Introduction

The purpose of this laboratory is to introduce the student to the General Purpose Instrumentation Bus (GPIB) equipment. All of the instruments will be included in this experiment. The necessary programs and instructions for the GPIB equipment are given in appendices I and II. Carefully read this material and spend extra time studying the applicable portions before coming to the Laboratory period.

Part I. Power Supply

Examine the front panel controls on the power supply. Try lighting the light bulb with 10V at 1A maximum using the positive 25V power supply. Read the actual voltage and current. One of these readings will be different from the set value because the resistance of the light bulb determines the relationship between the current and voltage. Observe the following points and indicate if the supply is in the constant voltage (CV) or constant current (CC) mode:

<table>
<thead>
<tr>
<th>Vmax</th>
<th>Imax</th>
</tr>
</thead>
<tbody>
<tr>
<td>10V</td>
<td>1A</td>
</tr>
<tr>
<td>14V</td>
<td>1A</td>
</tr>
<tr>
<td>14V</td>
<td>.08A</td>
</tr>
</tbody>
</table>

Observe each point and add more points to illustrate current or voltage control.

NOTE: THE LIGHT BULB IS RATED AT 14 VOLTS.

Part II. Setting the Power Supply with the GPIB

The program PS1 will set the power supply to 5V. (Note: all of the necessary GPIB programs are already on the hard disk.) Use it to light the bulb with two or three different voltages, by changing the voltage setting in the C++ program. With no current setting specified the default value is 1A.

```c
#include <iostream.h>
#include <string.h>
#include <conio.h>
#include <windows.h>
#include <decl-32.h>

void main(void)
{
    char code[50];
    int a;
    long int length;
    int ps;
    ps = ibdev(0,5,0,T3s,1,0); //define and open the power supply
    strcpy(code,"APPL P25V, 5, .05;OUTP ON"); // code to set P25V supply to 5 volts
    length = strlen(code);  //length of code
    ibwrt(ps, code, length);  // output code to the power supply
    cout << "The positive 25V power supply should now be set to 5 volts";
    cout << "\n\n\nPress q (for quit) to return to the C++ program";
    a = 0;
    while ((a != 81) && (a != 113))
    {
        a = getch();
    }
}
```

This program sets the positive 25V power supply to 5 volts. The current limit will be 50ma. Place the bulb across the positive 25V supply.

This is the simplest program possible. There is no error detection included and the system may lock up for certain situations.

In order to compile this program the following files must be included in the .ide file.

```
PS1.CPP
BORLAN~1.OBJ
```

/*
#include <iostream.h>
#include <string.h>
#include <conio.h>
#include <windows.h>
#include <decl-32.h>
void main(void)
{
    char code[50];
    int a;
    long int length;
    int ps;
    ps = ibdev(0,5,0,T3s,1,0); //define and open the power supply
    strcpy(code,"APPL P25V, 5, .05;OUTP ON"); // code to set P25V supply to 5 volts
    length = strlen(code);  //length of code
    ibwrt(ps, code, length);  // output code to the power supply
    cout << "The positive 25V power supply should now be set to 5 volts";
    cout << "\n\n\nPress q (for quit) to return to the C++ program";
    a = 0;
    while ((a != 81) && (a != 113))
    {
        a = getch();
    }
}
The program **PS2** has several features added. It has bus error detection, a voltage loop, a current setting, and a timer statement. Modify this program to light the bulb with voltages of 8V to 14V in steps of 2V for a current of 1A. Carefully observe each point.

```cpp
#include <gpib108.h>

void set_voltage(int ps, double volts)
{
    char code1[100]; // code to set voltage
    long int length;
    sprintf(code1, "%s%8.3f%s", "APPL P25V, ", volts, ";OUTP ON"); // string to set volts
    length = strlen(code1); // find length of code1
    ibwrt(ps,code1,length); //sent code to power supply
    if(ibsta & ERR) error_trap("while setting the power supply"); // checks for errors
    return;
}
```

Part III. V-I Characteristic Curves

Use the program **BULB** to sweep the light bulb voltage from 0 to 14 volts and plot the V-I characteristic curve. This program plots two curves. The first time through the program takes the data as fast as possible. The second time through the programs waits for the light bulb to equalize before taking the data. This demonstrates that data cannot always be taken as fast as the measuring system is capable. A hard copy of this set of curves is required.
Part IV. Other Instruments

Apply a 1KHz, 5Vp-p sine wave from the function generator to the oscilloscope. Adjust the oscilloscope for a stable waveform with about three cycles showing on the screen.

Use the GPIB program, PLOT, to obtain a hardcopy of the oscilloscope waveform.

No Pre-Lab Required
Before leaving lab today hard copies of the two curves are required.