MICROCHIP
WebSeminars

Wireless Communication using the IrDA® Standard Protocol

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 1

Thank You

My name is Mark Palmer and I am the application engineer supporting Microchip's family of IrDA Standard Protocol Handling devices. Today we will be giving a brief introduction to wireless communication using the IrDA Standard protocol.

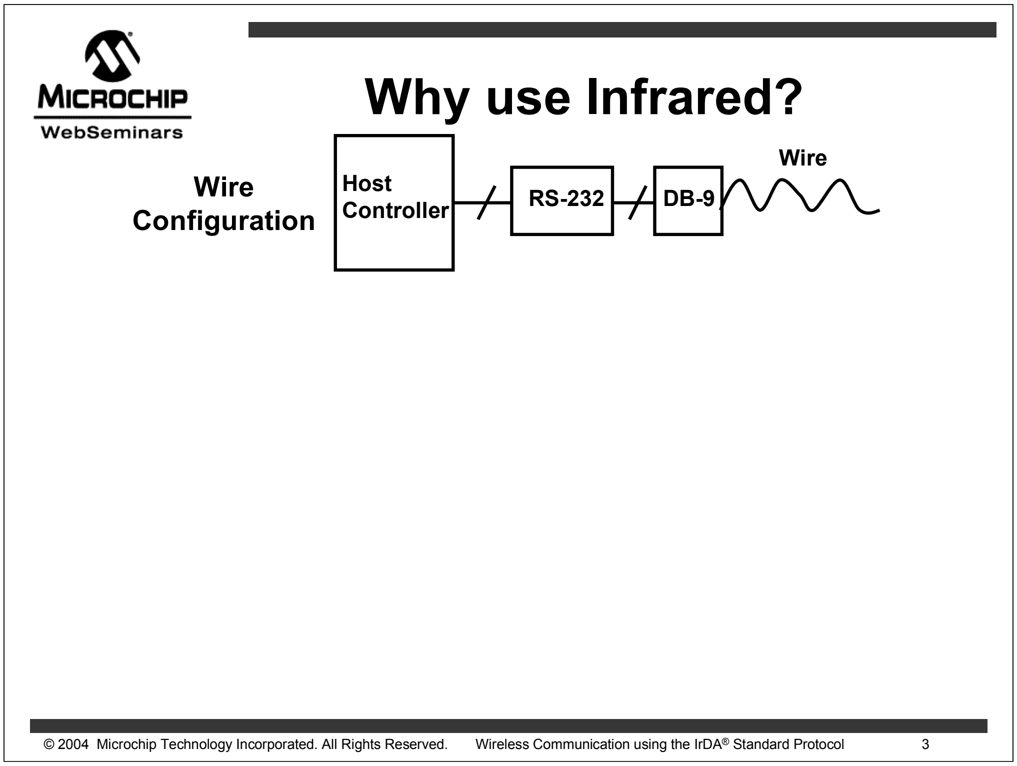
MICROCHIP
WebSeminars

Agenda

- Why use infrared?
- Typical applications
- An IrDA® Standard system
- IrDA protocol stack
- Connection sequence
- Host UART interface
- Complete support
- Summary

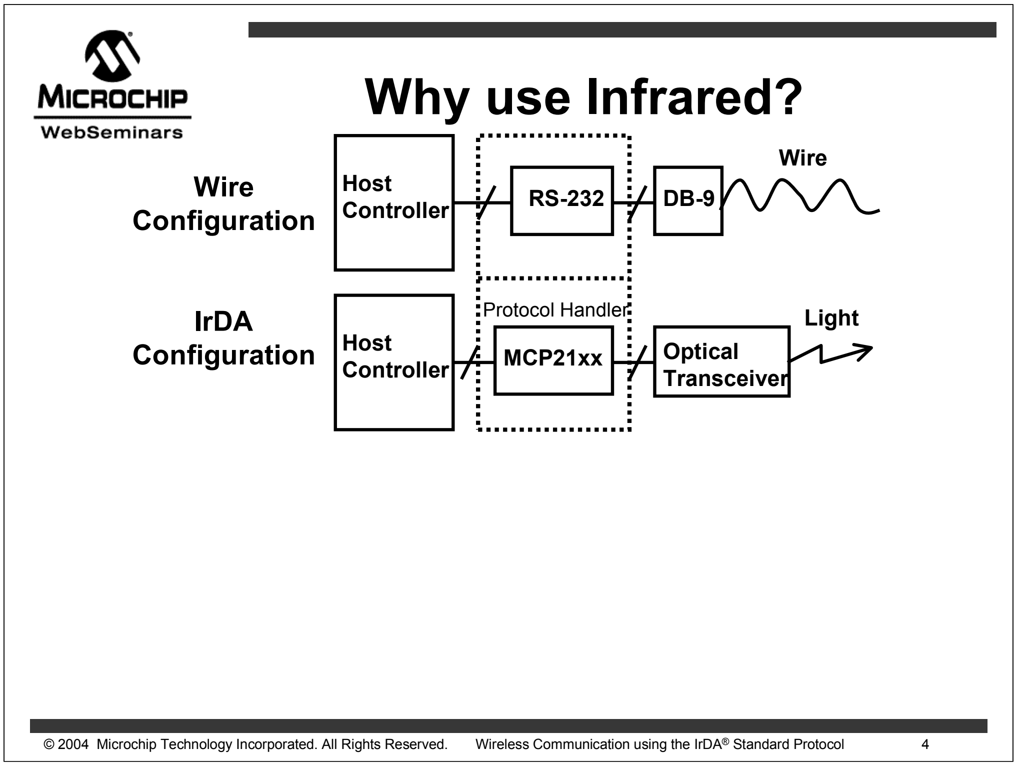
© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 2

This agenda gives you an idea of the topics we will cover. First we will discuss why use infrared, and what are some of the typical applications. We will then give a high level look at an IrDA Standard system. From this system we will discuss the IrDA protocol stack, the connection sequence that occurs on the IR interface, and then the Host UART interface between the Microchip Protocol Handler (MCP215x/MCP2140) and the Host Controller. We will then wrap up with what Microchip offers to support this product family.



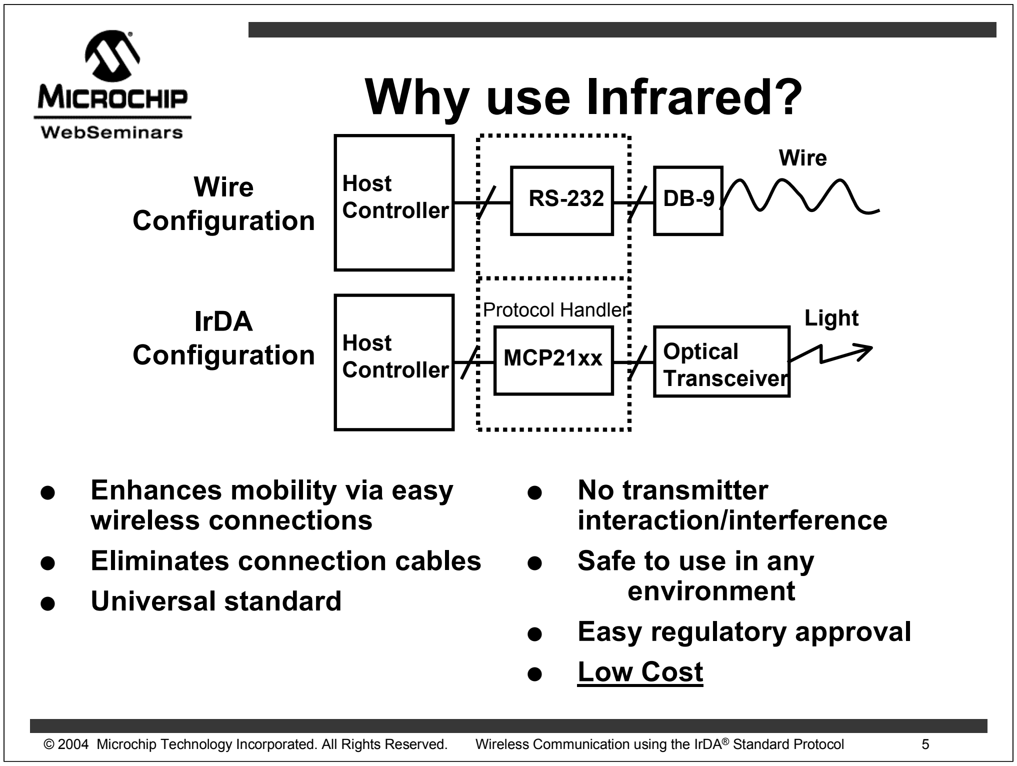
In a typical embedded application, the interface to the outside world is a serial cable. This embedded application will have the Host controller, an RS-232 line driver, the DB-9 connector, and then the serial cable.

Microchip’s Protocol Handler devices allow an analogous visualization:



Here the RS-232 line driver is replaced by the MCP21xx Protocol Handler, The DB-9 connector is replaced by the Optical Transceiver circuitry, and the Serial cable is replaced by light.

So what are the advantages of IrDA



Enhances mobility since cables are no longer required. Think how useful it would be if to transfer data between PDAs you had to have a cable. Would data (address contacts, photos, programs, ..) be transferred as often?

Since it is a Universal standard (PDAs, notebook PC, mobile phones, ...), ensures interoperability with many existing devices

Some other wireless technologies (such as RF) radiate noise, they can cause interaction and interference. Since IR uses light, which is directional, these issues don’t occur. This lack of interference means that IR is safe to use in any environment (planes, hospitals, …). This also eases regulatory approval.

Lastly, the cost of a wireless solution can be implemented near the same cost as a wired solution.

Typical IrDA Standard Applications

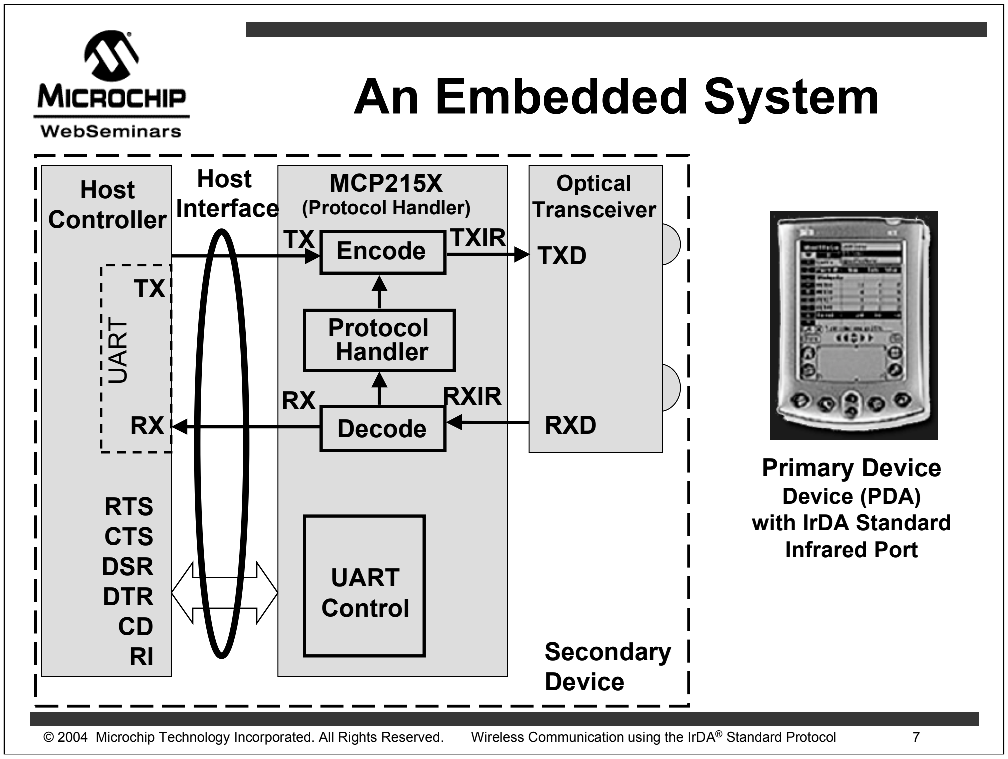
IrDA Standard compliant ports are available on:

- PDAs
- Notebook Computers
- Mobile phones
- Printers
- Pagers
- Watches (Heart Rate Monitor)

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 6

Some of the common devices where you will find IrDA standard ports are:

PDAs, Notebook PC, some printers, Mobile Phones and pagers, and the one that I use regularly a Heart Rate monitor watch, which I use to log my exercise routines. The IrDA interface is used to download the exercise data to my PC.



Here we have the IrDA Standard system.

Starting on the right we have the Primary Device, in this case a PDA.

Left of the PDA we have the embedded system, which is the IrDA Standard Secondary Device. This Embedded System can be broken into three parts:

1. The IR side, with Optical Transceiver
2. The Protocol Handler
3. The Host Controller side, includes Host UART

MICROCHIP WebSeminars

The IrDA Protocol "Stack"

Asynchronous Serial IR (2, 3, 4) (SIR) (9600 - 115200 Baud)	Synchronous Serial IR (1.15 MBaud)	Synchronous Fast IR (FIR) (4 MBaud)

[shaded box] Implemented in the MCP215x and MCP2140

[white box] Optional IrDA Data Protocols not implemented in the MCP215X and MCP2140

Note 1: The MCP215X and MCP2140 implements the 9-wire "cooked" service class serial replicator
2: An Optical Transceiver is required
3: MCP2120 only implements the Encoding/Decoding portion of Physical Layer
4: MCP2140 operates at a fixed 9600 Baud

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 8

So first we look at the Protocol Stack that is implemented in the Protocol Handler.

The IrDA Standard is a network protocol. The IrDA Organization has remapped the Standard OSI network model into these Protocol layers for their documentation purposes.

At the bottom of the Stack is the Physical layer. This specifies how the data looks like (bit format, IR/differential/….). IrDA has 3 specifications. Most Devices, including PDAs use the Asynchronous Serial IR, or SIR. SIR supports transmission speed of 115200 Baud. PCs and some printers implement Synchronous Fast IR, or FIR. FIR supports transmission speed of 4 Mbaud, but to ensure backwards compatible with SIR it also starts negotiation at 9600 Baud (that is why PCs can talk to PDAs).

MICROCHIP WebSeminars

The IrDA Protocol "Stack"

LM-IAS	Tiny Transport Protocol (Tiny TP)	
IR Link Management - Mux (IrLMP)		
IR Link Access Protocol (IrLAP)		
Asynchronous Serial IR (2, 3, 4) (SIR) (9600 - 115200 Baud)	Synchronous Serial IR (1.15 MBaud)	Synchronous Fast IR (FIR) (4 MBaud)

Implemented in the MCP215x and MCP2140

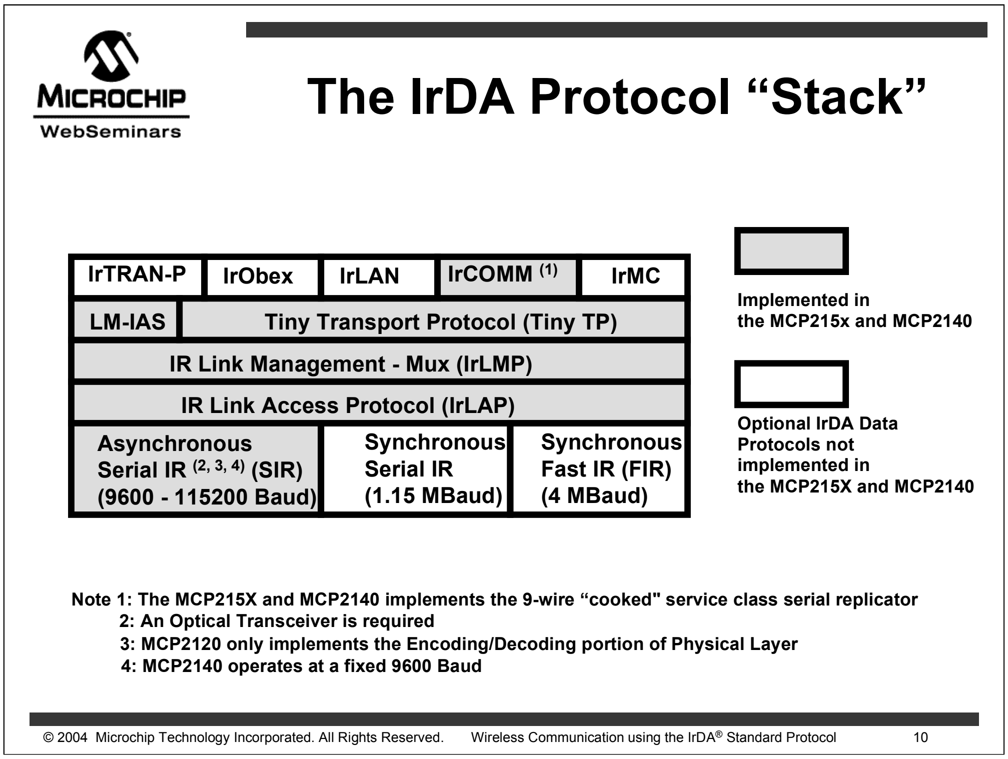
Optional IrDA Data Protocols not implemented in the MCP215X and MCP2140

Note 1: The MCP215X and MCP2140 implements the 9-wire "cooked" service class serial replicator
2: An Optical Transceiver is required
3: MCP2120 only implements the Encoding/Decoding portion of Physical Layer
4: MCP2140 operates at a fixed 9600 Baud

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 9

Now we move to the middle area of the Protocol Stack.

These layers handle the Routing of the Data, Error Correction for the Data Packet, Link Management, other communication issues, as well as structuring the information for the Application Layer of the Protocol Stack.

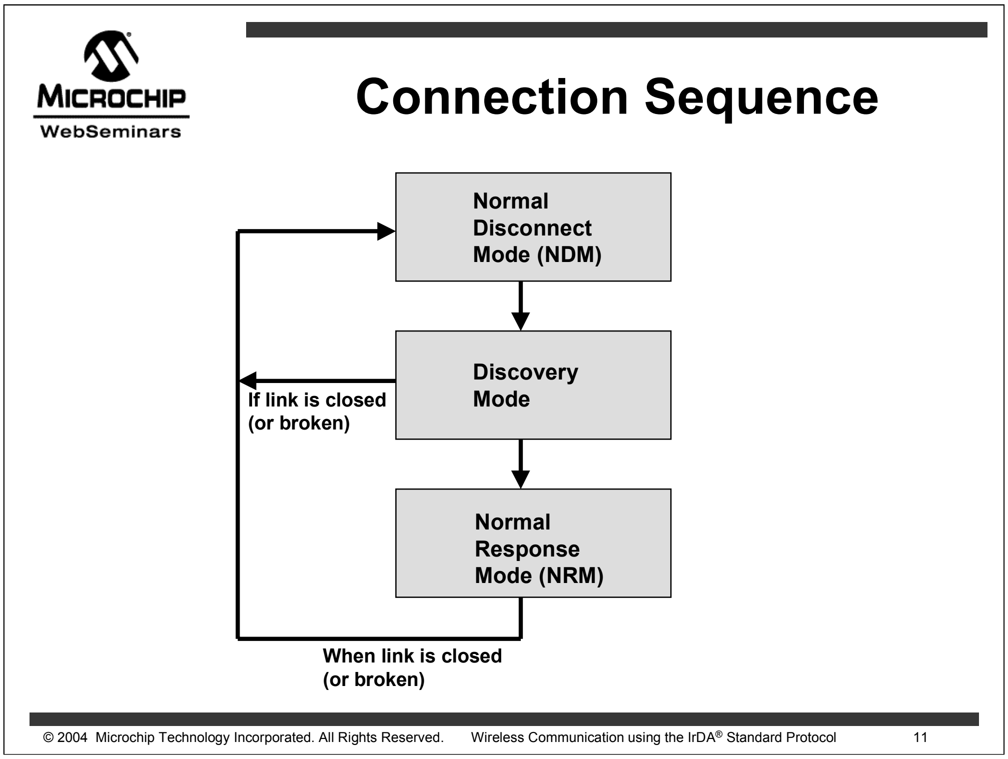


Then we add the Application layer. At the Application layer is where there are different Application Protocols. You can think of these application protocols as languages. If two devices try to communicate, and they are not using the same protocol (language) then they will not be successful.

When many people think infrared, they think "Beaming". That is when they select an Object to "beam" to another device. This technique actually uses IrObex, which stands for IR Object Exchange. In this case the characteristics of the "Object" (file, program, photo, ….) are know before the object is transferred.

IrCOMM is the IrDA Standard specification for the replacement of the communication ports (Serial and Parallel). The characteristics of the information that will be transferred is unknown.

THE MCP215x/MCP2140 HANDLES ALL THIS PROTOCOL, The applications Host Controller is only required to send and receive UART data and follow the Flow Control signals of the Host UART interface.



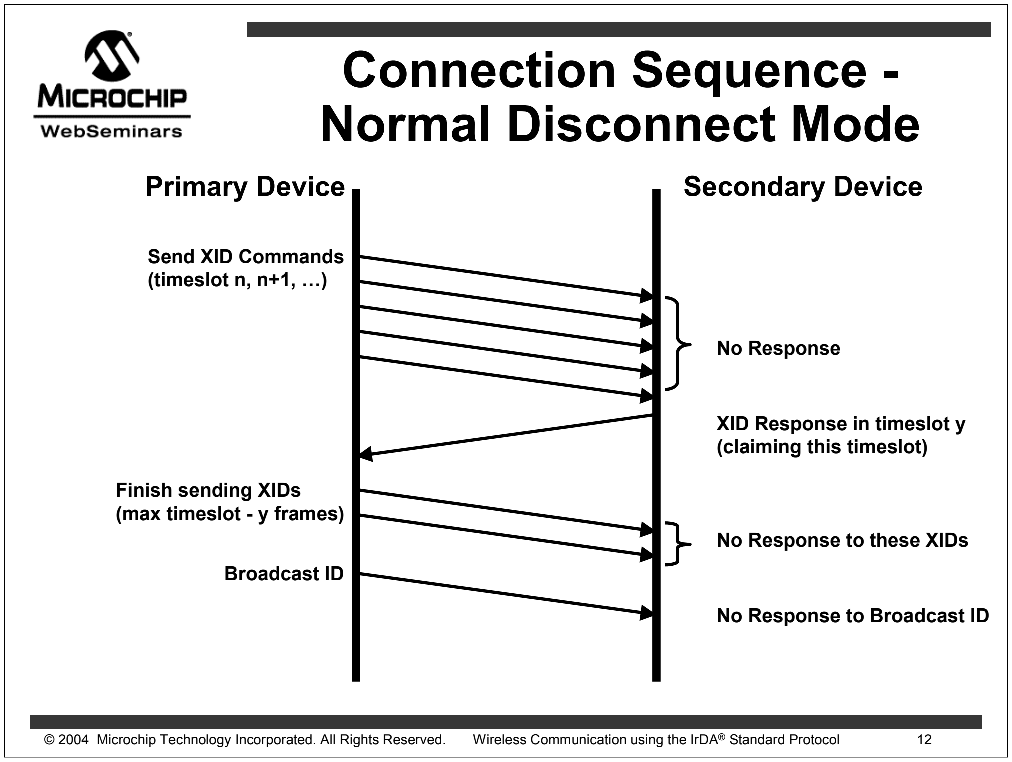
There are three states that the IrDA Standard protocol stack could be in. These are:

* Normal Disconnect Mode (NDM)
* Discovery Mode
* Normal Response Mode (NRM)

NDM is the state where the Primary Device is searching for other IrDA Standard devices (with the same Application Layer Protocol) and the Secondary Device is waiting for interrogation from a Primary Device

Discovery is the state where the Primary Device and the Secondary Device determine each others capabilities (and use the Highest common denominator)

NRM is the state where Data and Status information is transferred back and forth. Status information is also the mechanism to verify if a link is still connected (not blocked).

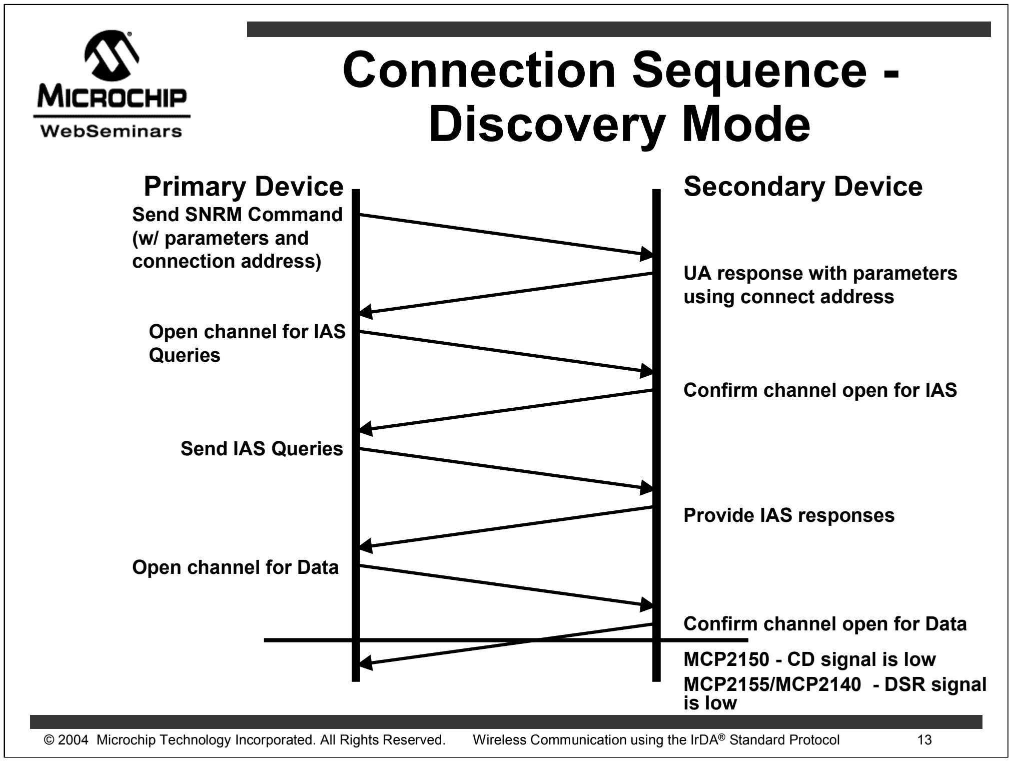


Here the Primary Device sends out XID commands. The XID command will indicate a timeslot (typically 0 to 7 - dependant on the IrDA Standard Stack used).

When the Secondary Device gets in range of the Primary Device, it will respond to one of these XID commands - claiming that timeslot. This Secondary Device will then ignore the remaining XID commands.

The IrDA Standard protocol allows the Primary Device to identify up to 8 devices, The MCP215x and MCP2140 devices are designed for point-to-point communication (not multi-point).

The Primary Device will then transmit a Broadcast ID which no Secondary Device will respond to.



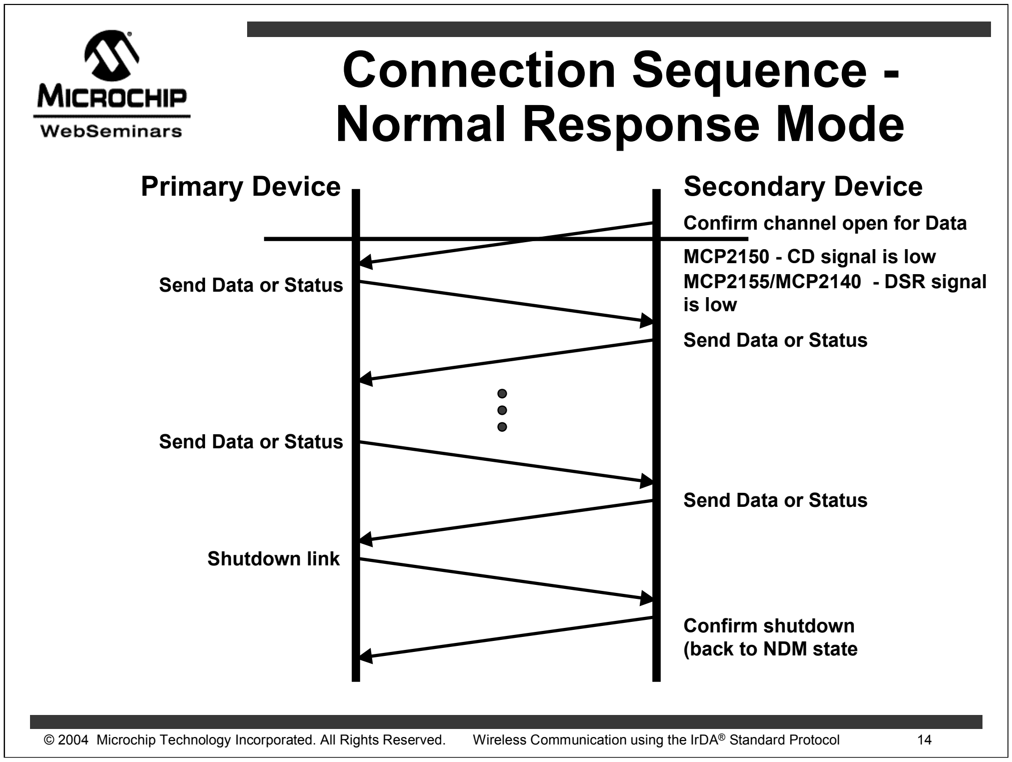
Discovery is the state where the Primary Device and the Secondary Device determine each others capabilities

First the Primary Device transmits an SNRM command with parameters and connection addresses. The Secondary Device will then issue a UA response with parameters using the specified connection address.

The Primary Device then Opens a channel for IAS Queries, which the Secondary Device will confirm.

Now it is time for the capabilities of each device to be determined. These capabilities include: IR Baud Rate, Data Packet Size, Turn Around times, and a few other things. So the Primary Device will send its capabilities, and the Secondary Device will respond with its capabilities. The Highest common denominator will then be used to optimize system performance.

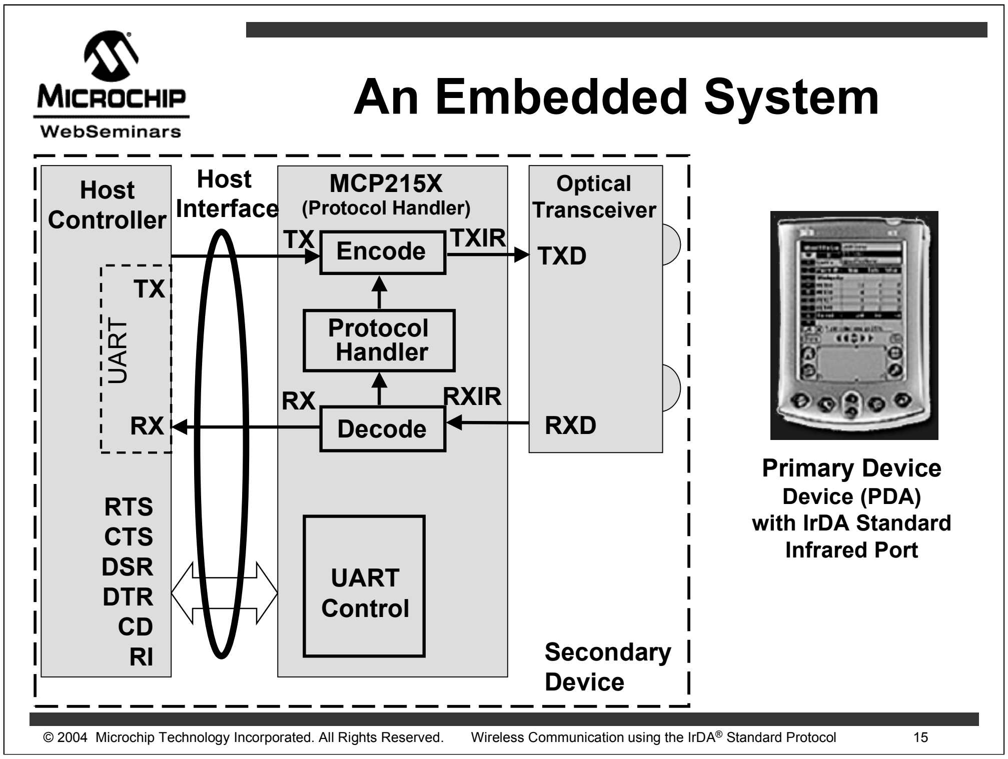
Now it is the responsibility of the Primary Device to send the Open Channel for Data. On a PC this will only occur after a PC program has “attached” to the IR port - such as HyperTerminal. On a PDA this occurs when the 1st byte (packet) of data is ready to be transferred. The Secondary Device will confirm that the channel is open for data. When the MCP215x or MCP2140 confirm that the channel is open for data, they will also drive the appropriate signal (CD or DSR) low to indicate that the link is now open, and Data may be transferred



NRM is the state where Data and Status information is transferred back and forth. Status information is also the mechanism to verify if a link is still connected (not blocked).

If the link is blocked for longer then the Link timeout, then the devices will reset to the NDM state.

Whenever communication is completed, the Primary Device may shut down the link. The Secondary Device will confirm this command and both devices will return to the NDM state.



Now we are going to look at the interface between the Host Controller and the Protocol Handler, or the Host UART interface.

Host UART Hardware Handshaking - continued

- The CTS and RTS pins are used for the control of the data flow between the MCP215x/MCP2140 and the Host Controller
  - CTS indicates when the Host Controller may send data to the Protocol Handler
  - RTS indicates when the Protocol Handler may send data to the Host Controller
- The MCP215x CTS signal is windowed
  - Protocol Handler is either operating on IR interface or Host UART interface

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 16

The CTS signal must be monitored by the Host Controller, so the Host Controller knows when it can transfer data to the Protocol Handler device. Any data transferred while the CTS signal is high is ignored, and will be "lost". That is will not be transferred to the Primary Device.

Occasionally the Host Controller may be busy and not able to receive things from the Protocol Handler. For this, the Host Controller can control the state of the RTS signal. This indicates when the Protocol Handler can transmit the data it has received from the Primary Device on to the Host Controller. If the RTS signal is de asserted too long, while the Protocol Handler has data pending for the Host Controller, the link will be closed.

Now the MCP215x devices window the time between the IR interface and the Host UART interface. So the CTS signal can only be valid during the window where the Protocol Handler is operating on the Host UART

MICROCHIP WebSeminars

Host UART Hardware Handshaking - continued

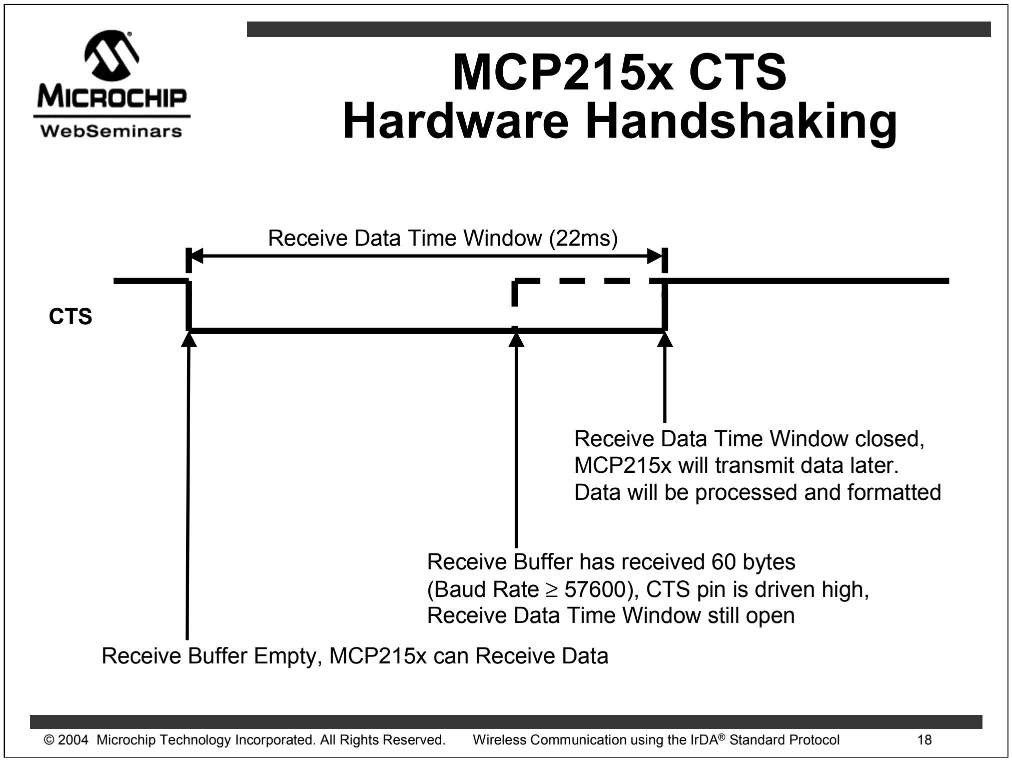
- MCP215x
  - Window is 22ms
  - Once “Communications” is ok (CTS is Low), Data can be transmitted for 22ms, or for 64 bytes (Host UART Receive Buffer size)

Host UART Baud Rate	Bytes Transfer (in 22 ms)	
115200	253	(> 64 byte limit)
57600	126	(> 64 byte limit)
19200	42	(< 64 byte limit)
9600	21	(< 64 byte limit)

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 17

This Window is 22ms and the MCP215x Host UART Receive Buffer is 64 Bytes. So the Host Controller can transfer data for this 22ms up to a maximum of 64 Bytes.

So the table shows how many bytes can be transferred in 22ms at a given baud rate. Baud rates 19200 Baud and below will not be able to transfer the 64 Bytes in the Window time. With a Baud rate 57600 and Greater, the 64 Byte limit can be reached



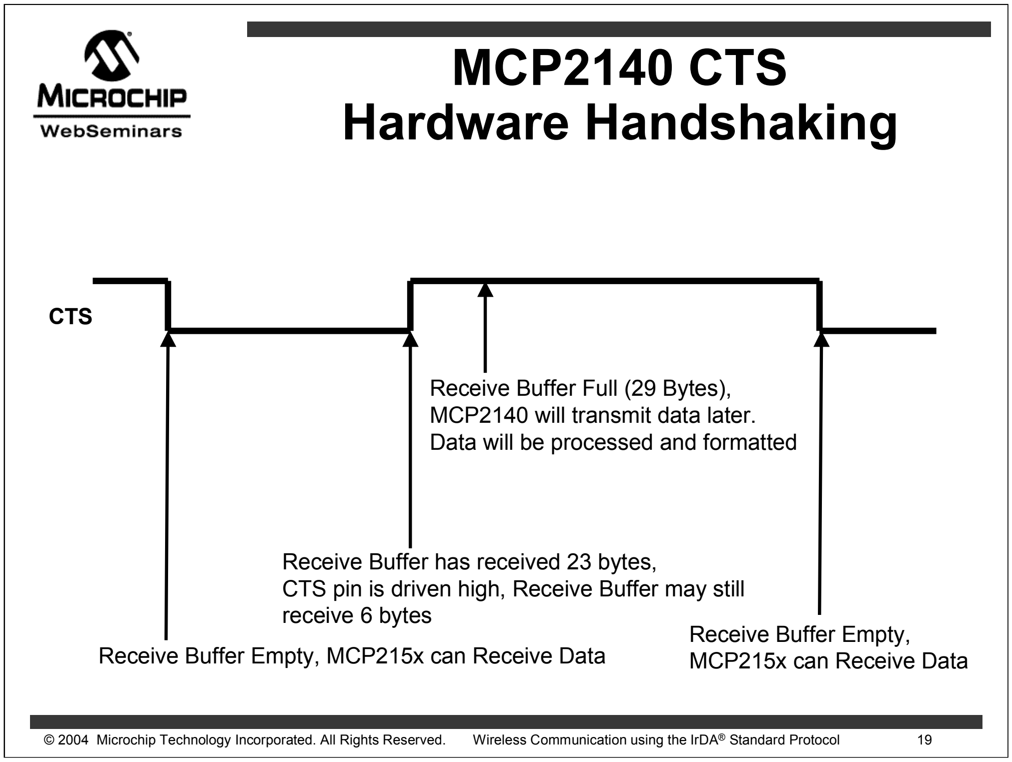
Here is a visualization of the CTS and Receive Data Time Window operation.

Assuming that the Host controller Baud rate is 57600, that there is no Byte to Byte spacing, and minimal latency when CTS goes low to Data to start transfer.

When CTS goes low, the MCP215x Host UART Receive Buffer is empty. Once 60 Bytes have been received, the CTS signal will be driven high. This is done to indicate to Host Controllers that use a transmit FIFO on the UART. The MCP215x can receive an additional 4 bytes. When 64 bytes are received, the Data will be processed, formatted, and then transmitted to the Primary Device.

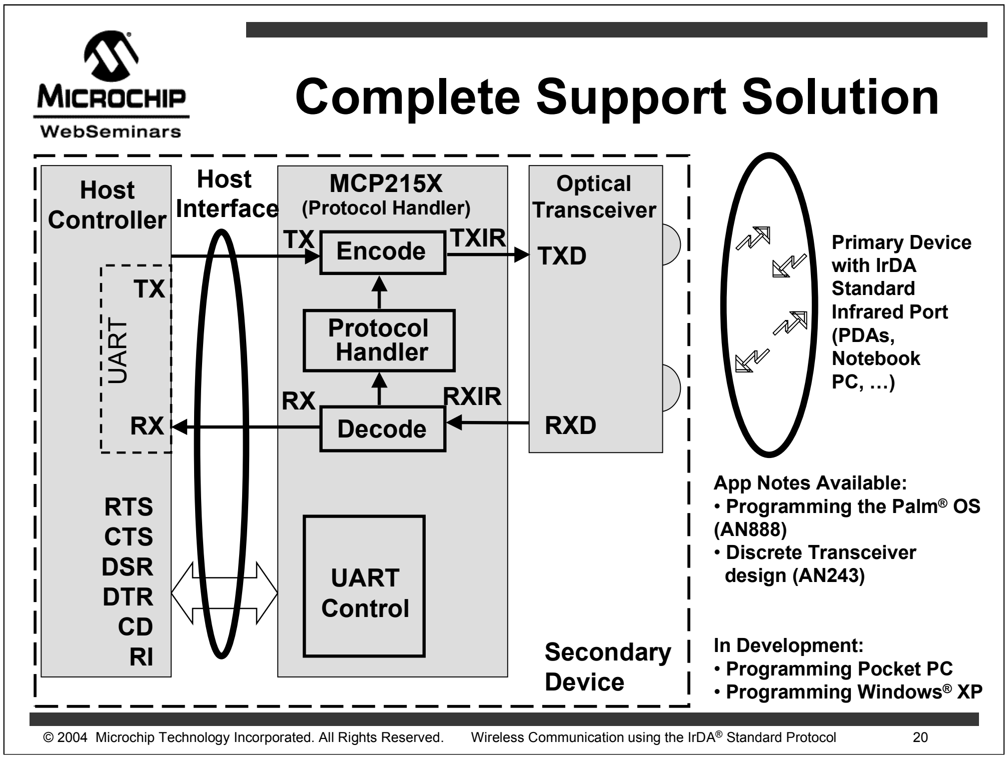
If the Host UART Baud rate is 19200, then the Receive Data Time window will close, which forces CTS High, before 64 Bytes could be received. The Data in the Host UART Receive Buffer will be processed, formatted, and then transmitted to the Primary Device.

Any data that is being received when CTS is driven high will not be lost.



For the MCP2140, there is no Receive Data Time Window. CTS is low when the Host UART Receive Buffer has space available.

When CTS falls, the entire 29 Bytes of the Host UART Receive Buffer are available. CTS will be driven high when 23 bytes are in the Host UART Receive Buffer. Again this is to “give notice” to Host Controllers that use a transmit FIFO on the UART.



This visual shows where Application Notes are done to assist the customer in the development of their system.

We have:

1 Ap Note on the interface between the MCP215x and a Host Controller

1 Ap Note for the Fundamentals of Optical Transceivers

1 Ap Note on Programming Primary Device
Programming the Palm OS for Embedded IR Applications

In Development are:

2 Ap Notes on Programming Primary Device

Programming the PocketPC OS for Embedded IR Applications
Programming the Windows XP OS for Embedded IR Applications

MICROCHIP
WebSeminars

Summary

- Microchip's family of IrDA® Standard Protocol Handles enables wireless connectivity to easily be added to embedded applications.
- This universal interface allows commonly available low cost devices, such as PDAs, to interface to the embedded system.

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 21

Microchip's IrDA Standard Protocol Handler devices can easily be added to systems currently using a serial connection with flow control. This when allows these embedded systems to interface to the many existing devices with an IrDA port.

List of some of the available Application Notes, Data Sheets, and other reference material is available on the next couple of slides. For a complete list of Information Please visit the Infrared Communication Design page of the Microchip Web Site.

Documentation Support

- Data sheets:
  - MCP2150 Data Sheet - DS21655A
  - MCP2155 Data Sheet - DS21690A
  - MCP2140 Data Sheet - DS21790A
  - MCP2120 Data Sheet - DS21618A
- Application Notes:
  - AN858 - Interfacing the MCP215x to a Host Controller
  - AN888 - Programming the Palm OS® for Embedded IR Applications
  - AN243 - Fundamentals of the Infrared Physical Layer
  - AN758 - Using the MCP2150 to add IrDA Standard Wireless Connectivity
  - AN756 - Using the MCP2120 for Infrared Communication

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 22

Here is a list of documentation that is available from our website including a number of application notes.

Other Reference Material

- Microchip Infrared Connectivity Internet Design Center
  - www.microchip.com/infrared
- IrDA Standards download page http:/www.irda.org/standards/specifications
- Microsoft documentation http:/www.microsoft.com/hwdev/infrared

© 2004 Microchip Technology Incorporated. All Rights Reserved. Wireless Communication using the IrDA® Standard Protocol 23

In addition there are a number of web sites that offer additional information on wireless communication using the IrDA Standard.

The Microchip name and logo, the Microchip logo, are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
PICDEM is a trademark of Microchip Technology Incorporated in the U.S.A. and other countries.

IrDA is a registered trademark of the Infrared Data Association

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

All rights reserved. Copyright © 2003 Microchip Technology Incorporated.
Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.