

Advanced computing in industrial Automation*Saurabh Saxena¹ and Rajeev Kumar²*

¹Assistant Professor & Head of Department
Department of Computer Science, Indraprastha Institute of technology
Gajraula (J.P Nagar) India

²Computer Science Department, College of Engineering
Teerthanker Mahaveer University, Moradabad (U.P.) India

ABSTRACT: The fundamental problem of every industry that they need more human power for their work such that to handling machine, carry item from one machine to another machine. the Grid research and development community is seeking to solve is how to coordinate distributed resources amongst a dynamic set of individuals and organizations in order to solve a common collaborative goal. The problem of service discovery in a Grid environment arises through the heterogeneity, distribution and sharing of the resources in different virtual organizations. This paper proposes a service discovery framework which is based on Automation. It gives an example of the industrial automation to reduce the man power.

[Saurabh Saxena, Rajeev Kumar. Advanced Computing in Industrial Automation. New York Science Journal 2011;4(7):40-44]. (ISSN: 1554-0200). <http://www.sciencepub.net/newyork>.

KEY WORDS: Advanced computing in industrial automation

Introduction

The recent trend of merging control systems associated with both factory and process automation demands knowledge of diverse fields. Automation applications span plant automation, discrete and batch process control, embedded machine control and manufacturing production line automation. Industrial automation applications include automation of time critical systems that demand precise real-time reading and control.

A PC printer port is an inexpensive but highly useful device to run several computer controlled hardware projects. The port provides eight TTL outputs, five inputs and four bi-directional leads. Some applications of the port include running driver circuits, DC and stepping motors (as we intend to in this project), infrared and radio remote controlling

2. Related Work

Parallel port is a simple and inexpensive tool for building computer controlled devices and projects. The simplicity and ease of programming makes parallel port popular in electronics hobbyist world. The parallel port is often used in Computer controlled robots, Atmel/PIC programmers, home automation, etc. Here a simple tutorial on parallel port interfacing and programming with some examples.

Everybody knows what is parallel port, where it can be found, and for what it is being used. the primary use of parallel port is to connect printers to computer and is specifically designed for this purpose. Thus it is often called as printer Port or Centronics port (this name came from a popular printer manufacturing

company 'Centronics' who devised some standards for parallel port). You can see the parallel port connector in the rear panel of your PC. It is a 25 pin female (DB25) connector (to which printer is connected). On almost all the PCs only one parallel port is present, but you can add more by buying and inserting ISA/PCI parallel port cards.

3. Printer Port Programming**Parallel port modes**

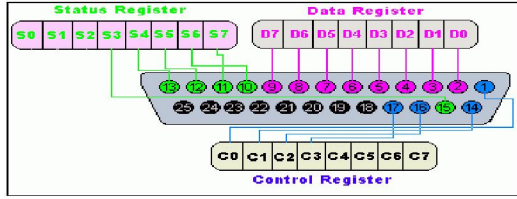
The IEEE 1284 Standard which has been published in 1994 defines five modes of data transfer for parallel port. They are,

- 1) **Compatibility Mode**
- 2) **Nibble Mode**
- 3) **Byte Mode**
- 4) **EPP**
- 5) **ECP**

The programs, circuits and other information found in this tutorial are compatible to almost all types of parallel ports and can be used without any problems (Not tested, just because of confidence!). More information on parallel port operating modes can be found here.

Hardware

The pin outs of DB25 connector is shown in the picture below:-



(a) Port Addresses :

Each printer port consists of three port addresses; data, status and control port. These addresses are always in sequential order. For e.g. in our project we use the LPT2 port.

The Data Port address is 0x0378, the Status Port address is 0x0379 and the Control

Port address is 0x037a. LPT1 is the other printer port. It's Data Port address is 0x03bc, the Status Port address is 0x03bd and the Control Port address is 0x03be. The addresses are always in sequential order.

The lines in DB25 connector are divided in to three groups, they are:

- 1) **Data lines (data bus)**
- 2) **Control lines**
- 3) **Status lines**

As the name refers , data is transferred over data lines , Control lines are used to control the peripheral and of course , the peripheral returns status signals

back computer through Status lines. These lines are connected to Data, Control And Status registers internally. The details of parallel port signal lines are given below:

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	-	-	-

5. TheDLLinput32

The functions in the DLL are implemented in two source files, "input32drv.cpp" and "osversion.cpp". osversion.cpp checks the version of operating system. "input32drv.cpp" does installing the kernel mode driver, loading it , writing/ reading parallel port etc... The two functions exported from input32.dll are

- 1) Inp32(), reads data from a specified parallel port register.
 - 2) Out32(), writes data to specified parallel port register.
- the other functions implemented in Inpout32.dll are

- 1) DllMain(), called when dll is loaded or unloaded. When the dll is loaded , it checks the OS version and loads hwinterface.sys if needed.
- 2) Closedriver(), close the opened driver handle. called before unloading the driver.
- 3) Opendriver(), open a handle to hwinterface driver.
- 4) inst() , Extract 'hwinterface.sys' from binary resource to 'systemroot\drivers' directory and creates a service. This function is called when 'Opendriver' function fails to open a valid handle to 'hwinterface' service.
- 5) start() , starts the hwinterface service using Service Control Manager APIs.
- 6) SystemVersion() Checks the OS version and returns appropriatecode.

4. Parallel port registers

As you know, the Data, Control and status lines are connected to there corresponding registers inside the computer. So by manipulating these registers in program , one can easily read or write to parallel port with programming languages like 'C' and BASIC.

The registers found in standard parallel port are,

- 1) data register
- 2) Status register
- 3) Control register

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

6. Programming Part

```
#include <pic.h>
__CONFIG (INTIO & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT & UNPROTECT & BORDIS &
IESODIS & FCMDIS);
int i;
int j;
unsigned char OutputVal = 1 << 5;
int i, j, k, n, Flag;
const int ones = 225;
const int fiftymys = 2790;
const int twentymys = 1000;
const int fourms = 118;
main ()
{
PORTA = 0x3F;
PORTC = 0x3F;
CMCON0 = 7;
ANSEL = 0;
TRISA = 0b001100;
TRISC0 = 0;
while (1==1)
{
INTCON = 0;
Flag = 0; // Flag Pressed Button
for (i = 0; i < twentymys; ) // Debounce Button Press
for (i = 0; (i < twentymys) && (0 == RA2); i++)
if (0 == RA3)
Flag = 1;
PORTA = 0x00; // Turn off LEDs
PORTC = 0x00;
for (i = 0; i < twentymys; )
// Debounce Button Release
for (i = 0; (i < twentymys) && (1 == RA2); i++)
if (0 == RA3)
Flag = 1;
for (i = 0; (i < ones) && (0 == Flag); i++) // 1s Dlay
for (j = 0; j < ones; j++)
if (0 == RA3)
Flag = 1;
k = TMR0; // Get Start Random Value
```

```

for (i = 0; (i < k) && (0 == Flag); i++) // Random Dlay
for (j = 0; j < fourms; j++)
if (0 == RA3)
Flag = 1;
for (i = 0; i < 255; i++)
for (j = 0; j < 129; j++);
OutputVal = (OutputVal & 0x3c) >> 1;
if ((1 << 1) == OutputVal)
OutputVal = 1 << 5;
PORTC = OutputVal;
INTCON = 0; // Disable/Reset ALL
Interrupts
                Flag = 0; // Flag Pressed Button
                for (i = 0; i < twentyms; ) // Debounce Button Press
for (i = 0; (i < twentyms) && (0 == RA3); i++)
if (0 == RA2)
Flag = 1;
                PORTA = 0x00; // Turn off LEDs
                PORTC = 0x00;
for (i = 0; i < twentyms; ) // Debounce Button Release
for (i = 0; (i < twentyms) && (1 == RA3); i++)
if (0 == RA2)
Flag = 1;

for (i = 0; (i < ones) && (0 == Flag); i++)
// 1s Dlay
for (j = 0; j < ones; j++)
if (0 == RA2)
Flag = 1;
k = TMR0; // Get Start Random Value
for (i = 0; (i < k) && (0 == Flag); i++)
// Random Dlay
for (j = 0; j < fourms; j++)
if (0 == RA2)
Flag = 1;
for (i = 0; i < 255; i++)
for (j = 0; j < 129; j++);
OutputVal = (OutputVal & 0x3c) << 1;
if ((1 << 6) == OutputVal)
OutputVal = 1 << 2;
PORTC = OutputVal;
}
}

```

7. Conclusion

The recent trend of merging control systems associated with both factory and process automation demands knowledge of diverse fields. Automation applications span plant automation, discrete and batch process control, embedded machine control and manufacturing production line automation. Industrial automation applications include automation of time critical systems that demand precise real-time reading and control.

Computer-controlled stepper motors are one of the most versatile forms of positioning systems. They are typically digitally controlled as part of an open loop system, and are simpler and more rugged than closed loop servo systems.

Industrial applications are in high speed pick and place equipment and multi-axis machine CNC machines often directly driving lead screws or ballscrews. In the field of lasers and optics they are frequently used in precision positioning equipment such as linear actuators, linear stages, rotation stages,

goniometers, and mirror mounts. Other uses are in packaging machinery, and positioning of valve pilot stages for fluid control systems.

References

1. "New Computer Alliance Forms". The New York Times. 1991-04-08. Retrieved 2007-05-03.
2. "New breed of computers based on new standard UNIX/RISC software debuts: Compaq's Rod Canion says 'ACE' destined to be environment of choice for the 1990s.". Software Industry Report. 1991-04-15. Retrieved 2010-02-20.
3. "Apache Group of True Blue UNIX System V.4 ACE Renegades Ready to Show Their Hand". Computer Business Review. 1991-10-23. Retrieved 2007-05-03.
4. Jonathan Cassell; Gerry Khermouch, Craig Stedman, Stuart Zipper (1992-05-04). "Is ACE consortium in the hole as Compaq, SCO throw in cards?". Electronic News. Retrieved 2007-05-03.
5. Search engine www.google.com

6/5/2011