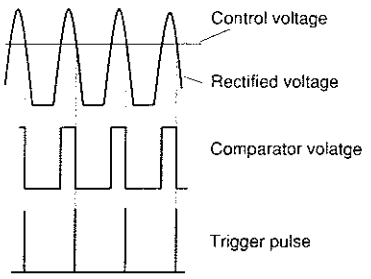
Linear, 90°-180° scr trigger

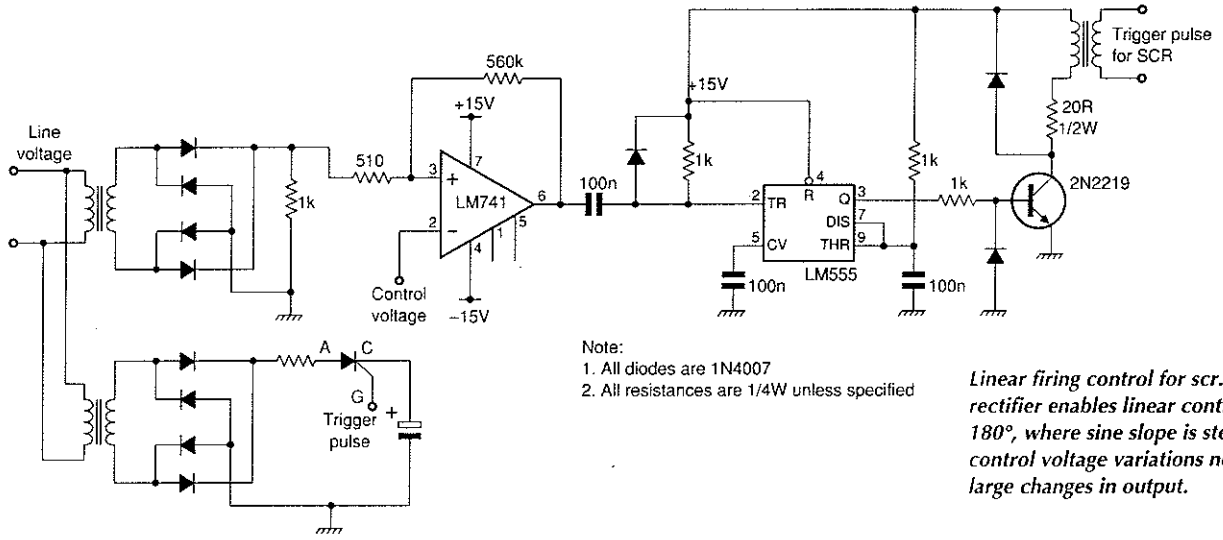
When 90°-180° scr firing control is needed in an scr-controlled power supply and a simple capacitor filter is in use, this arrangement reduces non-linearity of control characteristic near 180°, where the slope of the sine wave is steep.

A comparator 741 compares the rectified output of the bridge with the control voltage, its output falling edge triggering a 555-based monostable to trigger the scr. Firing angle now varies bwteen 90° and 180° for a control voltage varying from the peak of the full-wave rectified ac to zero.

Output voltage on the capacitor is now stable to within about 0.05%, even at a firing angle near 180°.

M Revathi, Ved Prakash and R Yogesh
Cat-Indore
India





Note:
1. All diodes are 1N4007
2. All resistances are 1/4W unless specified

Linear firing control for scr. Full-wave rectifier enables linear control even near 180°, where sine slope is steep and small control voltage variations normally give large changes in output.

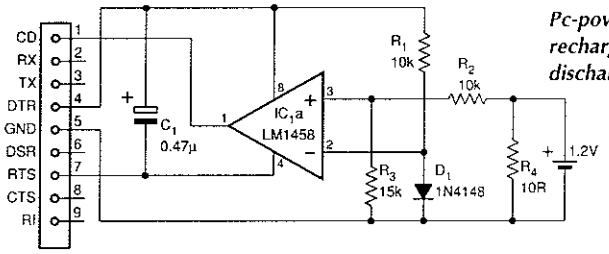
Pc tests rechargeable battery capacity

This little circuit and some software, plus the pc, constitute a tester for rechargeable batteries.

The circuit shown takes its power from the pc's serial port, the voltages being set up in software. Diode D1 holds the inverting input of the op-amp to 0.6V so that a voltage over 1V from the battery, divided by $R_{2,3}$, takes the op-amp output high.

Resistor R_4 discharges the battery at 100mA, the op-amp output going low when the battery voltage falls below 1V. The C program measures the time during which the voltage is greater than 1V and, as soon as it falls below this figure, calculates the battery capacity and displays it on screen.

Yongping Xia
Torrance
California
USA



Pc-powered battery tester for rechargeables. Program measures discharge time and calculates capacity.

Listing for checking cells on the pc

```
#include <stdio.h>
#include <dos.h>
#include <time.h>
#include <conio.h>
#define DISCHARGE_CURRENT  0.12
#define MCR          4
#define MSR          6
struct time t;
time_t start_time, read_time;
int i, base_add1=0x3f8, base_add2=0x2f8;
double bat_time;
void set_port(void)
{
outportb(base_add1+MCR, 0x01:
delay(1000);
}
int read_port(void)
{
int data;
data=(inportb(base_add1+MSR)&0x80)/128;
return (data);
}
void dis_data(void)
{
long run_time, run_hour, run_min, run_sec,j;
read_time=time(NULL);
run_time=(long)difftime(read_time,
                        start_time);
run_hour=run_time/3600;
run_min=(run_time-run_hour*3600)/60;
run_sec=run_time-run_hour*3600-
                        run_min*60;
bat_time=DISCHARGE_CURRENT*(float)
                        run_time/3.6;
gotoxy(2, 1);
printf("Battery has %.2fmAH", bat_time);
gettime(&t);
gotoxy(1,24);
if (run_hour<10)
  printf ( "0" );
printf("%d:", run_hour);
if (run_min<10)
  printf("0");
printf("%d:", run_min);
if (run_sec<10)
  printf("0");
printf("%d", run_sec);
}
void main(void)
{
  int read_data;
  clrscr();
  set_port();
  start_time=time(NULL);
  gotoxy(60,24);
  printf ("Hit any key to  quit");
  bat_time=0;
   do{
    read_data= (read_port ());
    dis_data();
    delay(1000);
   } while(!kbhit() && read_data!=0);
  if (read_data==0)
  {

  gotoxy(1,24);
  printf("Test is done");
  getch();
  }
}
```