



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



## IrDA<sup>®</sup> Standard Protocol Stack Controller Supporting DCE Applications

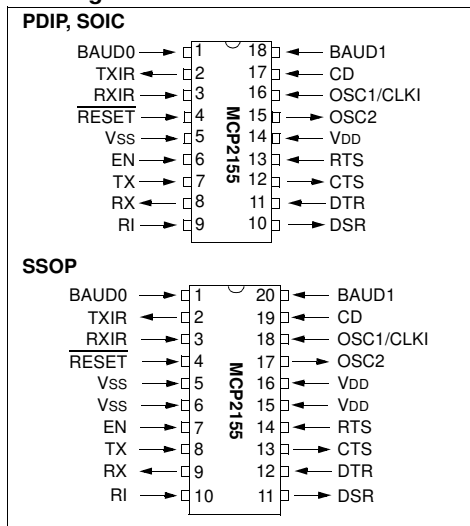
### Features

- Implements the IrDA<sup>®</sup> standard including:
  - IrLAP
  - IrLMP
  - IAS
  - TinyTP
  - IrCOMM (9-wire "cooked" service class)
- Provides IrDA standard physical signal layer support including:
  - Bi-directional communication
  - CRC implementation
  - Data communication rates up to 115.2 kbaud
- Includes UART to IrDA standard bit encoder/decoder functionality:
  - Easily interfaces to industry standard UARTs and infrared transceivers
- UART interface for connecting to Data Communicating Equipment (DCE) systems
- Transmit/Receive formats (bit width) supported:
  - 1.63  $\mu$ s
- Hardware baud rate selection for UART
  - 9.6 kbaud
  - 19.2 kbaud
  - 57.6 kbaud
  - 115.2 kbaud
- Infrared baud rates supported
  - 9.6 kbaud
  - 19.2 kbaud
  - 38.4 kbaud
  - 57.6 kbaud
  - 115.2 kbaud
- 64 Byte Data Packet Size
- Programmable Device ID String
- Operates as Secondary Device

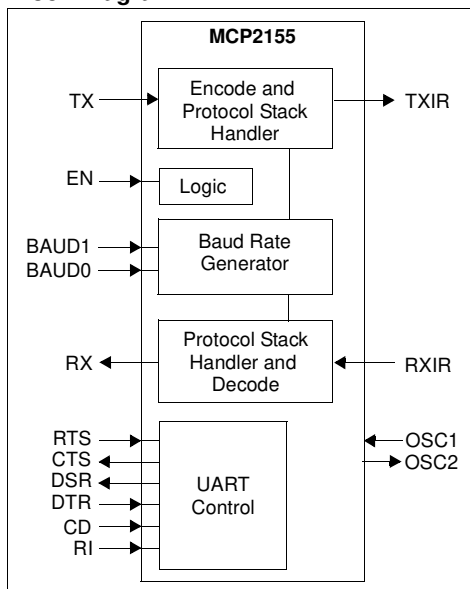
### CMOS Technology

- Low-power, high-speed CMOS technology
- Fully static design
- Low voltage operation
- Industrial temperature range
- Low power consumption
  - < 1 mA @ 3.3V, 11.0592 MHz (typical)
  - 3 mA typical @ 5.0V when disabled

### Pin Diagrams



### Block Diagram



# MCP2155

---

---

NOTES:

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the following device:

- MCP2155

The MCP2155 is a cost effective, low pin count (18-pin), easy to use device for implementing IrDA standard wireless connectivity. The MCP2155 handles for the IrDA standard protocol “stack” plus bit encoding/decoding. The MCP2155 operates in Data Communication Equipment (DCE) applications and sits between a UART and an infrared optical transceiver.

The Serial interface baud rates are user selectable to one of four IrDA standard baud rates between 9600 baud and 115.2 kbaud (9600, 19200, 57600, 115200). The IR baud rates are user selectable to one of five IrDA standard baud rates between 9600 baud and 115.2 kbaud (9600, 19200, 38400, 57600, 115200). The serial interface baud rate will be specified by the BAUD1:BAUD0 pins, while the IR baud rate is specified by the Host Controller. This means that the baud rates do not need to be the same.

The MCP2155 encodes an asynchronous serial data stream, converting each data bit to the corresponding infrared (IR) formatted pulse. IR pulses that are received are decoded, and then handled by the protocol handler state machine. The protocol handler will then send the appropriate data bytes to the host controller in UART formatted serial data.

The MCP2155 supports “point-to-point” applications. That is one Primary device and one Secondary device. The MCP2155 is a secondary device and does not support “multi-point” applications.

Sending data using IR light requires some hardware and the use of specialized communications protocols. These protocols and hardware requirements are described in detail by the IrDA standard specifications. The encoding/decoding functionality of the MCP2155 is designed to be compatible with the physical layer component of the IrDA standard. This part of the standard is often referred to as “IrPHY”.

The complete IrDA standard specifications are available for download from the IrDA website ([www.IrDA.org](http://www.IrDA.org)).

# MCP2155

## 1.1 Applications

The MCP2155 IrDA standard protocol stack controller supporting the IrDA standard for IrCOMM 9-wire “cooked” service class which enables embedded system designers the easiest way to implement IrDA standard wireless connectivity. Figure 1-1 shows a typical application block diagram. Table 1-2 shows the pin definitions.

**TABLE 1-1: OVERVIEW OF FEATURES**

Features	MCP2155
Serial Communications:	UART, IR
Baud Rate Selection:	Hardware
Low Power Mode:	Yes
Resets (and Delays):	RESET, POR (PWRT and OST)
Packages:	18-pin DIP, SOIC, 20-pin SSOP

Infrared communication is a wireless two-way data connection using infrared light generated by low-cost transceiver signaling technology. This provides reliable communication between two devices.

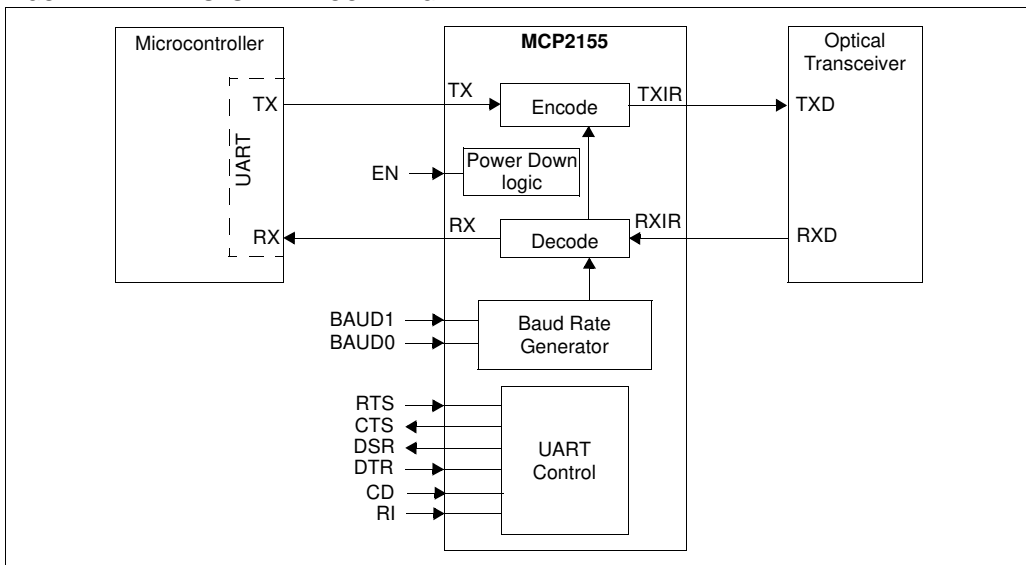
Infrared technology offers:

- Universal standard for connecting portable computing devices
- Easy, effortless implementation
- Economical alternative to other connectivity solutions
- Reliable, high speed connection
- Safe to use in any environment; can even be used during air travel
- Eliminates the hassle of cables
- Allows PC's and other electronic device's (such as PDA's, cell phones, ....) to communicate with each other
- Enhances mobility by allowing users to easily connect

The MCP2155 allows the easy addition of IrDA standard wireless connectivity to any embedded application that uses serial data. Figure 1-1 shows typical implementation of the MCP2155 in an embedded system.

The IrDA protocols for printer support are not included in the IrCOMM 9-wire “cooked” service class.

**FIGURE 1-1: SYSTEM BLOCK DIAGRAM**



**TABLE 1-2: PIN DESCRIPTION**

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	SOIC	SSOP			
BAUD0	1	1	1	I	ST	BAUD1:BAUD0 specify the baud rate of the device. For more information see <a href="#">Section 2.5.1</a> .
TXIR	2	2	2	O	—	Asynchronous transmit to Infrared transceiver.
RXIR	3	3	3	I	ST	Asynchronous receive from Infrared transceiver.
$\overline{\text{RESET}}$	4	4	4	I	ST	Resets the device.
Vss	5	5	5, 6	—	P	Ground reference for logic and I/O pins.
EN	6	6	7	I	TTL	Device enable. 1 = Device is enabled 0 = Device is disabled (low power)
TX	7	7	8	I	TTL	Asynchronous receive; from Host Controller UART.
RX	8	8	9	O	—	Asynchronous transmit; to Host Controller UART.
RI	9	9	10	I	TTL	Ring Indicator. The state of this bit is communicated to the IrDA Primary Device. 1 = No Ring Indicate Present 0 = Ring Indicate Present
DSR	10	10	11	O	—	Data Set Ready. Indicates that the MCP2155 has established a valid link with a Primary Device. This signal is locally emulated and not related to the DTR bit of the IrDA Primary Device. 1 = An IR link has not been established (No IR Link) 0 = An IR link has been established (IR Link)
DTR	11	11	12	I	TTL	Data Terminal Ready. Indicates that the Embedded device connected to the MCP2155 is ready for IR data. The state of this bit is communicated to the IrDA Primary Device, via the IrDA bit carried by IrCOMM. 1 = Embedded device not ready 0 = Embedded device ready At device power-up, this signal is used with RTS to enter device ID programming. 1= Enter Device ID programming mode (if RTS is cleared) 0= Do not enter Device ID programming mode
CTS	12	12	13	O	—	Clear to Send. Indicates that the MCP2155 is ready to receive data form the Host Controller. This signal is locally emulated and not related to the CTS/RTS bit of the IrDA Primary Device. 1 = Host Controller should not send data 0 = Host Controller may send data
RTS	13	13	14	I	TTL	Request to Send. Indicates that the Host Controller is ready to receive data from the MCP2155. This signal is locally emulated and not related to the CTS/RTS bit of the IrDA Primary Device. 1 = Host Controller not ready to receive data 0 = Host Controller ready to receive data At device power-up, this signal is used with CTS to enter device ID programming. 1= Do not enter Device ID programming mode 0= Enter Device ID programming mode (if DTR is set)
VDD	14	14	15, 16	—	P	Positive supply for logic and I/O pins.
OSC2	15	15	17	O	—	Oscillator crystal output.
OSC1/CLKIN	16	16	18	I	CMOS	Oscillator crystal input/external clock source input.

Legend: TTL = TTL compatible input  
I = Input  
P = Power

ST = Schmitt Trigger input with CMOS levels  
O = Output  
CMOS = CMOS compatible input

# MCP2155

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	SOIC	SSOP			
CD	17	17	19	I	ST	Carrier Detect. The state of this bit is communicated to the IrDA Primary Device. 1 = No Carrier Present 0 = Carrier Present
BAUD1	18	18	20	I	ST	BAUD1:BAUD0 specify the baud rate of the device. For more information see <a href="#">Section 2.5.1</a> .

Legend: TTL = TTL compatible input      ST = Schmitt Trigger input with CMOS levels  
I = Input                                      O = Output  
P = Power                                      CMOS = CMOS compatible input

## 1.1.1 SIGNAL DIRECTIONS

[Table 1-3](#) shows the direction of the MCP2155 signals. The MCP2155 is designed for use in Data Communication Equipment (DCE) applications.

**TABLE 1-3: MCP2155 SIGNAL DIRECTION**

DB-9 Pin #	Signal	Direction	Comment
1	CD	HC → MCP2155	Carrier Detect
2	RX	MCP2155 → HC	Received Data
3	TX	HC → MCP2155	Transmit Data
4	DTR	HC → MCP2155	Data Terminal Ready
5	GND	—	Ground
6	DSR	MCP2155 → HC	Data Set Ready
7	RTS	HC → MCP2155	Request to Send
8	CTS	MCP2155 → HC	Clear to Send
9	RI	HC → MCP2155	Ring Indicator

Legend: HC = Host Controller

## 2.0 DEVICE OPERATION

The MCP2155 is a cost effective, low pin count (18-pin), easy to use device for implementing IrDA standard wireless connectivity. The MCP2155 provides support for the IrDA standard protocol “stack” plus bit encoding/decoding. The Serial interface and IR baud rates are independently selectable.

### 2.1 Power-up

Any time that the device is powered up ([parameter D003](#)), the Power-up timer delay ([parameter 33](#)) occurs, followed by an Oscillator Start-up Timer (OST) delay ([parameter 32](#)). After these two delays complete, communication with the device may be initiated. This communication is from both the infrared transceiver’s side as well as the controller’s UART interface.

### 2.2 Device Reset

The MCP2155 is forced into the reset state when the RESET pin is in the low state. After the RESET pin is brought to a high state, the Device Reset sequence occurs. Once the sequence completes, functional operation begins.

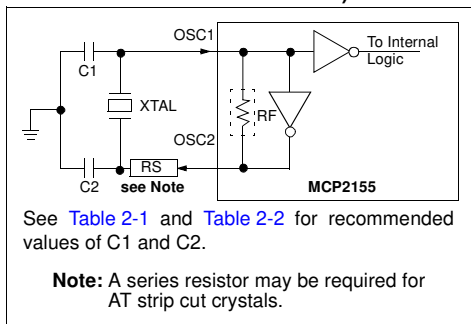
### 2.3 Clock Source

The MCP2155 requires a clock source to operate. The frequency of this clock is 11.0592 MHz (electrical specification [parameter 1A](#)). This clock can be supplied by either a crystal/resonator or as an external clock input.

#### 2.3.1 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

A crystal or ceramic resonator can be connected to the OSC1 and OSC2 pins to establish oscillation ([Figure 2-1](#)). The MCP2155 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer’s specifications.

**FIGURE 2-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR)**



**TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

Freq	OSC1 (C1)	OSC2 (C2)
11.0592 MHz	10 - 22 pF	10 - 22 pF

Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

**TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

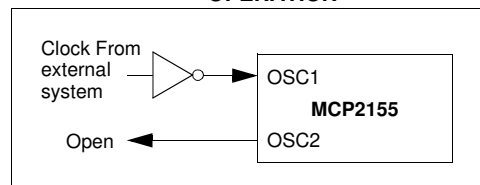
Freq	OSC1 (C1)	OSC2 (C2)
11.0592 MHz	15 - 30 pF	15 - 30 pF

Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

#### 2.3.2 EXTERNAL CLOCK IN

For applications where a clock is already available elsewhere, users may directly drive the MCP2155 provided that this external clock source meets the AC/DC timing requirements listed in [Section 4.3](#). [Figure 2-2](#) below shows how an external clock circuit should be configured.

**FIGURE 2-2: EXTERNAL CLOCK INPUT OPERATION**





# MCP2155

---

## 2.4 Bit Clock

The device crystal is used to derive the communication bit clock (BITCLK). There are 16 BITCLKs for each bit time. The BITCLKs are used for the generation of the start bit and the eight data bits. The stop bit uses the BITCLK when the data is transmitted (not for reception).

This clock is a fixed frequency, and has minimal variation in frequency (specified by crystal manufacturer).

## 2.5 UART Interface

The UART interface communicates with the "controller". This interface is a half duplex interface, meaning that the system is either transmitting or receiving, but not both at the same time.

### 2.5.1 BAUD RATE

The baud rate for the MCP2155 serial port (the TX and RX pins) is configured by the state of the BAUD1 and BAUD0 pins. These two device pins are used to select the baud rate that the MCP2155 will transmit and receive serial data (not IR data). [Table 2-3](#) shows the baud rate configurations.

**TABLE 2-3: SERIAL BAUD RATE SELECTION VS. FREQUENCY**

BAUD1:BAUD0	Baud Rate @ 11.0592 MHz	Bit Rate
00	9600	Fosc / 1152
01	19200	Fosc / 576
10	57600	Fosc / 192
11	115200	Fosc / 96

### 2.5.2 TRANSMITTING

When the controller sends serial data to the MCP2155, the controller's baud rate is required to match the baud rate of the MCP2155's serial port.

### 2.5.3 RECEIVING

When the controller receives serial data from the MCP2155, the controller's baud rate is required to match the baud rate of the MCP2155's serial port.

## 2.6 Modulation

The data that the MCP2155 UART received (on the TX pin) that needs to be transmitted (on the TXIR pin), will need to be modulated. This modulated signal drives the IR transceiver module. Figure 2-3 shows the encoding of the modulated signal.

**Note:** The signal on the TXIR pin does not actually line up in time with the bit value that was transmitted on the TX pin as shown in Figure 2-3. The TX bit value is shown to represent the value to be transmitted on the TXIR pin.

Each bit time is comprised of 16-bit clocks. If the value to be transmitted (as determined by the TX pin) is a logic low, then the TXIR pin will output a low level for 7-bit clock cycles, a logic high level for 3-bit clock cycles or a minimum of 1.6  $\mu\text{s}$  (see parameter IR121), and then the remaining 6-bit clock cycles (or difference up to the 16-bit clock time) will be low. If the value to transmit is a logic high, then the TXIR pin will output a low level for the entire 16-bit clock cycles.

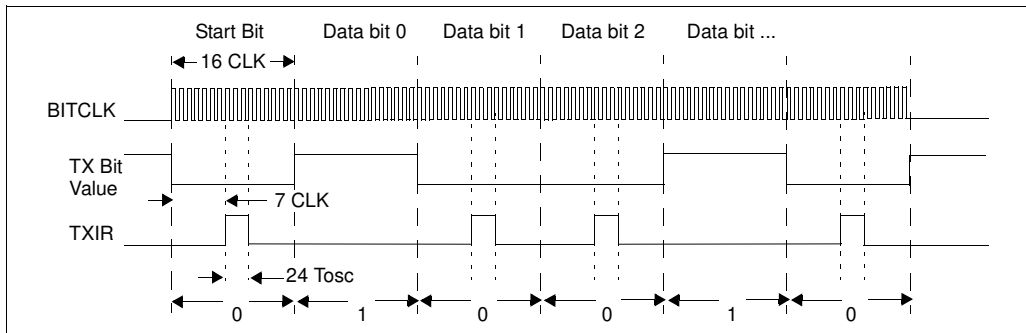
## 2.7 Demodulation

The modulated signal (data) from the IR transceiver module (on RXIR pin) needs to be demodulated to form the received data (on RX pin). After demodulation of the data byte occurs, the data that is received is transmitted by the MCP2155 UART (on the RX pin). Figure 2-4 shows the decoding of the modulated signal.

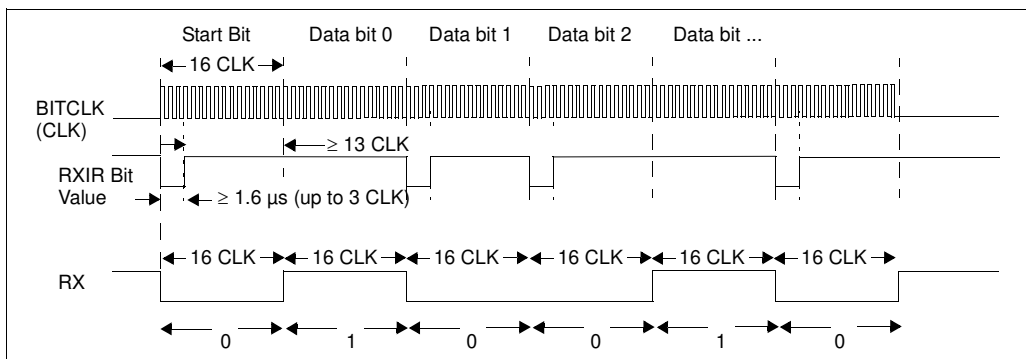
**Note:** The signal on the RX pin does not actually line up in time with the bit value that was received on the RXIR pin as shown in Figure 2-4. The RXIR bit value is shown to represent the value to be transmitted on the RX pin.

Each bit time is comprised of 16 bit clocks. If the value to be received is a logic low, then the RXIR pin will be a low level for the first 3-bit clock cycles or a minimum of 1.6  $\mu\text{s}$ , and then the remaining 13-bit clock cycles (or difference up to the 16-bit clock time) will be high. If the value to be received is a logic high, then the RXIR pin will be a high level for the entire 16-bit clock cycles. The level on the RX pin will be in the appropriate state for the entire 16 clock cycles.

**FIGURE 2-3: ENCODING**



**FIGURE 2-4: DECODING**



# MCP2155

## 2.8 Minimizing Power

The device can be placed in a low power mode by disabling the device (holding the EN pin at the low state). The internal state machine is monitoring this pin for a low level, and once this is detected the device is disabled and enters into a low power state.

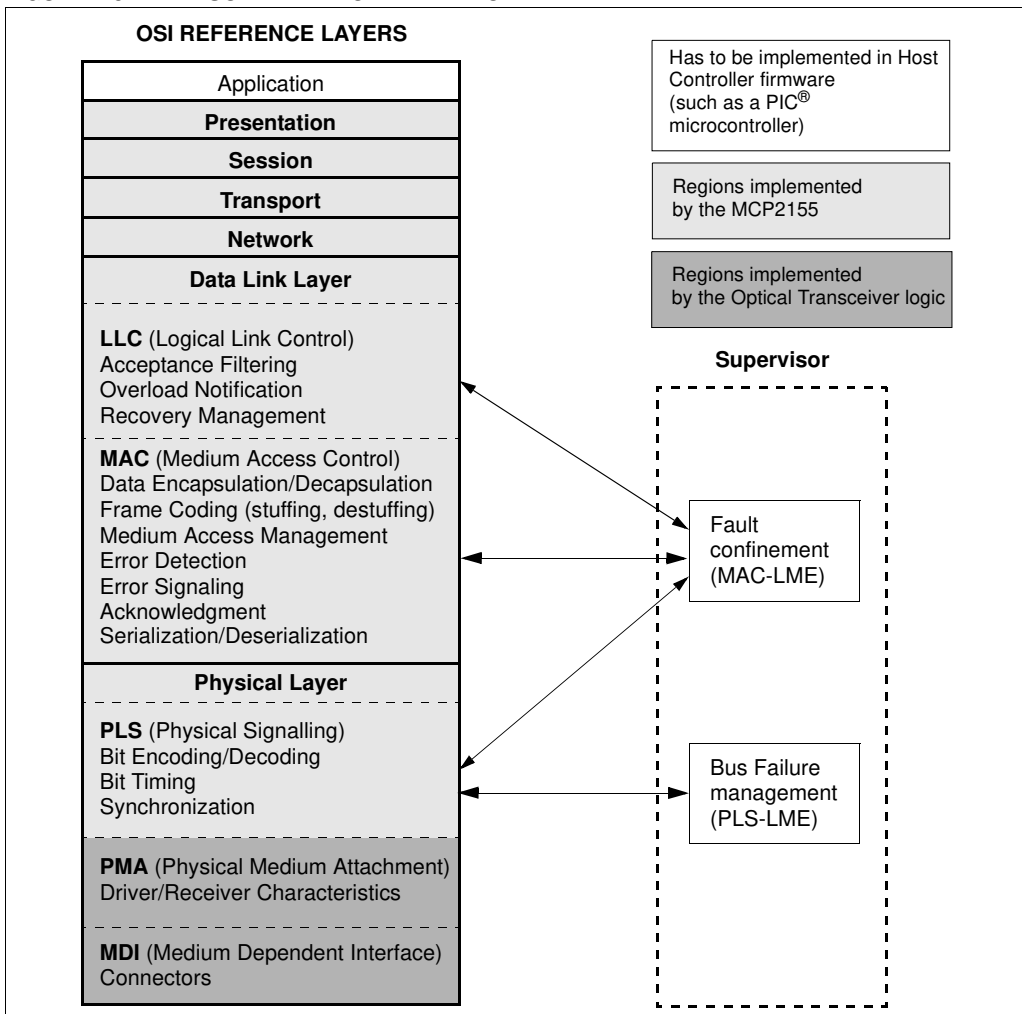
### 2.8.1 RETURNING TO DEVICE OPERATION

When disabled, the device is in a low power state. When the EN pin is brought to a high level, the device will return to the operating mode. The device requires a delay of 1024 T<sub>osc</sub> before data may be transmitted or received.

## 2.9 Network Layering Reference Model

Figure 2-5 shows the ISO Network Layering Reference Model. The shaded areas are implemented by the MCP2155, the cross-hatched area is implemented by an infrared transceiver, and the unshaded areas need to be implemented by the Host controller.

**FIGURE 2-5: ISO REFERENCE LAYER MODEL**



The IrDA standard specifies the following protocols:

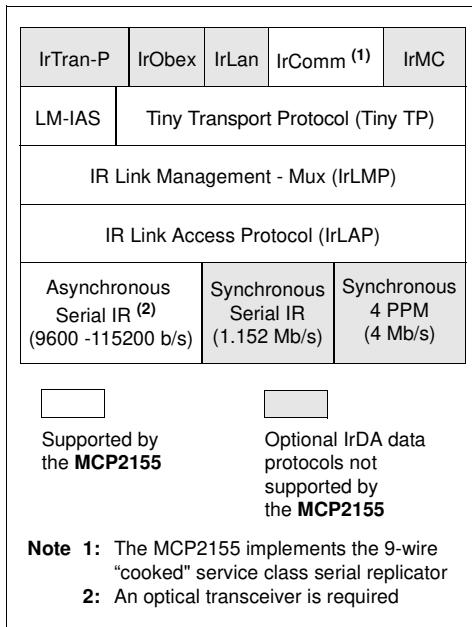
- Physical Signaling Layer (PHY)
- Link Access Protocol (IrLAP)
- Link Management Protocol/Information Access Service (IrLMP/IAS)

The IrDA data lists optional protocols. These are:

- Tiny TP
- IrTran-P
- IrOBEX
- IrLAN
- IrCOMM
- IrMC
- IrDA Lite

Figure 2-6 shows the IrDA data protocol stack and which components are implemented by the MCP2155.

**FIGURE 2-6: IRDA DATA - PROTOCOL STACKS**



## 2.9.1 IrDA DATA PROTOCOLS SUPPORTED BY MCP2155

The MCP2155 supports these required IrDA standard protocols:

- Physical Signaling Layer (PHY)
- Link Access Protocol (IrLAP)
- Link Management Protocol/Information Access Service (IrLMP/IAS)

The MCP2155 also supports some of the optional protocols for IrDA data. The optional protocols that the MCP2155 implements are:

- Tiny TP
- IrCOMM

### 2.9.1.1 Physical Signal Layer (PHY)

The MCP2155 provides the following Physical Signal Layer specification support:

- Bi-directional communication
- Data Packets are protected by a CRC
  - 16-bit CRC for speeds up to 115.2 kbaud
- Data Communication Rate
  - 9600 baud minimum data rate (with primary speed/cost steps of 115.2 kbaud)

The following Physical Layer Specification is dependant on the optical transceiver logic used in the application. The specification states:

- Communication Range, which sets the end user expectation for discovery, recognition and performance
  - Continuous operation from contact to at least 1 meter (typically 2 meters can be reached)
  - A low power specification reduces the objective for operation from contact to at least 20 cm (low power and low power) or 30 cm (low power and standard power).

# MCP2155

## 2.9.1.2 IrLAP

The MCP2155 supports the IrLAP protocol. The IrLAP protocol provides:

- Management of communication processes on the link between devices.
- A device-to-device connection for the reliable, ordered transfer of data.
- Device discover procedures.
- Hidden node handling. (Not supported by MCP2155)

Figure 2-7 identifies the key parts and hierarchy of the IrDA protocols. The bottom layer is the Physical layer, IrPHY. This is the part that converts the serial data to and from pulses of IR light. IR transceivers can't transmit and receive at the same time. The receiver has to wait for the transmitter to finish sending. This is sometimes referred to as a "Half-Duplex" connection. The IR Link Access Protocol (IrLAP) provides the structure for packets or "frames" of data to emulate data that would normally be free to stream back and forth.

**FIGURE 2-7: IRDA STANDARD PROTOCOL LAYERS**

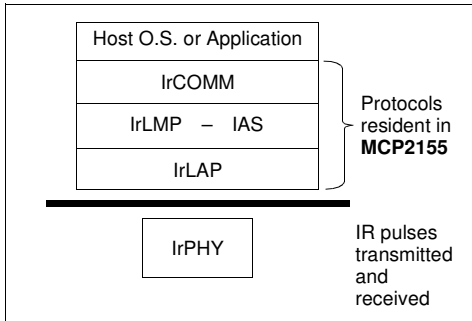


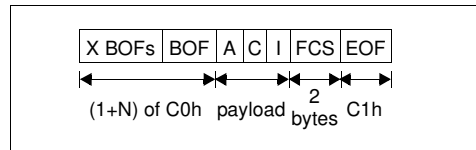
Figure 2-8 shows how the IrLAP frame is organized. The frame is preceded by some number of Beginning of Frame characters, (BOFs). The value of the BOF is generally 0xC0, but 0xFF may be used if the last BOF character is a 0xC0. The purpose of multiple BOFs is to give the other station some warning that a frame is coming.

The IrLAP frame begins with an address byte ("A" field), then a control byte ("C" field). The control byte is used to differentiate between different types of frames and is also used to count frames. Frames can carry status, data, or commands. The IrLAP protocol has a command syntax of its own, and these commands are part of the control byte. Lastly, IrLAP frames carry data. This data is the information or "I" field. The integrity of the frame is ensured with a 16-bit CRC, referred to as the Frame Check Sequence (FCS). The 16-bit CRC value is transmitted LSB first. The end of the frame is marked with an EOF character which is always a 0xC1. The frame structure described here is used for all versions of IrDA protocols used for serial wire replacement for speeds up to 115.2 kbaud.

**Note 1:** Another IrDA standard which is entering general usage is IR Object Exchange (IrOBEX). This standard is not used for serial connection emulation.

**2:** IrDA communication standards faster than 115.2 kbaud use a different CRC method and physical layer.

**FIGURE 2-8: IRLAP FRAME**



In addition to defining the frame structure, IrLAP provides the "housekeeping" function of opening and closing connections, and maintaining connections once they're open. The critical parameters that determine the performance of the link are part of this function. These parameters control how many BOFs are used, identify the speed of the link, how fast either party may change from receiving to transmitting, etc. IrLAP has the responsibility of negotiating these parameters to the highest common set so that both sides can communicate as fast and as reliably as possible.

## 2.9.1.3 IrLMP

The MCP2155 implements the IrLMP protocol. The IrLMP protocol provides:

- Multiplexing of the IrLAP layer. This allows multiple channels above an IrLAP connection
- Protocol and service discovery. This is via the Information Access Service (IAS)

When two devices that contain the IrDA standard feature are connected, there is generally one device that has something to do, and the other device has the resource to do it. For example, a laptop may have a job to print and an IrDA standard compatible printer has the resources to print it. In IrDA standard terminology, the laptop is a Primary device and the printer is the Secondary device. When these two devices connect, the Primary device must determine the capabilities of the Secondary device to determine if the Secondary device is capable of doing the job. This determination is made by the Primary device asking the Secondary device a series of questions. Depending on the answers to these questions the Primary device may or may not elect to connect to the Secondary device.

The queries from the Primary device are carried to the Secondary device using IrLMP. The responses to these queries can be found in the Information Access Service (IAS) of the Secondary device. The IAS is a list of the resources of the Secondary device. The Primary device compares the IAS responses with its requirements and then makes the decision if a connection should be made.

The MCP2155 identifies itself to the Primary device as a modem.

**Note:** The MCP2155 identifies itself as a modem to ensure that it is identified as a serial device with a limited amount of memory.

## 2.9.1.4 Link Management - Information Access Service (LM-IAS)

The MCP2155 implements the LM-IAS. Each LM-IAS entity maintains an information data base to provide:

- Information on services for other devices that contain the IrDA standard feature (Discovery)
- Information on services for the device itself
- Remote accessing of another device's information base

This is required so that clients on a remote device can find configuration information needed to access a service.

## 2.9.1.5 Tiny TP

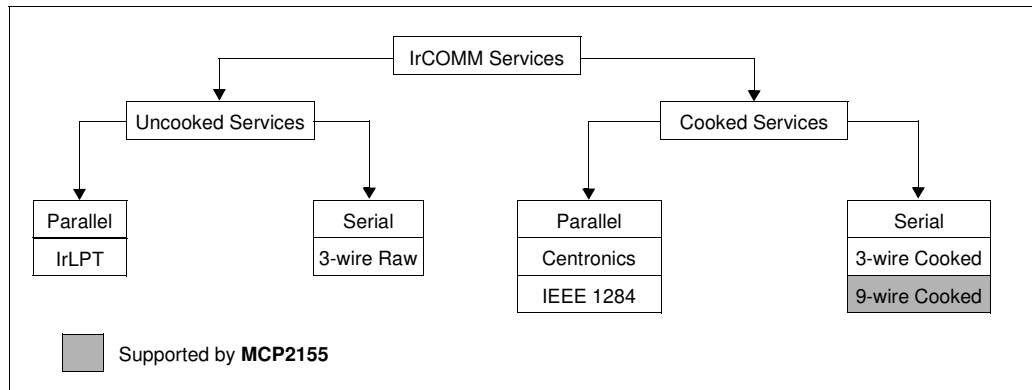
Tiny TP provides the flow control on IrLMP connections. An optional service of Segmentation and Reassembly can be handled.

## 2.9.1.6 IrCOMM

IrCOMM provides the method to support serial and parallel port emulation. This is useful for legacy COM applications, such as printers and modem devices.

The IrCOMM standard is simply a syntax that allows the Primary device to consider the Secondary device as a serial device. IrCOMM allows for emulation of serial or parallel (printer) connections of various capabilities. The MCP2155 supports the 9-wire "cooked" service class of IrCOMM. Other service classes supported by IrCOMM are shown in [Figure 2-9](#).

**FIGURE 2-9: IRCOMM SERVICE CLASSES**



# MCP2155

---

## 2.9.2 OTHER OPTIONAL IrDA DATA PROTOCOLS

Other IrDA data protocols have been developed to specific application requirements. These optional protocols are not supported in the MCP2155. These IrDA data protocols are briefly described in the following sub-sections. For additional information, please refer to the IrDA website ([www.IrDA.org](http://www.IrDA.org)).

### 2.9.2.1 IrTran-P

IrTran-P provides the protocol to exchange images with digital image capture devices/cameras.

### 2.9.2.2 IrOBEX

IrOBEX provides OBJECT EXchange services. This is similar to HTTP.

### 2.9.2.3 IrLAN

IrLAN describes a protocol to support IR wireless access to a Local Area Network (LAN).

### 2.9.2.4 IrMC

IrMC describes how mobile telephony and communication devices can exchange information. This information includes phonebook, calendar, and message data.

Also how call control and real-time voice are handled (RTCON).

### 2.9.2.5 IrDA Lite

IrDA Lite describes how to reduce the application code requirements, while maintaining compatibility with the full implementation.

## 2.9.3 HOW DEVICES CONNECT

When two devices implementing the IrDA standard feature establish a connection using the IrCOMM protocol, the process is analogous to connecting two devices with serial ports using a cable. This is referred to as a "point-to-point" connection. This connection is limited to half-duplex operation because the IR transceiver cannot transmit and receive at the same time. The purpose of the IrDA protocol is to allow this half-duplex link to emulate, as much as possible, a full-duplex connection. In general, this is done by dividing the data into "packets", or groups of data. These packets can then be sent back and forth when needed without risk of collision. The rules of how and when these packets are sent constitute the IrDA protocols. The MCP2155 supports elements of this IrDA protocol to communicate with other IrDA standard compatible devices.

When a wired connection is used, the assumption is made that both sides have the same communications parameters and features. A wired connection has no need to identify the other connector, because it is assumed that the connectors are properly connected. In the IrDA standard, a connection process has been defined to identify other IrDA compatible devices and establish a communication link. There are three steps that these two devices go through to make this connection. These are:

- Normal Disconnect Mode (NDM)
- Discovery Mode
- Normal Connect Mode (NCM)

Figure 2-10 shows the connection sequence.

### 2.9.3.1 Normal Disconnect Mode (NDM)

When two IrDA standard compatible devices come into range they must first recognize each other. The basis of this process is that one device has some task to accomplish and the other device has a resource needed to accomplish this task. One device is referred to as a Primary device and the other is referred to as a Secondary device. This distinction between Primary device and Secondary device is important. It is the responsibility of the Primary device to provide the mechanism to recognize other devices. So the Primary device must first poll for nearby IrDA standard compatible devices. During this polling, the default baud rate of 9600 baud is used by both devices.

For example, if you want to print from an IrDA equipped laptop to an IrDA printer utilizing the IrDA standard feature, you would first bring your laptop in range of the printer. In this case, the laptop is the one that has something to do and the printer has the resource to do it. The laptop is called the Primary device and the printer is the Secondary device. Some data-capable cellphones have IrDA standard infrared ports. If you used such a cellphone with a Personal Dig-

ital Assistants (PDAs), the PDA that supports the IrDA standard feature would be the Primary device and the cellphone would be the Secondary device.

When a Primary device polls for another device, then a nearby Secondary device may respond. When a Secondary device responds, the two devices are defined to be in the Normal Disconnect Mode (NDM) state. NDM is established by the Primary device broadcasting a packet and waiting for a response. These broadcast packets are numbered. Usually 6 or 8 packets are sent. The first packet is number 0, the last packet is usually number 5 or 7. After all the packets are sent, the Primary device then sends an ID packet which is not numbered.

The Secondary device waits for these packets, and then responds to one of the packets. The packet it responds to determines the "time slot" to be used by the Secondary device. For example, if the Secondary device responds after packet number 2, then the Secondary device will use time slot 2. If the Secondary device responds after packet number 0, then the Secondary device will use time slot 0. This mechanism allows the Primary device to recognize as many nearby devices as there are time slots. The Primary device will continue to generate time slots and the Secondary device should continue to respond, even if there's nothing to do.

**Note 1:** The MCP2155 can only be used to implement a Secondary device.

**2:** The MCP2155 supports a system with only one Secondary device having exclusive use of the IrDA standard infrared link (known as "point-to-point" communication).

**3:** The MCP2155 always takes time slot 2.

**4:** If another Secondary device is nearby, the Primary device may fail to recognize the MCP2155, or the Primary device may not recognize either of the devices.

During NDM, the MCP2155 handles all of the responses to the Primary device (see Figure 2-10), without any communication with the Host controller. The Host controller is inhibited by the CTS signal, of the MCP2155, from sending data to the MCP2155.



# MCP2155

## 2.9.3.2 Discovery Mode

Discovery mode allows the Primary device to determine the capabilities of the Secondary device. The discovery mode is entered after the MCP2155 (Secondary device) has sent an XID response to the Primary device, and the Primary device has completed sending the XIDs and then sends a Broadcast ID. If this sequence is not completed, then a Primary device and a Secondary device can stay in NDM indefinitely.

When the Primary device has something to do, it then initiates Discovery. Discovery has two parts. these are:

- Link initialization
- Resource determination

The first step is for the Primary device and Secondary device to determine and then adjust to each other's hardware capabilities. These capabilities are parameters like:

- Data rate
- Turn around time
- Number of packets without a response
- How long to wait before disconnecting

Both the Primary device and Secondary device begin communications at 9600 baud, which is the default baud rate. The Primary device sends its parameters, then the Secondary device responds with its parameters. For example, if the Primary supports all data rates up to 115.2 kbaud and the Secondary device only supports 19.2 kbaud then the link will be established at 19.2 kbaud.

**Note:** The MCP2155 is limited to a data rate of 115.2 kbaud.

After the hardware parameters are established, the Primary device must determine if the Secondary device has the resources it requires. If the Primary device has a job to print, then it has to know if it's talking to a printer, not a modem or other device. This determination is made using the Information Access Service, (IAS). The job of the Secondary device is to respond to IAS queries made by the Primary device. The Primary device must ask a series of questions like:

- What is the name of your service?
- What is the address of this service?
- What are the capabilities of this device?

When all the Primary device's questions are answered, the Primary device can access the service provided by the Secondary device.

During Discovery Mode, the MCP2155 handles all of the responses to the Primary device (see [Figure 2-10](#)), without any communication with the Host controller. The Host controller is inhibited by the CTS signal, of the MCP2155, from sending data to the MCP2155.

## 2.9.3.3 Normal Connect Mode (NCM)

After discovery has been completed, the Primary device and MCP2155 (Secondary device) can freely exchange data.

The MCP2155 can receive IR data or serial data, but not both at the same time. The MCP2155 uses a hardware handshake to stop the local serial port from sending data while the MCP2155 is receiving IR data.

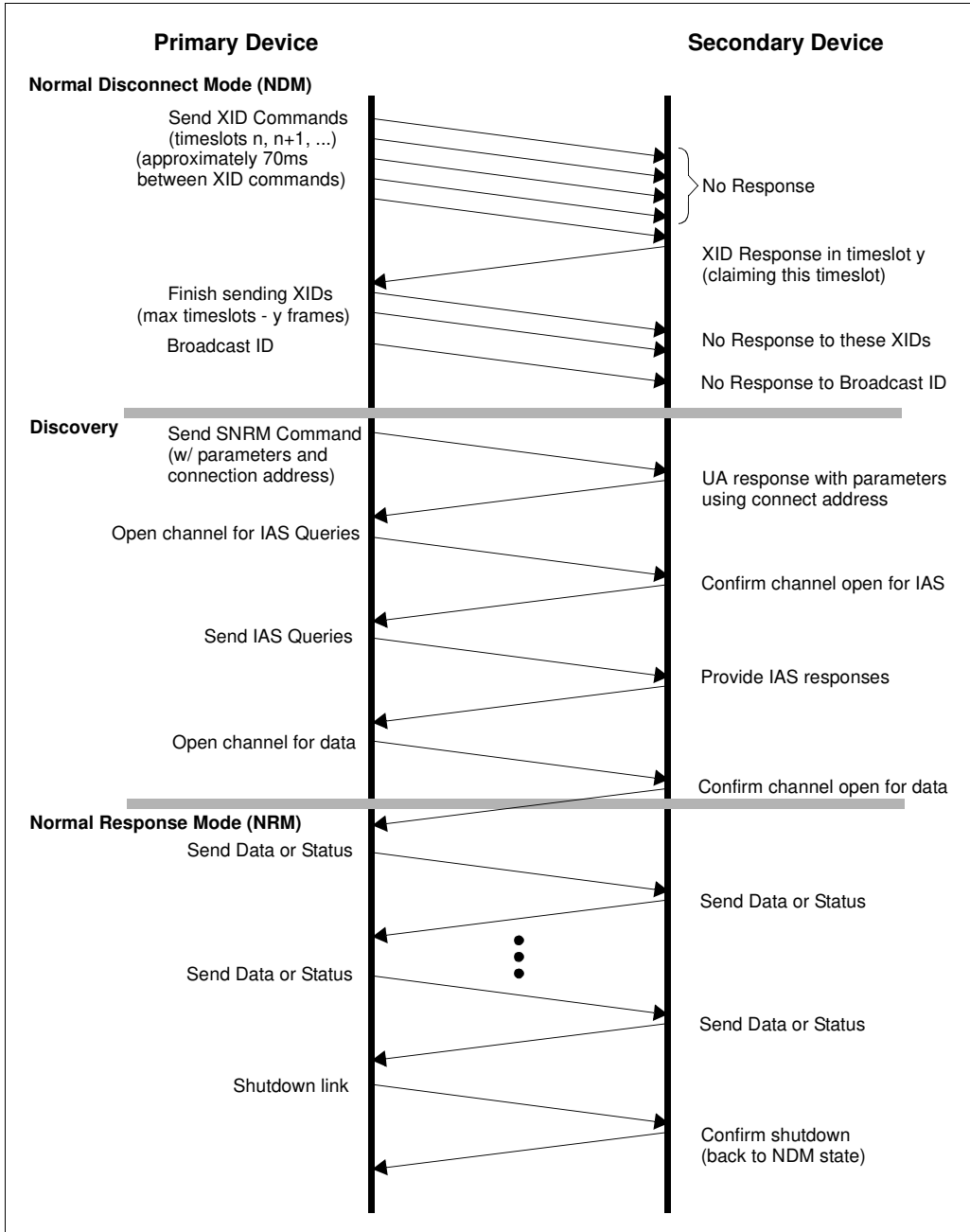
**Note:** Data loss will result if this hardware handshake is not observed.

Both the Primary device and MCP2155 (Secondary device) check to make sure that data packets are received by the other without errors. Even when data is required to be sent the Primary device and Secondary device will still exchange packets just to make sure that the connection hasn't unexpectedly been dropped. When the Primary device has finished, it then transmits the close link command to the MCP2155 (Secondary device). The MCP2155 will confirm the close link command, and then both the Primary device and MCP2155 (Secondary device) will revert to the NDM state.

**Note:** If the NCM mode is unexpectedly terminated for any reason (including the Primary device not issuing a close link command), the MCP2155 will revert to the NDM state 10 seconds after the last frame has been received.

It is the responsibility of the Host controller program to understand the meaning of the data received, and how the program should respond to it. This is the same as if the data was being received by the host controller from a UART.

**FIGURE 2-10: CONNECTION SEQUENCE**



# MCP2155

## 2.10 Operation

The MCP2155 emulates a modem connection. The application on the DCE device sees a virtual serial port. This serial port emulation is provided by the IrDA standard protocols. The link between the DCE device and the embedded application is made using the MCP2155. The connection between the MCP2155 and the embedded application should be wired as a modem connection.

The Data Set Ready (DSR) signal of the MCP2155 is used to indicate if a valid IrDA standard infrared link has been established between the MCP2155 and the Primary device (DCE host). The DSR signal should be monitored closely to make sure that any communication tasks can be completed.

To indicate that the MCP2155 has powered-up, successfully initialized, and is ready for service, monitor the CTS signal for a High level. The CTS signal is driven High during the NDM and Discovery states, and may be either High or Low during the NCM state.

The MCP2155 generates the CTS signal locally.

**Note:** The RTS and CTS signals are local emulations.

### 2.10.1 HARDWARE HANDSHAKING

The MCP2155 uses a 64-byte buffer for incoming data from the IR Host. Another 64-byte buffer is provided to buffer data from the UART serial port. When an IR packet begins the IrComm, the MCP2155 handles IR data exclusively. So the UART serial port buffer is not available. A hardware handshaking pin (CTS) is provided to inhibit the host controller from sending serial data while IR Data is being sent or received.

**Note:** When the CTS output from the IrComm is high, no data should be sent from the Host controller. The UART FIFO will store up to 2 bytes. Any additional data bytes will be lost.

### 2.10.2 BUFFERS AND THROUGHPUT

The maximum IR data rate of the MCP2155 is 115.2 kbaud. The actual throughput will be less due to several factors, the most significant of which are under the control of the developer. One factor beyond the control of the designer is the overhead associated with the IrDA standard. The MCP2155 uses a fixed data block size of 64-bytes. To carry 64 bytes of data the MCP2155 must send 72 bytes (64+8). The additional 8 bytes are used by the protocol. When the Primary device receives the frame it must wait for a minimum latency period before sending a packet of its own. This turnaround time is set by IrLAP when the parameters of the link are negotiated. A common turnaround time is 1ms, although longer and shorter times may be encountered. 1 ms represents approximately 12 byte times at a data rate of 115.2 kbaud. The minimum size frame that the Primary device can respond with is 6 bytes. The MCP2155 will add the 12 byte-time latency of its own, again assuming a 1ms latency. This means that the maximum throughput will be 64 data bytes out of a total of 64 + 38 byte times. Thus, the maximum theoretical throughput will be limited to about 64/(64+38)=63% of the IR data rate. Actual maximum throughput will be between 38.4 kbaud and 57.6 kbaud. This difference is due to processing time of the receiving station and other factors.

The most significant factor in data throughput is how well the data frames are filled. If only 1 byte is sent at a time, then the maximum throughput is 1/(1+38)=2.5% of the IR data rate. The best way to maximize throughput is to align the amounts of data with the packet size of the MCP2155. Throughput examples are shown in [Table 2-4](#).

**TABLE 2-4: IrDA STANDARD THROUGHPUT EXAMPLES @ 115.2 KBAUD**

MCP2155 Data Packet Size (Bytes)	Overhead (Bytes)	Primary Device Minimum Response (Bytes)	Primary Device Turn-around Time <sup>(1)</sup> (Bytes)	MCP2155 Turn-around Time <sup>(1)</sup> (Bytes)	Total Bytes Transmitted	Throughput % (Data/Total)
64	8	6	12	12	102	62.7%
1	8	6	12	12	39	2.6%

**Note 1:** Number of bytes calculated based on a common turnaround time of 1 ms.

## 2.11 Turnaround Latency

An IR link can be compared to a one-wire data connection. The IR transceiver can transmit or receive, but not both at the same time. A delay of one bit time is suggested between the time a byte is received and another byte is transmitted.

## 2.12 IR Port Baud Rate

The baud rate for the MCP2155 IR port (the TXIR and RXIR pins) is initially at the default rate of 9600 baud. The Host controller will determine the maximum baud rate that the MCP2155 can support. This information is used during NDM with the Primary device to set the baud rate of the IR link. The maximum IR baud rate is not required to be the same as the MCP2155's serial port (UART) baud rate (as determined by the BAUD1:BAUD0 pins).

## 2.13 Programmable Device ID

The MCP2155 has a flexible feature that allows the MCP2155 Device ID to be changed by the Host controller. The default ID is "Generic IrDA", and is stored in non-volatile electrically erasable programmable memory (EEPROM). The maximum ID String length is 19 bytes. The format of the ID EEPROM is shown in [Figure 2-11](#).

The ID String must only contain the ASCII characters from 20h to 7Ah (inclusive).

The MCP2155 enters into ID String programming when the MCP2155 exits the reset state and detects that the DTR pin is High and the RTS pin is Low.

A Host controller connected to the MCP2155 would typically do the following steps to place the MCP2155 into ID String programming mode:

1. Force the MCP2155 into reset (RESET pin forced Low).
2. Force the DTR pin High and the RTS pin Low.
3. Release the MCP2155 from reset (RESET pin forced High).

Once the MCP2155 is ready to receive data, the CTS pin will be forced low. Data may now be transferred, following the format in [Figure 2-11](#). The CTS pin determines the flow control, and the Host controller must monitor this signal to ensure that the data byte may be sent.

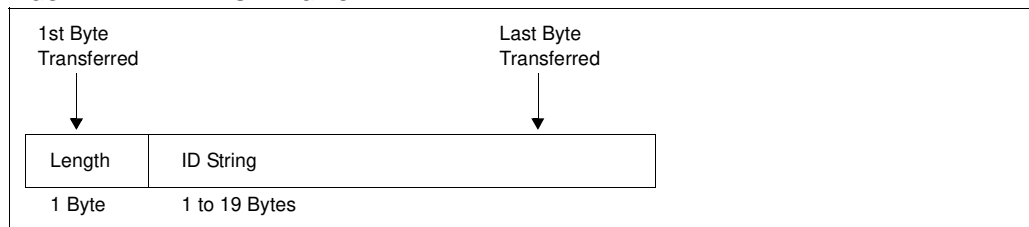
Once the Host controller has sent its last byte, the DTR pin must be set Low. This ensures that if another reset occurs, the MCP2155 will not re-enter ID String programming mode. The MCP2155 uses the String Length (1st byte transmitted) to determine when to ID String programming mode has completed. This returns the MCP2155 to normal operation.

**Note 1:** If a non-valid ID String (contains an ASCII character not in the valid range) is programmed, the MCP2155 will not create a link with a primary device.

**2:** The communication program that is supplied with Microsoft Windows operating system (called Hyper Terminal), may leave the DTR signal High and the RTS signals Low when the program disconnects, or is closed. Care should be taken to ensure that this could not accidentally cause the MCP2155 to enter Device ID String Programming.

[Example 2-1](#) show the firmware code for a PIC16Cxxx acting as the Host controller to modify the MCP2155 Device ID String.

**FIGURE 2-11: ID STRING FORMAT**



# MCP2155

## EXAMPLE 2-1: PIC16Fxx Code to Program the Device ID

```

;#define      dtr      PORTx, Pinx      ; Must specify which Port and Which Pin
;#define      cts      PORTx, Pinx      ; Must specify which Port and Which Pin
;#define      rts      PORTx, Pinx      ; Must specify which Port and Which Pin
;#define      clr      PORTx, Pinx      ; Must specify which Port and Which Pin
;
;*****
; String Table
; This table stores a string, breg is the offset.  The string
; is terminated by a null character.
;*****
string1  clrf    PCLATH          ; this routine is on page 0
         movf   breg, W          ; get the offset
         addwf  PCL, F          ; add the offset to PC
         DT     D'15'           ; the first byte is the byte count
         DT     "My IR ID String"
;
UpdateID
         call   deviceInit      ; Initialize the PIC16Fxxx
         bcf   clr              ; place the MCP2155 in reset
         bsf   dtr              ; Force the DTR pin High for program mode
         bcf   rts              ; Force the RTS pin Low for program mode
         call  delay1mS         ; delay for 1 ms.
         bsf   clr              ; allow the MCP2155 to come out of reset
;
         clrf  LoopCnt          ; LoopCnt = 0
ctsLp1   call  delay1mS         ; delay for 1 ms.
         btfs  cts              ; if cts=0 then we're ready to program
         goto  ctsLow           ; MCP2155 is ready to receive data
         decfsz LoopCnt, F      ;
         goto  ctsLp1           ; NO, wait for MCP2155 to be ready
         goto  StuckReset       ; The MCP2150 did not exit reset, do your recovery
                                   ; in this routine.

```

## EXAMPLE 2-1: PIC16Fxx Code to Program the Device ID (continued)

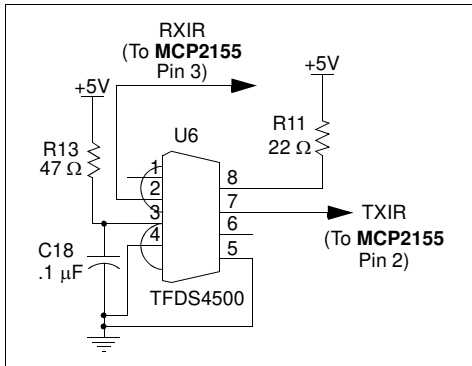
```
ctsLow  clrf    breg          ; clear the offset
        call   string1       ; get the byte count
        ;      (ID length byte + # bytes in string)
        movwf  creg          ; use creg as the loop counter
        incf   creg, f       ; add 1 to the loop count since
        ;      we're jumping into the middle
        movwf  areg          ; save the count in areg to send it
        goto   sndwt        ; start sending the count + ID string
;
sndlp   call   string1       ; get the byte
        movwf  areg          ; save the byte
sndwt   btfsc  cts           ; check the cts input
        goto   sndwt        ; wait if cts=1
        call   txser         ; send the byte using the Transmit Routine
        incf   breg, f       ; increment the table pointer
        decfsz creg, f       ; more bytes to send?
        goto   sndlp        ; YES, send more bytes
;
        bcf   clr           ; NO, place the MCP2155 in reset
        bcf   dtr          ; Force the DTR pin Low for normal mode
        bsf   rts          ; Force the RTS pin High for normal mode
        call  delay1mS     ; delay for 1 ms.
        bsf   clr          ; allow the MCP2155 to come out of reset
;
ctsLP2  btfss  cts          ; if cts=1 then MCP2155 is in Normal mode
        goto   ctsLP2       ; NO, wait for MCP2155 to be ready
        goto   NormalOperation ; The MCP2155 is now programmed with new ID,
        ; and is ready to establish an IR link
```

# MCP2155

## 2.14 Optical Transceiver

The MCP2155 requires an infrared transceiver. The transceiver can be a integrated solution. Table 2-5 shows a list of common manufacturers of integrated optical transceivers. A typical optical transceiver circuit using a Vishay/Temic TFDS4500 is shown in Figure 2-12.

**FIGURE 2-12: TYPICAL OPTICAL TRANSCEIVER CIRCUIT**



The optical transceiver logic can be implemented with discrete components, for component cost savings. Care must be taken in the design and layout of the photo detect circuit. This is due to the small signals that are being detected and their sensitivity to noise. A discrete implementation of the optical transceiver logic is implemented on the MCP2120 and MCP2150 Developer's Kit boards.

**Note:** The discrete optical transceiver implementation on the MCP2120 and MCP2150 Developer's Kit boards may not meet the IrDA specifications for the physical layer (IrPHY). Any discrete solution will require appropriate validation for the user's application.

## 2.15 References

The IrDA Standards download page can be found at:

<http://www.irda.org/standards/specifications>

Some common manufacturers of Optical Transceivers are shown in Table 2-5.

**TABLE 2-5: COMMON OPTICAL TRANSCEIVER MANUFACTURERS**

Company	Company Web Site Address
Infineon	<a href="http://www.infineon.com">www.infineon.com</a>
Agilent	<a href="http://www.agilent.com">www.agilent.com</a>
Vishay/Temic	<a href="http://www.vishay.com">www.vishay.com</a>
Rohm	<a href="http://www.rohm.com">www.rohm.com</a>

## 3.0 DEVELOPMENT TOOLS

The MCP2155 is supported by the MCP2120/MCP2150 Developer's Kit (order number DM163008). This kit allows the user to demonstrate the operation of the MCP2155 by doing some hardware "cuts" on the MCP2150 Developer's board.

Each kit (DM163008) comes with two MCP2120 Developer's boards and one MCP2150 Developer's Board to demonstrate transmission/reception of infrared data streams. Figure 3-1 shows a block diagram of the MCP2150 Developer's Board and the 4 lines that are to be "cut". The use of MCP2155 requires that the Header interface is used (SP3238E disconnected from system).

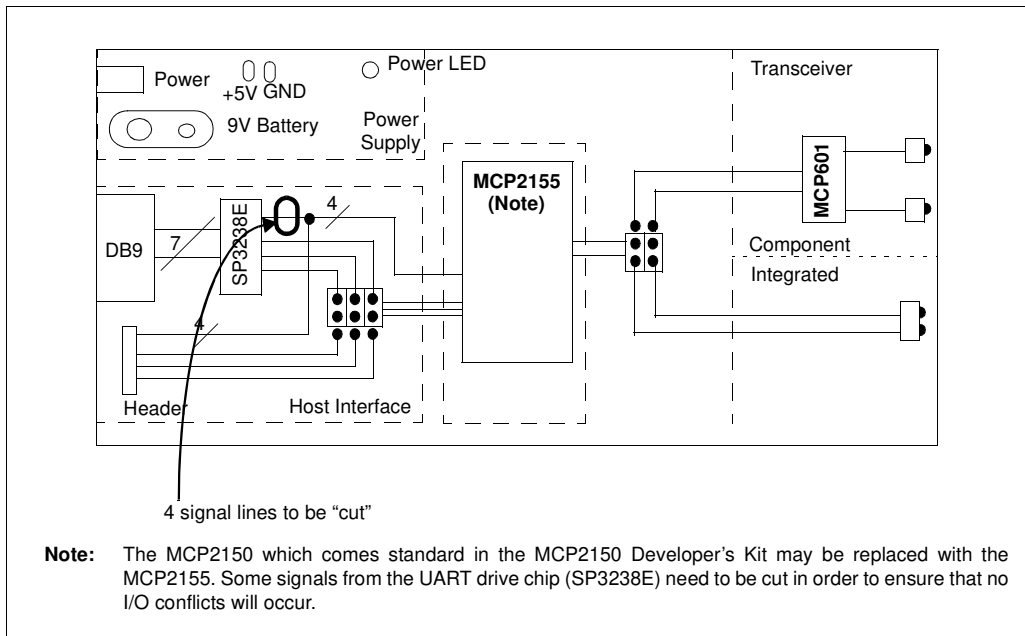
As can be seen, the user has jumper options for both the interface to the host controller (UART or Header) and the transceiver solution (Integrated or discrete component).

The UART interface allows a direct connection to a PC (use a terminal emulation program), or a header to allow easy connection to host prototypes (or one of the Microchip PICDEM™ boards).

The transceiver logic is jumpered to allow the selection of either a single chip transceiver solution, or a low cost discrete solution. This low cost discrete solution allows a lower system cost to be achieved. With the lower cost comes some trade-offs of the IrDA standard physical layer specifications. These trade-offs need to be evaluated to ensure the characteristics of the component solution meet the requirements of the system.

This kit comes with two identical MCP2120 Developer's Boards and a single MCP2150 Developer's board. This allows a complete system (Transmitter and Receiver) to be implemented with either system requirement (simple encoder/decoder or IrDA standard protocol stack plus encoder/decoder).

**FIGURE 3-1: MCP2150 DEVELOPER'S KIT BLOCK DIAGRAM**





# MCP2155

---

---

NOTES:

## 4.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings†

Ambient Temperature under bias .....	-40°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on V <sub>DD</sub> with respect to V <sub>SS</sub> .....	-0.3V to +6.5V
Voltage on $\overline{\text{RESET}}$ with respect to V <sub>SS</sub> .....	-0.3V to +14V
Voltage on all other pins with respect to V <sub>SS</sub> .....	-0.3V to (V <sub>DD</sub> + 0.3V)
Total Power Dissipation <sup>(1)</sup> .....	800 mW
Max. Current out of V <sub>SS</sub> pin .....	300 mA
Max. Current into V <sub>DD</sub> pin .....	250 mA
Input Clamp Current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>DD</sub> ) .....	±20 mA
Output Clamp Current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>DD</sub> ).....	±20 mA
Max. Output Current sunk by any Output pin.....	25 mA
Max. Output Current sourced by any Output pin.....	25 mA

**Note 1:** Power Dissipation is calculated as follows:

$$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

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