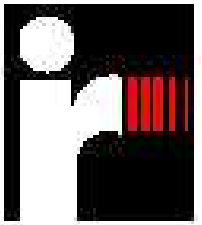
Using Infrared devices with OS/2

Oliver Stein
InnoTek GmbH
ostein@innotek.de



What is IrDA?

- IrDA is the current standard for Infrared communication
  - Standards defined through IrDA board (see www.irda.org)
  - peer-to-peer connection
  - Capable of up to 4MB/s transfer speed
  - no security or authentication

What can I do with IrDA?

- Transfer data between (mobile) devices
  - i.e. "beam" business cards between PDA's and cell phones
  - print to an IR capable printer
  - use your cell phone as a modem
  - configure your cell phone (images, phone book, ring tones, ...)
  - send messages through your cell phone
  - ...

OS/2's IrDA history...

- First IrDA support for IBM Thinkpads was available with Warp 4 (1996):
  - A device driver (IRDD.SYS) and a Ring 3 DLL (LMDLL.DLL) provided the basic interface to the TP specific IR hardware
  - Thinkpad 75x, 76x supported
  - PM/WPS could print over IrDA using a special port driver
  - LAN connectivity through NDIS driver

OS/2 and IrDA today

- IRDD.SYS was removed for CP1/CP2
  - no current hardware support anymore
- "new" IrDA stack from IBM was introduced in 2000
- source code of this stack was released to the DDK in 2002
- works with:
  - Thinkpads 38x, 56x, 600/E/X, 770/E/ED/X, T2x, T30, A2x, A3x, X23, X30.
  - most UART type IR implementations

The IBM IrDA stack for OS/2

- The stack consists of:
  - Physical driver (HW interface, currently implemented for UART chips in most notebooks/motherboards)
  - IrDA base driver
  - IAS driver
  - NDIS driver
  - IrCOMM driver
    - provides connectivity with cell phones, printers, etc.

CONFIG.SYS entries:

- REM BEGIN IBM OS/2 IRDA DRIVER SECTION
- DEVICE=C:\IRDD\IRDA_PHY.OS2 /P0x2E8 /I3 /D3
- DEVICE=C:\IRDD\IRDA_DD.OS2
- DEVICE=C:\IRDD\IAS_DD.OS2
- DEVICE=C:\IRDD\IRCOMM.SYS /COMPORT:COM3 /V
- REM DEVICE=C:\IBMCOM\MACS\IRDANDIS.OS2
- REM END IBM OS/2 IRDA DRIVER SECTION

-> make sure you have the right HW addresses set!

Determining the right HW settings

- Thinkpads with "GUI-BIOS": use PS2.EXE:
  - download from http://www.ibm.com/support/ddrivers.html
  - type "PS2 ? IR" to get current HW settings
- Newer Thinkpads with "Text-BIOS" and all other PC's: check your BIOS settings for IR
- Look for Port address, IRQ and DMA #'s
- Set IRCOMM.SYS virtual COM port to an unused number (i.e. COM3)

How to use the IR NDIS driver

- IR NDIS driver can be used to set up a wireless peer-to-peer connection between two PC's or a PC and an IR "hub" (no longer available)
- configure the NDIS driver in MPTS and use it like a "normal" NIC

How to use the IRCOMM driver

- start PMIRDA.EXE (IR indicator)
- put IR device next to the port
- wait for the IR indicator to start flashing
- now your IR device is available under the COM port you specified for IRCOMM.SYS
- you can now use any application that can talk to a COM port

How to use the IRCOMM driver (2)

- Printers:
  - install the right printer driver
  - choose the IRCOMM COM port as your output port in the settings object
  - start printing

Cell phones

- You can use the IR connection to your cell phone to:
  - connect to an ISP (mobile internet access)
  - configure the phone through a terminal program (AT commands)
  - use special config tools
  - write your own tools!

Alternative to PMIRDA.EXE

- use the IR widget in XWorkplace
- available in IRDD.EXE package on HOBBES
- same functionality, better looks!
- higher reliability

GSM Cell phone custom apps

- Phonebook Manager:
  - read, edit and write your phonebook
  - send and manage SMS's
- Tune Manager (Ericsson only):
  - compose your own ringtones or select from over 900 shipped melodies
  - transfer the tunes to your phone

Writing your own tools in REXX

- There is a DLL (RXIRDA.DLL) that exports the following functions as REXX API's:
  - RxIrDAQueryDevice() returns the name of the detected device or "NOT CONNECTED"
  - RxIrDAQueryCOMPort() returns the name of the port used by the serial device ("COMx:")
- These API's are initialized automatically when PMIRDA.EXE or the IR widget are loaded.
- see IRDD2.EXE on hobbes for details

Writing your own tools in REXX

- COM port handling:
  - use \mptn\dll\sliprexx.dll
  - a multi-threaded COM port library for REXX with ring buffering
  - Code snippet:
    - CALL RxFuncAdd 'SysLoadFuncs', 'SLIPREXX', 'SysLoadFuncs'
    - CALL SysLoadFuncs
    - ...
    - rc=slip_com_open(1, port)
    - cmd= 'AT+CBC'||X2C(d)
    - rc=slip_com_output(port,cmd)
    - rc=slip_com_close(port)