Electronics II
Physics 3620 / 6620

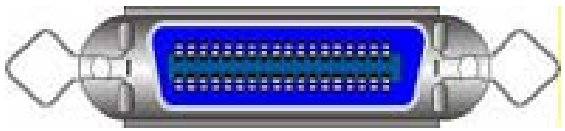
Feb 23, 2008

Part 3

Parallel (LPT) port interfacing in LabVIEW

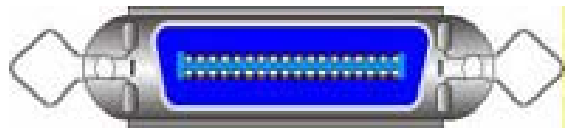
Introduction to the Parallel Port

- The other historical interface found on nearly every PC (still on some laptops) is the “printer” or “LPT” port.
- The “IBM PC” incorporated this port as the standard interface to printers in 1981 and thereby extended its useful existence.
- The interface specification was originally adopted by the Centronics Corp. (originally a division of Wang Laboratories) spun off as an independent company in 1971
- Centronics was primarily a manufacturer of printers
- The parallel interface was developed at Wang (the team included the legendary Dr. An Wang), using an amphenol 36-pin ribbon-cable connector (Wang had a surplus of 20,000 of these)

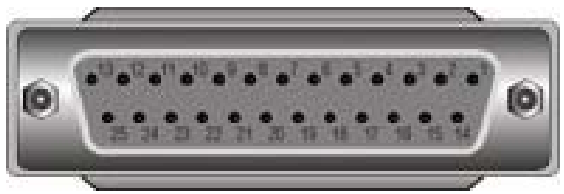


Male: on cable

Female: on printer



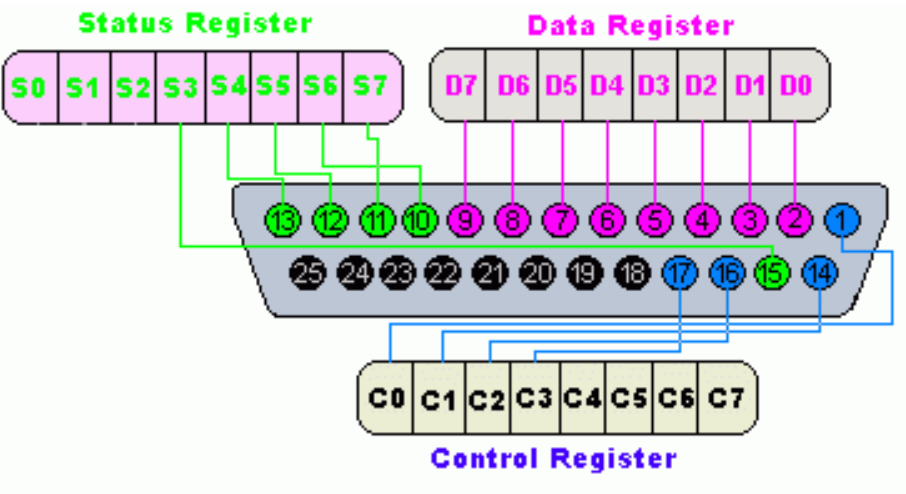
- When the IBM PC adopted the Centronics interface, a female DB25 connector was used at the computer end instead, and a DB-25 to Amphenol 36 cable became the standard printer interface cable until the very late 1990’s



DB25F PC LPT port

Parallel Interface Specifications

- The DB25F parallel port, as seen on the PC, contains three sets of lines
  - 8 bi-dir. data lines (D0-D7)
  - 5 input Status lines (S3-S7)
  - 3 output control lines (C0-C3)
  - Each is seen as part of an 8-bit register in the interface
  - The standard ("Centronics") parallel port was really a "write"-only port



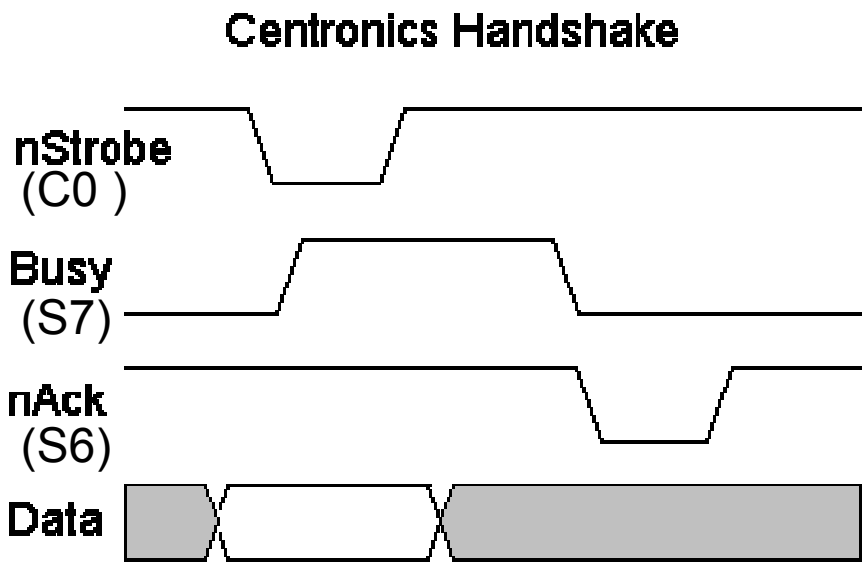
Pin No (DB25)	Signal name	Direction	Register bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

PC interface

- On an IBM PC the usual (though not mandatory) assignment for the address of the two printer ports are at:

Register	LPT1	LPT2
data registar(baseaddress + 0)	0x378	0x278
status register (baseaddress + 1)	0x379	0x279
control register (baseaddress + 2)	0x37a	0x27a

- Communication with the printer follows a simple "hand-shake" sequence:
  - PC Writes data to the D-register
  - PC asserts (low) the "nStrobe" to initiate
  - Printer acknowledges transfer by first asserting (high) "Busy"
  - When the data byte has been read the printer drops (low) "Busy" and then momentarily asserts (low) "nAck" to terminate transfer



Standard Parallel Port "write" cycle

Extended Modes

- In the BIOS setup of a PC you will often see options for the address and interrupt associated with the printer port, along with “mode”.
- Early on vendors hacked the interface to add read capabilities with bit 5 and 6 (no associated pins).
- The two common extensions are “Enhanced parallel Port” (EPP) and Enhanced Capabilities port (ECP) modes
- These use completely different functions on the control and status lines.
- Both are fully bi-diredctional
- WARNING: The SPP, EPP and ECP are not mutually compatible!

Pin	SPP Signal	EPP Signal	ECP Signal	IN/OUT
1	Strobe	Write	HostCLK	Out
2-9	Data 0-7	Data 0-7	Data 0-7	In-Out
10	Ack	Interrupt	PeriphCLK	In
11	Busy	Wait	PeriphAck	In
12	Paper Out / End	Spare	nAckReverse	In
13	Select	Spare	X-Flag	In
14	Auto Linefeed	Data Strobe	Host Ack	Out
15	Error / Fault	Spare	PeriphRequest	In
16	Initialize	Reset	nReverseRequest	Out
17	Select Printer	Address Strobe	1284 Active	Out

Using the LPT port in Labview

- There are two ways to use the LPT port under Labview:
  1. Treat the port(s) as generic I/O channels and use the “Out Port.vi” and “In Port.vi” subVI’s
     - Functions → Connectivity → “Out Port.vi” OR “In Port.vi”
     - The “In Port.vi” allows you to read in 8-, 16- and 32-bit chucnks
     - This is the most direct way of getting this done, but is platform dependent
     - You have to find out for yourself which address the parallel port is in
     - You are recommended to set the port to “normal” or “SPP” mode otherwise things may not work properly.
  2. Use the VISA “serial” tools like we had previously done:
     - Functions → Instrument I/O → Serial → “Configure”, “Read” or “Write”
     - There is a problem with this approach: the VISA drivers are designed to expect the hand-shaking associated with a “standard” parallel port.
     - If you do not have an active device on the other end of the cable then you need to tie pins 11 (Busy) and 12 (Paper Out) to ground, otherwise the attempts to write will simply time-out

Example of using In/Out Port.vi

- In this example we made a while loop enclosing separate read (32 bit) and write (8-bit)
- The address was left as a numerical control. Note that the format is set to “hexadecimal” (most addresses are quoted that way):
  - Right-click on “control” in front panel → Format and Representation, this will pop up a palette/menu
  - Under “Format and Precision Tab” select “Hexadecimal”
  - Under “Appearance” Tab, select “Show Radix”
  - Select OK, enter 378 (hex) into address