รายการอ้างอิง

1. Yoo, J., Fahrenbruch, A.L. and Bube, R.H., "Transport mechanisms in ZnO / CdS / $CuInSe_2$ solar cell " ,J. Appl. Phys. , 69(9) , pp. 4694 - 4699, 1990.
2. Rockett, A. and Birkmire, R.W., " $CuInSe_2$ for photovoltaic applications ", J. Appl. Phys., 70(7) , pp. 81 - 96, 1991.
3. Mitchell, K.W., and others., " $CuInSe_2$ cell and modules" , IEEE Transactions on electron devices , vol 37 , No.2, February 1990.
4. Grove, A.S., Physics and Technology of Semiconductor Devices , John Wiley & Sons, Inc., Singapore , 1967.
5. Miller, W.A. and Olsen, L.C., " Current Transport in Boing (Cd,Zn)/$CuInSe_2$ Solar Cells ", IEEE Transactions on electron devices ,Vol. ED-31 ,No. 5 , pp. 654 - 661, 1984.
6. Milnes, A.G., and Feucht, D.L., Heterojunctions and Metal-Semiconductor Junctions , Academic Press, Inc., New York , 1972.
7. Sharma, B.L., and Purohit, R.K., Semiconductor Heterojunctions , pp 1 - 37 , Pergamon Press Ltd., New York , 1974.
8. Shaw, M.P., Properties of junctions and barrier , pp 32 - 50 , Acadamic Press, Inc., 1978.
9. Sze, S.M., Physics of Semiconductor Devices , John Wiley & Sons, Inc., New York, 2nd ed., 1981.
10. กัลยา เอี้ยประเสริฐศักดิ์ , "การศึกษารอยต่อแบบพี-เอ็น โฮโมจังชันคอปเปอร์อินเดียมไดซิลิไนด์ ", วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาฟิสิกส์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย 2531.
11. Schroder, D.K., Semiconductor Material and Device Characterization, John Wiley & Sons, Inc., Singapore , 1990.
12. ดร. โกศล เพ็ชร์สุวรรณ์ และ ดร. มาซามอริ อีดะ , เทคโนโลยีสารกึ่งตัวนำ , หน้า 48 , สมาคมส่งเสริมความรู้ด้านเทคนิคระหว่างประเทศ , พ.ศ. 2522

13. Stupelman, V. and Filaretov, G., Semiconductor Devices, Mir Publishers, Moscow, 1976.

14. Hess, K., Advanced Theory of Semiconductor Devices , Prentice Hall International, Inc., Singapore , 1988.

15. ยุทธ อัครมาศ , ฟิสิกส์ของระบบเซลสุริยะ , ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย , พ.ศ. 2530

16. Commission of the European Communities Joint Research Center , Standard Procedures for Terrestrial Photovoltaic Performance Measurements , Ispra Establishment, Italy, 1981.

17. Ostoja, P., Characterization of Solar Cells , CNR - L.A.M.E.L. , Bologna , Italy , 1987.

18. Böer, K.W., "Polycrystalline semiconductor heterojunction modeling (CdS/$CuInSe_2$) ", Solar Cells, **16**, pp. 591 - 609 , 1986.

19. Christoforou, N., et al , " Characterization of CdS-$CuInSe_2$ solar cells by current-voltage, capacitance-voltage and capacitance-transient measurements", pp. 966 - 971 Can. J. Phys. , Vol **65**, 1987.

20. Tell, B., Wagner, S., Bridenbaugh, P.M., " Motion of p-n junctions in $CuInSe_2$ " , pp. 454 - 455, Appl. Phys. Let., Vol. 28, No.8, 1976.

21. ชาญวิทย์ จิตยุทธการ , " การเตรียมและศึกษาลักษณะเฉพาะทางไฟฟ้าของฟิล์มบางคอปเปอร์อินเดียมไดซีลีไนด์ ", วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาฟิสิกส์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย 2538.

22. Schock, H.W., " Solar cells based on $CuInSe_2$ and related compounds: recent progress in Europe ", Solar Energy Materials and Solar Cells , **34**, pp. 19 - 26, 1994.

23. Schock, H.W., " $CuInSe_2$ and Others Chalcopyrite-Based Solar Cells " , pp. 42 - 44 , MRS Bulletin, 1993.

24. Phillips, J.E., et al , " Polycrystalline Heterojunction Solar Cells : Device Perspective ", pp. 206 - 209, American Institute of Physics , 1992.

25. Arienzo, M. and Loferski, J.J., " Single-crystal solar cell heterojunctions involving N-cadmium sulfide", pp. 3393 - 3403, J. Appl. Phys., **51**(6), 1980.

26. สิงหเดช แตงจวง , " การเตรียมฟิล์มบางแคดเมียมซัลไฟด์ สำหรับเซลล์แสงอาทิตย์ โดยเทคนิคการเคลือบแบบอาบสารเคมี ", 2538. (เอกสารไม่ตีพิมพ์)

ภาคผนวก

โปรแกรมคอมพิวเตอร์ของระบบวัดลักษณะเฉพาะ

ทางไฟฟ้าของรอยต่อวิวิธพันธุ์

ในการใช้งานระบบวัดลักษณะเฉพาะทางไฟฟ้าที่จัดสร้างขึ้นนั้น จะต้องอาศัยชุดโปรแกรมสำหรับควบคุมระบบ ซึ่งจะประกอบด้วยแฟ้มข้อมูลหลักจำนวน 5 แฟ้มด้วยกัน คือ

- IVCMS2.PAS
- IEEEPAS.PAS
- IEEEPAS.TPU
- PAS488.OBJ
- IEEETPW.PAS

โดยแฟ้มข้อมูล IVCMS2.PAS จะเป็นโปรแกรมหลัก ซึ่งซอร์สโคด (source code) ของแฟ้มข้อมูล IVCMS2.PAS มีดังนี้

```
{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R-,S+,T-,V+,X+}
{$M 16384,0,655360}

PROGRAM IV_MEASUREMENT_SYSTEM;

{ # For measure I-V characteristic with GPIB IEEE488 Interface card
  and KEITHLEY Model 617 programmable electrometer.}

USES ieeepas,crt,dos;

CONST  K617 = 15;          { GPIB address of the instrument }
     dpath = 'C:\PASCAL7\KEITHLEY\PASCAL\DATA\';
VAR    fpt : text;


{------------------ end header ----------------}


Procedure MASKBIT7(Var Instring : String);
{ Use this Procedure in case of the GPIB line has some problems. }
Var  SLEN : Integer;
   OutStr : String;
   EachChar : Byte;
   ThisChar : Char;
Begin
  OutStr := '';
  SLEN := 1;
  Repeat
    ThisChar := Instring[SLEN];
    EachChar := Ord (ThisChar);
    EachChar := EachChar AND 127;
```

```
    OutStr := OutStr + Chr(EachChar);
    SLEN := SLEN + 1;
   Until SLEN > Length(Instring);
   Instring := OutStr;
  End;

  { ------------------- maskbit7 complete ------------}

  Procedure Beep(tb :integer);
  begin
    case tb of
      0 : begin
          sound(500) ; delay(30);      {General Beep 1}
          nosound;
          end;
      1 : begin
          sound(800); delay(30);     {General Beep 2}
          nosound;
          end;
      2 : begin
          sound(1000); delay(200);     {Calculateion finish Beep}
          sound(1600); delay(200);
          sound(2400); delay(100);
          nosound;
          end;
      3 : begin
          sound(900); delay(70);       {Return to main menu Beep}
          sound(1300); delay(50);
          nosound;
          end;
    end;
  end;

  {------------- Beep  complete ----------------}


  Procedure SetUp617(fn :string);
  Var
   Status : Integer;
   rstr,d : String;
   len8   : Word;

  Begin
   Status := 8;
   gotoxy(8,10);
   textcolor(28);
   writeln('                    INITIALIZING ');
   writeln;  textcolor(11);
   Repeat
    Initialize (21,0); Delay (1000);
    Send ( K617, 'F0XR1XO0XC0XZ0XN0XD0XB0XG1X', Status);  {Set status state}
    delay(1000);
    send(K617,'C1X',status); delay(2000);   {set zero}
    send(K617,'Z1X',status); delay(1000);   {zero check on}
    send(K617,fn,status); delay(1000);   {set function}
    send(K617,'R0X',status);             {set auto range}
    enter(d,30,len8,K617,status) ;
   Until Status = 0;
   textcolor(15);
   writeln('                         !!!  READY  !!! ');
   delay(300);
   Beep(1);
  End;

  {---------------- SetUp617  complete ---------------------}

  Procedure RELAY(OnOff :integer);
  begin
    case OnOff of
       1  : port[$378] := 255;        {### Relay  ON ###}

       0  : port[$378] := 0;          {!!! Relay OFF !!!}
```

```
    end;
  end;

  { ---------- Relay complete ---------- }

  Procedure SetVoltTo (Value : Real);
  Var
   StrVal : String;
   Stat1  : integer ;
  Begin
   Str (Value, StrVal);
   StrVal := 'V'+StrVal+'X' ;
   Send ( K617, StrVal, Stat1);
  End;

  {---------- SetVoltTo complete ----------}

  Procedure ReadCurrent (var loop :integer ; Var Current : Real);
  Var
   StrCur    : String;
   len       : Word;
   Stat2,k   : Integer;
   ValCur,SumI : Real;
  Begin
   SumI := 0 ;
   for k := 1 to loop do
   begin
   Enter (StrCur,30,len,K617,Stat2);
   Val (StrCur, ValCur, Stat2);
   SumI := SumI + ValCur ;
   end;
   Current := SumI/loop;
  End;

  { ---------- ReadCurrent complete ---------- }


  procedure RESISTANCE;
  var
     status,i,code          :integer;
     stat,Rdata,d,reset2,resis  :string;
     l                      :word;
     SumR,AvrgR,LVolt,Rnum      :real;
  const
       k = 20 ;   { Loops to repeat and average }
     dt = 100 ;   { set delay time}
  begin
   textbackground(1);
   clrscr;
   Resis := 'F2X' ;
   SetUp617(Resis);
   textbackground(0);
   clrscr;
   gotoxy(1,3);
   textcolor(9);
   writeln('              ++++++++++++++++++++++++++++++++++++++++++++ ');
   writeln('              + SAMPLE RESISTANCE MEASUREMENT SECTION + ');
   writeln('              ++++++++++++++++++++++++++++++++++++++++++++ ');
   gotoxy(22,10);
   write('Connect Load and Press a Key to Start ! ');
   readkey;
   gotoxy(1,10);
   clreol;
   textcolor(28);
   writeln('                    !!! Measuring Please Wait !!!');
   writeln;

    { Start measure resistance and then average }

   textcolor(15);
   writeln;
   delay(200);
```

```
    SumR := 0;
    send(K617,'C0X',status);
    for i := 1 to k  do
     begin
       send(K617,'F2X',status);
       enter(Rdata,30,l,K617,status) ;
       if (status =  0) then stat := 'OK' else stat:= 'ERROR';
       {MASKBIT7(Rdata);}    {!Use this line when GPIB line has problems.}
       val(Rdata,Rnum,code);
       gotoxy(10,15);
       writeln(' Resistance is ',Rnum,  ' Ohm.   Status is ',stat);
       Beep(1);
       SumR := SumR + Rnum ;
       delay(dt);
     end;
    writeln;
    writeln('                    zzzzzz  FINISH  zzzzzz ');
    Beep(2);
    AvrgR := SumR/k ;
    gotoxy(1,10);
    clreol;
    gotoxy(1,20);
    writeln(' The Average Resistance is ',AvrgR,' Ohm.');
    LVolt := AvrgR*0.002 ;
    writeln(' The Maximum Load Voltage for this Load is ñ ',LVolt,' Volt.');
    writeln;
    textcolor(Red);
    write(' ** Press a key to return to Main Menu ');
    readkey;
    Beep(3);
    send(K617,'C1F0R0Z1G0X',status);   { Return status state }
    enter(d,30,l,K617,status) ;
    initialize(21,0);
  end;

  {---------- Resistance Measurement complete ----------}


  Procedure Scan_I_V (WhichWay,Step,FBmax,RBmax : real; rt :integer);
  Const
       TimeDelay = 10 ;          { Set Speed and Current limit hear ! }
       I_default = 2.5e-03 ;
  Var
   Vsend, GetCurrent, Direction : Real;
   Stat,Hz                : Integer;
   len                    : Word;
   rstr                   : String;
  Begin
   Vsend := 0;
   Direction := WhichWay/Abs(WhichWay);
   Repeat
    SetVoltTo (Vsend);
    Delay (TimeDelay);

    ReadCurrent (rt,GetCurrent);

    WriteLn (fpt,'  Voltage is ',Vsend,'     Current is ',GetCurrent);

    Vsend := Round( (Vsend + Direction * Step) * 1000)/1000;
    if Direction = 1 then Hz := 0 else Hz := 1 ;
    Beep(Hz);
   Until (GetCurrent > I_default) or (GetCurrent < -1*I_default) or (Vsend > FBmax) or (Vsend < RBmax);
   Send ( K617, 'O1XV+0.000E+00X', Stat);
   Enter (rstr, 30, len, K617, Stat);
   { If current more (or less) than default setting current or voltage over
   the voltage maximam then stop working and Set V-Source to zero.  }
  End;

  { ---------- Scan_I_V  complete ----------}


  procedure IV_MEASURE;
```

```
const  FBside = +1;
       RBside = -1;


var
       fname,Idata,device,rstr,cond,Cur   :string;
       stepV,MaxFB,MaxRB                 :real;
       status                      :integer;
       len2                        :word;


begin
  clrscr;
  gotoxy(1,3);
  textcolor(10);
  writeln('              ++++++++++++++++++++++++++++++++++++++++ ');
  writeln('              ± I-V CHARACTERISTIC SCANING SECTION  ± ');
  writeln('              ++++++++++++++++++++++++++++++++++++++++ ');
  writeln; writeln;
  window(1,7,80,25); textbackground(blue);
  clrscr;
  Cur := 'F1X' ;
  SetUp617(Cur);
 repeat
  send(K617,'O1XV+0.000E+00X',status); delay(1000);
  clrscr;  textcolor(15);
  writeln(' ### Parameters Setting ### ');
  writeln(' Type " Quit " for go back to Main Menu !');
  writeln;
  write('          Device Code : '); textcolor(yellow);
  readln(device);          textcolor(15);
  if (device = 'QUIT') or (device = 'quit') or (device = 'Quit')
    then  writeln( '              ##### Teminated by User ##### ')
    else  begin
  write('          Condition   : '); textcolor(yellow);
  readln(Cond);  textcolor(15);
  write('          Filename for Saving Data => '); textcolor(yellow);
  readln(fname); textcolor(15);
  write('          Steping Voltage ,50mV Increment (V) => ');    textcolor(yellow);
  readln(stepV); textcolor(15);
  write('          Maximum Voltage for Forward Bias (V) => '); textcolor(yellow);
  readln(MaxFB); textcolor(15);
  write('          Maximum Voltage for Reverse Bias (V) => '); textcolor(yellow);
  readln(MaxRB); textcolor(13);
  writeln;
  write('          !!!! Press a key to start measurement !!!! ');
  readkey;
  assign(fpt,dpath+fname);
  rewrite(fpt);
  writeln(fpt,'              I-V CHARACTERISTIC DATA ');
  writeln(fpt);
  writeln(fpt,' Device Code : ',device);
  writeln(fpt,' Conditions  : ',cond);
  writeln(fpt);
  clrscr;
  send(k617,'R0XC0XD1X',status);
  enter (rstr, 30, len2, K617, Status);
  gotoxy(1,10);   textcolor(28);

                { Scan Forward Bias }

  writeln('                 ************************** ');
  writeln('                 * Measure Forward Bias *  ');
  writeln('                 ************************** ');
  Scan_I_V(FBside,stepV,MaxFB,MaxRB,10);
  gotoxy(1,10); clreol;

                { Scan Reverse Bias }

  writeln('                 ************************** ');
  writeln('                 * Measure Reverse Bias *');
```

```
  writeln('                    ************************** ');
  Scan_I_V(RBside,stepV,MaxFB,MaxRB,10);
  close(fpt);
  Beep(2);
  send(K617,'C1X',status);
  end;
 until (device = 'QUIT') or (device = 'quit') or (device = 'Quit') ;
  window(1,1,80,25); textbackground(black);
  send(K617,'V0O0F0R0C1Z1X',status);
  enter(rstr,30,len2,K617,status);
  initialize(21,0);
  Beep(3);
end;

{ ---------- I-V Scan mode  complete ----------}

Procedure Get_Temp;
var   stat              :integer;
      T1                :string;
      T2                :real;
      l                 :word;
begin
  send(k617,'F0XR0XG1X',stat); delay(700);
  send(k617,'C0X',stat);
  enter (T1,30,l, K617, stat);
  Beep(1);
  Val (T1,T2,Stat);
  writeln(fpt,'Voltage from Thermocouple is ',T2,' volt.');
end;

{ ---------- Get_Temp complete ---------- }

procedure IV_vs_T_MEASURE;
const   FBconst = +1 ;
var     fname,Idat,device,rstr,cond,V    :string ;
        stat                    :integer ;
        len3                    :word;
        Vstep,VMax,t1,t2                :real ;
        k                       :char;

begin
  clrscr;
  gotoxy(1,3);
  textcolor(10);
  writeln('              ++++++++++++++++++++++++++++++++++++++++++ ');
  writeln('              ± FORWARD BIAS I-V vs. T  SCANING SECTION ± ');
  writeln('              ++++++++++++++++++++++++++++++++++++++++++ ');
  gotoxy(27,13); textcolor(15);
  writeln;  window(1,7,80,25);
  textbackground(1); clrscr;
  V := 'F0X' ;              { Set up Keithley }
  SetUp617(V);
{  port[$378] := 0;          set Relay off}
  send(K617,'O1XV+0.000E+00X',stat); delay(1000);

   { Parameters setting section }
  clrscr; textcolor(15);
  writeln(' ### Parameters Setting ### ');
  writeln(' Type " Quit " for go back to Main Menu !');
  writeln;
  write('         Device Code : '); textcolor(yellow);
  readln(device);         textcolor(15);
  if (device = 'QUIT') or (device = 'quit') or (device = 'Quit')
    then writeln( '              ##### Teminated by User ##### ')
    else  begin
  write('         Condition   : '); textcolor(yellow);
  readln(Cond); textcolor(15);
  write('         Filename for Saving Data => '); textcolor(yellow);
  readln(fname); textcolor(15);
  write('         Steping Voltage ,50mV Increment (V) => ');    textcolor(yellow);
  readln(Vstep); textcolor(15);
  write('         Maximum Voltage for Forward Bias (V) => '); textcolor(yellow);
```

```
  readln(VMax); textcolor(15);

      {open and write data file}
  assign(fpt,dpath+fname);
  rewrite(fpt);
  rewrite(fpt);
  writeln(fpt,'       I-V FORWARD vs TEMPERATURE CHARACTERISTIC DATA ');
  writeln(fpt);
  writeln(fpt,' Device Code : ',device);
  writeln(fpt,' Conditions  : ',cond);
  writeln(fpt);

    { begin loop of measurement }
  repeat
  clrscr;
  textcolor(13);
  gotoxy(1,10);
  writeln('           !!!! Press a key to start measurement !!!! ');
  writeln('           ºº±²² or "Q" to Quit this Section ²²±ºº ');
  k := readkey;
  writeln;
  IF (k = 'q') or (k = 'Q') then
  writeln('                !! PROGRAM  TERMINATED !!')
  else
  begin
  clrscr;
  gotoxy(1,10);   textcolor(28);

            { Scan Forward Bias }

  writeln('                ************************** ');
  writeln('                * Measure Forward Bias *  ');
  writeln('                ************************** ');
{  port[$378] := 0;          set Relay off}
  textcolor(13);
  write('Thermocouple voltage start at (mV) : '); readln(t1);     {=====> Manual mode}
  writeln(fpt,'Thermocouple voltage start at : ',t1,' mV');       {=====> Manual mode}

{  Relay(1);
  write('                ³ON³');
  Get_Temp;
  Relay(0); }  writeln(fpt);
{  write('³OFF³ Measuring ');}

  send(K617,'F1XR0XZ1X',stat);
  send(k617,'C0XD1X',stat);
  Scan_I_V(FBconst,Vstep,VMax,-1,3);  writeln(fpt);
{  port[$378] := 0;          set Relay off}
  beep(1);  beep(1);
  write('Thermocouple voltage stop at (mV): '); readln(t2);     {=====> Manual mode}
  writeln(fpt,'Thermocouple voltage stop at : ',t2,' mV');     {=====> Manual mode}

{  Relay(1);
  write('³ON³');
  Get_Temp;
  Relay(0);} writeln(fpt);
{  writeln('³OFF³');}
  send(k617,'C1XD0XZ0X',stat);


  writeln(fpt,' ********** END of Loop at a Temperature Range ********** ');
  writeln(fpt);
  Beep(2);
  end;
  until  (k = 'q') or (k = 'Q');
  close(fpt);
  end;
  window(1,1,80,25); textbackground(black);
  send(K617,'V0XO0XF0XR0XC1XZ1X',stat);    {RETURN READY STATUS}
  enter(rstr,30,len3,K617,stat);
  initialize(21,0);
  window(1,1,80,25);
```

```
    textbackground(0);
    Beep(3);
  end;


  { ---------- I-V vs T complete ---------- }


  procedure SEND_RECIEVE;
  var   code,stat,data,d   :string;
        l                  :word;
        status,len,i       :integer;
  begin
    clrscr;
    gotoxy(1,3);
    textcolor(13);
    writeln('                ++++++++++++++++++++++++++++++++++++++++ ');
    writeln('                ±  DIRECT COMMAND CODING SECTION   ± ');
    writeln('                ++++++++++++++++++++++++++++++++++++++++ ');
    writeln('                              Refer to IEEE-488 Manual ');
    writeln;
    window(1,8,80,25);
    textbackground(1);
    clrscr;
    repeat
      initialize (21,0);     { make PC a controller at address 21 }
      textcolor(Yellow);
      write('INPUT COMMAND CODE or "EXIT" to Quit  : '); textcolor(15);
      readln(code);
      for i := 1 to Length(code) do
        code[i] := UpCase(code[i]);  {Make NoCap tp Cap }
      code := code+'X' ;
      if ( code <> 'EXITX') or ( code <> 'exitX') then
      begin

      send (K617,code,status);    { device command to set mode }

      if (status =  0) then stat := 'OK' else stat:= 'ERROR';
      write(' Status Respond is  ',stat);  Beep(0);
      writeln;
      textcolor(13);

      enter(data,30,l,K617,status) ;
      writeln(' Data Recieve is => ',data);
     { Maskbit7 (data); }                  {!Use this line when GPIB line has problems}
     { writeln(' Data Recieve is => ',data);}

      if (status =  0) then stat := 'OK' else stat:= 'ERROR';
      writeln(' Status Respond is  ',stat);  Beep(1);
      writeln;

      end;

    until ( code = 'EXITX') ;
    send(K617,'V0O0F0R0C1Z1X',status);
    enter(d,30,l,K617,status);
    initialize(21,0);
    Beep(3);
    window(1,1,80,25); textbackground(black);
  end;

  { ---------- end diract command section ---------- }


  {========== Below this , is the Relay control and checking part ==========}

  procedure relay_count;
   var  i :integer;
   begin
   i := 0;
   clrscr;
```

```
  port[$378] := 0;  {set realy off}
  repeat
   i := i+1 ;
   writeln('                NOW!  i = ',i);
   delay(500);
   port[$378] := 255;
   write('### Relay  ON ###');
   delay(500);
   port[$378] := 0;
   writeln('                !!! Relay OFF !!! ');
  until (i = 20);
   sound(1200); delay(1500); nosound;
   port[$378] := 0;
end;


procedure Relay_call;
var j :integer;
begin
 clrscr;
 port[$378] := 0 ;  {set relay off}
 j := 0;
 repeat
  write('Relay Order [0 for OFF , 1 for ON , 9 for EXIT] => ');
  readln(j);
  case j of
    0: port[$378] := 0    ;
    1: port[$378] := 255  ;
  end;
 until (j=9) ;
end;

{#################### relay control and checking complete ################}

procedure Time_Dependent;
var     fname,datum,datum2,mode,N     :string ;
        stat,pc,time,t4,t5            :integer ;
        len6                          :word;
        t2,t3                         :real ;

begin
  clrscr;
  gotoxy(1,3);
  textcolor(12);
  writeln('              ++++++++++++++++++++++++++++++++++++++++ ');
  writeln('              ±   TIME DEPENDENT MEASUREMENT SECTION ± ');
  writeln('              ++++++++++++++++++++++++++++++++++++++++ ');
  gotoxy(27,13); textcolor(15);
  writeln;   window(1,7,80,25);
  textbackground(1);  clrscr;
  N := 'F0X' ;
  SETUP617(N);
  repeat
  clrscr;
  gotoxy(1,2);
  writeln('       Press a Number of Parameter !');   writeln;  writeln;
  writeln('          [1] Voltage ');
  writeln('          [2] Current ');
  writeln('          [3] Resistance ');
  writeln('          [4] Voltage with Current ');    writeln;
  writeln('          [0] Return to Main Menu ');
  writeln;    writeln;
  write('          Select => ');  readln(pc);
   case pc of

   1 : begin
        writeln;
        write('  * Time Interval set to (sec.) : ');  readln(time);
        write('      Filename to Save ? =  '); readln(fname);
        clrscr;
        send(K617,'N1X',stat);
        send(K617,'C0X',stat);
```

```
    assign(fpt,dpath+fname);
    rewrite(fpt);
    t3 := 0;
    t4 := time*1000;
    repeat
    send (K617,'F0X',stat);    { device command to set mode }
    enter(datum,30,len6,K617,stat) ;
    writeln(fpt,' Time ',t3,' sec.  Voltage = ',datum,' V.');
    beep(0);  delay(t4);
    t3 := t3+time;
    until keypressed;
    close(fpt);
   end;

 2: begin
    writeln;
    write('  * Time Interval set to (sec.) : ');  readln(time);
    write('       Filename to Save ? =  '); readln(fname);
    send(K617,'C0X',stat);
    assign(fpt,dpath+fname);
    rewrite(fpt);
    t3 := 0;
    t4 := time*1000;
    repeat
    send (K617,'F1X',stat);    { device command to set mode }
    enter(datum,30,len6,K617,stat) ;
    writeln(fpt,' Time ',t3,' sec. Current = ',datum,' Amp.');
    beep(0);  delay(t4);
    t3 := t3+time;
    until keypressed;
    close(fpt);
   end;

 3: begin
    writeln;
    write('  * Time Interval set to (sec.) : ');  readln(time);
    write('       Filename to Save ? =  '); readln(fname);
    clrscr;
    send(K617,'C0X',stat);
    assign(fpt,dpath+fname);
    rewrite(fpt);
    t3 := 0;
    t4 := time*1000;
    repeat
    send (K617,'F2X',stat);    { device command to set mode }
    enter(datum,30,len6,K617,stat) ;
    writeln(fpt,' Time ',t3,' sec. Resistance = ',datum,' Ohm.');
    beep(0);  delay(t4);
    t3 := t3+time;
    until keypressed;
    close(fpt);
   end;
 4: begin
    writeln;
    write('  * Time Interval set to (sec.) : ');  readln(time);
    write('       Filename to Save ? =  '); readln(fname);
    clrscr;
    send(K617,'N1X',stat);
    send(K617,'C0X',stat);
    assign(fpt,dpath+fname);
    rewrite(fpt);
    t3 := 0;
    t4 := time*1000;
    t5 := time*500 ;
    repeat
    send (K617,'F0X',stat);    { device command to set mode }
    delay(t5);
    enter(datum,30,len6,K617,stat) ;
    send (K617,'F1X',stat);    { device command to set mode }
    delay(t5) ;
    enter(datum2,30,len6,K617,stat) ;
    writeln(fpt,' Time ',t3,' sec. Voltage = ',datum, ' Current = ',datum2);
```

```
        beep(0);
        t3 := t3+time;
        until keypressed;
        close(fpt);
       end;

   0:  beep(3);
   end;
  until pc=0 ;
  window(1,1,80,25);
  send(K617,'V0O0F0R0C1Z1X',stat);
  enter(datum,30,len6,K617,stat);
  initialize(21,0);
end;
{ ++++++++++ Time measurement complete here! ++++++++++++++++++}

procedure MAIN_MENU;
var c1    :char;

begin
  clrscr;
  c1 := 'y';
  repeat
   textbackground(0);
   clrscr;
   gotoxy(1,3);
   textcolor(14);
   writeln('            ========================================= ');
   writeln('            == I-V CHARACTERISTIC MEASUREMENT SYSTEM == ');
   writeln('            ========================================= ');
   textcolor(5);
   writeln('            SPRL                    version 2.0  ');
   textcolor(3);
   gotoxy(1,9);
   writeln('                    # MAIN MENU #');
   writeln;
   writeln('   [1] Measure Resistance (R) of the Sample. ');
   writeln('   [2] Measure I-V Characteristic (FB & RB Scan). ');
   writeln('   [3] Measure I-V Forward Bias Characteristic vs. Temperature.');
   writeln('   [4] Time Dependent Measurement . ');
   writeln('   [5] Direct Command by Send & Recieve GPIB Code. ');
   writeln('   [6] Relay Control and Checking. ');
   writeln;
   writeln('   [0] Exit the System. ');
   gotoxy(5,20);
   textcolor(15);
   write('Enter the Number =>  '); textcolor(12);
   readln(c1);
   case c1 of

    '1' :begin
      RESISTANCE;
      end;

    '2' :begin
      IV_MEASURE;
      end ;

    '3' :begin
      IV_vs_T_MEASURE;
      end  ;
    '4' :begin
      Time_Dependent;
      end;

    '5' :begin
      SEND_RECIEVE;
      end ;

    '6' :begin
      textcolor(14);
```

```
        RELAY_CALL;
        RELAY_COUNT;
        end;

    end;
  until (c1 = '0') ;
end;


{ ----------*** This is main program and the menu ***---------- }

BEGIN
  textbackground(0);
  MAIN_MENU;
  beep(0); beep(1); beep(0); beep(1); beep(0);
  beep(1); beep(0); beep(1); beep(0); beep(1);
END.
```

รหัสการสั่งงานอิเลคโตรมิเตอร์ ชนิดโปรแกรมได้ Keithley 617 ผ่านทางแผงวงจรเชื่อมต่อ GPIB IEEE488 โดยตรง แสดงได้ดังตารางข้างล่าง

Function:	F0 = Volts F1 = Amps F2 = Ohms F3 = Coulombs F4 = External Feedback F5 = V/I

Range:

	Volts	Amps	Ohms	Coul	X Fdbk	V/I
R0 =	Auto on	Auto on	Auto on	Auto on	Auto on	Auto on
R1 =	200mV	2 pA	2 kΩ	200pC	200mV	200TΩ
R2 =	2 V	20 pA	20 kΩ	2nC	2 V	20TΩ
R3 =	20 V	200 pA	200 kΩ	20nC	20 V	2TΩ
R4 =	200 V	2 nA	2MΩ	20nC	20 V	200GΩ
R5 =	200 V	20 nA	20MΩ	20nC	20 V	20GΩ
R6 =	200 V	200 nA	200MΩ	20nC	20 V	2GΩ
R7 =	200 V	2 μA	2GΩ	20nC	20 V	200MΩ
R8 =	200 V	20 μA	20GΩ	20nC	20 V	20MΩ
R9 =	200 V	200 μA	200GΩ	20nC	20 V	2MΩ
R10 =	200 V	2mA	200GΩ	20nC	20 V	200KΩ
R11 =	200 V	20mA	200GΩ	20nC	20 V	200KΩ
R12 =	Auto off	Auto off	Auto off	Auto off	Auto off	Auto off

Zero Check:	C0 = Off C1 = On
Zero Correct:	Z0 = Off Z1 = On
Suppress:	N0 = Off N1 = On
Display:	D0 = Electrometer D1 = Voltage Source
Read Mode:	B0 = Electrometer B1 = Data Store B2 = Highest (Max) Buffered Reading B3 = Lowest (Min) Buffered Reading B4 = Voltage Source
***Execute:**	X = Execute Device-Dependent Commands *Use "X" to force a sequence. Example: To zero the instrument, send C1XZ1XC0X.

Data Store Control:	Q0 = Every Reading Q1 = 1 reading per second. Q2 = 1 reading per 10 seconds. Q3 = 1 reading per minute. Q4 = 1 reading per 10 minutes. Q5 = 1 reading per hour. Q6 = Every time TRIG button is pressed. Q7 = Off
V-Source Value:	V ± n.nnnnnE ± nn n = Voltage Source Value
V-Source Operate:	O0 = Off O1 = On
Calibration Value:	A ± n.nnnnnE ± nn A = Calibration Value
Store:	L1 = Store Calibration Constants
***Trigger:**	T0 = Continuous, Talk T1 = Single, Talk T2 = Continuous, GET T3 = Single, GET T4 = Continuous, "X" T5 = Single, "X" T6 = Continuous, External T7 = Single, External *Trigger always starts a new reading, regardless of mode (single/continuous).
Status:	U0 = Send machine status word. U1 = Send error status word. U2 = Send data status word.
Data Format:	G0 = Send prefix with data. G1 = Do not send prefix with data. G2 = Send prefix and buffer address with data (if in B1).
SRQ:	M0 = Clear SRQ Mask M1 = Reading Overflow M2 = Buffer Full M8 = Reading Done M16 = Ready M32 = Error Example: M10 programs the 617 to SRQ when a reading is done or the buffer is full.

ข้อมูลทางเทคนิคของ

Keithley model 617 Programable Electrometer

SPECIFICATIONS

VOLTS

RANGE	RESOLUTION	ACCURACY (1 Yr.)* 18°-28°C ±(%rdg+counts)	TEMPERATURE COEFFICIENT 0°-18°C & 28°-50°C ±(%rdg+counts)/°C
200mV	10 μV	0.05+4	0.004+3
2 V	100 μV	0.05+1	0.004+0.3
20 V	1mV	0.05+1	0.005+0.1
200 V	10mV	0.07+1	0.007+0.1

*When properly zeroed.

NMRR: Greater than 80dB on 200mV, 60dB on 2V and 20V, 55dB on 200V range, at 50Hz or 60Hz ±0.1%.

CMRR: Greater than 120dB at dc, 50Hz or 60Hz.

INPUT IMPEDANCE: Greater than 200TΩ in parallel with 20pF (< 2pF guarded).

AMPS

RANGE	RESOLUTION	ACCURACY (1 Yr.)* 18°-28°C ±(%rdg+counts)	TEMPERATURE COEFFICIENT 0°-18°C & 28°-50°C ±(%rdg+counts)/°C
2 pA	100aA	1.6 +66	0.15 +8
20 pA	1 fA	1.6 + 7	0.15 +1
200 pA	10 fA	1.6 + 1	0.15 +0.1
2 nA	100 fA	0.25+ 5	0.015+3
20 nA	1pA	0.25+ 1	0.015+0.3
200 nA	10pA	0.25+ 1	0.015+0.1
2 μA	100pA	0.15+ 4	0.005+3
20 μA	1nA	0.15+ 1	0.005+0.3
200 μA	10nA	0.15+ 1	0.006+0.1
2mA	100nA	0.15+ 4	0.005+3
20mA	1μA	0.15+ 1	0.005+0.3

*When properly zeroed.

INPUT BIAS CURRENT: Less than 5fA (5×10⁻¹⁵A) at 23°C.

INPUT VOLTAGE BURDEN: Less than 1mV except 3mV on 20mA range.

PREAMP SETTLING TIME (to 1% of final value): 2.5s on pA, 15ms on nA, 5ms on μA and mA ranges.

NMRR: Greater than 95dB on pA, 60dB on nA, μA and mA ranges at 50Hz or 60Hz ±0.1%.

COULOMBS

RANGE	RESOLUTION	ACCURACY (1 Yr.)* 18°-28°C ±(%rdg+counts)	TEMPERATURE COEFFICIENT 0°-18°C & 28°-50°C ±(%rdg+counts)/°C
200pC	10 fC	0.4+4	0.02+3
2nC	100 fC	0.4+1	0.02+0.3
20nC	1pC	0.4+1	0.02+0.1

*When properly zeroed.

INPUT BIAS CURRENT: Less than 5fA (5×10⁻¹⁵A) at 23°C.

OHMS

RANGE	RESOLUTION	ACCURACY (1 Yr.)* 18°-28°C ±(%rdg+counts)	TEMPERATURE COEFFICIENT 0°-18°C & 28°-50°C ±(%rdg+counts)/°C	TEST CURRENT ±1.5%
2 kΩ	100 mΩ	0.20+4	0.01+3	100μA
20 kΩ	1 Ω	0.15+1	0.01+0.3	100μA
200 kΩ	10 Ω	0.25+1	0.01+0.3	10μA
2MΩ	100 Ω	0.25+1	0.02+0.3	1μA
20MΩ	1 kΩ	0.25+1	0.02+0.3	100nA
200MΩ	10 kΩ	0.30+1	0.02+0.3	10nA
2 GΩ	100 kΩ	1.5 +1	0.04+0.3	1nA
20 GΩ	1MΩ	1.5 +1	0.04+0.1	1nA
200 GΩ	10MΩ	1.5 +1	0.04+0.1	1nA

*When properly zeroed.

MAXIMUM OPEN CIRCUIT VOLTAGE: 300V dc.

PREAMP SETTLING TIME (To 0.1% of final value, unguarded, with less than 100pF input capacitance): 2kΩ through 20MΩ, 15ms; 200MΩ, 150ms. (To 1% of final value with Input Guard on and less than 1pF of unguarded input capacitance): 2GΩ, 10ms, 20GΩ, 100ms; 200GΩ, 1s.

V/I MODE: Used with V-source, displays resistance (5×10⁵ to 10¹⁵Ω) calculated from measured current. V/I Ohms accuracy equal to accuracy of V-Source plus accuracy of selected Amps range.

VOLTAGE SOURCE

OUTPUT: −102V to +102V in 50mV steps.

ACCURACY (1 Yr., 18°-28°C): ±(0.2% + 50mV).

TEMPERATURE COEFFICIENT: ±(0.005% + 1mV)/°C.

MAXIMUM OUTPUT CURRENT: ±2mA; active current limit at less than 4mA with annunciation.

SETTLING TIME: Less than 3ms to rated accuracy.

NOISE: < (1ppm of output voltage + 200μV) p-p from 0.1Hz to 10Hz.

IEEE BUS IMPLEMENTATION

MULTILINE COMMANDS: DCL, LLO, SDC, GET, GTL, UNT, UNL, SPE, SPD

UNILINE COMMANDS: IFC, REN, EOI, SRQ, ATN.

INTERFACE FUNCTIONS: SH1, AH1, T5, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT1, C0, E1.

PROGRAMMABLE PARAMETERS: Function, Range, Zero Check, Zero Correct, Zero Suppress, EOI, Trigger, Terminator, 100-rdg Store and Retrieval, Calibration, V-Source Output, Display Format, SRQ, Status (including V-Source I-Limit), Output Format.

ADDRESS MODES: TALK ONLY and ADDRESSABLE.

TRIGGER TO READING DONE: 350ms typical.

GENERAL

DISPLAY: 4½-digit numeric LEDs with appropriate decimal point and polarity indication: signed two-digit alphanumeric exponent.

OVERRANGE INDICATION: Display reads "OL".

CONVERSION TIME: 330ms. RANGING: Automatic or manual.

DATA STORE and MIN/MAX: 100-reading store capacity; records data at one of six selectable rates from every reading to 1 reading/hour, or by manual triggering. Also detects and stores maximum and minimum readings continuously while in the Data Store mode.

PROGRAMS: Provide front panel access to IEEE address, choice of engineering units or scientific notation, and digital calibration.

MAXIMUM INPUT: 250V peak, dc to 60Hz sine wave; 10s per minute max on mA ranges.

MAXIMUM COMMON-MODE VOLTAGE (dc to 60Hz sine wave): Electrometer 500V peak; V-Source, 100V peak.

INPUT CONNECTOR: Two lug triaxial on rear panel.

OUTPUT CONNECTORS: 5-way binding posts on rear panel for V-source, preamp, and analog outputs. Rear panel BNC for External Trigger and Meter Complete.

2V ANALOG OUTPUT: 2V for full range input. Inverting in Volts and Ohms modes. Output impedance 10kΩ.

PREAMP OUTPUT: Provides a guard output for Volts and Ohms measurements. Can be used as an inverting output or with external feedback in Amps and Coulombs modes. Output Impedance: 100Ω.

EXTERNAL TRIGGER: TTL compatible External Trigger and Electrometer Complete

V, Ω GUARD SWITCH: OFF Position: Inner shield of triax is Input LO, input capacitance is less than or equal to 20pF. ON Position: Inner shield of triax is Guard (follows Input HI). Input capacitance is less than or equal to 2pF. Use Analog Output COM for Input LO connection.

ENVIRONMENT: Operating: 0°-50°C. Relative Humidity: 70% non-condensing, up to 35°C. Storage: −25° to +65°C.

SHIELDING: Double shielded.

WARMUP: 2 hours to rated accuracy.

POWER: 105-125V, or 210-250V (internal switch selected), 90-110V available: 50-60Hz, 25 VA.

DIMENSIONS, WEIGHT: 127mm high × 216mm wide × 359mm deep (5 in. × 8½ in. × 14⅛ in.). Net weight 3.6kg (8 lbs.).

ACCESSORY SUPPLIED: Model 6011 Triaxial Input Cable.

ACCESSORIES AVALIABLE:

Model 1019A: Universal Fixed Rack Mounting Kit
Model 1019S: Universal Slide Rack Mounting Kit
Model 6011: Triaxial Input Cable (3 ft.)
Model 6011-10: Triaxial Input Cable (10 ft.)
Model 6103C: Voltage Divider Probe (1000:1)
Model 6012: Triaxial-to-Coaxial UHF Adapter
Model 6104: Test Shield
Model 6105: Resistivity Chamber
Model 6146: Triaxial Tee Adapter
Model 6147: Triaxial-to-BNC Adapter
Model 6171: 3 Lug Male-to-2 Lug Female Triaxial Adapter
Model 6172: 2 Lug Male-to-3 Lug Female Triaxial Adapter
Model 7008-3: IEEE-488 Digital Cable (3 ft.)
Model 7008-6: IEEE-488 Digital Cable (6 ft.)
Model 7023: Female Triaxial Connector
Model 7024-3: Triaxial Cable (3 ft.)
Model 7024-10: Triaxial Cable (10 ft.)
Model 8573: IEEE-488 Interface to IBM PC

Specifications subject to change without notice.

ประวัติผู้เขียน

นาย สุคคเณศ ตุงคะสมิต เกิดเมื่อวันที่ 14 มกราคม พ.ศ. 2515 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (ฟิสิกส์) จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2536 และได้เข้าศึกษาต่อในระดับปริญญามหาบัณฑิต สาขาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในระหว่างการศึกษา ได้รับทุนผู้ช่วยสอนจากฝ่ายวิชาการ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2536 - 2537 และได้รับเลือกให้นำผลงานออกแสดงในงานการประชุมวิชาการประจำปี 2536 และ 2538 ซึ่งจัดโดย คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปัจจุบัน รับราชการในตำแหน่ง อาจารย์ประจำภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย