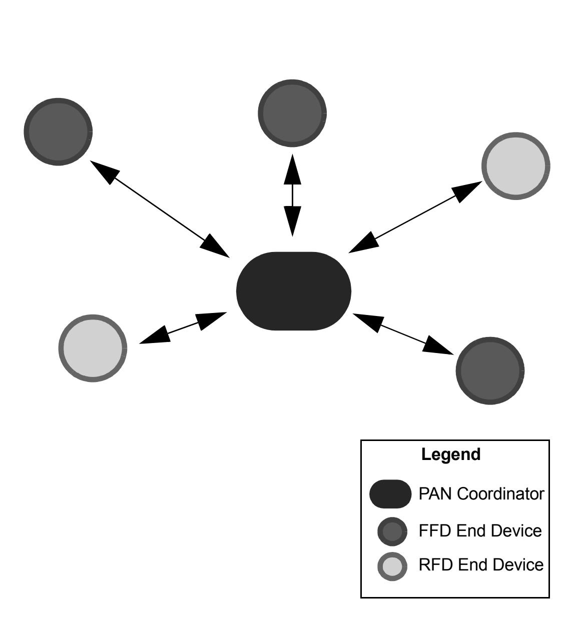
The IEEE 802.15.4 and the MiWi P2P protocol support two topologies: star and peer-to-peer.

STAR TOPOLOGY

A typical star topology is shown in Figure 1. From a device role perspective, the topology has one Personal Area Network (PAN) Coordinator that initiates communications and accepts connections from other devices. It has several end devices that join the communication. The end devices can establish connections only with the PAN Coordinator.

As to functionality type, the PAN Coordinator of the star topology is a Full Function Device (FFD). The end device can be an FFD with its radios ON all the time, or a Reduced Function Device (RFD) with its radio OFF when it is Idle. Regardless of its functional type, the end devices can only communicate to the PAN Coordinator.

FIGURE 1: STAR TOPOLOGY

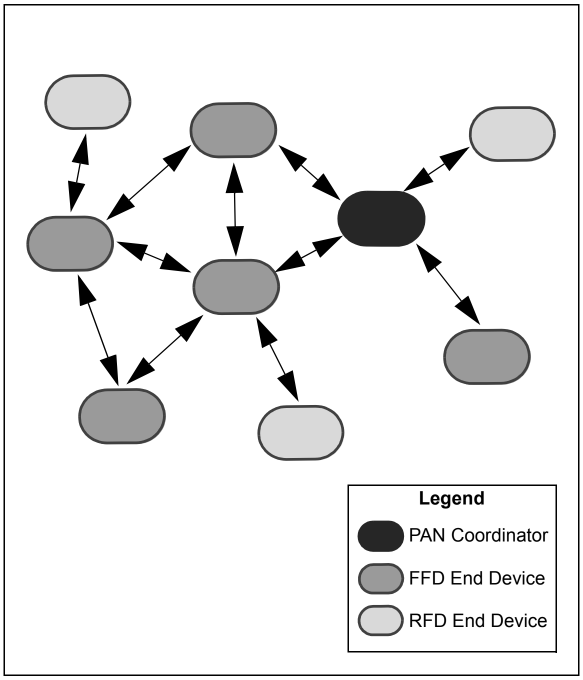


PEER-TO-PEER (P2P) TOPOLOGY

A typical P2P topology is shown in Figure 2. From a device role perspective, this topology also has one PAN Coordinator that starts communication from the end devices. When joining the network, however, end devices do not have to establish their connection with the PAN Coordinator.

As to functional types, the PAN Coordinator is an FFD and the end devices can be FFDs or RFDs. In this topology, however, end devices that are FFDs can have multiple connections. Each of the end device RFDs, however, can connect to only one FFD and cannot connect to another RFD.

FIGURE 2: PEER-TO-PEER TOPOLOGY



Network Types

The IEEE 802.15.4 specification has two types of networks: Beacon and Non-Beacon.

In a Beacon network, devices can transmit data only during their assigned time slot. The PAN Coordinator assigns the time slots periodically by sending a superframe (Beacon frame). All devices are supposed to synchronize with the Beacon frame and transmit data only during their assigned time slot. Beacon networks reduce the power consumption of all devices because every device periodically turns off its radio.

In a Non-Beacon network, any device can transmit data at any time when the energy level (noise) is below the predefined level. Non-Beacon networks increase the power consumption by FFD devices as radios are turned on all the time. These networks reduce the power consumption of RFD devices as the RFDs do not have to perform the frequent synchronizations.

The MiWi P2P protocol supports only Non-Beacon networks.

Network Addressing

The IEEE 802.15.4 specification defines two kinds of addressing mechanisms:

- Extended Organizationally Unique Identifier (EUI) or long address – an 8-byte address that is unique for each device, worldwide. The upper three bytes are purchased from IEEE by the company that releases the product. The lower five bytes are assigned by the device manufacturer as long as the EUI of each device is unique. The 8-byte unique address is usually called the MAC address of the wireless device/node and is predominantly associated with the node hardware.
- Short Address – a 2-byte address that is assigned to the device by its parent when it joins the network. The short address must be unique within the network.

The MiWi P2P protocol supports only one-hop communication, hence it transmits messages through EUI or long address. Short addressing is used only when the stack transmits a broadcast message as there is no predefined broadcast long address defined in the IEEE 802.15.4 specification.

For Microchip proprietary transceivers, the unique address length can be between 2 to 8 bytes, depending on the application needs.

Message Format for IEEE 802.15.4 Compliant Transceiver

The message format of the MiWi P2P protocol is a subset of the message format of the IEEE 802.15.4 specification. Figure 3 illustrates the packet format of the stack and its fields.

FRAME CONTROL

Figure 4 illustrates the format of the 2-byte Frame Control field.

The 3-bit Frame Type field defines the type of packet with the following values:

- Data frame = `001`
- Acknowledgement = `010`
- Command frame = `011`

The Security Enabled bit indicates if the current packet is encrypted. There is an additional security header if encryption is used. For more information, refer to **Section "Security Features"**.

The Frame Pending bit is used only in the Acknowledgement packet handled by the MRF24J40 radio hardware. This bit indicates if an additional packet follows the Acknowledgement after a data request packet is received from a RFD end device.

The Intra-PAN bit indicates if the message is within the current PAN. If this bit is set to '`1`', the Source PAN ID field in the addressing fields is omitted. In the stack, this bit is always set to '`1`', but it can be set to '`0`' to enable inter-PAN communication. Resetting the bit to '`0`' can be done in the application layer, if it is necessary.

The Destination Address mode can be either 16-bit Short Address mode = `10` or 64-bit Long Address mode = `11`

In the MiWi P2P protocol, the Destination Address mode is usually set to the Long Address mode. The Short Address mode is used only for a broadcast message. For broadcast messages, the Destination Address field in the addressing fields is fixed to `0xFFFF`.

The Source Address mode for the MiWi P2P protocol can only be the 64-bit Long Address mode.

SEQUENCE NUMBER

The sequence number is 8 bits long. It starts with a random number and increases by one each time a data or command packet is sent. The number is used in the Acknowledgement packet to identify the original packet. The sequence number of the original packet and the Acknowledgement packet must be the same.

FIGURE 3: MiWi™ P2P WIRELESS PROTOCOL PACKET FORMAT

Bytes	2	1	2	2/8	0/2	0/8	Variable	2
	Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Pay Load	Frame Check Sequence

Addressing Fields

FIGURE 4: FRAME CONTROL

Bits	3	1	1	1	1	3	2	2	2
	Frame Type	Security Enabled	Frame Pending	Acknowledgement Request	Intra PAN	(Reserved)	Destination Address Mode	(Reserved)	Source Address Mode

DESTINATION PAN ID

This is the PAN identifier for the destination device. If the PAN identifier is not known, or not required, the broadcast PAN identifier (`0xFFFF`) is used.

DESTINATION ADDRESS

The destination address can either be a 64-bit long address or a 16-bit short address. The destination address must be consistent with the Destination Address mode defined in the Frame Control field. If the 16-bit short address is used, it must be the broadcast address of `0xFFFF`.

SOURCE PAN ID

The source PAN identifier is the PAN identifier for the source device and must match the intra-PAN definition in the Frame Control field. The source PAN ID exists in the packet only if the intra-PAN value is '`0`'.

In the current MiWi P2P protocol implementation, all communication is intra-PAN. As a result, all packets do not have a Source PAN ID field.

However, the stack reserves the capability for the application layer to transmit the message inter-PAN. If a message needs to transmit inter-PAN, the source PAN ID is used.

SOURCE ADDRESS

The Source Address field is fixed to use the 64-bit extended address of the source device.

Message Format for Microchip Proprietary Transceiver

The message format for Microchip proprietary RF transceiver is defined in the MiMAC interface. For more information, refer to the Application Note *"AN1283 Microchip Wireless MiWi™ Media Access Controller – MiMAC"* (DS00001283).

Transmitting and Receiving

TRANSMITTING MESSAGES

There are two ways to transmit a message: Broadcast and Unicast.

Broadcast packets have all devices in the radio range as their destination. The IEEE 802.15.4 defines a specific short address as the broadcast address, but has no definition for the long address. As a result, for IEEE 802.15.4 compliant transceiver, broadcasting is the only situation when the MiWi P2P stack uses a short address.

There is no Acknowledgement for broadcasting messages. Unicast transmissions have only one destination and use the long address as the destination address. The MiWi P2P protocol requires Acknowledgement for all unicast messages.

If the transmitting device has at least one device that turns off its radio when Idle, the transmitting device saves the message in RAM and wait for the sleeping device to wake-up and request the message. This kind of data transmitting is called *Indirect Messaging*.

If the sleeping device fails to acquire the indirect message, it expires and becomes discarded. Usually, the indirect message time-out needs to be longer than the pulling interval for the sleeping device.

RECEIVING MESSAGES

In the MiWi P2P protocol, only the messaged device is notified by the radio. If the messaged device turns off its radio when Idle, it can only receive a message from the device to which it is connected.

For the idling device with the turned off radio to receive the message, the device must send a data request command to its connection peer. Then, it acquires the indirect message if there is one.

In star topology, only the PAN Coordinator is enabled for connections and End Devices (FFD/RFD) are all connected to the PAN Coordinator. Hence, the End Devices in star topology have single connections.

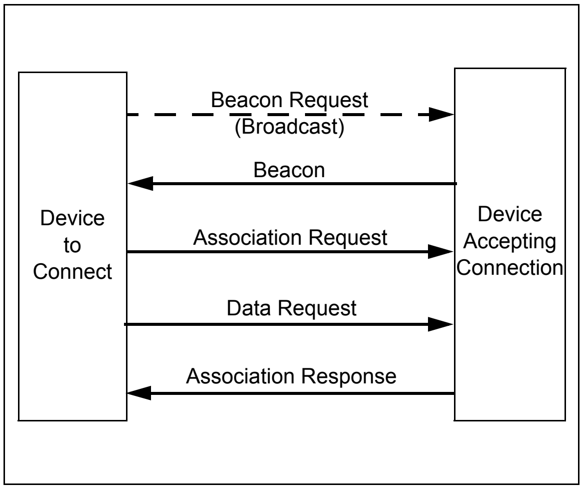
VARIATIONS FOR HANDSHAKING

The major difference of the MiWi P2P Wireless Protocol from the IEEE 802.15.4 specification is in the process of handshaking. Under IEEE 802.15.4, the first step of the device after powering up is to do a handshake with the rest of the world.

Figure 5 shows the handshaking process of the IEEE 802.15.4 specification and described as follows:

1. The device that is seeking to communicate sends out a Beacon request.
2. All devices capable of connecting to other devices responds with a Beacon message.
3. The initiating device collects all of the beacons. (To accommodate multiple responses, the device waits until the active scan requests a time-out). The device determines which beacon to use to establish the handshake and sends out an association request command.
4. After a predefined time, the initiating device issues a data request command to get the association response from the other side of the intended connection.
5. The device on the other side of the connection sends the association response.

FIGURE 5: TYPICAL HANDSHAKING IN IEEE 802.15.4™



Handshaking is the elaborate process of joining a network. A device can join only a single device as its parent and hence, the initial handshaking is the actual process of choosing a parent.

Choosing the parent requires the following steps:

1. Listing all the possible parents.
2. Choosing the right one as its parent.

The Beacon frames do not use CSMA-CA detection before transmitting to meet the timing requirement of the active scan time-out. As a result, the Beacon frames may be discarded due to a packet collision.

The MiWi P2P protocol is designed for simplicity and direct connections in star and P2P communication topologies. Some IEEE 802.15.4 requirements obstruct that design:

- The five-step handshaking process, plus two time-outs, requires a more complex stack.
- The association process uses one-connection communication rather than the multi-connection concept of peer-to-peer topology.

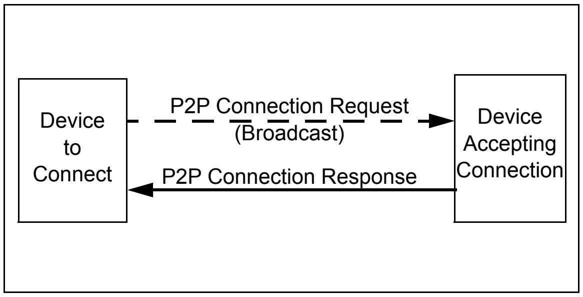
For the preceding reasons, the MiWi P2P protocol uses its own two-step handshaking process as shown in Figure 6:

1. The initiating device sends out a P2P connection request command.
2. Any device within radio range responds with a P2P connection response command that finalizes the connection.

This is a one-to-many process that may establish multiple connections, where possible, to establish a peer-to-peer topology. Since this handshaking process uses a MAC layer command, CSMA-CA is applied for each transmission. This reduces the likelihood of packet collision.

RFDs may receive the Connection Request command from several FFDs, but can connect to only one FFD. An RFD chooses the FFD, from which it receives the first P2P connection response as its peer.

FIGURE 6: HANDSHAKING PROCESS FOR MIWI™ P2P WIRELESS PROTOCOL



Custom MAC Commands for MiWi P2P Wireless Protocol

The MiWi P2P protocol extends the functionality of the IEEE 802.15.4 specification by using custom MAC commands for removing the connection between two devices. Table 3 lists all of the custom MAC commands of the protocol.

P2P CONNECTION REQUEST

The P2P connection request (0x81) is broadcasted to establish a P2P connection with other devices after powering up. The request can also be unicast to a specific device to establish a single connection.

When the transmitting device receives a P2P connection response (0x91) from the other end, a P2P connection is established.

The P2P connection request custom command can also start an active scan to determine what devices are available in the neighborhood.

When a P2P connection request command is sent for active scan purposes, the capability information and optional payload is not attached. The receiving device uses the attachment, or absence of capability information, and an optional payload to determine if the command is a request to establish a connection or just an active scan.

The MiWi P2P protocol can enable or disable a device to allow other devices to establish connections. After a device is disabled from making connections, any new P2P connection request is discarded, except under the following conditions:

- The P2P connection request is coming from a device in which the receiving end established a connection.
- The P2P connection request is an active scan.

Figure 7 shows the format of the P2P connection request command frame.

TABLE 3: CUSTOM MAC COMMANDS FOR MIWI™ P2P WIRELESS PROTOCOL

Command Identifier	Command Name	Description
0x81	P2P Connection Request	Request to establish a P2P connection. Usually broadcast to seek P2P connection after powering up. Alternately, unicast to seek an individual connection.
0x82	P2P Connection Removal Request	Removes the P2P connection with the other end device.
0x83	P2P Data Request	Similar to the IEEE 802.15.4™ specification Data Request command (0x04), a request for data from the other end of a P2P connection if the local node had its radio turned off. Reserved for the previously sleeping device to request the other node to send the missed message (indirect messaging).
0x84	Channel Hopping	Request to change operating channel to a different channel. Usually used in the feature of frequency agility.
0x87	Active Scan Request	Checks available nodes in the current and accessible channels.
0x91	P2P Connection Response	Response to the P2P connection request. Also can be used in active scan process.
0x92	P2P Connection Removal Response	Response to the P2P connection removal request.
0x97	Active Scan Response	Response returns the node information including the Channel, PAN ID and Node ID as populates and index table.

Note: See **Section "Active Scan"** for details on Active Scan Request and Active Scan Response.

FIGURE 7: P2P CONNECTION REQUEST COMMAND FORMAT

Octets	15/21	1	1	1 (Optional)	Various (Optional)
	MAC Header	Command Identifier (0x81)	Operating Channel	Capability Information	Optional payload to identify the node. It is not required for the stack, but may be useful for applications.

The operating channel is used to bypass the effect of subharmonics that may come from another channel. It avoids the false connections with devices that operate on different channels. The capability information byte, as shown in Figure 7 uses a format as illustrated in Figure 8.

The optional payload of the P2P connection request is provided for specific applications. A device may need additional information to identify itself, either its unique identifier or information about its capabilities in the application. With the optional payload, no additional packets are required to introduce or identify the device after the connection is established. The optional payload is not used in the stack.

P2P CONNECTION REMOVAL REQUEST

The P2P connection removal request (0x82) is sent to the other end of the connection to remove the P2P connection. Figure 9 shows the format of the request.

DATA REQUEST

The data request (0x83) command is the same as the data request (0x04) command of the IEEE 802.15.4 specification. Figure 10 shows the format of the request.

If one side of a P2P connection node is able to Sleep when Idle, and that node can receive a message while in Sleep, the active side of the connection must store the message in its RAM. The active side delivers the message when the sleeping device wakes up and requests the message.

If an application involves such conditions, the `ENABLE_INDIRECT_MESSAGE` feature needs to be activated. The sleeping node must send the data request command after it wakes up.

FIGURE 8: CAPABILITY INFORMATION FORMAT

Bits	0	1	2	3	4-7
	Receiver ON when Idle	Request Data on Wake-up	Need Time Synchronization (Reserved)	Security Capable	(Reserved)

FIGURE 9: P2P CONNECTION REMOVAL REQUEST FORMAT

Octets	15/21	1
	MAC Header: Send to the other end of the P2P connection to cut the communication	Command Identifier (0x82)

FIGURE 10: DATA REQUEST FORMAT

Octets	21	1
	MAC Header: Unicast from extended source address to extended destination address	Command Identifier (0x83 or 0x04)

CHANNEL HOPPING

The channel hopping command (0x84) requests the destination device to change the operating channel to another channel. Figure 11 shows the format of the command.

This command is usually sent by the frequency agility initiator which determines when to change channels and what channel to select.

This command is usually broadcasted to notify all devices, with their radios ON when Idle, to switch channels. To ensure that every device receives this message, the frequency agility initiator performs the broadcast three times and all the FFD devices performs the rebroadcast.

When the channel hopping sequence is carried out and all FFDs hop to a new channel, RFDs have to perform resynchronization to restore connection to their respective FFD peers.

P2P CONNECTION RESPONSE

The P2P connection response (0x91) command is used to respond to the P2P connection request. Figure 12 shows the format of the command.

The P2P connection response command can be used to establish a connection. Alternately, the command can be used by a device responding to an active scan, identifying itself as active in the neighborhood.

If the P2P connection request command received had a capability information byte and an optional payload attached, it is requesting a connection. The capability information and optional payload, if any, is attached to the P2P connection response.

Once the response is received by the other end of the connection, a P2P connection is established. Now, the two ends of the connection now can exchange packets.

If the P2P connection request command received do not have a capability information byte and optional payload, the command is an active scan. The P2P connection response, therefore, do not have any capability information or optional payload attached.

In the case of the active scan connection request, no connection is established after the message exchange.

The format of the capability information response is shown in Figure 8.

The optional payload is provided for specific applications. Its format and usage is the same as the optional payload attached to P2P connection request command (see Figure 9).

P2P CONNECTION REMOVAL RESPONSE

The P2P connection removal response command (0x92) is used to respond to the P2P connection removal request. It notifies the other end of the P2P connection that a P2P connection request is received early and the resulting connection is removed. Figure 13 shows the format of the command.

FIGURE 11: CHANNEL HOPPING FORMAT

Octets	15/21	1	1	1
	MAC Header: Broadcast or unicast from the Frequency Agility Starter	Command Identifier (0x84)	Current Operating Channel	Destination Channel to Jump to

FIGURE 12: P2P CONNECTION RESPONSE FORMAT

Octets	21	1	1	1 (Optional)	Various (Optional)
	MAC Header: Unicast from extended source address to extended destination address.	Command Identifier (0x91)	Status. 0x00 means successful. All other values are error codes.	Capability Information	Optional payload to identify the node. Not required for the stack, but possibly useful for applications.

FIGURE 13: P2P CONNECTION REMOVAL RESPONSE FORMAT

Octets	21	1	1
	MAC Header: Unicast from extended source address to extended destination address	Command Identifier (0x92)	Status. • 0x00 means successful. • All other values are error codes

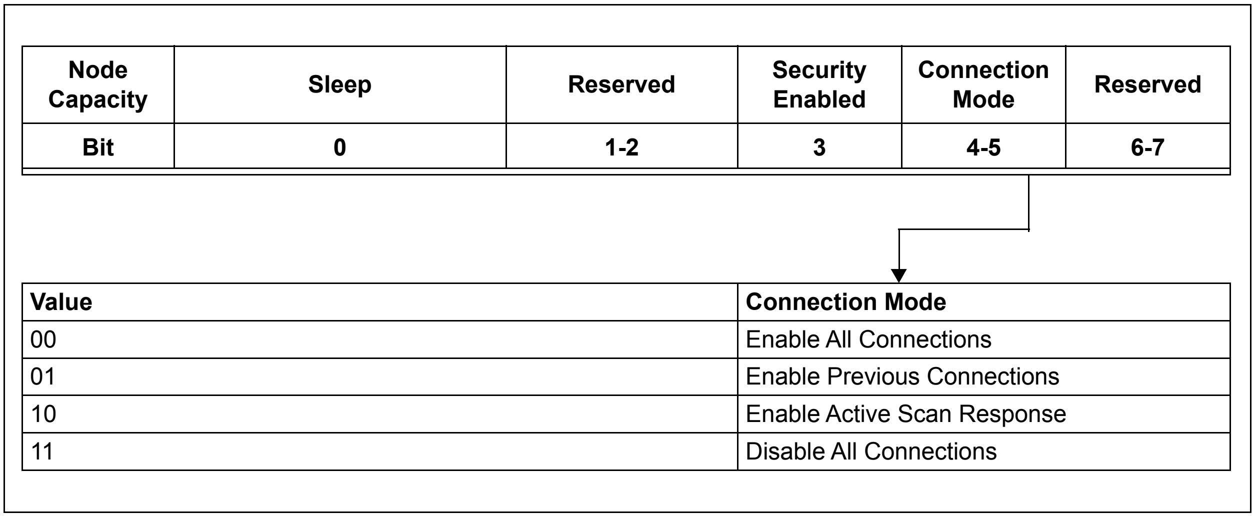
The MiWi Star protocol extends the functionality of the IEEE 802.15.4 specification by using custom MAC commands for removing the connection between two devices. Table 4 lists all of the custom MAC commands of the protocol.

Figure 14 shows the modified connection mode details in the star protocol.

TABLE 4: CUSTOM MAC COMMANDS FOR MIWI™ STAR WIRELESS PROTOCOL

Command Identifier	Command Name	Description
0xCC	Forward Packet CMD with Payload	0XCC (1 byte) Command. Destination End Device Address (3 bytes). Data Payload.
0xDA	Software ACK to END Device	N/A
0x7A	LINK STATUS	N/A
0x77	Connection Table Broadcast Command	0x77 (1 byte) Command. Total number of End Devices in the Network.

FIGURE 14: MODIFIED CONNECTION MODE DETAILS IN STAR PROTOCOL



Node Capacity	Sleep	Reserved	Security Enabled	Connection Mode	Reserved
Bit	0	1-2	3	4-5	6-7

Value	Connection Mode
00	Enable All Connections
01	Enable Previous Connections
10	Enable Active Scan Response
11	Disable All Connections

UNIQUE FEATURES OF THE MiWi P2P WIRELESS PROTOCOL

The MiWi P2P protocol supports a reduced functionality, point-to-point, direct connection and a rich set of features. All features can be enabled or disabled and compiled in and out of the stack according to the needs of the wireless application.

This section describes the unique features of the MiWi P2P protocol as follows:

- Small programming size
- Support for Idle devices to turn off radio
- Indirect messaging
- Special security features
- Active scan for finding existing PANs on different channels
- Energy scan for finding the channel with the least noise
- Frequency agility (channel hopping)
- Messaging Types

Small Programming Size

To address many cost constraints of the wireless applications, the MiWi P2P protocol is small in size. Enabling the stack to target the smallest programming size can reduce the code size to over 3 Kbytes. A simple application can easily fit into a microcontroller with only around 4 Kbytes of programming memory.

To activate this feature, "`TARGET_SMALL`" must be defined in the file, `miwi_config.h`.

The feature supports bidirectional communication between devices, but communication between PANs is disabled. If the security feature is used, the freshness check is disabled. For more information on freshness check, refer to **Section "Security Features"**.

Idle Devices Turning Off Radios

For devices operating on batteries, reducing power consumption is essential. This can be done by having the devices turn off its radios when not transmitting data. The MiWi P2P protocol includes features for putting radios into Sleep mode and then waking up the device.

To activate this feature, "`ENABLE_SLEEP`" must be defined in the file, `miwi_config.h`.

To determine as to when a device is put into Sleep mode is identified by the specific application. The possible triggers include:

- Length of radio Idle time
- Receipt of a packet from a connected FFD, requesting the device to go to Sleep mode

The conditions for awakening a device can be determined by the specific application. Possible triggers include:

- An external event like a button is pressed
- Expiration of a predefined timer

While a device is sleeping, its peer device may need to send a message. If no message is sent, no additional feature must be enabled by the peer device.

If the peer device sends a message to the sleeping device, the peer device must store the message in its volatile memory until the sleeping device wakes up and acquires the message. Since the message is not directly delivered to the sleeping device, this process is called an *Indirect Message*.

If an indirect message is delivered, the peer device of the sleeping node must define "`ENABLE_INDIRECT_MESSAGE`" in the, `miwi_config.h` file.

If indirect messaging is enabled, there must be a specified maximum number of indirect messages that can be stored in the volatile memory. The maximum size of the message depends on the free RAM memory available in the peer device and from the number of RFDs connected to the same parent FFD.

The maximum number of indirect messages is defined by the "`INDIRECT_MESSAGE_SIZE`" in the, `miwi_config_p2p.h` file. For indirect messaging, the time-out period for the indirect messages also needs to be defined. If a time-out period is not defined and an RFD device is inactive or not visible, the indirect message remains forever in the volatile memory.

The indirect message time-out period is defined by the "`INDIRECT_MESSAGE_TIMEOUT`" in the, `miwi_config_p2p.h` file, with seconds as the unit of measurement.

Security Features

MiWi P2P protocol has the security feature handled in MiMAC layer. For more information, refer to the Microchip Application Note "*AN1283 Microchip Wireless MiWi™ Media Access Controller - MiMAC*" (DS00001283).

Active Scan

Active scan is the process of acquiring information about the local PAN. The active scan determines:

- The device's operating channel
- The device's signal strength in the PAN
- The PAN's identifier code for IEEE 802.15.4 compliant transceiver

Active scan is particularly useful if there is no predefined channel or PAN ID for the local devices.

The maximum number of PANs that an active scan can acquire is defined, in the stack, as `ACTIVE_SCAN_RESULT_SIZE`.

The scan duration and channels to be scanned are determined before the active scan begins.

The scan duration is defined by the IEEE 802.15.4 specification and its length of time, measured in symbols, is calculated with the formula shown in Equation 1 (One second equals 62,500 symbols.).

EQUATION 1: SCAN DURATION

$$\text{Scan Time Period} \equiv 960 \bullet (2^{\text{ScanDuration}} + 1)$$

Note: ScanDuration = The user-designated input parameter for the scan. An integer is from 1 to 14.

A scan duration of 10 results in a scan time period of 61,500 symbols or about 1 second. A scan duration of 9 is about half second.

The scan channels are defined by a bitmap with each channel number represented by its comparable bit number in the double word. Channel 11 is `b'0000 0000 0000 0000 0000 1000 0000 0000`. Channels 11 to 26, supported in the 2.4 GHz spectrum, is `b'0000 0111 1111 1111 1111 1000 0000 000` or 0x07FFF800.

When an active scan broadcasts a P2P connection request command, it expects any device in radio range to answer with a P2P connection response command. The active scan determines only what PANs are available in the neighborhood, not how many individual devices are available for new connections. Every device responds to the scan, including those devices that do not allow new connections.

To invoke the active scan feature, "`ENABLE_ACTIVE_SCAN`" must be defined in the, `miwi_config.h` file.

Energy Scan

On each frequency band, there may have multiple channels, but a PAN must operate on one. The best channel to use is the one with the least amount of energy or noise.

Energy scan is used to scan all available channels and determine the channel with the least noise.

The scan duration and channels to be scanned are determined before the energy scan is performed.

The scan duration is defined by the IEEE 802.15.4 specification and its length of time, measured in symbols, is calculated with the formula as shown in Equation 1. For more information on measurement, see **Active Scan**.

After the scan is complete, the channel identifier with the least noise is returned. To activate the Energy Scan feature, "`ENABLE_ED_SCAN`" must be defined in the `miwi_config.h` file.

Messaging Type

Both broadcasting and unicast messages are supported by the MiWi P2P Protocol stack.

Broadcasting may be useful for some applications, but it requires more effort for peer devices. When a peer device can broadcast a message to an RFD, the "`ENABLE_BROADCAST`" must be defined in the `miwi_config.h` file.

Unicast Messages are sent using unicast function calls available in the `miwi_api.h` file.

Frequency Agility

Frequency agility enables the MiWi P2P Protocol PAN to move to a different channel if required by the operating conditions.

In implementing this feature, the affected devices fall into one of these two roles:

- Frequency Agility Initiators – these are devices that determine whether channel hopping is necessary and which new channel is applicable to use.
- Frequency Agility Followers – these are devices that change to another channel when directed.

FREQUENCY AGILITY INITIATORS

Each PAN can have one or more devices as a frequency agility initiator. An initiator must be an FFD.

Each initiator must have the energy scanning feature enabled to determine the optimal channel for the hop. The initiator broadcasts a channel hopping command to the other devices on the PAN.

FREQUENCY AGILITY FOLLOWERS

A frequency agility follower can be an FFD or an RFD device.

The FFD makes the channel hop by performing one of the following steps:

- Receiving the channel hopping command from the initiator.
- Resynchronizing the connection, if data transmissions continuously fail.

An RFD device makes the message hop using the resynchronization method, that reconnects to the PAN when communication fails.

ENABLE FREQUENCY AGILITY FEATURE

The application determines when to perform a frequency agility operation. Frequency agility is usually triggered by continuous transmission failure, either by CCA failure or no Acknowledgement received.

To activate the frequency agility feature, the "`ENABLE_FREQUENCY_AGILITY`" must be defined in the `miwi_config.h` file.

MiWi STAR OVERVIEW

MiWi Star protocol is an extension of the MiWi P2P protocol which is defined by Microchip. From a device role perspective, the topology has one PAN Coordinator that initiates communications and accepts connections from other devices. It can have several end devices that join the communication. End devices can establish connections only with the PAN Coordinator. As to functionality type, the star topology's PAN Coordinator is a Full Function Device (FFD). An end device can be an FFD with its radios ON all the time, or a Reduced Function Device (RFD) with its radio OFF when Idle. Regardless of its functional type, the end devices can only communicate to the PAN Coordinator.

Unique Features of the MiWi Star Wireless Protocol

The star topology supported by the MiWi P2P Protocol stack provides all the features supported by the peer-to-peer topology, however, star topology supports several more features based on the device roles.

PAN Coordinator supports the following features:

- Shares peer device connection (FFDs and RFDs) information to all the peer devices
- Forwards data packet from one end device to another end device
- Checks network health periodically (optional)
- Transmits data packet to End Devices
- Handles Indirect Messages for Sleeping End Devices (RFDs)
- Supports software ACK to indicate successful data transmission

The FFDs (End Devices) or RFDs (Sleeping End Devices) support the following features:

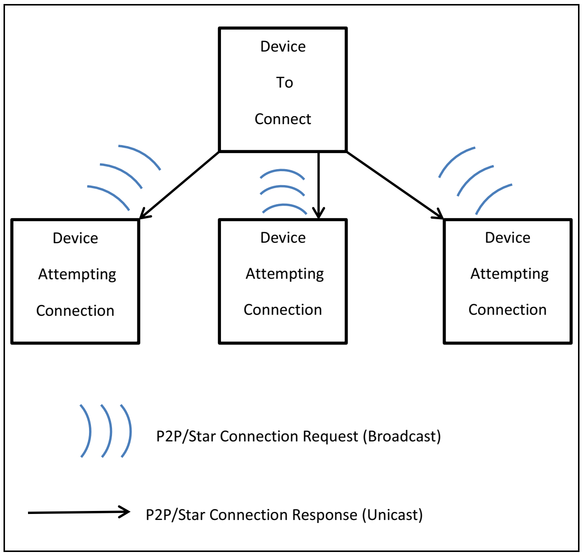
- Link Status
- Leave Network command

HANDSHAKING IN MiWi STAR WIRELESS PROTOCOL

MiWi Star Routing

Figure 15 shows that a MiWi Star network consists of two types of devices (PAN Coordinator, FFDs or RFDs). PAN Coordinator creates the network while the End Devices (FFDs or RFDs) join the PAN Coordinator. The PAN Coordinator can send messages to all the End devices in the network in a single hop. If an end device wants to communicate to another end device which may or may not be in the vicinity, the source end device must first send the packet to the PAN Coordinator and then the PAN Coordinator forwards that packet to the destination end device (2 hops).

FIGURE 15: MiWi™ STAR ROUTING

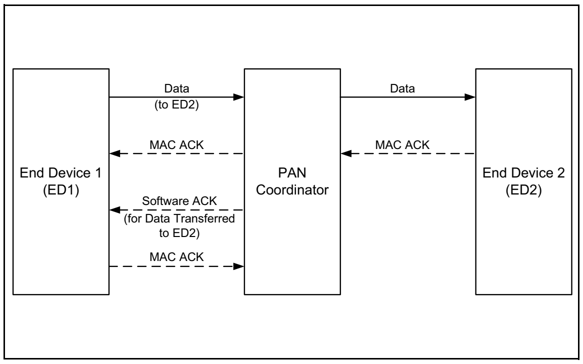


In a MiWi Star network, it is the responsibility of the PAN Coordinator to share the peer connections (End Device Addresses). In this way, all the end devices in the network know about the existence of every other device in network. When an end device wants to send a message to another end device, the source end device includes the address of the destination end device in the data payload. The source end device payload comprises of the type of packet (0xCC), Destination End Device Address (only first 3 bytes) and the Data Payload. When this packet is received by the PAN Coordinator, it indicates that this packet is intended for another end device, hence, it forwards the packet to the destination end device.

MiWi Star Data Transfer

The connection requests and responses are similar to that of P2P between the nodes. However in MiWi Star, the PAN Coordinator forms the network, connects the End Devices and also supports the End Devices to communicate between each devices (via PAN Cooordinator). Figure 16 shows a simple data transfer between the End Devices in MiWi Star network.

FIGURE 16: DATA TRANSFER BETWEEN END DEVICES IN MiWi™ STAR NETWORK



APPLICATION PROGRAMMING INTERFACES (APIs)

MiWi P2P protocol uses MiApp as its application programming interface. For more information on MiApp interface, refer to the Application Note "*AN1284* "*Microchip Wireless MiWi™ Application Programming Interface – MiApp*" (DS00001284).

APPLICATION FLOWCHART

A typical MiWi P2P protocol application starts by initializing the hardware and MiWi P2P protocol and then it tries to establish a connection and enter the normal operation mode of receiving and transmitting data. Figure 17 illustrates the typical flow of the MiWi P2P protocol applications.

After a connection is established, the procedures for most MiWi P2P protocol applications remain the same. Due to different stack configuration, variation takes place during the establishment of the connections.

Figure 18 shows the simplest P2P connection application for establishing connections.

FIGURE 17: FLOWCHART FOR MiWi™ P2P WIRELESS PROTOCOL APPLICATIONS

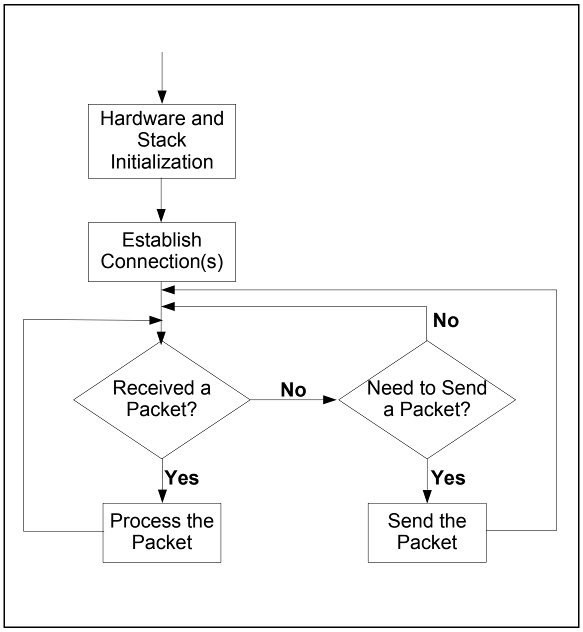
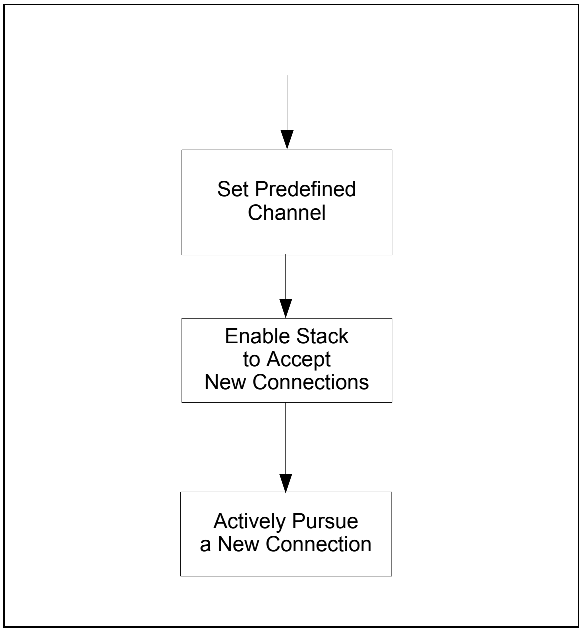


FIGURE 18: FLOWCHART TO ESTABLISH CONNECTIONS IN SIMPLE MODE


Set Predefined Channel
Enable Stack to Accept New Connections
Actively Pursue a New Connection

The complex applications require active scan capability. The steps for the active scan to establish connections differ between the PAN Coordinator and end devices. Figure 19 illustrates how to establish connections when active scan is enabled for both categories of devices.

For applications with energy scan enabled, the steps after connection also differs for the PAN Coordinator and end devices. Figure 20 shows how to establish connections when energy scan is enabled.

Figure 21 illustrates the process for establishing connections when active scan and energy scan are both enabled.

FIGURE 19: FLOWCHART TO ESTABLISH CONNECTIONS WHEN ACTIVE SCAN IS ENABLED

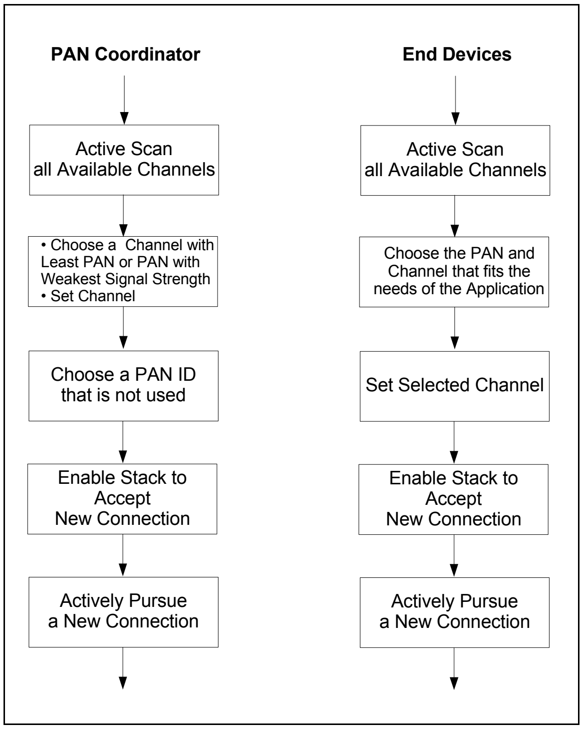

PAN Coordinator
Active Scan all Available Channels
• Choose a Channel with Least PAN or PAN with Weakest Signal Strength
• Set Channel
Choose a PAN ID that is not used
Enable Stack to Accept New Connection
Actively Pursue a New Connection
End Devices
Active Scan all Available Channels
Choose the PAN and Channel that fits the needs of the Application
Set Selected Channel
Enable Stack to Accept New Connection
Actively Pursue a New Connection

FIGURE 20: FLOWCHART TO ESTABLISH CONNECTIONS WHEN ENERGY SCAN IS ENABLED

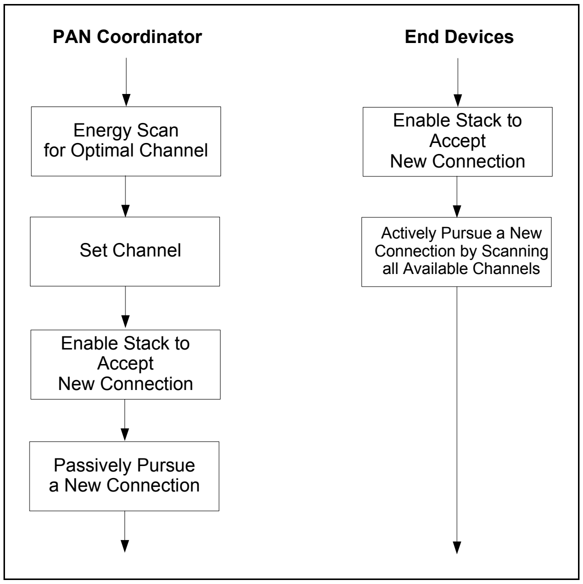
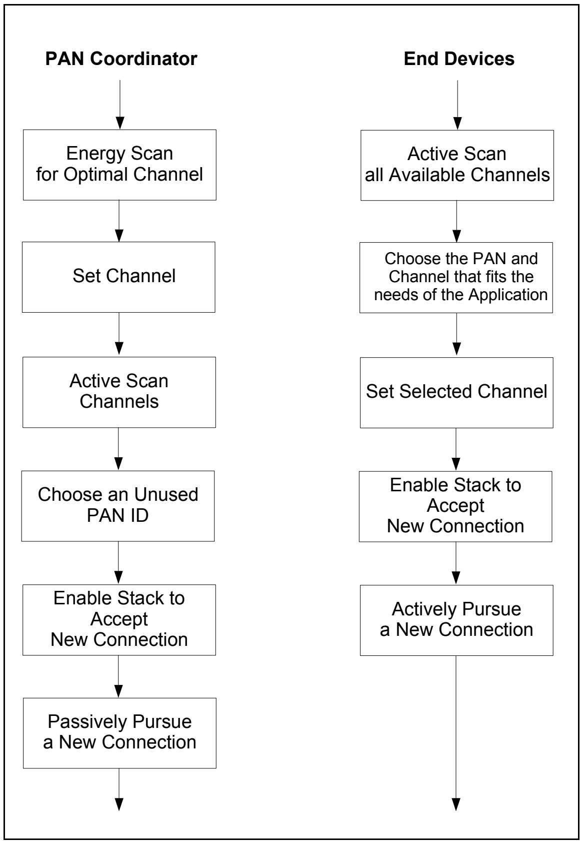

PAN Coordinator
Energy Scan for Optimal Channel
Set Channel
Enable Stack to Accept New Connection
Passively Pursue a New Connection
End Devices
Enable Stack to Accept New Connection
Actively Pursue a New Connection by Scanning all Available Channels

FIGURE 21: FLOWCHART TO ESTABLISH CONNECTIONS WITH ACTIVE AND ENERGY SCAN


PAN Coordinator
Energy Scan for Optimal Channel
Set Channel
Active Scan Channels
Choose an Unused PAN ID
Enable Stack to Accept New Connection
Passively Pursue a New Connection
End Devices
Active Scan all Available Channels
Choose the PAN and Channel that fits the needs of the Application
Set Selected Channel
Enable Stack to Accept New Connection
Actively Pursue a New Connection

SYSTEM RESOURCES REQUIREMENT

The MiWi P2P Wireless Protocol has a rich set of features. Enabling a feature set increases the system requirements for the microcontrollers.

The MiWi P2P and MiWi Star Protocol Stack are part of the Microchip Libraries for Applications (MLA). The protocol stack is available for download from the Microchip website at http://www.microchip.com/mla.

TABLE 5: PIC18 MEMORY REQUIREMENTS FOR MiWi™ P2P WIRELESS PROTOCOL

Configuration	Program Memory (Bytes)	RAM (Bytes)
Target Small Stack Size	< 4K	100 + RX Buffer Size + TX Buffer Size + (9 * P2P Connection Size)

TABLE 6: PIC18 MEMORY REQUIREMENTS FOR MiWi™ P2P WIRELESS PROTOCOL FEATURES[1]

Configuration	Additional Program Memory (Bytes)	Additional RAM (Bytes)
Enable Intra-PAN Communication	462	0
Enable Sleep	186	0
Enable Security (Without Frame Freshness Checking)	500	48
Enable Security (With Frame Freshness Checking)	1,488	54
Enable Active Scan	1,070	69
Enable Energy Scan	752	0
Enable Indirect Message	950	Indirect Message Size * TX Buffer Size
Enable Indirect Message with Capability of Broadcasting	1,228	Indirect Message Size * TX Buffer Size

Note 1: These requirements are for the PIC18 family of microcontrollers. The stack is also capable of supporting PIC16, PIC24, dsPIC33 and PIC32 microcontrollers, but requirements of the devices may vary. These requirements are for the initial release of the stack and are subject to change. Refer to the latest stack release notes for the specific device or platform support.

Table 5 gives the requirements of a basic P2P configuration. Additional MiWi P2P protocol features require more program memory and RAM. Table 6 lists the system requirements for features above a basic configuration.

Table 7 gives the requirements of a basic Star Configuration. Additional MiWi Star protocol features require more program memory and RAM. Table 8 lists the system requirements for features above a basic configuration.

TABLE 7: PIC18 MEMORY REQUIREMENTS FOR MIWI™ STAR WIRELESS PROTOCOL BASIC STACK

Configuration	Program Memory (Bytes)	RAM (Bytes)
Target Small Stack Size	< 6K	100 + RX Buffer Size + TX Buffer Size + (9 * STAR Connection Size) + (4 * STAR Connection Size) + 50 bytes (Global Variables)

TABLE 8: PIC18 MEMORY REQUIREMENTS FOR MIWI™ STAR WIRELESS PROTOCOL STACK FEATURES[1]

Configuration	Additional Program Memory (Bytes)	Additional RAM (Bytes)
Enable Sleep	186	0
Enable Security	2,070	54
Enable Periodic Connection Table Share	300	6
Enable Link Status	702	22
Enable Active Scan	1,070	69
Enable Energy Scan	400	0
Enable Indirect Message	950	Indirect Message Size * TX Buffer Size
Enable Indirect Message with Capability of Broadcasting	1,228	Indirect Message Size * TX Buffer Size

Note 1: These requirements are for the PIC18 family of microcontrollers. The stack is also capable of supporting PIC24, dsPIC33 and PIC32 microcontrollers, but requirements of the devices may vary. These requirements are for the initial release of the stack and are subject to change. Refer to the latest stack release notes for the specific device or platform support.

CONCLUSION

For wireless applications that require a star or peer-to-peer topology, the MiWi™ P2P Wireless Protocol is a good solution. The stack provides all the benefits of the IEEE 802.15.4 specification with a simple yet robust solution for both P2P and Star type of network communication. If an application is more complex, the Microchip MiWi Mesh Wireless Protocol stack must be considered as it provides support for a real network with more active nodes and message hops.

REFERENCES

- "*Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)",* IEEE Std 802.15.4™-2003,New York: IEEE, 2003.
- IEEE Std 802.15.4™-2006, (Revision of IEEE Std 802.15.4-2003). New York: IEEE, 2006.
- "*AN1283 Microchip Wireless MiWi™ Media Access Control – MiMAC"* (DS00001283), Yifeng Yang, Pradeep Shamanna, Derrick Lattibeaudiere, and Vivek Anchalia, Microchip Technology Inc., 2009-2017.
- "*AN1284 Microchip Wireless MiWi™ Application Programming Interface – MiApp"* (DS00001284), Yifeng Yang, Pradeep Shamanna, Derrick Lattibeaudiere, and Vivek Anchalia, Microchip Technology Inc., 2009-2017.
- "*AN1066 MiWi™ Wireless Networking Protocol Stack"* (DS00001066), David Flowers and Yifeng Yang, Microchip Technology Inc., 2007-2010.
- "*AN1232 Microchip ZigBee-2006 Residential Stack Protocol*" (DS00001232), Derrick Lattibeaudiere, Microchip Technology Inc., 2007.
- "*AN1255 Microchip ZigBee PRO Feature Set Protocol Stack"* (DS01255), Derrick Lattibeaudiere, Microchip Technology Inc., 2009.
- "*AN1371 Microchip MiWi™ PRO Wireless Networking Protocol"* (DS00001371), Yifeng Yang, Microchip Technology Inc., 2011.

APPENDIX A: SOURCE CODE FOR MIWI P2P AND STAR WIRELESS NETWORKING PROTOCOL STACK

All of the software covered in this application note are available through Microchip Libraries for Applications (MLA). The MLA suite/archive can be downloaded from the Microchip website at: www.microchip.com/mla or www.microchip.com.

REVISION HISTORY

Revision A (May 2008)

This is the initial released of this document.

Revision B (July 2010)

This revision incorporates the following updates:

- Updated the following sections:
  - Introduction
  - MiWi™ P2P wireless protocol's unique features
  - Message Format for IEEE 802.15.4 compliant transceiver
  - References
- Updated Figure 8.
- Updated Table 3.
- Additional minor corrections such as language and formatting updates are incorporated throughout the document.

Revision C (August 2017)

- Added the following sections:
  - **Messaging Type**
  - MiWi Star Overview
  - Handshaking in MiWi Star Wireless Protocol
- Updated the following sections:
  - **Protocol Features**
  - Unique Features of the MiWi Star Wireless Protocol
- Added Table 4, Table 7, and Table 8.
- Updated Table 1 and Table 3.
- Added Figure 14.
- Updated Figure 3.
- Updated Equation 1.
- Incorporated minor updates to text and corrected formatting throughout the document.

NOTES:

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.

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

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 ═

Trademarks

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

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

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

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

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

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

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

ISBN: 978-1-5224-2003-3

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

Austin, TX
Tel: 512-257-3370

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

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

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

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

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

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

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

EUROPE

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

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

Finland - Espoo
Tel: 358-9-4520-820

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

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700
Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

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

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

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

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

11/07/16