AN1204

Microchip MiWi™ P2P Wireless Protocol

Author: Yifeng Yang
Microchip Technology Inc.

INTRODUCTION

The demand is growing for more and more applications to move to wireless communication.

The benefits are reduced costs and ease implementation. Wireless communication does not require cabling and other hardware, and the associated installation costs. It also can 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, 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.

Microchip MiWi™ P2P Wireless Protocol is one of the wireless protocols that are 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). Microchip MiWi P2P protocol stack is now expanded beyond IEEE 802.15.4 specification to support Microchip proprietary transceivers (MRF49XA, MRF89XA and future proprietary transceivers from Microchip), while using IEEE 802.15.4 Media Access Control (MAC) layer design as the reference.

The protocol provides reliable direct wireless communication through an 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 the MiWi P2P Protocol and its differences from IEEE 802.15.4. The document details the supported features and how to implement them.

For more information, please refer to the Microchip application note AN1283 *"Microchip Wireless Media Access Controller - MiMAC"* (DS01283) and AN1284 *"Microchip Wireless Application Programming Interface - MiApp"* (DS01284).

This application note assumes that readers know C programming. However, it also recommends that readers review the IEEE 802.15.4 specification and Microchip MiMAC/MiApp interfaces before starting this application note or working with the MiWi P2P wireless protocol.

Protocol Overview

The MiWi P2P protocol modifies the IEEE 802.15.4 specification's Media Access Control (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 detect 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:

- Operates on Microchip PIC18, PIC24, dsPIC33 and PIC32 platforms
- Supports Microchip C18, C30 and C32 compilers
- Functions as a state machine (not RTOS-dependent)
- Supports a sleeping device at the end of the communication
- Enables Energy Detect (ED) scanning to operate on the least-noisy channel
- Provides active scan for detecting existing connections
- Enables frequency agility (channel hopping)

Protocol Considerations

The MiWi P2P protocol is a variation of IEEE 802.15.4 and supports both peer-to-peer and star topologies. It has no routing mechanism, so the wireless communication coverage is defined by the radio range.

Guaranteed Time Slot (GTS) and beacon networks are not supported, hence both the sides of the communication cannot go to Sleep Mode simultaneously.

If the application requires wireless routing instead of P2P communication; or interoperability with other vendors' devices; or a standard-based solution, for marketability, refer to the AN1066 *"MiWi™ Wireless Networking Protocol Stack"* (DS1066), AN1232 "*Microchip ZigBee-2006 Residential Stack Protocol*" (DS01232A) and AN1255 "*Microchip ZigBee PRO Feature Set Protocol Stack*" (DS01255A).

IEEE 802.15.4™ SPECIFICATION AND MiWi™ P2P WIRELESS PROTOCOL

After the initial 2003 release of the IEEE specification, a 2006 revision was published to clarify 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.

Most of the products in the market, however, use the original IEEE 802.15.4a specification, also known as IEEE 802.15.4-2003 or Revision A.

In this document, references to IEEE 802.15.4 means Revision A of the specification. MiWi™ P2P protocol takes IEEE 802.15.4 specification as the design reference and expand 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	Receiver Idle Configuration	Data Reception Method
Full Function Device (FFD)	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

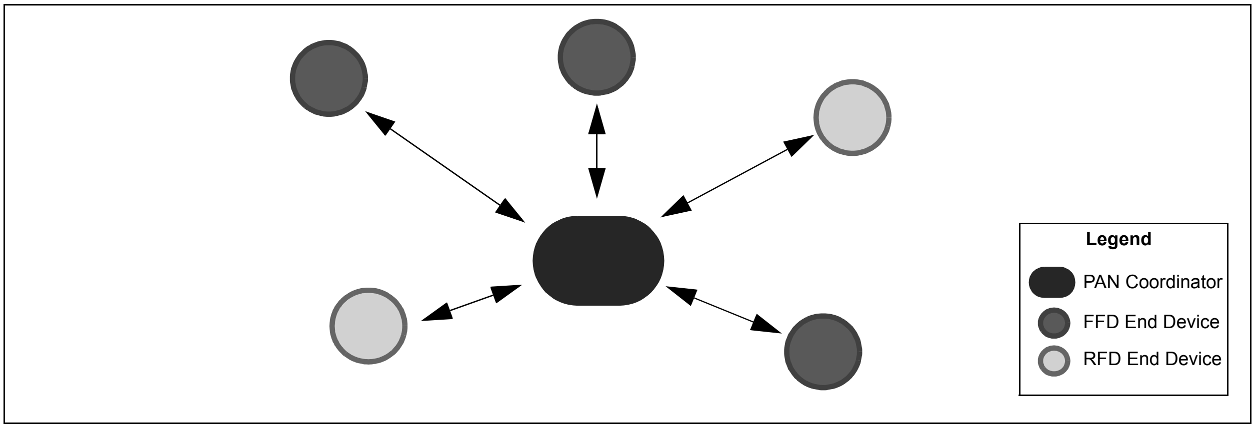
IEEE 802.15.4 and the MiWi P2P protocol support two topologies: star and peer-to-peer.

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. 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 it is Idle. Regardless of its functional type, end devices can only communicate to the PAN coordinator.

FIGURE 1: STAR TOPOLOGY

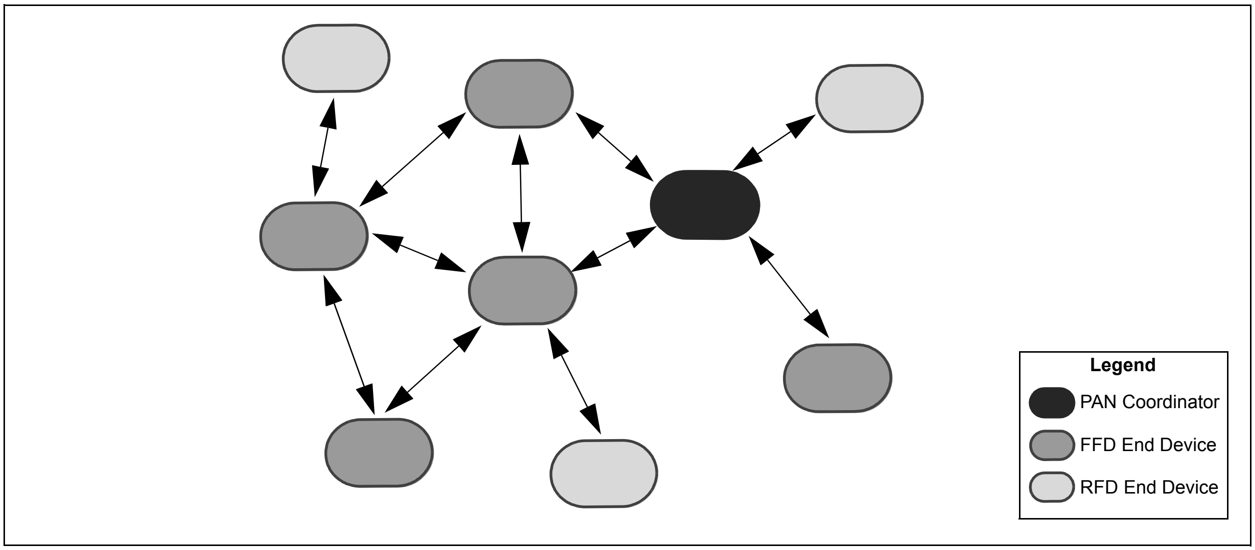


PEER-TO-PEER (P2P) TOPOLOGY

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.

In a non-beacon network, any device can transmit data at any time when the energy level (noise) is below the predefined level.

Beacon networks reduce all devices' power consumption because all of the devices turn off their radios periodically.

Non-beacon networks increase the power consumption by FFD devices because they must have their radios on all the time. These networks reduce the power consumption of RFD devices, however, because 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 eight-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 each device's EUI is unique.
- Short Address: A two-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. This is because 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 IEEE 802.15.4 specification's message format. Figure 3 illustrates the stack's packet format and its fields.

FIGURE 3: MiWi™ P2P WIRELESS PROTOCOL PACKET FORMAT

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

FRAME CONTROL

Figure 4 illustrates the format of the two-byte frame control field.

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

The three-bit frame type field defines the type of packet. The values are:

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

The security enabled bit indicates if the current packet is encrypted. If encryption is used, there will be an additional security header which will be detailed in later sections on the security feature.

The frame pending bit is used only in the Acknowledgement packet handled by the MRF24J40 radio hardware. The bit indicates if an additional packet will follow 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 will be 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 will be 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.

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) can be 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 will exist 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 will be 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 has been defined in MiMAC interface. For more information, refer to the Microchip application note AN1283 "*Microchip Wireless Media Access Controller - MiMAC*" (DS01283A).

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. 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 will save 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 will expire and be 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 will be 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 will acquire the indirect message if there is one.

VARIATIONS FOR HANDSHAKING

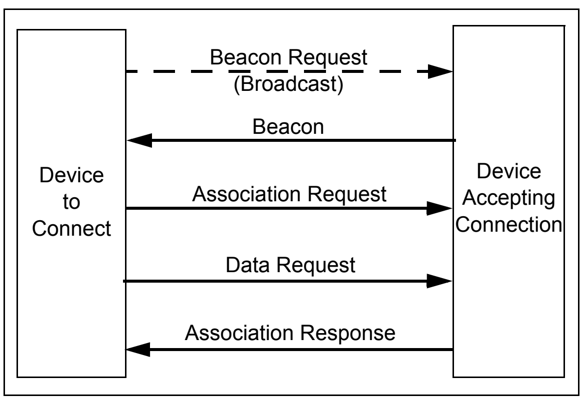
The MiWi P2P Wireless Protocol's major difference from the IEEE 802.15.4 specification is in the process of handshaking.

Under IEEE 802.15.4, a device's first step after powering up is to do a handshake with the rest of the world.

The specification's handshaking process, shown in Figure 5, is as follows:

1. The device that seeking to communicate sends out a beacon request.
2. All devices capable of connecting to other devices will respond with a beacon message.
3. The initiating device collects all of the beacons. (To accommodate multiple responses, the device waits until the active scan request times out). The device decides 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 complex process of joining a network. A device can join only a single device as its parent, hence the initial handshaking actually is the process of choosing a parent.

Choosing the parent requires:

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 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

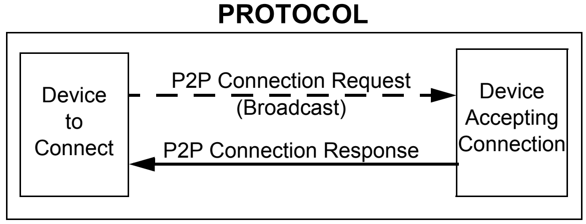
For the preceding reasons, the MiWi P2P protocol uses its own two-step handshaking process as shown in Figure 6:

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 IEEE 802.15.4 specification's functionality by using custom MAC commands for removing the connection between two devices. All of the protocol's custom MAC commands are listed in Table 3.

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. Also used for active scan functionality. (See **Section "Active Scan"** on Page 14).
0x82	P2P Connection Removal Request	Removes the P2P connection with the other end device.
0x83	Data Request	Similar to the IEEE 802.15.4™ specification's 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.
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.

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 will not be 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 will be discarded, except under the following conditions:

- The P2P connection request is coming from a device with which the receiving end already has had an established connection.
- The P2P connection request is an active scan.

The format of the P2P connection request command frame is shown in Figure 7.

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 will avoid the false connections with devices that operate on different channels.

The capability information byte, shown in Figure 7, is formatted as shown in Figure 8.

FIGURE 8: CAPABILITY INFORMATION FORMAT

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

The P2P connection request's optional payload 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 will not be used in the stack itself.

P2P CONNECTION REMOVAL REQUEST

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

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)

DATA REQUEST

The data request (0x83) command is the same as the IEEE 802.15.4 specification's data request (0x04) command. Its format is shown in Figure 10.

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

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

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 one. The command's format is shown in Figure 11.

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

This command usually is 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 will broadcast three times and all FFD devices will rebroadcast this message.

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.

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

P2P CONNECTION RESPONSE

The P2P connection response (0x91) command is used to respond to the P2P connection request. The command's format is shown in Figure 12.

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 that was received had a capability information byte and an optional payload attached, it is requesting a connection. The capability information and optional payload, if any, would be 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 did not have a capability information byte and optional payload, the command is an active scan. The P2P connection response, therefore, would have no capability information or optional payload attached.

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

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.

The format of the response's capability information 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 Page 10).

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 had been received early and whether the resulting connection has been removed.

The command's format is shown in Figure 13.

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

MiWi™ P2P WIRELESS PROTOCOL'S UNIQUE FEATURES

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. These include:

- 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)

Small Programming Size

To address many wireless applications' cost constraints, the MiWi P2P protocol is as small as possible. 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 4 Kbytes of programming memory.

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

The feature supports bidirectional communication between devices, but communication between PANs is disabled. If the security feature is used, the freshness check will be disabled. (For more information on freshness check, refer to the **Section "Security Features"** on Page 14.)

Idle Devices Turning Off Radios

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

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

The decision as to when a device is put into Sleep mode is made by the specific application. Possible triggers could 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 decided 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 it a message. If no message needs to be sent, no additional feature must be enabled by the peer device.

If the peer device needs to send 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 being delivered directly to the sleeping device, this process is called an indirect message.

If an indirect message needs to be delivered, the peer device of the sleeping node needs to define "`ENABLE_INDIRECT_MESSAGE`" in the, `ConfigApp.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. That message maximum 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 "`INDIRECT_MESSAGE_SIZE`" in the, `ConfigApp.h` file. For indirect messaging, the time-out period for the indirect messages also needs to be defined. If a time-out period was not defined and an RFD device was dead, the indirect message would remain forever in the volatile memory.

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

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, "`ENABLE_BROADCAST`" must be defined in the, `ConfigApp.h` file.

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 Media Access Controller - MiMAC*" (DS01283A).

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:

$$\text{Scan Time Period} \equiv 60 \bullet (2^{\text{ScanDuration}} + 1)$$

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

A scan duration of 10 would result 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 would be `b'0000 0000 0000 0000 0000 1000 0000 0000`. Channels 11 to 26, supported in the 2.4 GHz spectrum, would be `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. That is because every device responds to the scan, even those that will not allow new connections.

To activate the active scan feature, "`ENABLE_ACTIVE_SCAN`" must be defined in the, `ConfigApp.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 shown in Equation 1.

For more information on measurement, see **Section "Active Scan"** on Page 14.

After the scan is complete, the channel identifier with the least noise will be returned.

To activate the Energy Scan feature, "`ENABLE_ENERGY_SCAN`" must be defined in the, `ConfigApp.h` file.

Frequency Agility

Frequency agility enables the MiWi P2P protocol PAN to move to a different channel if operating conditions so require.

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

- Frequency agility initiators: The devices that decide whether channel hopping is necessary and which new channel to use.
- Frequency agility followers: Devices that change to another channel when so 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. That is because the initiator must do an energy scan to determine the optimal channel for the hop. Then, 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:

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

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

IMPLEMENTING, ACTIVATING FEATURE

When to perform a frequency agility operation is decided by the application. Usually frequency agility is triggered by continuos transmission failure: either by CCA failure or no acknowledgement received.

To activate the frequency agility feature, “`ENABLE_FREQUENCY_AGILITY`” must be defined in the, `ConfigApp.h` file.

APPLICATION PROGRAMMING INTERFACES (APIs)

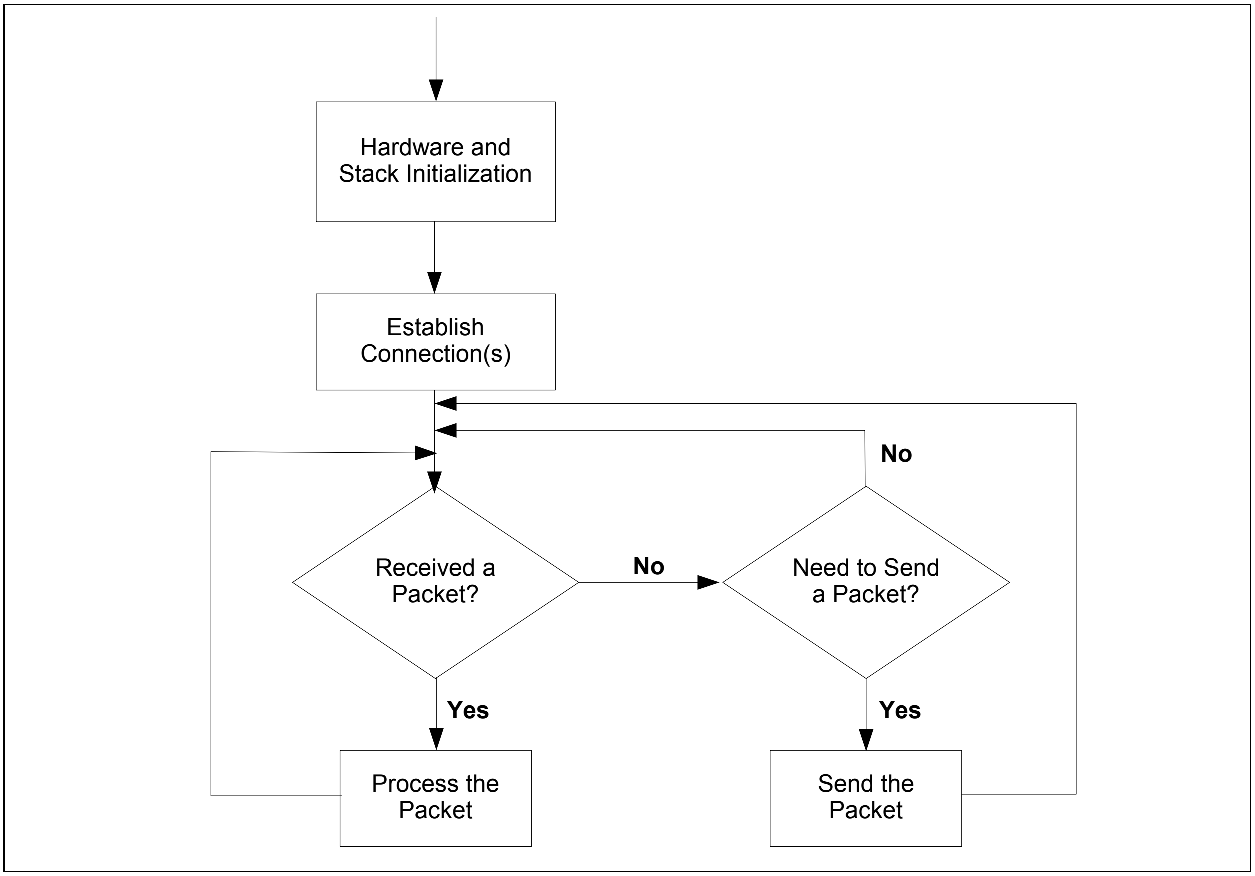
MiWi P2P protocol uses MiApp as its application programming interface. For more information on MiApp interface, refer to the Microchip application note AN1284 "*Microchip Wireless Application Programming Interface - MiApp*" (DS01284A).

APPLICATION FLOWCHART

A typical MiWi P2P protocol application starts by initializing the hardware and MiWi P2P protocol. Then, it tries to establish a connection and enter the normal operation mode of receiving and transmitting data.

Figure 14 illustrates the typical flow of the MiWi P2P protocol applications.

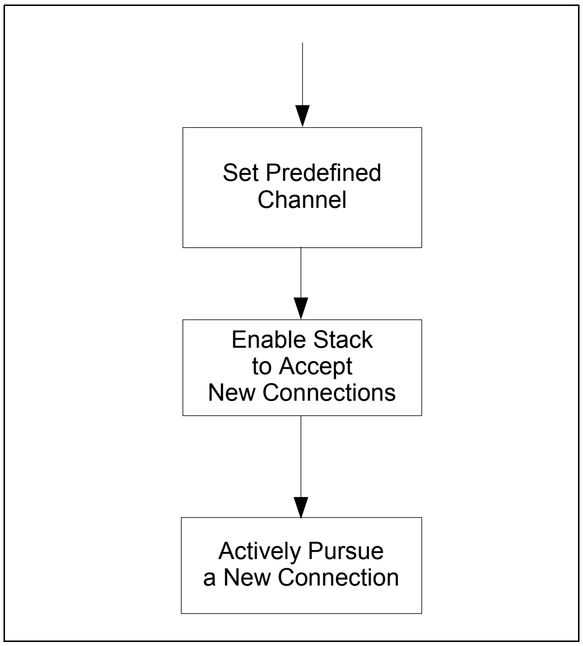
FIGURE 14: FLOWCHART FOR MiWi™ P2P WIRELESS PROTOCOL APPLICATIONS



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

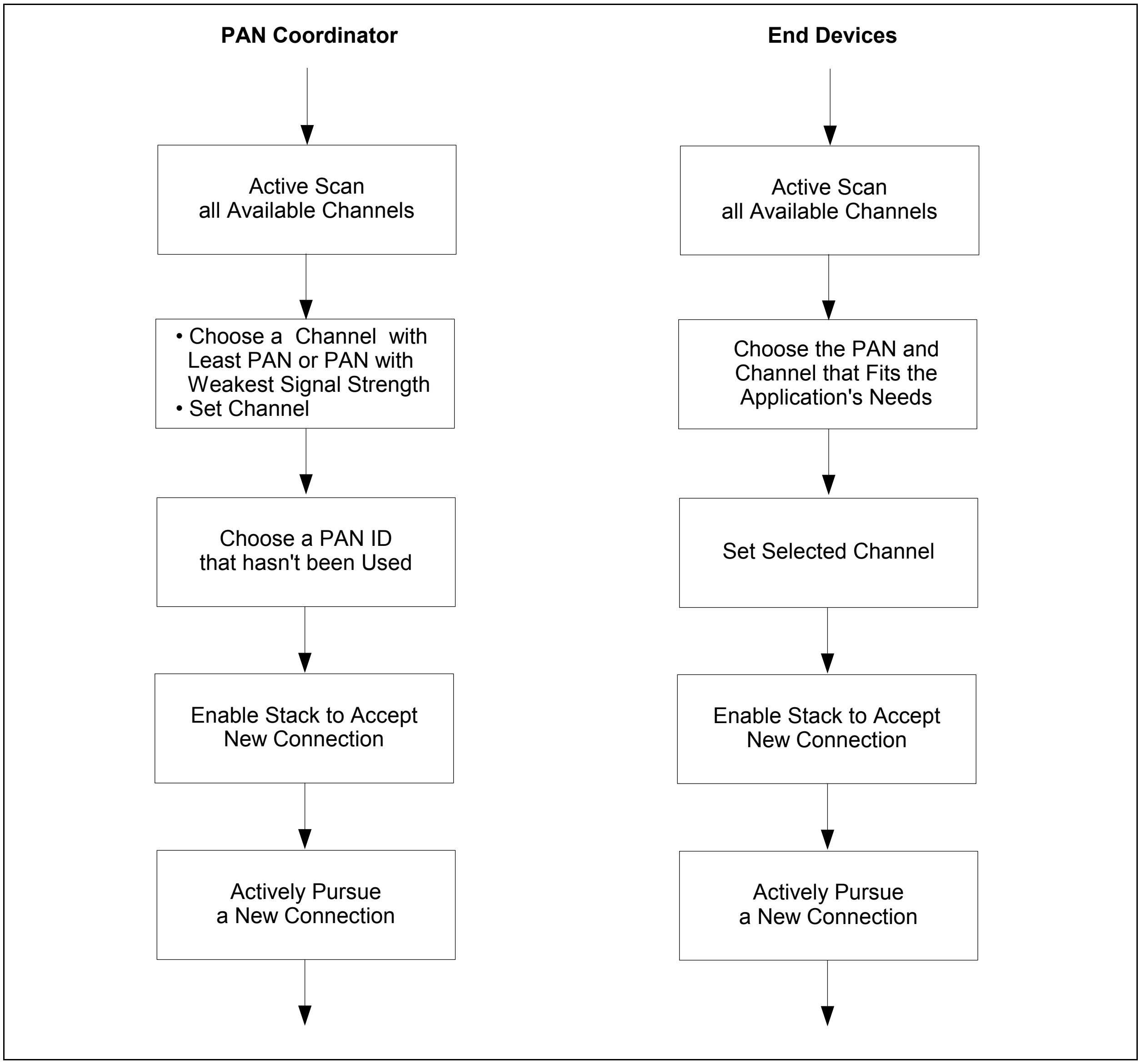
The simplest P2P connection application for establishing connections is shown in Figure 15.

FIGURE 15: FLOWCHART TO ESTABLISH CONNECTION(S) IN SIMPLE MODE



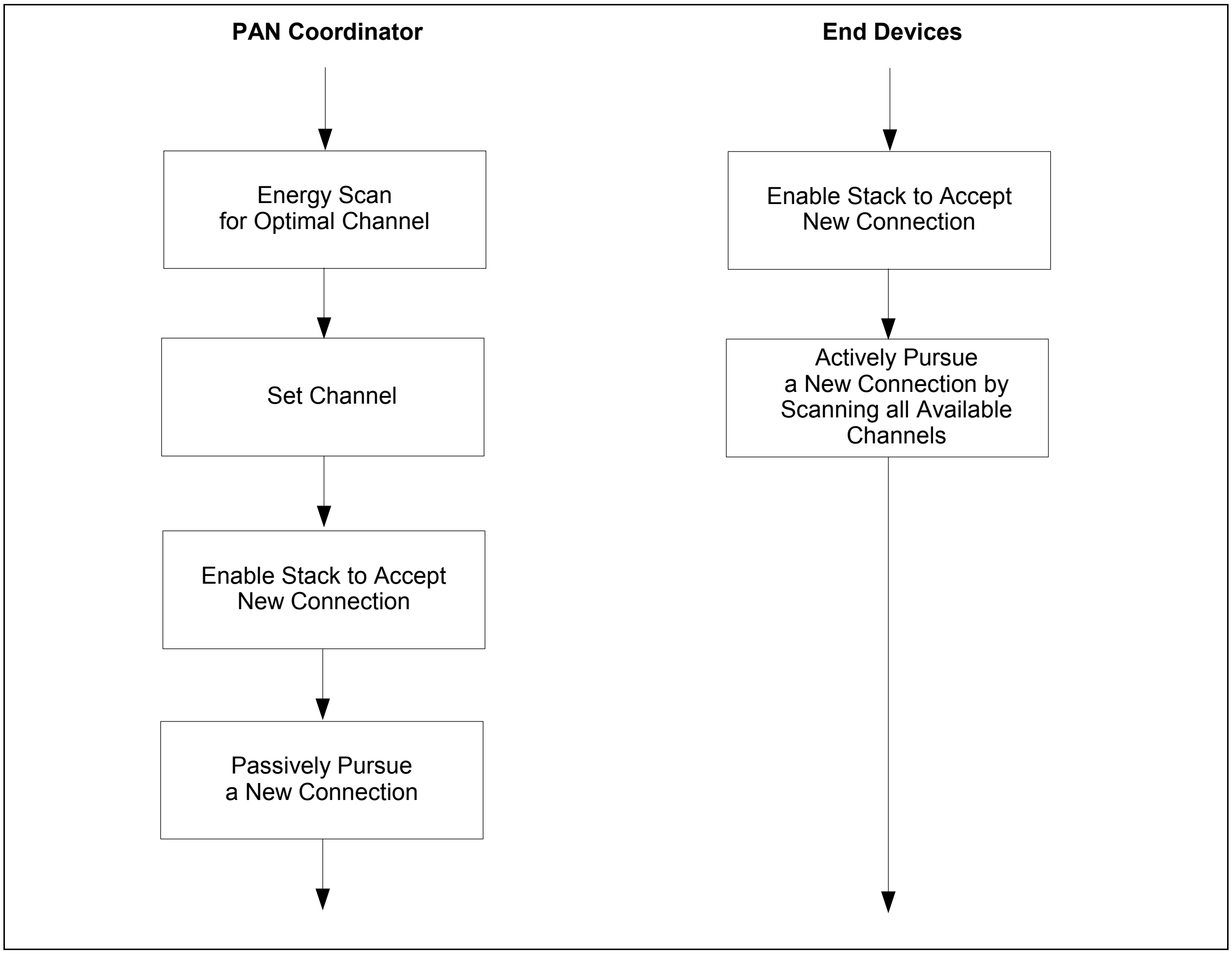
The complex applications require active scan capability, the active scan steps for establishing connections differ between the PAN coordinator and end devices. Figure 16 illustrates the active scan steps for both categories of devices.

FIGURE 16: FLOWCHART TO ESTABLISH CONNECTIONS WHEN ACTIVE SCAN IS ENABLED



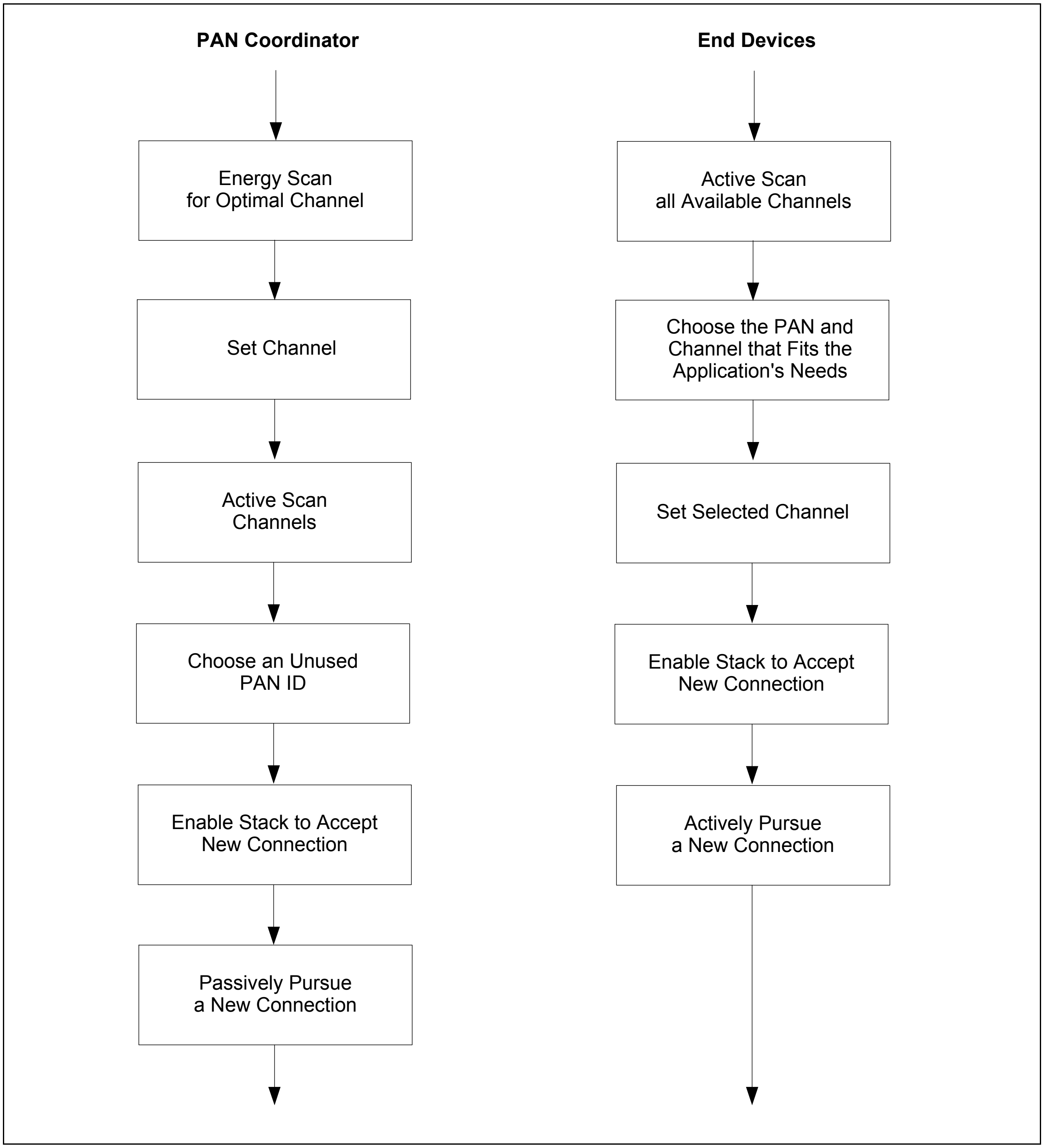
For applications with energy scan enabled, the steps after connection also differs for the PAN coordinator and end devices, as shown in Figure 17.

FIGURE 17: FLOWCHART TO ESTABLISH CONNECTIONS WHEN ENERGY SCAN IS ENABLED



The process for establishing connections with active scan and energy scan enabled is shown in Figure 18.

FIGURE 18: FLOWCHART TO ESTABLISH CONNECTIONS WITH ACTIVE AND ENERGY SCAN



SYSTEM RESOURCES REQUIREMENT

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

Table 4 gives the requirements of a basic configuration.

TABLE 4: PIC18 MEMORY REQUIREMENTS FOR MiWi™ P2P WIRELESS PROTOCOL

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

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

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

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

† These requirements are for the PIC18 family of microcontrollers. The stack also supports PIC24, dsPIC33 and PIC32 microcontrollers, but those devices' requirements may vary.

These requirements are for the initial release of the stack and are subject to change.

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.

If an application is more complex, the Microchip MiWi™ Wireless Protocol stack should be considered. That stack provides support for a real network with up to 1,024 active nodes across as many as four hops. For more information on this protocol, refer to the application note AN1066*, “MiWi™ Wireless Networking Protocol Stack”* (DS1066).

For a complex network or interoperability, implementing Microchip’s ZigBee protocol specification is an option. For more information ZigBee protocol, refer to the application notes AN1232 “*Microchip ZigBee-2006 Residential Stack Protocol*” (DS01232A) and AN1255 “*Microchip ZigBee PRO Feature Set Protocol Stack*” (DS01255A).

REFERENCES

D. Flowers and Y. Yang, *AN1066,* “*MiWi™ Wireless Networking Protocol Stack”* (DS1066), Microchip Technology Inc., 2007.

Derrick Lattibeaudiere, *AN1232* “*Microchip ZigBee-2006 Residential Stack Protocol”* (DS01232A).

Derrick Lattibeaudiere, *AN1255* “*Microchip ZigBee PRO Feature Set Protocol Stack”* (DS01255A).

IEEE Std 802.15.4-2003™, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2003.

Yifeng Yang, *AN1283* “*Microchip Wireless Media Access Controller - MiMAC”* (DS01283A).

Yifeng Yang, *AN1284* “*Microchip Wireless Application Programming Interface - MiApp”* (DS1284A).

REVISION HISTORY

Revision A (May 2008)

This is the initial release of the 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
- Figures
  - Updates **FIGURE 8: "Capability Information Format"**
- Tables
  - Updated **TABLE 3: "Custom MAC Commands for MiWi™ P2P Wireless Protocol"**
- Additional minor corrections such as language and formatting updates are incorporated 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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Trademarks

Printed on recycled paper.

ISBN: 978-1-60932-347-9

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
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

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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: 86-10-8528-2100
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 - 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: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

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

01/05/10