48511 Warm Springs Blvd., Suite 206, Fremont, CA 94539
Tel: (510) 490-8024 Fax: (510) 623-7268
Website: http://www.actisys.com E-mail: irda-info@actisys.com

ACT-IR8250P

IrDA Compliant Protocol Processor

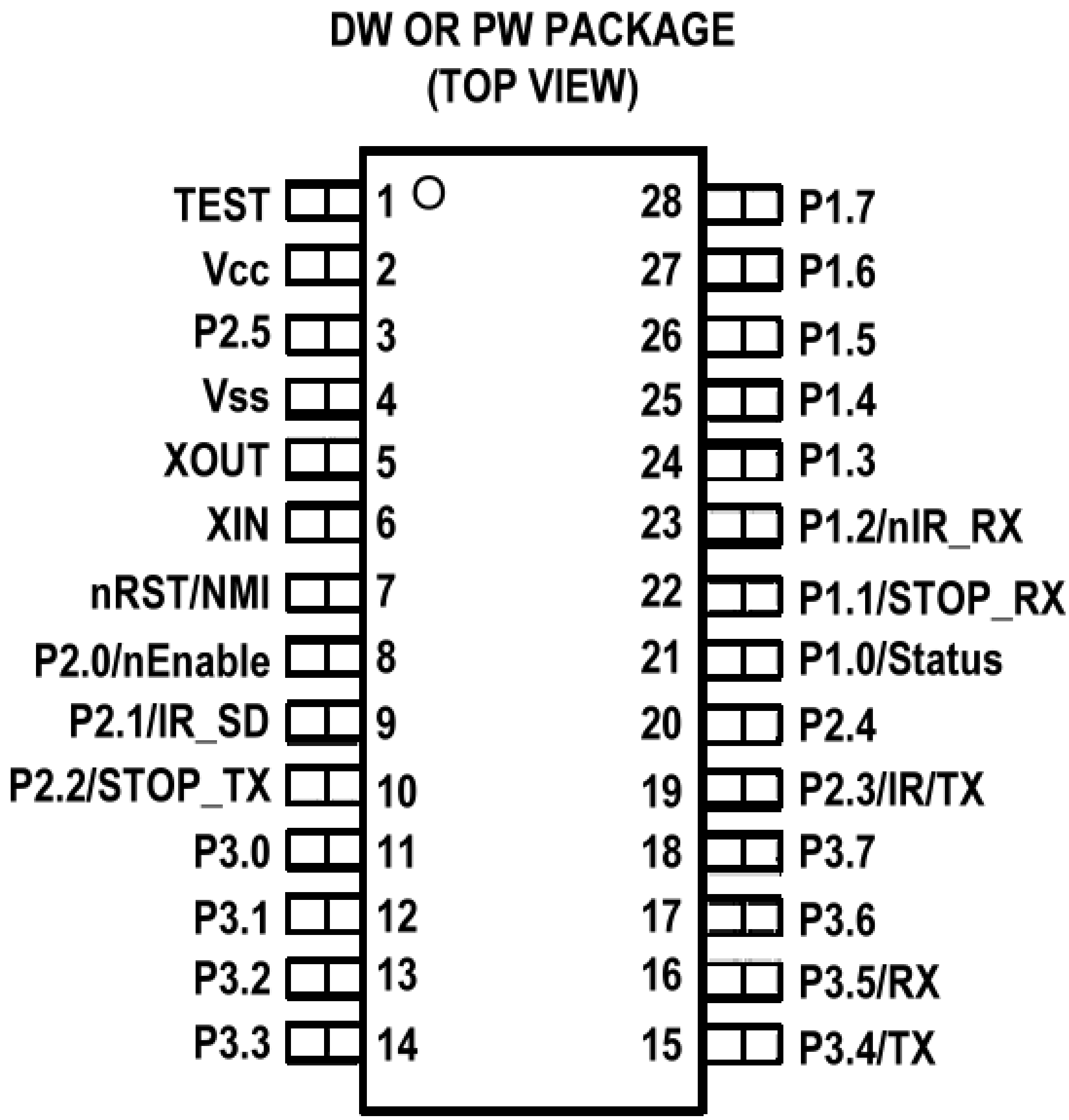
Specification

Revision History		
Revision	**Date**	**Comment**
Rev. 0.1	08/21/2003	Preliminary Design Specification
Rev. 0.2	08/26/2003	Removed Chapter 5 and added Appendix 1
Rev. 0.3	12/08/2003	Updated reference circuit & IrDA transceiver model #. Updated implementation examples. Added eval. kit PCB dimensions and circuit schematics.

Table of Contents

1. Features

- A complete IrDA Protocol stack in a single chip, includes mandatory protocols (IrPHY, IrLAP, IrLMP) and optional protocols (IrLPT, IrCOMM+TinyTP, OBEX Transport, or TinyTP with customized IAS class name for IrSocket).
- Also Includes IrPHY encoding/decoding, and interfaces directly to Infrared transceivers for data rate from 9.6kbps up to 115.2kbit/s. Only an external Infrared transceiver is needed to complete an IrDA compliant infrared communication subsystem.
- ACT-IR8250P supports IrDA Secondary mode only.
- Supports 64 bytes data packet for IrDA IrLAP frame.
- Interfaces to host device via a full function UART port.
- Supports host baud rate from 300kbit/s to 115.2kbit/s selecting by programmable settings.
- Programmable Device name, IAS class name and data format setting.
- Available in programmed and tested chip (IR8250P), assembled & tested board (IR8250PDB), or assembled & tested evaluation kit (IR8250PEK).
- IR8250P, housed in embedded RS232 adapter, already passed the tough IrReady certification tests, assuring broad compatibility with many IrDA-compliant devices, and error free, efficient IR data transfer. It is listed on IrDA website, IrReady Qualified Products.
- Low supply voltage, 3.0 V to 3.6 V.
- Low current consumption; 2uA standby, 3mA active.
- Small low profile plastic 28-pin SSOP/TSSOP package.
- In-system-programmable FLASH, facilitates firmware changes or updates.
- The evaluation kit, IR8250PEK includes: AC power supply, RS232 level translators, and PC software for in-system re-programming of firmware and future firmware options.
- ACT-IR8250PEK consists of: IR82x0PMB (motherboard) + IR8250PDB (daughterboard) + Self-downloadable SW to program IR8250P firmware.
  1) IR82x0PMB: RS232 level converter, DB9 connector, probe pins and IR8250PDB connector.
  2) IR8250PDB (daughter board for direct connection to your embedded PCB): Self-contained, full-function IrDA module which consists of: IR8250P protocol IC + IrDA transceiver + 3.3V interface connector.
- A very useful tool kit, Evaluation Kit Full Set is ACT-IR8250PEKK, which is: IR8250PEK + IR4000US (notebook/desktop USB-IrDA adapter). This is to test IR8250P (connected to your device), to exchange IrDA data with IR4000US (connected to PC USB port), running hyper-terminal on top of Windows IrDA driver. To avoid debugging multiple issues: e.g. PDA application IrDA SW issues (activated and behaves properly, the proper protocol layer)? IR8250P to host interface issues (UART data rates, flow control, data bit/parity/stop bit, UART signal pins, power levels)? Performance issues (throughput, distance, error rate/dropping bits)?


DW OR PW PACKAGE
(TOP VIEW)
TEST 1
Vcc 2
P2.5 3
Vss 4
XOUT 5
XIN 6
nRST/NMI 7
P2.0/nEnable 8
P2.1/IR_SD 9
P2.2/STOP_TX 10
P3.0 11
P3.1 12
P3.2 13
P3.3 14
28 P1.7
27 P1.6
26 P1.5
25 P1.4
24 P1.3
23 P1.2/nIR_RX
22 P1.1/STOP_RX
21 P1.0/Status
20 P2.4
19 P2.3/IR/TX
18 P3.7
17 P3.6
16 P3.5/RX
15 P3.4/TX

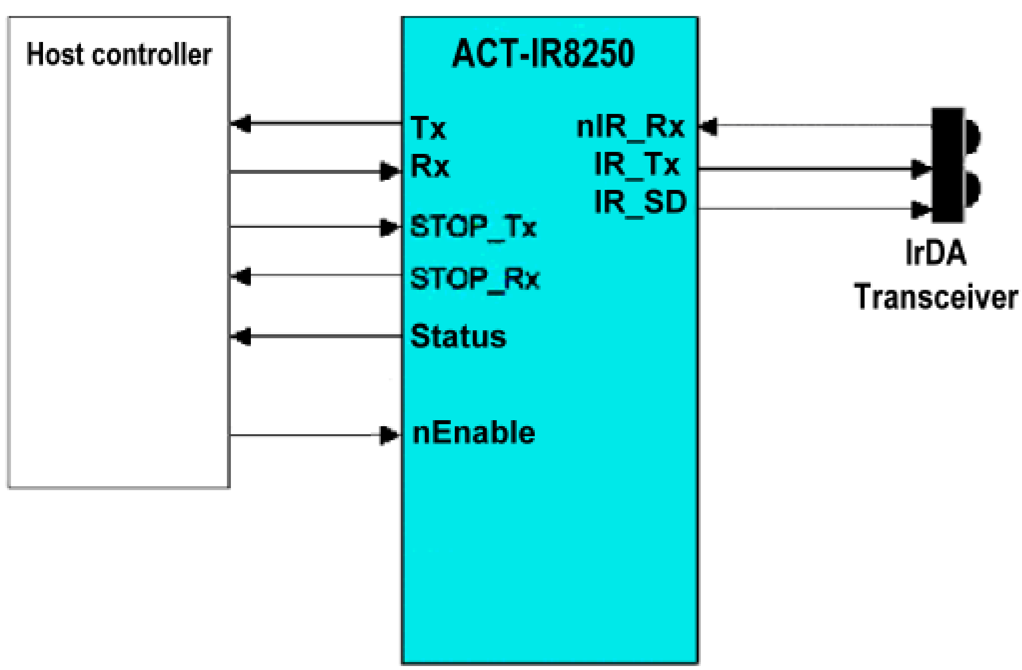
ACT-IR8250P

2. Overview

The ACT-IR8250P is a low cost, small pin-count, high-integration and in-system-programmable micro-controller, with on-chip IrDA protocol stack and on-chip Infrared physical encoder/decoder. It provides a serial interface to a host device that intends to have Infrared communication capability. The only external component needed to make a complete IrDA-compliant device is IrDA transceiver.

The host device can be any equipment or device that needs to communicate with IrDA enabled portable or tablet PC, PDA, cellular phone and hand held data terminal, via IrDA beam and protocol but has only a wired serial interface. The ACT-IR8250P will handle all the detail of IrDA protocol. It sends/receives only user data to/from the host device via the wired serial interface with hardware flow-control. IrDA has two modes; one is Primary, and the other is Secondary. The difference between these two modes is that a Primary mode device initiates the discovery, negotiation and connection sequence to Secondary mode device, and decides IrDA protocol parameters. Secondary mode device always waits for commands from Primary mode device. Both modes can run different protocols, and both may send or receive user data. ACT-IR8250P supports Secondary mode only. Fig.1 is system diagram.

Fig.1



3. PIN Description

Symbol	Pin No.	I/O Type	Descriptions
TEST	1	I	Reserved. Pull low with 30k ohm resistor.
VCC	2	Pwr	Digital supply voltage, positive terminal.
P2.5	3	I/O	Reserved. Pull high with 30k ohm resistor.
VSS	4		Digital supply voltage, negative terminal. Ground,
XOUT	5	I/O	Output terminal of crystal oscillator.
XIN	6	I	Input port for crystal oscillator. Standard crystals can be connected.
nRST/NMI	7	I	Reset input. (n: Active low)
P2.0/nEnable	8	I	Power control from Host. (n: Active low) High=Power down IR8250P, Low=Enable and power on.
P2.1/IR_SD	9	O	Shut down transceiver. High=Shut down, Low=Enable.
P2.2/STOP_TX	10	I	TX flow control from host. High=Host not ready to accept TX, Low=Host ready to accept TX.
P3.0	11	I/O	Reserved. Keep it open.
P3.1	12	I/O	Reserved. Keep it open.
P3.2	13	I/O	Reserved. Keep it open.
P3.3	14	I/O	Reserved. Keep it open.
P3.4/TX	15	O	Serial data to Host. High=Idle / Stop bit / 1-bit. Low=Start bit / 0-bit.
P3.5/RX	16	I	Serial data from Host. High=Idle / Stop bit / 1-bit. Low=Start bit / 0-bit.
P3.6	17	I/O	Reserved. Keep it open.
P3.7	18	I/O	Reserved. Keep it open.
P2.3/IR_TX	19	O	Transmit data to transceiver. High= IR on, Low=IR off.
P2.4	20	I/O	Reserved. Keep it open.
P1.0	21	O	IrDA status to Host. High=No IrDA connection. Low=IrDA connection active.
P1.1/STOP_RX	22	O	RX flow control to Host. High=IR8250P not ready to accept RX, Low=IR8250P ready to
P1.2/nIR_RX	23	I	Received data from transceiver. (n: Active low) High=No IR, Low=IR detected.

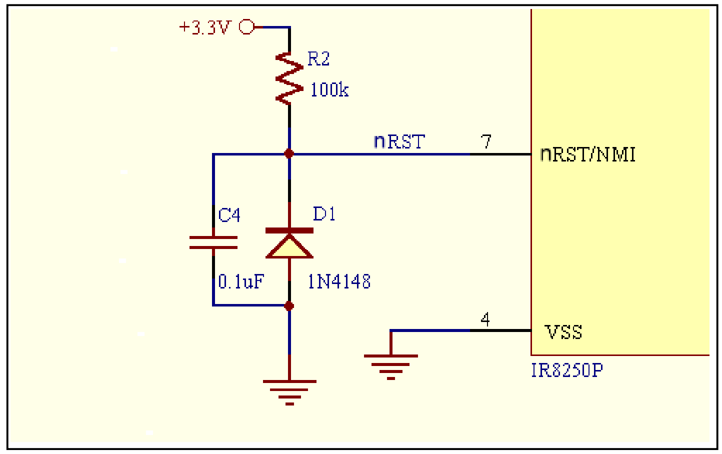
PIN Description (Continued)

Symbol	Pin No.	I/O Type	Descriptions
P1.3	24	I/O	Reserved. Keep it open.
P1.4	25	I/O	Reserved. Keep it open.
P1.5	26	I/O	Reserved. Keep it open.
P1.6	27	I/O	Reserved. Keep it open.
P1.7	28	I/O	Reserved. Keep it open.

4. Device Operation

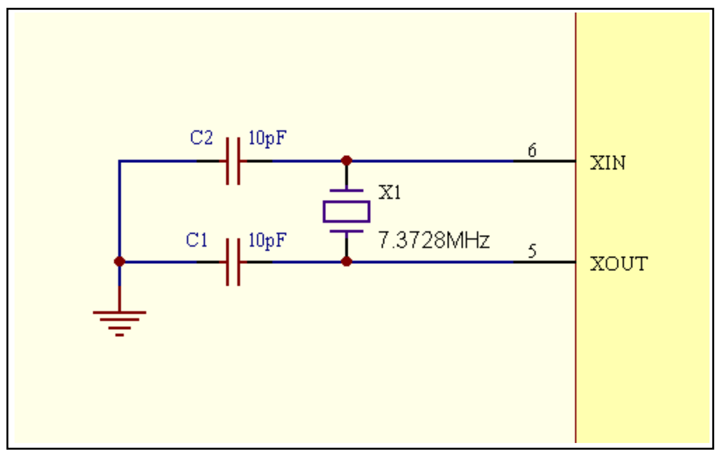
4.1 RESET circuit

IR8250P will be reset when nRST is pulled low. It needs a 100K ohm resistor, 0.1uF capacitor and a diode to implement the reset circuit. Please refer to right figure.



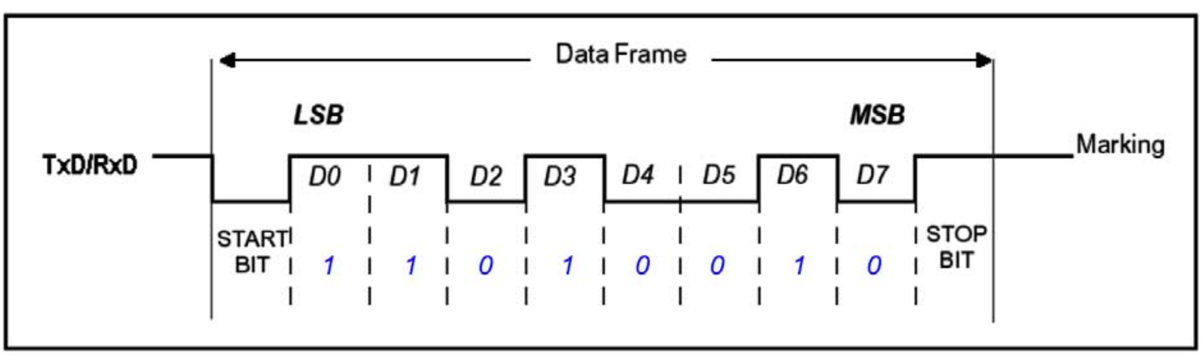
4.2 Crystal circuit

IR8250P needs a specified clock to operate, please refer to right figure.



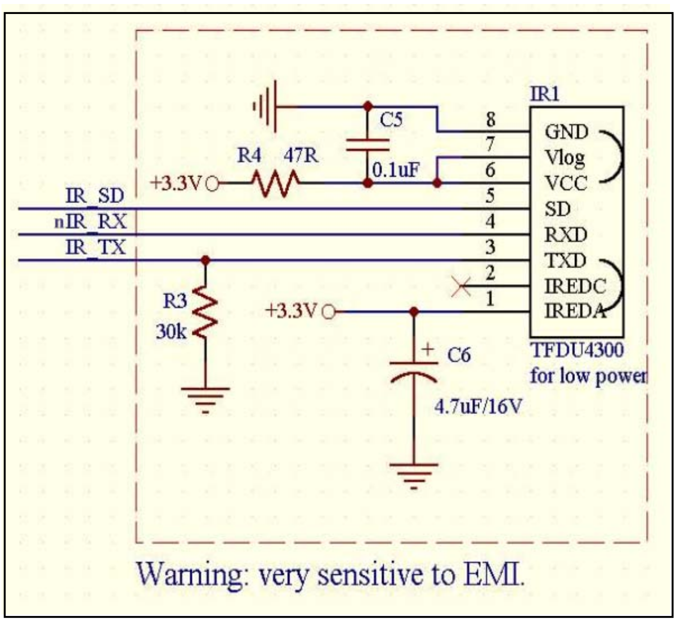
4.3 Host Interface

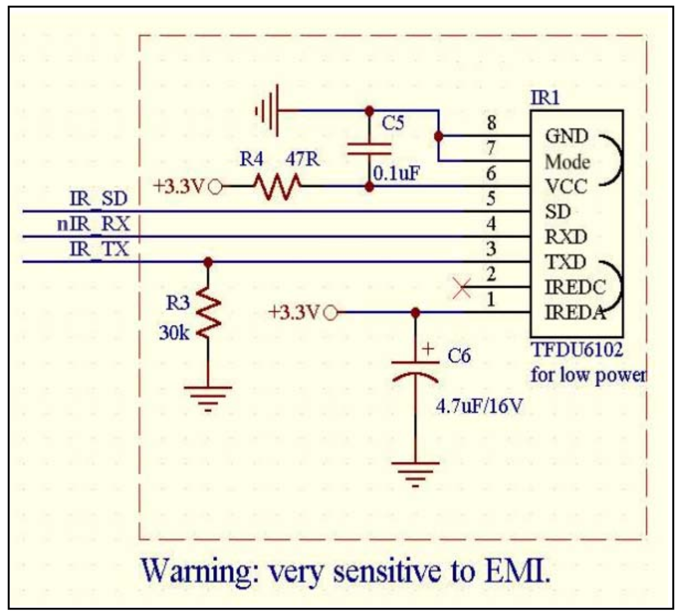
The host interface of IR8250P is a full-duplex asynchronous serial data interface. The data bytes are transmitted via TX and received via RX. Each data byte consists of one start bit (0), 8 data bits (LSB first, MSB last) and a stop bit (1).

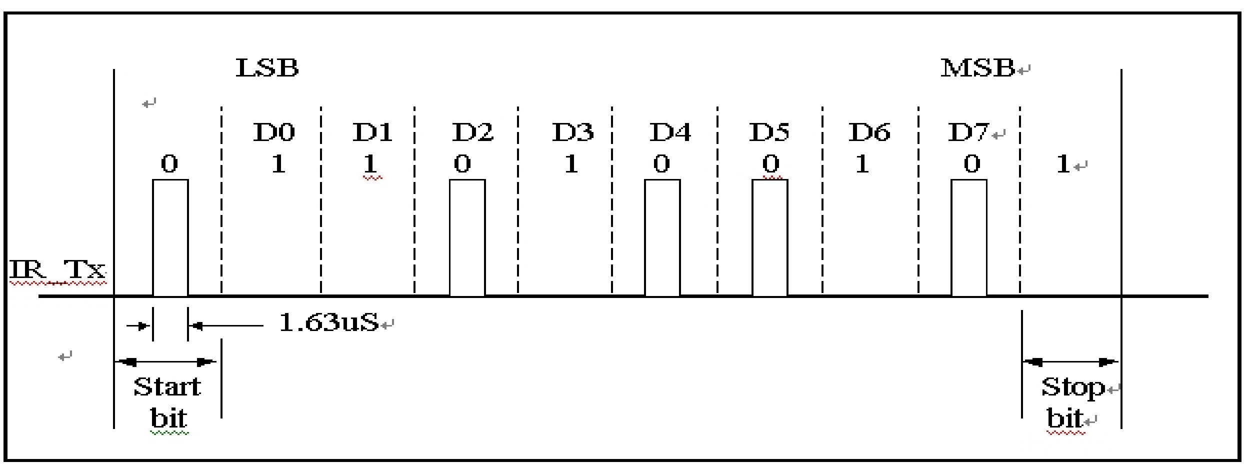


4.4 IR port interface

The IR port of IR8250P can be connected to most transceiver. The data is transmit by pin IR_TX and data received by nIR_RX. IR_SD can shut down transceiver when IR8250P is disabled. This pin is optional. The below figure is the signal specification of IR_TX, it sends 1.63uS pulse infrared signal out.







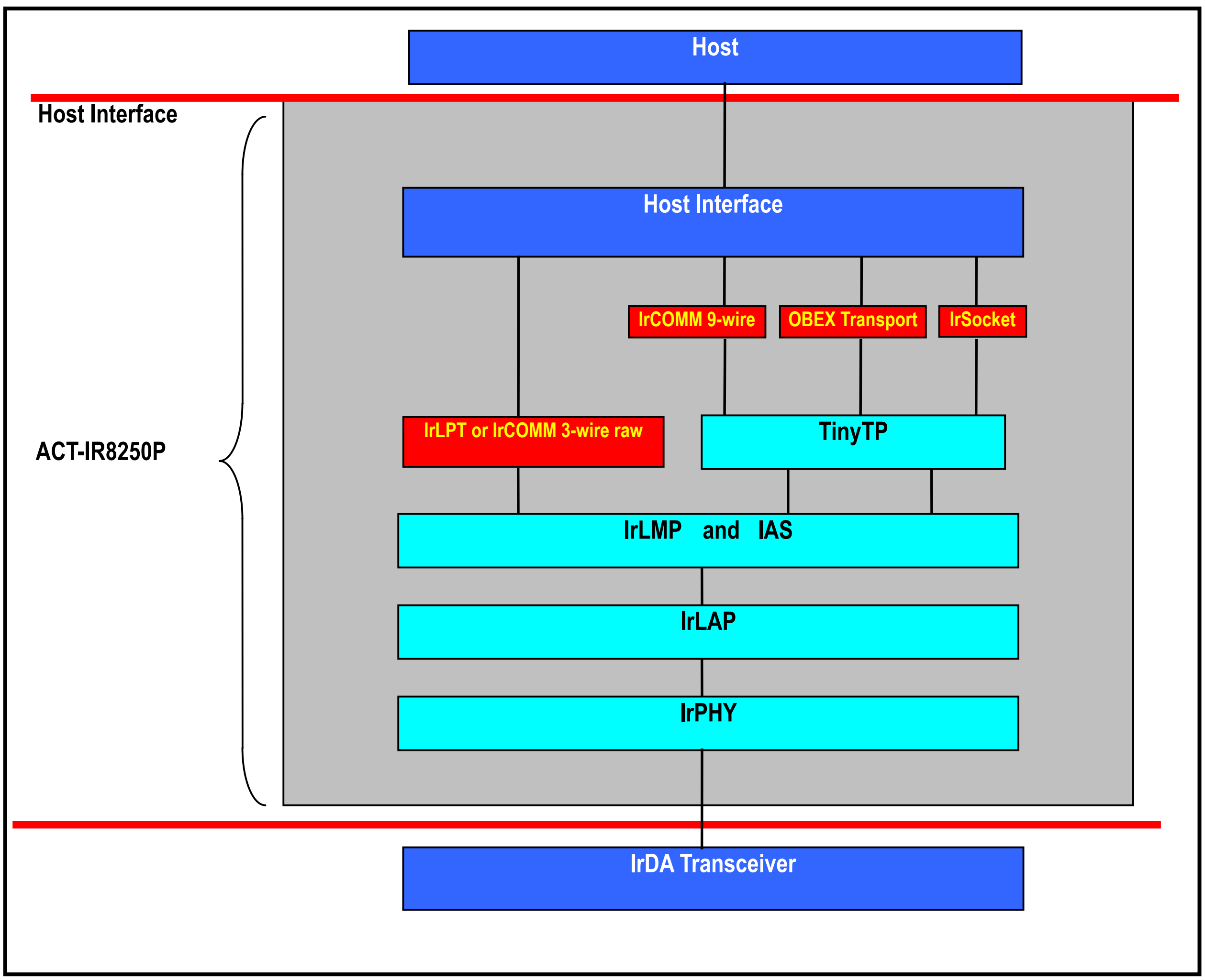
4.5 Operating modes

When nEnable is high, ACT IR8250P drives IR_SD high (to shutdown the IR transceiver) and puts itself to sleep. In this sleep state, it can be awaken only by nEnable going low.

When nEnable is low, ACT IR8250P drives IR_SD low (to power up the IR transceiver) and puts itself to sleep. In this sleep state, it can be awaken by any incoming IR signal. Once awaken, it will either actively communicate with an IrDA Primary device (PC, PDA, Cell phone, etc.), or go back to sleep if no IrDA Primary is there.

5. IrDA Protocols Supported

The following figure is the block diagram of ACT-IR8250P protocols architecture. The red colored blocks represent different optional upper layers. Only one of the red colored blocks can be included at a time.



6. IR8250P Parts ID and Firmware Number System

ACT-IR8250P-####

(Please contact irda-info@actisys.com or info@actisys.com.tw for updated firmware and parts #)

7. Connect ACT-IR8250P to a Host Device with RS232 Port

Devices that use serial cables for their communication are split into two categories. These are DCE (Data Communications Equipment) and DTE (Data Terminal Equipment.) Data Communications Equipment are devices such as your modem, TA adapter, plotter etc. while Data Terminal Equipment is your Computer or Terminal. The following shows how to connect IR8250P to a DTE or a DCE. Note: A RS232 level converter chip is needed when IR8250P is connected to a RS232 interface.

There are a total of 7 signals between the host and the IR8250P. The pin assignment and name of each signal at the chip levels are summarized in Table 1 below. Please note that DTE denotes signals on the RS232 board or device. In order to connect them to another DTE, a null-modem converter is used in-between. The pin assignment and name are listed in this table.

Table : Host interface signals

ACT-IR8250P			RS232 DTE device			RS232 DCE device		
Pin No.	Type	Name	Pin No.	Type	Name	Pin No.	Type	Name
15	O	TX	2	I	RX	3	I	RX
16	I	RX	3	O	TX	2	O	TX
8	I	nEnable	4	O	DTR	6	O	DSR
21	O	Status	6	I	DSR	4	I	DTR
10	I	STOP_TX	7	O	RTS	8	O	CTS
22	O	STOP_RX	8	I	CTS	7	I	RTS
4		GND	5		GND	5		GND

8. Implementation Notice

Following signals lists the description of implementation notes when designed into board.

(1) Power supply

Pin	Name	Comments
2	VCC	Digital power. Connect a 0.1uF bypass capacitor at this pin.
4	VSS	Digital ground

(2) Crystal Oscillator

Pin	Name	Comments
6	XTIN	Input port for crystal oscillator. Only standard crystals can be connected.
5	XTOUT	Output terminal of crystal oscillator

Both of these two pins connect a 20pF capacitor.

(3) Reset

Pin	I/O type	Name	Comments
7	I	nRST	Active low reset. This signal can be generated by the host, a RC circuit or a RESET chip. It supports Brown-out feature.

(4) IrDA Transceiver

Pin	I/O type	Name	Comments
19	O	Ir_TX	IrDA Transmitter, high active. Connect to TXD pin of IrDA transceiver.
23	I	nIr_RX	IrDA Receiver, active low. Connect to RXD pin of IrDA transceiver.
9	O	IR_SD	Transceiver power control.

Please see the transceiver's specification to get more detail information about transceiver.
Place IR8250P chip next to transceiver module as close as possible. And keep chip away from high power consumption components.

9. ACT-IR8250P Evaluation Kit (IR8250PEK and IR8250PEKK)

It is recommended that customer verify the compatibility with your host system by using ACT-IR8250P evaluation kit, IR8250PEK. It is a self-contained unit, with ACT-IR8250P, IrDA transceiver, RS232-level converter and external AC power connector, all built into a compact package. ACT-IR8250 evaluation kit enables your host system to be IrDA (IrReady) certifiable, immediately. Moreover, this kit can be a full PCB with RS232 interface, or as half-PCB with UART interface, without the burden of RS232 interface circuitry.

ACT-IR8250PEK package consists of: IR82x0PMB (motherboard) + IR8250PDB (daughterboard) + Self-downloadable SW to program IR8250P firmware.

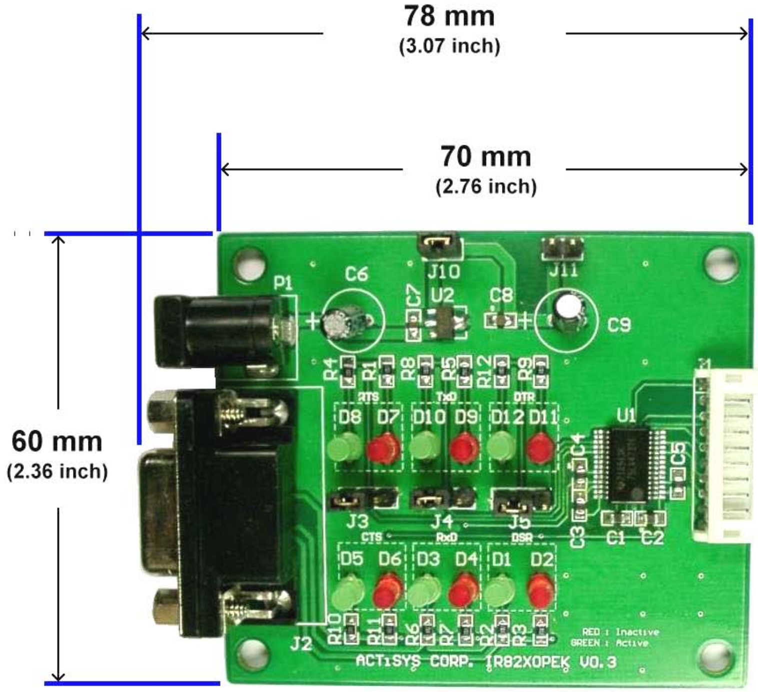
1) IR82x0PMB: RS232 level converter, DB9 connector, probe pins and IR8250PDB connector.
2) IR8250PDB (daughter board for direct connection to your embedded PCB):
   Self-contained, full-function IrDA module which consists of: IR8250P protocol IC + IrDA transceiver + 3.3V interface connector.

A more useful tool kit, Evaluation Kit Full Set is ACT-IR8250PEKK, which is: IR8250PEK + IR4000US (notebook/desktop USB-IrDA adapter). This is to test IR8250P (connected to your device), to exchange IrDA data with IR4000US (connected to PC USB port), running hyper-terminal (assigned to the same virtual comm port as the one used by Windows IrDA driver) on top of Windows IrDA driver. This avoids debugging multiple issues: PDA application IrDA SW issues (IrDA driver activated and behaves properly, the proper IrDA protocol layer)? IR8250P to host interface issues (UART data rates, flow control, data bit/parity/stop bit, signal pins, power levels)? Performance issues (throughput, distance, error rate/dropping bits)?

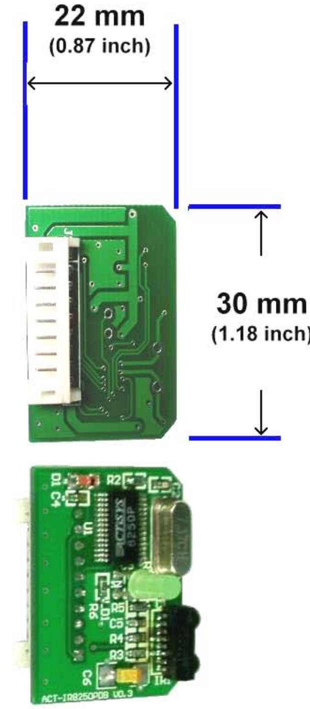
The following figure is IR8250P Evaluation Kit, IR8250PEK. It consists of two boards, IR82x0PMB on the left, and IR8250PDB on the right, connected to each other by 10pin connector/cable. You can connect this evaluation kit to your host RS232 port, or use IR8250PDB board to connect to your host CMOS level UART signals.

The corresponding circuit schematics for IR82x0PMB and IR8250PDB are included below.

ACT-IR8250PEK PCB Photos and Dimensions:



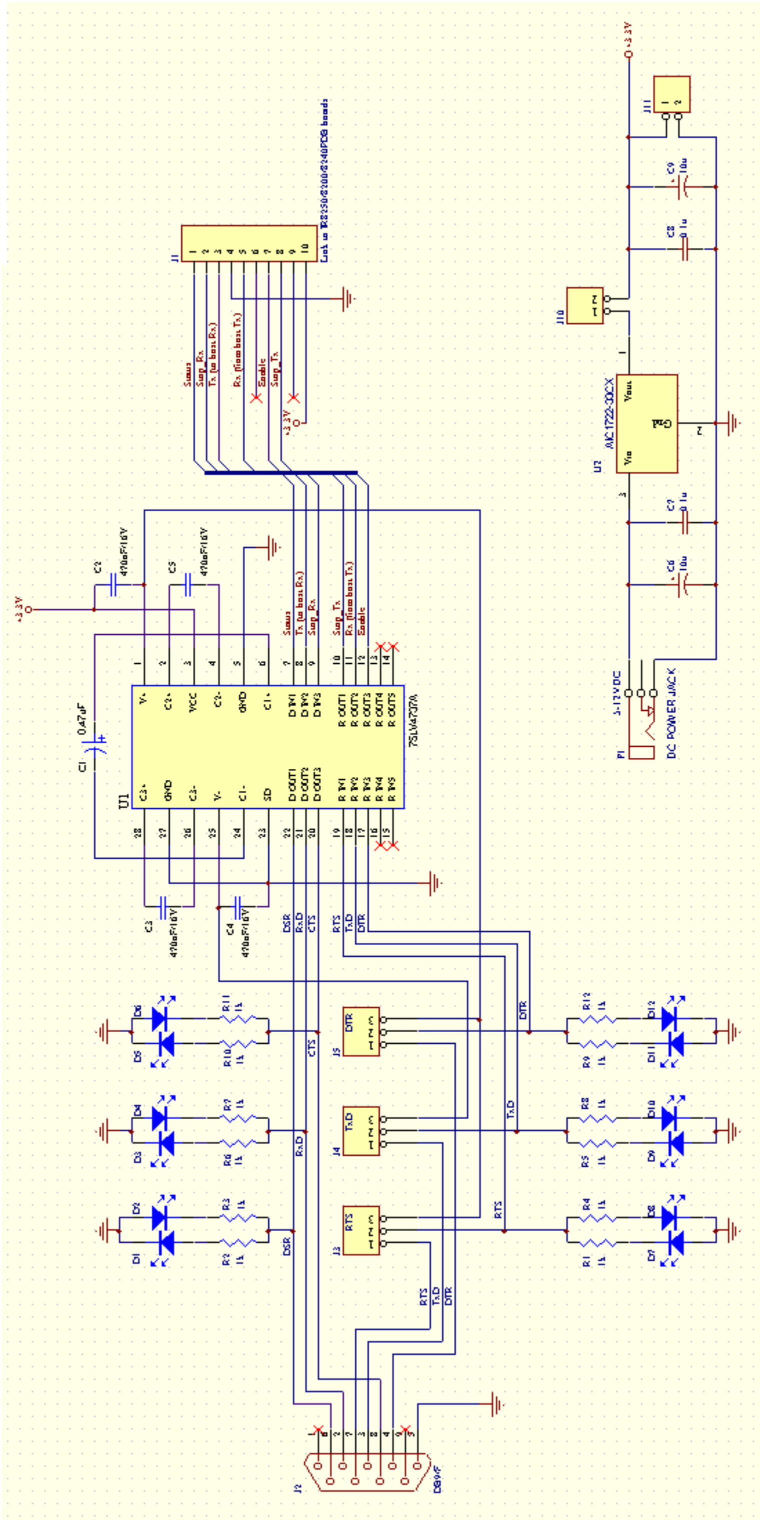
IR82x0PMB



IR8250PDB

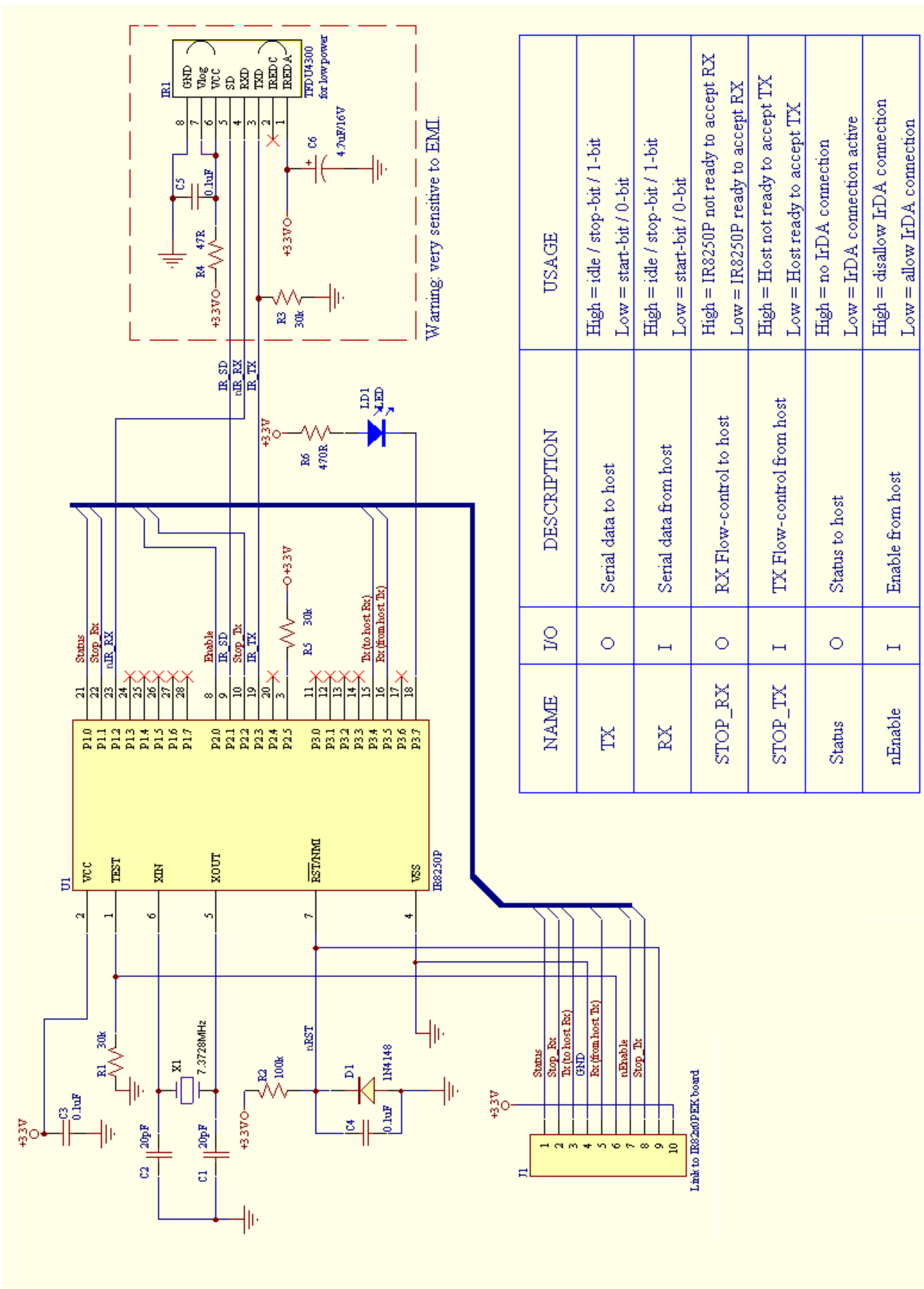
ACT-IR8250PEK Circuit Schematics:

1) ACT-IR82x0PMB Motherboard Circuit Schematic:



2) ACT-IR8250PDB Daughter Board Circuit Schematic:
(Same as Application Reference Circuit in Section 11)

If your design uses daughter board configuration for convenient placement against front panel, with cable connection to your system board, it is advisable for you to maintain the 10 pin connector as in this ACT-IR8250PDB reference circuit here. This enables in-system IR8250P firmware re-programming with new firmware if such need arises. This flexibility can extend your product lifetime, improve compatibility with new IrDA devices, add or change IrDA functionality, indeed a valuable and useful feature.



10. Characteristics and Specification

Absolute maximum ratings†

Voltage applied at VCC to VSS.. –0.3 V to 4.1 V
Voltage applied to any pin (referenced to VSS) ……………………………………………–0.3 V to VCC+0.3 V
Diode current at any device terminal ……………………………………………………………………………±2 mA
Storage temperature, Tstg (unprogrammed device) ………………………………………………–55°C to 150°C
Storage temperature, Tstg (programmed device) ……………………………………………….. –40°C to 85°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
NOTE: All voltages referenced to VSS.

Recommended operating conditions

Parameter	MIN.	TYPICAL	MAX.	Units
Supply voltage during program execution, VCC	**1.8**	**3.3**	**3.6**	**V**
Supply voltage, VSS		**0**		**V**
Operating free-air temperature range, TA	**-40**		**85**	℃
DC current (Shut down mode)		**2**		**UA**
DC current (Active mode)		**2**		**MA**
Crystal frequency		**7.3728**		**MHz**

The DC current in active mode is not included transceiver operating current.

Electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

Schmitt-trigger inputs Port P1 to Port P3; P1.0 to P1.7, P2.0 to P2.5, P3.0 to P3.7

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
VIT+ Positive going input threshold voltage	**VCC = 2.2V**	**1.1**		**1.5**	**V**
	VCC = 3.0V	**1.5**		**1.9**	
VIT- Negative going input threshold voltage	**VCC = 2.2V**	**0.4**		**0.9**	**V**
	VCC = 3.0V	**0.9**		**1.3**	
Vhys Input voltage hysteresis (VIT+ VIT-)	**VCC = 2.2V**	**0.3**		**1.1**	**V**
	VCC = 3.0V	**0.5**		**1.0**	

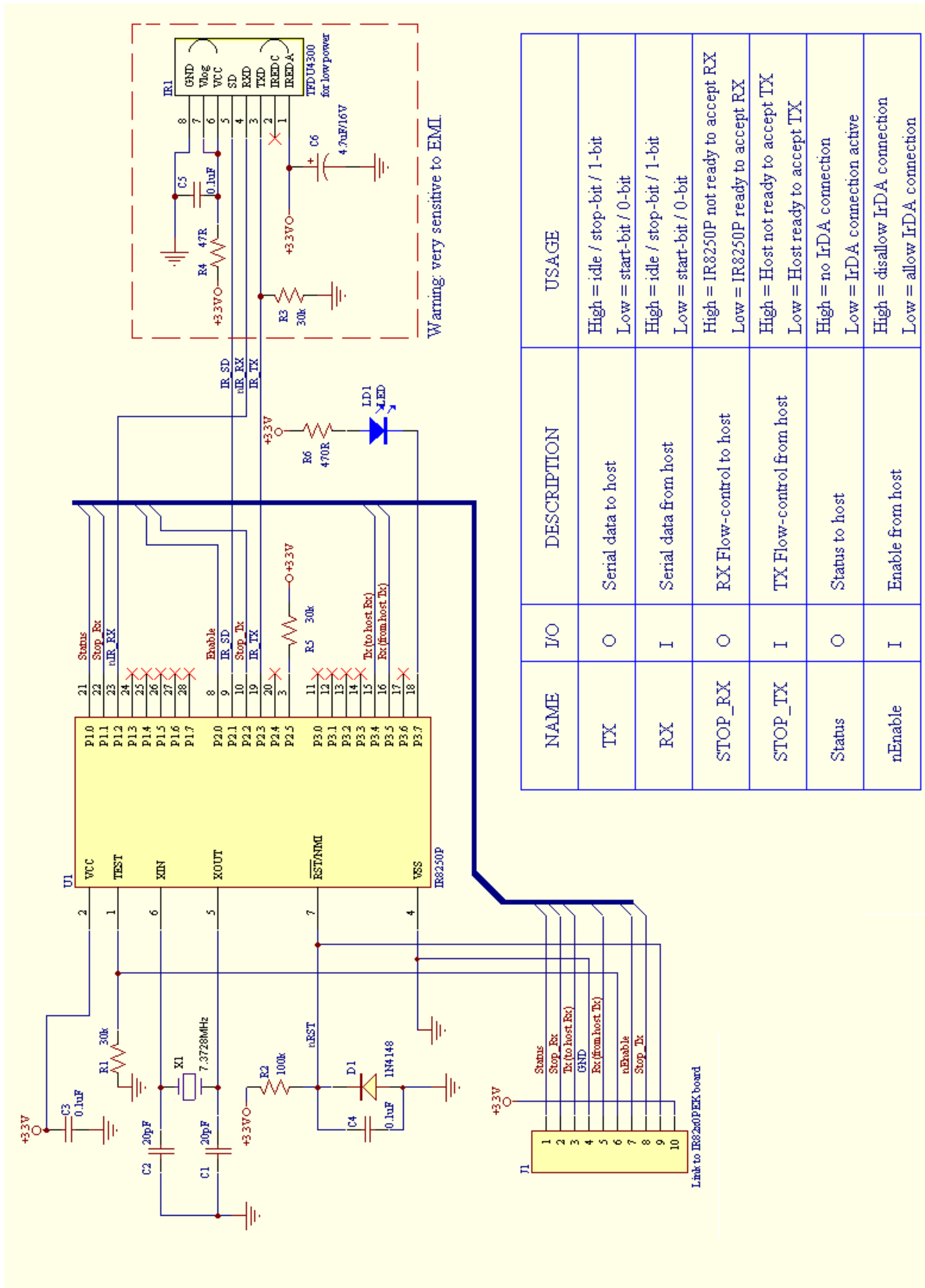
outputs Port 1 to P3; P1.0 to P1.7, P2.0 to P2.5, P3.0 to P3.7

PARAMETER		TEST CONDITIONS			MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	$I_{(OHmax)}$ = –1.5 mA	V_{CC} = 2.2 V	See Note 8	V_{CC}–0.25		V_{CC}	V
		$I_{(OHmax)}$ = –6 mA		See Note 9	V_{CC}–0.6		V_{CC}	
		$I_{(OHmax)}$ = –1.5 mA	V_{CC} = 3 V	See Note 8	V_{CC}–0.25		V_{CC}	
		$I_{(OHmax)}$ = –6 mA		See Note 9	V_{CC}–0.6		V_{CC}	
V_{OL}	Low-level output voltage	$I_{(OLmax)}$ = 1.5 mA	V_{CC} = 2.2 V	See Note 8	V_{SS}		V_{SS}+0.25	V
		$I_{(OLmax)}$ = 6 mA		See Note 9	V_{SS}		V_{SS}+0.6	
		$I_{(OLmax)}$ = 1.5 mA	V_{CC} = 3 V	See Note 8	V_{SS}		V_{SS}+0.25	
		$I_{(OLmax)}$ = 6 mA		See Note 9	V_{SS}		V_{SS}+0.6	

NOTES: 8. The maximum total current, I_{OHmax} and I_{OLmax}, for all outputs combined, should not exceed ±12 mA to hold the maximum voltage drop specified.
9. The maximum total current, I_{OHmax} and I_{OLmax}, for all outputs combined, should not exceed ±48 mA to hold the maximum voltage drop specified.
10. One output loaded at a time.

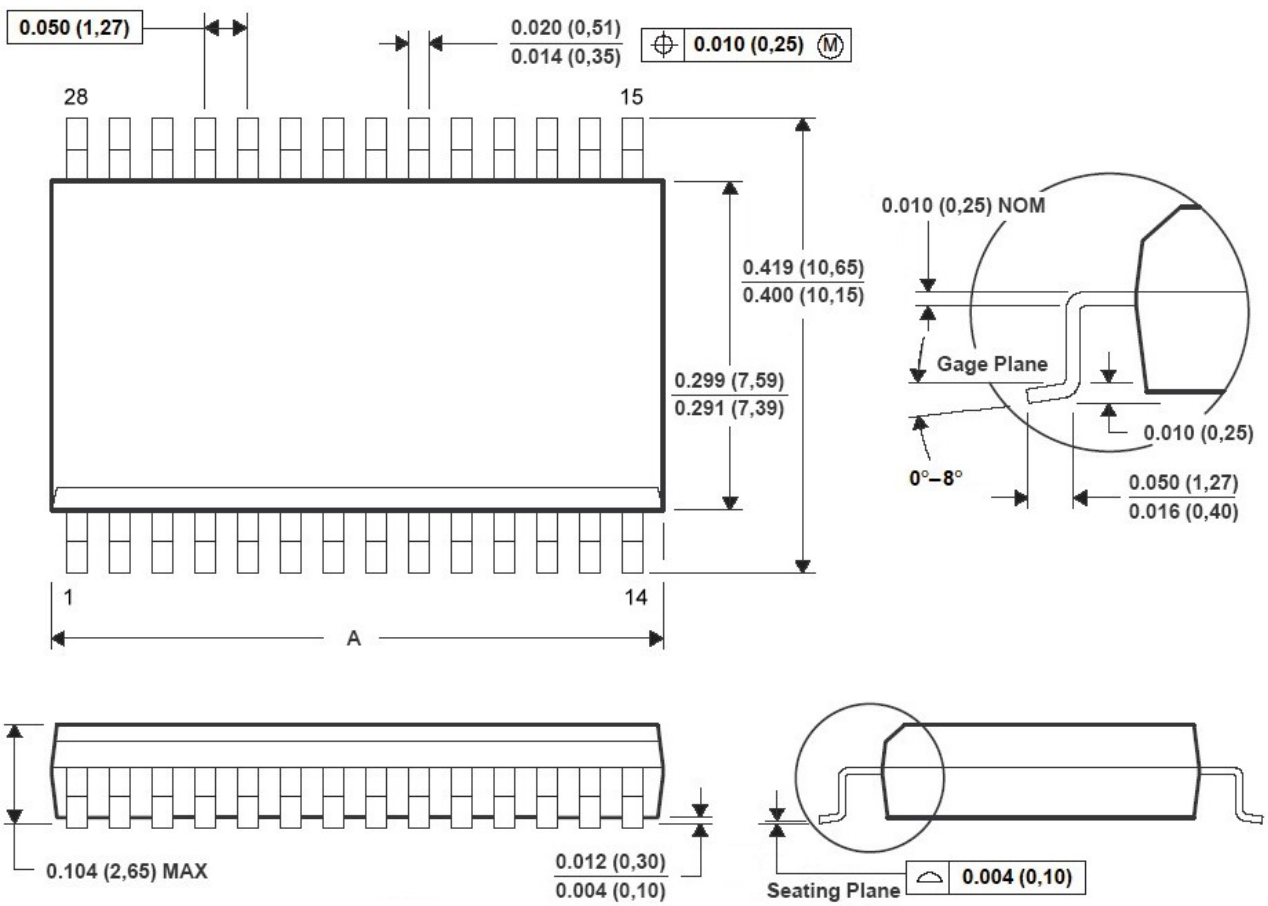
11. Application Circuit

(Same as Circuit Schematic of IR8250PDB in Section 9)



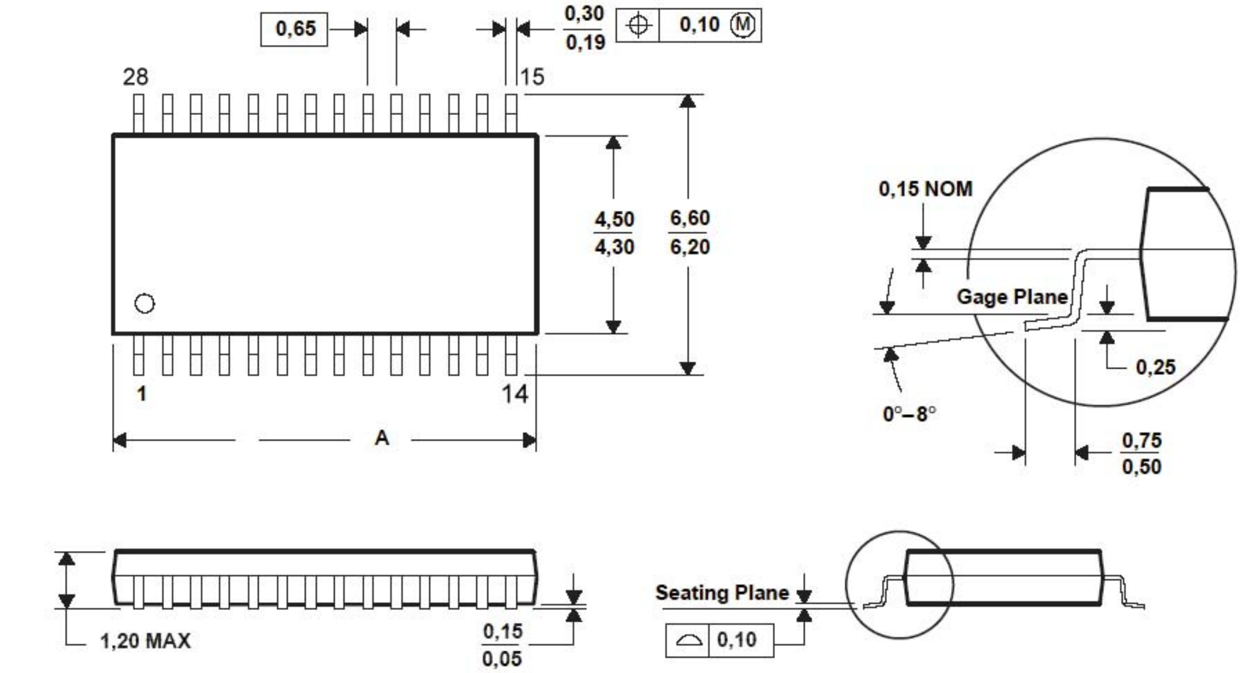
12. Package Dimensions

SOP28



PINS ** / DIM	16	20	24	28
A MAX	0.410 (10,41)	0.510 (12,95)	0.610 (15,49)	0.710 (18,03)
A MIN	0.400 (10,16)	0.500 (12,70)	0.600 (15,24)	0.700 (17,78)

TSSOP28

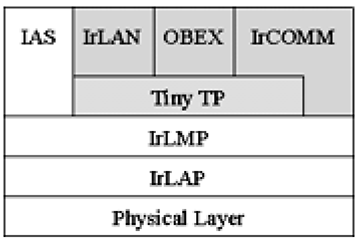


PINS ** / DIM	8	14	16	20	24	28
A MAX	3,10	5,10	5,10	6,60	7,90	9,80
A MIN	2,90	4,90	4,90	6,40	7,70	9,60

Appendix 1. What is an IrDA Protocol Stack?

Communications protocols deal with many issues, and so are generally broken into layers, each of which deals with a manageable set of responsibilities and supplies needed capabilities to the layers above and below. When you place the layers on top of each other, you get what is called a protocol stack, rather like a stack of pancakes or a stack of plates. An IrDA protocol stack is the layered set of protocols particularly aimed at point-to-point infrared communications and the applications needed in that environment.

Below is a picture of the IrDA protocol layers. This layering will serve as the overall structure for much of the remaining discussion.



The layers within this stack can be divided into two groups -- required and optional protocols.

A.1 Required IrDA Protocols

The required layers of an IrDA protocol stack are in unshaded boxes in the diagram above and include the following:

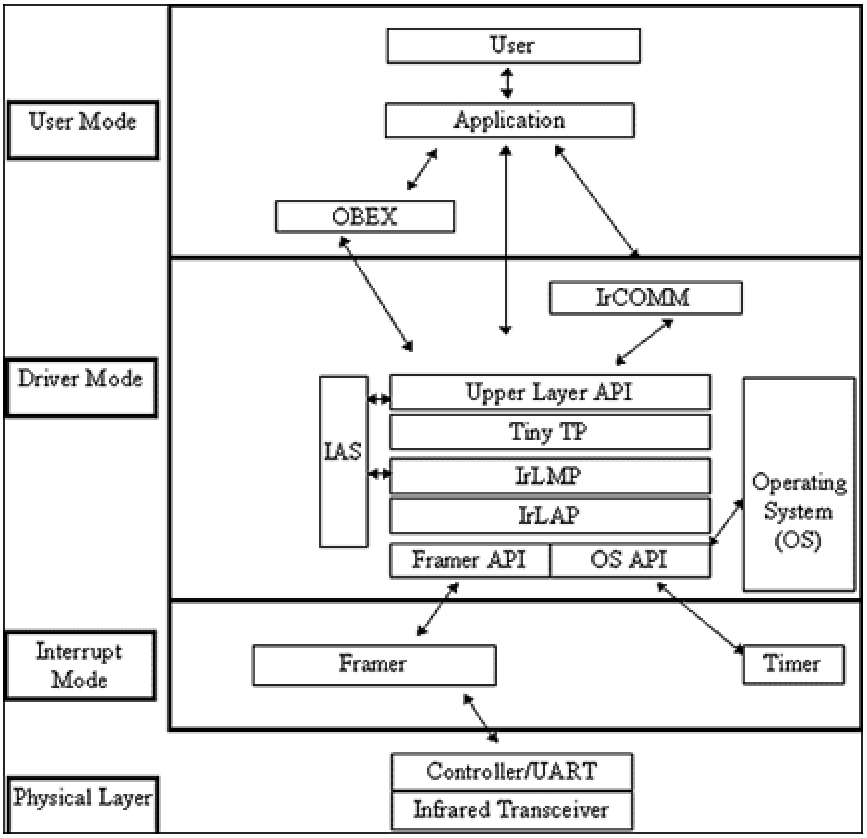
- Physical Layer: Specifies optical characteristics, encoding of data, and framing for various speeds.
- IrLAP: Link Access Protocol. Establishes the basic reliable connection.
- IrLMP: Link Management Protocol. Multiplexes services and applications on the LAP connection.
- IAS: Information Access Service. Provides a "yellow pages" of services on a device.

A.2 Optional Protocols

The optional protocols are shown above in shaded boxes. The use of the optional layers depends upon the particular application. The optional protocols are:

- TinyTP: Tiny Transport Protocol. Adds per-channel flow control to keep things moving smoothly. This is a very important function and is required in many cases.
- IrOBEX: The Object Exchange protocol. Easy transfer of files and other data objects.
- IrCOMM: Serial and Parallel Port emulation, enabling existing apps that use serial and parallel communications to use IR without change.
- IrLAN: Local Area Network access, enabling walk-up IR LAN access for laptops and other devices.

When the stack layers shown above are integrated into an embedded system, the picture may look more like the following:



A.3 Physical Layer and the Framer

5.3.1 Framer Overview

The Physical layer includes the optical transceiver, and deals with shaping and other characteristics of infrared signals including the encoding of data bits, and some framing data such as begin and end of frame flags (BOFs and EOFs) and cyclic redundancy checks (CRCs). This layer must be at least partially implemented in hardware, but in some cases is handled entirely by hardware.

In order to isolate the remainder of the stack from the ever-changing hardware layer, a software layer called the framer is created. Its primary responsibility is to accept incoming frames from the hardware and present them to the Link Access Protocol layer (IrLAP). This includes accepting outgoing frames and doing whatever is necessary to send them. In addition, the framer is responsible for changing hardware speeds at the bidding of the IrLAP layer, using whatever magic incantations the hardware designer invented for that purpose (these signals have not yet been standardized).

A.4 IrLAP - Link Access Protocol

A.4.1 IrLAP Overview

Immediately above the framer we encounter the IrLAP layer, also known as the Link Access Protocol, or LAP for short. IrLAP is a required IrDA protocol corresponding to OSI layer 2 (data link protocol). It is based on High-Level Data Link Control (HDLC) and Synchronous Data Link Control (SDLC) with extensions for some unique characteristics of infrared communications.

IrLAP provides reliable data transfer using the following mechanisms:

- Retransmission.
- Low-level flow control. (TinyTP provides high-level flow control and should almost always be used in place of IrLAP flow control.)
- Error detection.

By dealing with reliable data transfer at a low level, upper layers are free from this concern and can be assured that their data will be delivered (or at least that they will be informed if it was not). Data delivery might fail if the beam path were blocked. For instance, someone could put a coffee cup in the path of the infrared beam. IrLAP alerts the upper layer so that higher-level layers can deal with the problem appropriately. As an example, an application sitting on the stack could be alerted of an interruption in data flow, allowing it to alert the user through some interface. The user could then potentially remedy the problem (by moving the coffee cup) without dropping the connection or losing the data transferred to that point.

A.4.2 Environmental Characteristics

Several environmental factors influenced the development of the IrLAP layer. These include the following:

- *Point-to-point*. Connections are one-to-one, such as camera to PC or data collector to printer. The range is typically zero to one meter, although extended range up to 10 meters or more is under development. This is not like a local area network (many-to-many) protocol.
- *Half-duplex*. Infrared light, and therefore data, is sent in one direction at a time. However, the link changes directions frequently and can simulate full duplex in cases where timing is not extremely sensitive.
- *Narrow infrared cone*. The infrared transmission is directional within a 15 degree half angle in order to minimize interference with surrounding devices.
- *Hidden Nodes*. Other IR devices approaching an existing connection may not be immediately aware of the connection if they approach from behind the current transmitter. They must wait and see if the link turns around before stepping in. Interference. IrLAP must overcome interference from fluorescent lights, other IR devices, sunlight, moonbeams, and so forth.

- *No collision detection*. The design of the hardware is such that collisions are not detected, so the software must handle cases where collisions cause lost data with methods such as random back off.

A.4.3 Roles within a LAP connection

The two parties to a LAP connection have a master-slave relationship with differing responsibilities (and resulting code complexity). The IrDA terms for this are Primary (master) and Secondary (slave).

Primary Station:

- Sends Command frames - initiates connections and transfers.
- Responsible for organization and control of data flow.
- Deals with unrecoverable data link errors.
- Typical primary devices include PCs, PDAs, cameras, and anything that needs to print (printers are currently all secondaries).

Secondary Station:

- Sends Response frames - only speaks when spoken to.
- Typical secondary devices are printers and other peripherals, and resource-constrained devices (secondaries are smaller and less complex).

In any connection one device must play the primary role. The other device must play the secondary role, but its protocol stack may be either a secondary or another primary—most primaries can play a secondary role. Once started, the two sides take turns talking with the primary leading off. No side can talk for more than 500 milliseconds at a time before allowing the other side a chance to talk (even if just to say it has nothing to send for the moment). Note that the issue of master versus slave becomes much less obvious at the higher protocol layers—once two devices are connected, an application on a secondary (slave) can initiate an operation just as easily as an application on the primary side.

A.4.4 Modes

IrLAP is built around two modes of operation, corresponding to whether or not a connection exists.

A.4.4.1 Normal Disconnect Mode (NDM)

NDM is also known as contention state, and is the default state of disconnected devices. In this mode a device must observe a set of media access rules. Of utmost importance, a device in NDM must check whether other transmissions are occurring (a condition known as media busy) before transmitting. This is accomplished by listening for activity. If no activity is detected for greater than 500 milliseconds (the maximum time for the link to turn around), the media is considered to be available for establishment of a connection.

A great ease-of-use feature is provided by the NDM communications rules. A classic problem is getting both sides of the link configured with the same communications parameters—frequently users get completely stuck. This can be particularly difficult on embedded devices that don't have a user interface for setting or reviewing communications parameters. This problem is completely absent with IrDA solutions—all NDM communications use the following link parameters: ASYNC, 9600 bps, 8 bits, no parity. During the connection process, the two sides exchange capability information, and subsequently shift to the best parameters supportable by both sides.

A.4.4.2 Normal Response Mode (NRM)

NRM is the mode of operation for connected devices. Once both sides are talking using the best possible communication parameters (established during NDM), higher stack layers use normal command and response frames to exchange information.

A.4.5 IrLAP Frame Format

The basic IrLAP frame format is as follows:

Address	Control	Information

While there are too many details about frame formats to discuss here, it is worth noting that the Address and Control fields require only two bytes total—the IrDA protocols add very little overhead to the user data.

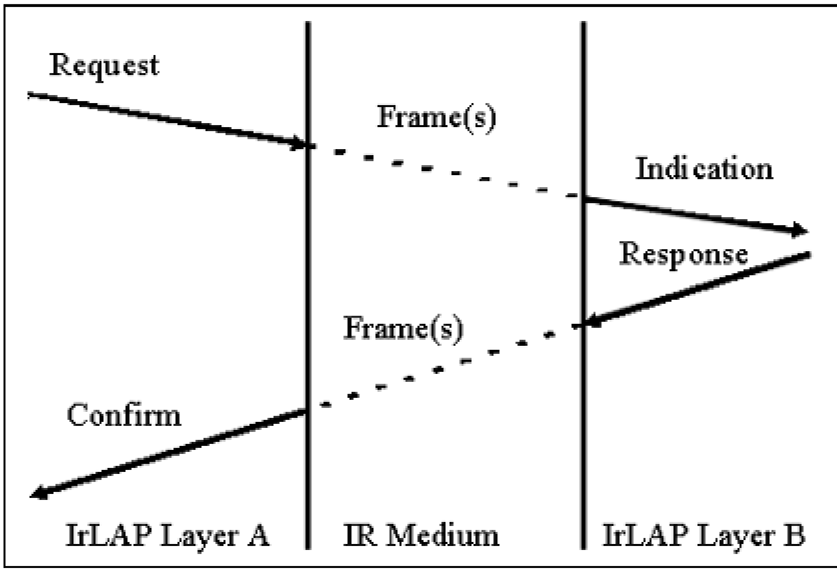
A.4.6 IrLAP Frame Wrappers

Before sending, a frame is wrapped with framing information. Three different frame wrappers are used by IrLAP, depending upon the speed of the connection.

- Asynchronous (ASYNC) Framing: 9600 bps - 115.2 kbps
- Synchronous (SYNC) HDLC Framing: 576 kbps and 1.152 Mbps
- Synchronous 4 PPM Framing: 4 Mbps

A.4.7 Service Primitive Diagram

IrLAP operations are described in the specification using service primitives. You can think of the service primitive as a conceptual model of an API for an operation that IrLAP performs (actual APIs for IrLAP services are completely up to the developer). This diagram illustrates an operation: it starts with a service request, travels across the link as a frame, is reported as an indication (frequently an up-call) on the receiving side; the receiver then formulates a response which travels back as a frame, finally resulting in a confirm (often an up-call) to the original requester.



A.4.8 IrLAP Services

A number of services are defined in the IrLAP specification. Not all services are necessary for all devices, and the IrLAP specification (along with the IrDA Lite standard) describes the minimum requirements. The most important services include the following:

- Device Discovery: Explores the nearby IR-space to see who is present and get some hint as to what they can do
- Connect: Chooses a specific partner, negotiates the best possible communication parameters supported by both sides, and connects.
- Send Data: The whole reason for all this effort—used by higher layer protocols for almost all of their work.
- Disconnect: Closes down and returns to the NDM state, ready for a new connection.

A.4.9 IrLMP - Link Management Protocol 4.1 IrLMP Overview

The IrLMP layer depends upon the reliable connection and negotiated performance provided by the IrLAP layer. IrLMP is a required IrDA layer, and provides the following functionality:

- Multiplexing - LMP allows multiple IrLMP clients to run over a single IrLAP link.
- Higher level discovery, consisting of:
- Address conflict resolution on IrLAP Discovery. Handles the case of multiple devices with the same IrLAP address by telling them to generate new addresses.
- Information Access Service (IAS). A "yellow pages" describing the services available on a device.

A.5 IrLMP Terminology

In order to have multiple IrLMP connections on a single IrLAP connection, there must be some higher level addressing scheme. The following terminology is used to describe this addressing:

- LSAP (Logical Service Access Point): The point of access to a service or application within IrLMP (for example, a printing service). It is referenced with a simple one byte number, the LSAP-SEL (described next).
- LSAP-SEL (LSAP Selector): A one byte number that corresponds to an LSAP. Think of this as the address of a service within the LMP multiplexor. This byte is broken into ranges—0x00 is the IAS server, 0x01 through 0x6F are legal LMP connections, 0x70 are for connectionless services (not discussed in this paper), and the rest are reserved for future use.

Given the limited number of LSAP-SEL values, services are not assigned fixed "port addresses" as in TCP/IP. Instead, services have fixed published names, and the LMP IAS (yellow pages) is used to look up the LSAP-SEL for a desired service.

A.5.1 IrLMP Services

Here are the services defined in the IrLMP specification. Not all services are necessary in all devices, and the specification (along with the IrDA Lite standard) describes the minimum requirements. Notice that this set is identical to the set listed in the IrLAP section above—it is a common feature of protocol stacks for operations to propagate upward like this, with each layer adding its particular contribution.

- Device Discovery: Finds out additional information about devices in the IR space.
- Connect: Establishes a connection between a pair of services at the LMP level.
- Data - send the data back and forth
- Disconnect: Closes this LMP connection. Note that this does not necessarily close the LAP connection—other LMP connections may still be open.

A.5.2 Frame Format

The IrLMP layer adds the following 2 bytes of information to frames in order to perform its basic operations:

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
C	DLSAP-SEL							r	SLSAP-SEL						

- C: Distinguishes between control and data frames.
- r: Reserved.
- DLSAP-SEL: LSAP-SEL (service address) of the destination of the current frame.
- SLSAP-SEL: LSAP-SEL for the sender of the current frame.

A.6 IAS - The Information Access Service

A.6.1 IAS - The Yellow Pages for Services and Applications

The IAS, or Information Access Service, acts as the "yellow pages" for a device. All of the services/applications available for incoming connections must have entries in the IAS which can be used to determine the service address (LSAP-SEL). The IAS can also be queried for additional information about services.

A full IAS implementation consists of client and server components. The **client** is the component that makes inquiries about services on the other device using the Information Access Protocol (IAP, used only within the IAS). The **server** is the component that knows how to respond to inquiries from an IAS client. The server uses an information base of objects supplied by the local services/applications. In fixed-purpose embedded systems this may be a hard-coded collections of objects, while in a PDA there may be APIs for registering and de-registering services. Note that devices which never initiate LMP connections might include an IAS server only.

A.6.2 IAS Information Model

The IAS Information Base is a collection of objects that describe the services available for incoming connections. The information base is used by the IAS server to respond to incoming IAS queries.

Information base objects consist of a class name and one or more attributes. They are quite similar to entries in the yellow pages of a phone book. The class name is equivalent to the business name in the phone book—it is the official published name of the service or application. IAS clients will inquire about a service using this name. The attributes contain information analogous to phone number, address, and other characteristics of the business. The one essential attribute for every entry is the LSAP-SEL (or service address), which is required in order to make a LMP connection to the service.
An IAS object is made of the following pieces:

- Class Name (up to 60 octets).
- Named attributes (up to 60 octet names):
    - Up to 256 attributes.
    - Attributes value types:
        - User string (up to 256 octets)
        - Octet Sequence (up to 1024 octets)
        - Signed Integer (32-bit).

A.6.3 Getting information using the IAS

There are a number of IAS operations defined in the IrLMP standard, but the most used and only required one is called GetValueByClass. The inquiring party gives the class name (for example, Printer) and the name of the attribute it wants (for example, the LSAP-SEL), and

receives a response that consist of one or more answers (for example, the LSAP-SELs for any printing services in the responding party's information base) or an indication that the service or attribute does not exist.
IAS Query Arguments:

- Class Name Length (1 octet).
- Class Name (Length octets).
- Attribute Name Length (1 octet).
- Attribute Name (Length octets).

Results:

- Return Code:
    - 0: Success, results follow.
    - 1: No such class, no results follow.
    - 2: No such attribute, no results follow.

If the result code indicates success, the call returns the following information:

- List Length (2 octets).
- List of results:
    - Object Identifier (2 octets).
    - Attribute value (based on attribute type).

A.7 TinyTP - the Tiny Transport Protocol

A.7.1 TinyTP Overview

To put TinyTP in its proper perspective, it would help to review the layers covered so far:

- Physical layer defines the hardware requirements and low-level framing of the data.
- IrLAP provides reliable, sequenced data, and trouble-free connection at agreed upon parameters with automatic negotiation to best common parameters.
- IrLMP provides multiplexing of services onto the LAP connection and the IAS "yellow pages" of services available for incoming connection.

TinyTP (TTP for short) is an optional IrDA layer, although it is so important that it should generally be considered a required layer (except in the case of current printing solutions). TTP provides two functions:

- Flow control on a per-LMP-connection (per-channel) basis.
- SAR (segmentation and reassembly).
-

TTP adds one byte of information to each IrLMP packet to perform its task.

A.7.2 Tiny TP Flow Control

Per-channel flow control is currently the most important use of TinyTP. You may recall that IrLAP offers flow control and wonder why another flow control mechanism is needed. To illustrate the need, suppose that a LAP connection is established, and two LMP connections are made on top of the LAP connection using the LMP multiplexing capability. If one side turns LAP flow control on, the flow of data on the LAP connection (which carries all the LMP connections) is completely cut in that direction, and the other side cannot get the data it wants until LAP flow control is turned off. The work of the second side may be seriously disrupted (especially if timers are involved).

If flow control is applied on a per-LMP-connection basis using TinyTP (or other mechanisms), then one side can stop to digest information without negatively affecting the other side.

A.7.3 How TTP flow control works

TTP is a credit-based flow control scheme. It works as follows:

- At connection, some credit is extended by each side. One credit corresponds to permission to send one LMP packet. If you send a credit, you must be able to accept a maximum sized packet. You can see that the number of credits you extend depends entirely on how much buffer space you have. As long as you have buffers, you can send anywhere from 1 to 127 credits.
- Sending data causes credit to be used up (1 unit of credit per packet sent).
- Periodically, the receiver issues more credit. This "credit policy" is entirely at the receiver's discretion, but the policy can make a big difference in the performance of the link. If the sender is constantly running out of credit and having to wait for more, throughput will suffer.
- If a sender has no credit, no data movement can occur, but…
- A credit-only packet can always be sent—it is not subject to flow control.

Although the above description talks about the sender and receiver as if those roles are fixed, it is common for both sides of a LMP connection to send and receive, hence both sides will be issuing and using credit. Note that the credit byte normally travels as part of a LMP data packet, so LMP packets are not needed just for sending credit as long as there is data to send and credit to send with. Obviously a little head scratching is in order to set up an efficient credit policy.

A.7.4 Segmentation and Reassembly

The other TTP function is called SAR (segmentation and re-assembly). The basic idea is that TTP breaks large data into pieces (segmentation), and puts it back together on the other side (re-assembly). The entire piece of data being chopped up and re-constituted is called an SDU, or Service Data Unit, and the maximum SDU size is negotiated when the TTP/LMP connection is first made.

A.7.5 Tiny TP Service Primitives

As with the IrLAP and IrLMP layers, TTP operations are characterized as service primitives:

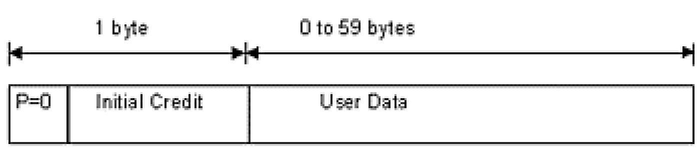
- Connect: Negotiate maximum SDU size.
- Disconnect.
- Data: Reliable sequenced data.
- Local Flow Control: Stop data delivery.
- Udata: Unreliable, unsequenced (pass through to IrLMP).

TTP service primitives focus on the core of the LMP primitives—connect, send, and disconnect, adding a means to exert flow control.

TTP Frame Formats

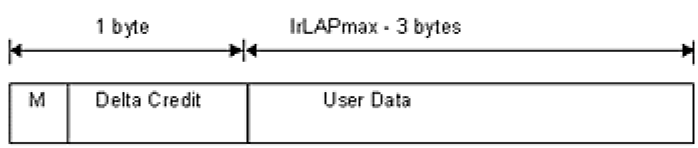
The two frame formats used by TTP are the connect packet (carried with the IrLMP connect packet, hence the limited data length), and the data packet, carried with IrLMP data packets.

A.7.6 Connect Packet



P=Parameter bit: 0-no paramters 1-parameters included

Data Packet



M=More bit: 0-Last segment 1-More segments to follow (not last)

A.8 IrOBEX - Object Exchange Protocol

A.8.1 IrOBEX overview

IrOBEX is an optional application layer protocol designed to enable systems of all sizes and types to exchange a wide variety of data and commands in a resource-sensitive standardized fashion. It addresses one of the most common applications on either PCs or embedded systems: take an arbitrary data object (a file, for instance), and send it to whoever the infrared device is pointing to. It also provides some tools to enable the object to be recognized and handled intelligently on the receiving side. The potential range of objects is wide, encompassing not only traditional files, but also pages, phone messages, digital images, electronic business cards, database records, hand-held instrument results, or diagnostics and programming. The common thread is that the application doesn't need or want to get involved in managing connections or dealing with the communications process at all. Just take the object and ship it to the other side with the least fuss possible. It is very similar to the role that HTTP serves in the Internet protocol suite, although HTTP is very "pull"-oriented in its fundamental design, while OBEX is more evenly balanced.

A.8.2 Characteristics of IrOBEX protocol

OBEX was created to "package" an IrDA communications transaction as completely as possible and thereby dramatically simplify the development of communications-enabled applications. It was further designed to meet the following criteria:

- Simple: Supports most-needed operations/applications.
- Compact: Under 1K code on small system.
- Flexible: Supports data handling for both industry standard and custom types.
- Extensible and Debug-able.
- Works on IrDA, but is transport independent.

A.8.3 Components of IrOBEX protocol

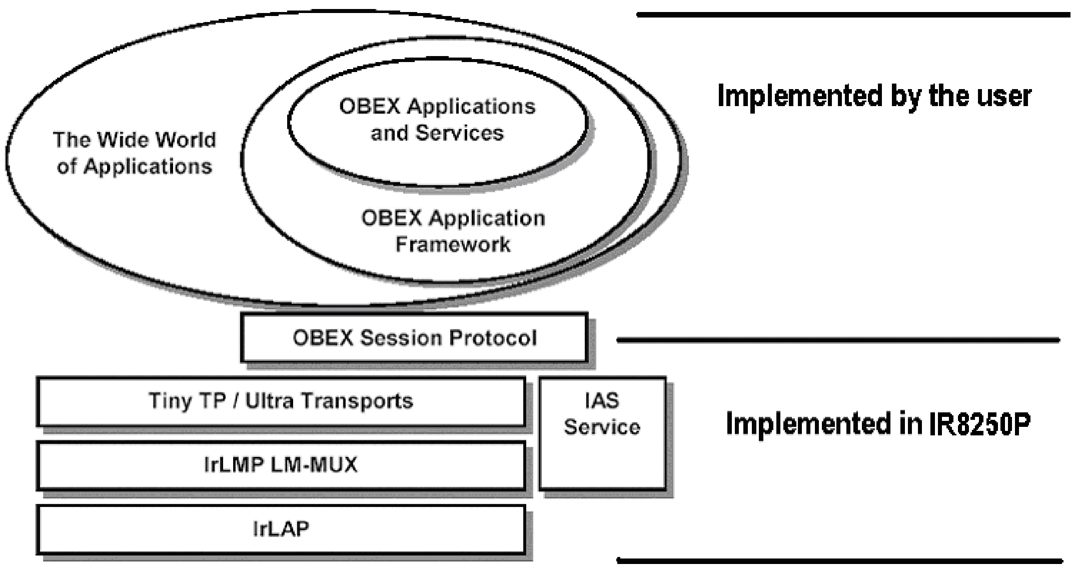
The OBEX standard consists of the following pieces:

- Session model: The rules of conversation governing the exchange of objects. Includes optional negotiation during connection, a set of operations such as Put and Get. Allows terminating the transfer of an object without closing the connection. Supports graceful close of a connection.
- Object model: Provides a flexible and extensible representation for objects and information describing the object.
- Guidelines for use and extension:
    - Defining new session operations.
    - Defining new object types
- IAS entry for a default OBEX server, and suggestions for its capability.

A.8.4 OBEX protocol supported by IR8250P

The OBEX specification consists of two major parts: a protocol and an application framework. This is also illustrated graphically below. The "application framework" is represented in ellipses inside the "wide word of applications" at the upper half of this figure. The "protocol" part is presented in five rectangles at the lower half of this figure.

ACT-IR8250P doesn't and can't provide the "application framework" part of OBEX in the "wide world of applications" (the ellipses in Figure). The host system using IR8250P must implement that part itself.



A.9 IrCOMM - Serial and Parallel Port Emulation

A.9.1 IrCOMM Overview

When the IrDA standards were developed, there was a strong desire to allow existing PC applications that use serial and parallel ports to operate via infrared without change. These applications, collectively known as "legacy applications", included printing, file transfer applications such as LapLink or Carbon Copy, and modem communications.

However, IrDA infrared communications differs significantly from serial and parallel communications. For instance, both serial and parallel cables have individual circuits over which signals can be sent independently and concurrently. By contrast, infrared has a single beam of light, and all information must be fitted into LMP or higher layer packets in a serial stream.

The IrCOMM standard was developed to solve these problems and allow legacy applications to be used over infrared with a minimum of hassle. The key feature of IrCOMM is the definition of a so-called control channel to carry the non-data circuit information. In the stack picture, IrCOMM rests on top of IrLMP and TinyTP.

IrCOMM is an optional IrDA protocol that applies only to certain applications. In general, new applications are better served if they avoid IrCOMM and use other IrDA applications protocols such as IrOBEX, IrLAN, or TinyTP directly. This is because IrCOMM masks some of the useful features built into the lower protocols. After all, its job is to make IrDA look like serial and parallel media that do not have handy features like automatic negotiation of best common parameters and a "yellow pages" of available services.

IrCOMM Service Types

A.9.2 IrCOMM Service Types

Because different applications use the non-data circuits of serial and parallel communications to varying degrees, four service types are defined in IrCOMM:

- 3-Wire Raw (Parallel and Serial Emulation): Sends data only, no non-data circuit information and hence no control channel. Runs directly on IrLMP.
- 3-Wire (Parallel and Serial Emulation): Minimal use of control channel. Uses Tiny TP.
- 9-Wire (Serial emulation only): Uses control channel for status of standard RS-232 non-data circuits. Uses Tiny TP.
- Centronics (Parallel emulation only): Uses control channel for status of Centronics non-data circuits. Uses Tiny TP.