CO326: Industrial Networks

Pre Lab

Lab 01 - Parallel Port I/O

Introduction

In computing, the parallel port is an interface used on computers for connecting peripheral devices such as printers (in the early days). In a parallel port, data is sent parallelly; multiple bits of data at once. i.e.: parallel data communication.

Figure 01: Parallel port interface.

Figure 02 shows the pin diagram of the parallel port

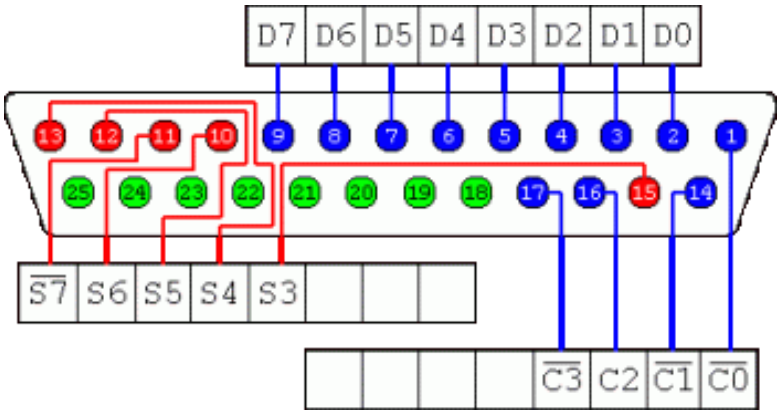


Figure 02: Pin diagram of the parallel port

The D7..D1 Register is the data register and S7..S3 is the status register which we will be using to give outputs and inputs respectively.

As you can see, the pin numbers are not connected to the registers in consecutive order. Also, sometimes they are interchanged, and sometimes they are inverted. You should pay attention carefully to these pins when you program and connect the port into a circuit. You may have to interchange wires and use shift operations/bitwise operations to handle inverted input/output.

Parallel Port Pinout Diagram is as follows (Figure 03)

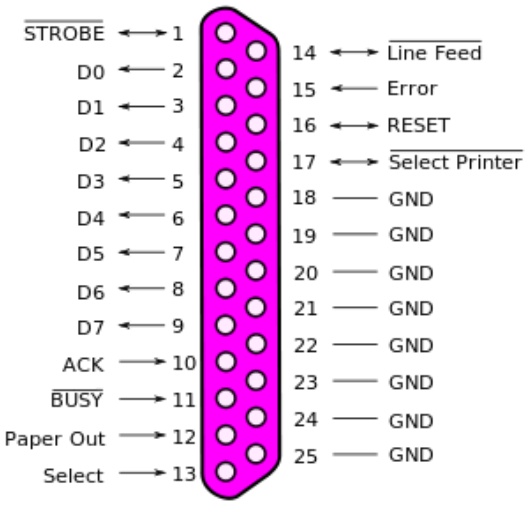


Figure 03: Parallel port pinout

Programming the parallel port

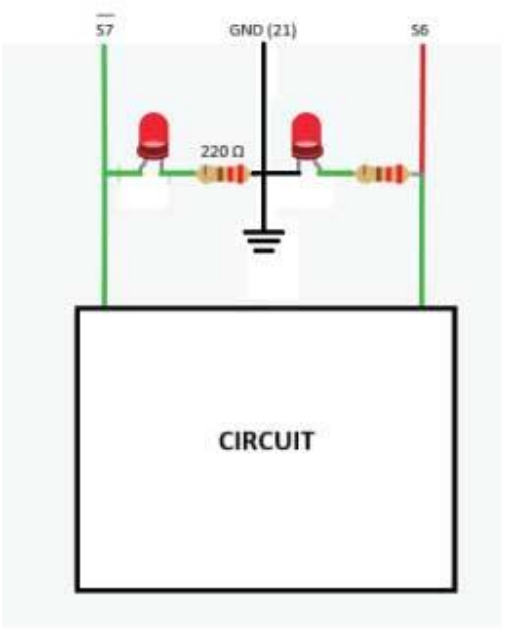
This is a sample program which shows you how to use the parallel port. It reads the value of the status port register and directly writes that value to the data port register, without taking into consideration about bit inversions.

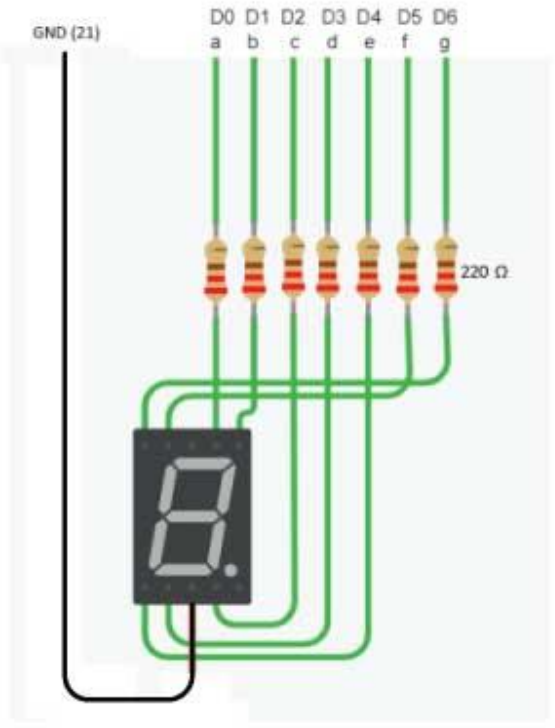
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/io.h>

#define DATA_PORT 0x378         /* parallel port base address */
#define STATUS_PORT DATA_PORT+1
#define CONTROL_PORT DATA_PORT+2

unsigned char status, data;


void main(){

        if (ioperm(DATA_PORT, 1, 1)){
            fprintf(stderr, "Access denied to %x\n", DATA_PORT), exit(1);
        }

        if (ioperm(STATUS_PORT, 1, 1)){
            fprintf(stderr, "Access denied to %x\n", STATUS_PORT), exit(1);
        }

        status = inb(STATUS_PORT);
        data = status;
        outb(data, DATA_PORT);
}
```

ioperm function set port input/output permissions. *inb* does the port input and *outb* does the port output. Please refer to Linux man pages for more information about arguments and return values.

Lab Exercises - Pre Lab

Consider the following circuit diagram for the lab exercise part 01 and 02





Part 01

1. Write a program to light up each segment of SSD one by one.

Part 02: Display 0-9 numbers on a single 7 segment display

1. Identify the difference between the common anode and common cathode 7-segment display.
2. Improve the previous implementation as counter.
    - Can count numbers from 0 to 9.
    - Count Up should increase the value shown in the 7 segment display by 1.
    - Count Down should decrease the value shown in the 7 segment display by 1.

    Count Up signal is wired to S7 port and Count Down is wired to S6 port.

    Write a program for this task also.