ภาคผนวก

ภาคผนวก ก.

ตารางแสดงค่าสัมประสิทธิ์การคำนวณหาความเข้มรังสีอาทิตย์

ตารางที่ ก-1 ค่าความเข้มรังสีอาทิตย์และค่าสัมประสิทธิ์ของชั้นบรรยากาศ [11]

เดือน	**A** (Btu/hr-ft^2)	**B**	**C**
มกราคม	390	0.142	0.058
กุมภาพันธ์	385	0.144	0.060
มีนาคม	376	0.156	0.071
เมษายน	360	0.180	0.097
พฤษภาคม	350	0.196	0.121
มิถุนายน	345	0.205	0.134
กรกฎาคม	344	0.207	0.136
สิงหาคม	351	0.201	0.122
กันยายน	365	0.177	0.092
ตุลาคม	378	0.160	0.073
พฤศจิกายน	387	0.149	0.063
ธันวาคม	391	0.142	0.057

ตารางที่ ก-2 แสดงค่าสัมประสิทธิ์ของสมการอนุกรมกำลัง [4]

i	a_i	b_i
0	1.1049	0.9864
1	-1.4354	-0.2001
2	-1.0720	-1.1883
3	6.6849	3.3705
4	-13.8990	-5.7674
5	13.0798	3.7206
6	-4.4631	-0.9217

ตารางที่ ก-3 ค่าสัมประสิทธิ์ของจังหวัดเชียงใหม่ [5]

วันที่	a_0	b_0
14 ม.ค. - 26 ก.พ.	0.321	0.678
27 ก.พ. - 12 เม.ย.	0.326	0.673
13 เม.ย - 28 พ.ค.	0.400	0.599
29 พ.ค. - 15 ก.ค.	0.395	0.604
16 ก.ค. - 31 ส.ค.	0.393	0.606
1 ก.ย. - 15 ต.ค.	0.345	0.654
16 ต.ค. - 29 พ.ย.	0.361	0.638
30 พ.ย. - 13 ม.ค.	0.366	0.633

ตารางที่ ก-4 ค่าสัมประสิทธิ์ของกรุงเทพมหานคร [5]

วันที่	$\mathbf{a}_0$	$\mathbf{b}_0$
14 ม.ค. - 26 ก.พ.	0.368	0.631
27 ก.พ. - 12 เม.ย.	0.324	0.675
13 เม.ย - 28 พ.ค.	0.352	0.647
29 พ.ค. - 15 ก.ค.	0.404	0.595
16 ก.ค. - 31 ส.ค.	0.410	0.589
1 ก.ย. - 15 ต.ค.	0.363	0.636
16 ต.ค. - 29 พ.ย.	0.382	0.617
30 พ.ย. - 13 ม.ค.	0.382	0.617

ภาคผนวก ข.

โปรแกรมการวิเคราะห์ขนาดระบบโฟโตโวลตาอิก

โปรแกรมที่ 1 คำนวณหาค่าเฉลี่ยชั่วโมงความยาวนานแสงแดดตลอดปี

```
%Program Calculated mean sunshine hours per Day.
%Name mean_sun.m
                                   load sunshine1.dat;%size(5,44)
                                   load sunshine2.dat;%size(5,46)
                                   load sunshine3.dat;%size(5,46)
                                   load sunshine4.dat;%size(5,48)
                                   load sunshine5.dat;%size(5,47)
                                   load sunshine6.dat;%size(5,45)
                                   load sunshine7.dat;%size(5,45)
                                   load sunshine8.dat;%size(5,45)
format bank;
%data_day = input('Please select type of day either during 1 or 2   =   ');
data_day = 1;
if     data_day == 1;
 sunshine1 = sunshine1 ;
 sunshine2 = sunshine2 ;
 sunshine3 = sunshine3 ;
 sunshine4 = sunshine4 ;
 sunshine5 = sunshine5 ;
 sunshine6 = sunshine6 ;
 sunshine7 = sunshine7 ;
 sunshine8 = sunshine8 ;
else        data_day == 2;
  data1 = zeros(10,44); % size(10,44)
  data1([1:5],[1:2:43]) = sunshine1(:,[1:2:43]); % sunshine1 Row  1-5
  data1([6:10],[1:2:43]) = sunshine1(:,[2:2:44]);% sunshine1 Row  6-10
  data1(:,[2:2:44]) = data1(:,[1:2:43]);% Double Column
  sunshine1 = data1; % data sunshine1 size (10,44)
  data2 = zeros(10,46); % size(10,46)
  data2([1:5],[1:2:45]) = sunshine2(:,[1:2:45]);
  data2([6:10],[1:2:45]) = sunshine2(:,[2:2:46]);
  data2(:,[2:2:46]) = data2(:,[1:2:45]);
  sunshine2 = data2; % data sunshine2 size (10,46)
```

```
    data3 = zeros(10,46); % size(10,46)
    data3([1:5],[1:2:45]) = sunshine3(:,[1:2:45]);
    data3([6:10],[1:2:45]) = sunshine3(:,[2:2:46]);
    data3(:,[2:2:46]) = data3(:,[1:2:45]);
    sunshine3 = data3; % data sunshine3 size (10,46)
    data4 = zeros(10,48); % size(10,48)
    data4([1:5],[1:2:47]) = sunshine4(:,[1:2:47]);
    data4([6:10],[1:2:47]) = sunshine4(:,[2:2:48]);
    data4(:,[2:2:48]) = data4(:,[1:2:47]);
    sunshine4 = data4; % data sunshine4 size (10,48)
    data5 = zeros(10,47); % size(10,47)
    data5([1:5],[1:2:45]) = sunshine5(:,[1:2:45]);
    data5([6:10],[1:2:45]) = sunshine5(:,[2:2:46]);
    data5(:,[2:2:46]) = data5(:,[1:2:45]);
    data5([1:5],47) = sunshine5([1:5],47);
    data5([6:10],47) = sunshine5([1:5],47);
    sunshine5 = data5; % data sunshine5 size (10,47)
    data6 = zeros(10,45); % size(10,45)
    data6([1:5],[1:2:43]) = sunshine6(:,[1:2:43]);
    data6([6:10],[1:2:43]) = sunshine6(:,[2:2:44]);
    data6(:,[2:2:44]) = data6(:,[1:2:43]);
    data6([1:5],45) = sunshine6([1:5],45);
    data6([6:10],45) = sunshine6([1:5],45);
    sunshine6 = data6; % data sunshine6 size (10,45)
    data7 = zeros(10,45); % size(10,45)
    data7([1:5],[1:2:43]) = sunshine7(:,[1:2:43]);
    data7([6:10],[1:2:43]) = sunshine7(:,[2:2:44]);
    data7(:,[2:2:44]) = data7(:,[1:2:43]);
    data7([1:5],45) = sunshine7([1:5],45);
    data7([6:10],45) = sunshine7([1:5],45);
    sunshine7 = data7; % data sunshine7 size (10,45)
    data8 = zeros(10,45); % size(10,45)
    data8([1:5],[1:2:43]) = sunshine8(:,[1:2:43]);
    data8([6:10],[1:2:43]) = sunshine8(:,[2:2:44]);
    data8(:,[2:2:44]) = data8(:,[1:2:43]);
    data8([1:5],45) = sunshine8([1:5],45);
    data8([6:10],45) = sunshine8([1:5],45);
    sunshine8 = data8; % data sunshine8 size (10,45)
end
```

```
  % save mean daily of sunshine duration data (total 366 day in a year)
save mean1.dat mean1 -ASCII;%size(1,44)
save mean2.dat mean2 -ASCII;%size(1,46)
save mean3.dat mean3 -ASCII;%size(1,46)
save mean4.dat mean4 -ASCII;%size(1,48)
save mean5.dat mean5 -ASCII;%size(1,47)
save mean6.dat mean6 -ASCII;%size(1,45)
save mean7.dat mean7 -ASCII;%size(1,45)
save mean8.dat mean8 -ASCII;%size(1,45)
```

โปรแกรมที่ 2 คำนวณหาค่าสัมประสิทธิ์ตามละติจูดของสถานที่ติดตั้งระบบ

```
% Program Calculation Coefficient of Regression
% Name cal_coeff.m
global latitude;
load coeff_cm.dat;
load coeff_bk.dat;
latitude = input('latitude of station  =  ');
if     latitude >= 18.47;
  coefficient = coeff_cm;
elseif latitude <= 13.44;
       coefficient = coeff_bk;
else (latitude < 18.47) & (latitude > 13.44);
      slope_a0 = (coeff_bk(:,1)-coeff_cm(:,1))./-(5.03);
      coeff_a0 = coeff_bk(:,1)-(slope_a0.*(13.44-latitude)); % Coefficient a0
      slope_a1 = (coeff_bk(:,2)-coeff_cm(:,2))./-(5.03);
      coeff_a1 = coeff_bk(:,2)-(slope_a1.*(13.44-latitude)); % Coefficient a1
      coefficient = zeros(8,2);
      coefficient(:,1) = coeff_a0(:,:);
      coefficient(:,2) = coeff_a1(:,:);
end
  save coefficient.dat coefficient -ASCII; % save coefficient value is ascii file
```

โปรแกรมที่ 3 คำนวณหาค่าสัมประสิทธิ์ตามละติจูดของสถานที่ติดตั้งระบบ

```
% Program Calculate solar hour angle , zenith angle
% name  zenith.m
global longitude;
global latitude;
global tilt;
longitude = input('Longitude of  local = ');
latitude = input('Latitude of  local = ');
tilt = input('Module tilt angle = ');
ts = zeros(365,12);
t = [390 450 510 570 630 690 750 810 870 930 990 1050];%time
w = zeros(365,12);
delta = zeros(1,365);
for i = 1:365;
  % Solar declination Angle
      delta(i) = (23.45)*sin(((360/365)*(284+(i)))*(pi/180));%size(1,365)
     B(i) = (360/364)*(i-81);
      EOT(i) = (9.87*sin(2*B(i)*(pi/180)))-(7.53*cos(B(i)*(pi/180)))-(1.5*sin(B(i)*(pi/180)));%unit (minute)
  for j = 1:12;
     % solar time = standard time+/-4(Lstd-Llocal)+EOT ; + for west , - for east
      x1(i,j)= -(4*(105-98.98))+EOT(i);%solar time 0-24 hour
      x2(i,j) = fix(x1(i,j)+ t(j));%round off second
      x(i,j) = fix(x2(i,j)/60);%time = hour
      ts(i,j) = x(i,j)+(((x2(i,j))-(x(i,j)*60))/100);%time = hour:minute
      if ts(i,j) < 12;
           w(i,j) = -((12-ts(i,j))*15);% solar hour angle +180 to -180 , 0 degree at solar noon
      elseif ts(i,j) > 12;
      w(i,j) = (ts(i,j)-12)*15;
      else  ts(i,j) == 12;
      w(i,j) = 0;
      end
  end
end
cos_z = zeros(365,12);
z = zeros(365,12);% zenith angle
zeta = zeros(365,12);%angle between beam tilt and beam normal
nn = zeros(365,12);%variable
for i = 1:365;
  for j = 1:12;
```

```
        cos_z(i,j) = (sin(latitude*(pi/180))*sin((delta(i))*(pi/180)))+...
       (cos(latitude*(pi/180))*cos((delta(i))*(pi/180))*cos((w(i,j))*(pi/180)));
       z(i,j) = (acos(cos_z(i,j)))*(180/pi);%size(365,12)
     if z(i,j) > 90;
       z(i,j) = 90;
     else
       z(i,j)=z(i,j);
     end
   % find incident angle (zeta)
   nn(i,j) = cos((latitude - tilt)*(pi/180))*cos((delta(i))*(pi/180))*cos((w(i,j))*(pi/180))+...
          (sin((latitude - tilt)*(pi/180))*sin((delta(i))*(pi/180)));
   zeta(i,j) = (acos(nn(i,j)))*180/pi;%size(365,12)
   end
end
% save data
save z.dat z -ASCII;%zenith angle size(365,12)      [degree] (zenith + alfa = 90 degree)
save zeta.dat zeta -ASCII;%beam between normal to incident angle size(365,12) [degree]
save w.dat w -ASCII;%solar hour angle size(365,12) [degree]
```

โปรแกรมที่ 4 คำนวณหาองค์ประกอบความเข้มรังสีอาทิตย์ที่ตกกระทบบนพื้นราบ

```
% Program calculate solar insolation on horizontal surface  from clear sky
% Name Exell_clear.m
load z.dat; % size(365,12)
load Exell.dat; % size(7,2)
Exell = Exell*1000;
Gc = zeros(365,12);
Ic = zeros(365,12);
Dc = zeros(365,12);
for i = 1:365;
  F(i) = 1-(0.0335*sin(((360*(i-94))/365)*(pi/180)));
  for j = 1:12;
    G(:,:) = 0;
    I(:,:) = 0;
    for k = 1:7;
      x = [0:1:6];              % z = zenith angle [degree]
      G(j,k) = (Exell(k,1))*((z(i,j)/90).^(2*x(k)));
      I(j,k) = (Exell(k,2))*((z(i,j)/90).^(2*x(k)));
    end
    Gc(i,j) = sum(G(j,:)');% Total solar insolation on horizontal surface from clear sky
    Ic(i,j) = sum(I(j,:)');% Direct solar insolation on horizontal surface from clear sky
    Gc(i,j) = Gc(i,j)*F(i);
    Ic(i,j) = Ic(i,j)*F(i);
    if z(i,j) >= 90;
      Gc(i,j) = 0;
      Ic(i,j) = 0;
    else
      Gc(i,j) = Gc(i,j);
      Ic(i,j) = Ic(i,j);
    end
      Dc(i,j) = Gc(i,j) - (Ic(i,j)*(sin((90-z(i,j))*(pi/180))));%Diffuse solar insolation on horizontal surface from clear sky
    % size Gc , Ic , Dc (365,12)
  end
end
save Gc.dat Gc -ASCII; %Total solar insolation size(365,12) watt/sq.metre
%size(365,12)
save Ic.dat Ic -ASCII; %Direct solar insolation size(365,12) watt/sq.metre
%size(365,12)
save Dc.dat Dc -ASCII; %Diffuse solar insolation size(365,12) watt/sq.metre
%size(365,12)
```

โปรแกรมที่ 5 คำนวณหาพลังงานแสงอาทิตย์ที่ตกกระทบบนแผงเซลล์แสงอาทิตย์ในสภาพท้องฟ้ามีเมฆ

```
% Program calculate solar insolation on tilt surface  from all sky
% Name klucher_allsky.m
load mean1.dat;%size(1,44)
load mean2.dat;%size(1,46)
load mean3.dat;%size(1,46)
load mean4.dat;%size(1,48)
load mean5.dat;%size(1,47)
load mean6.dat;%size(1,45)
load mean7.dat;%size(1,45)
load mean8.dat;%size(1,45)
load coefficient.dat;%size(8,2)
load Gc.dat;%size(365,12)
load Ic.dat;%size(365,12)
load Dc.dat;%size(365,12)
load z.dat;% zenith angle (365,12)
load zeta.dat;%beam between normal to incident angle(365,12)
load w.dat;% solar hour angle (365,12)
% adjust data sunshine duration
s = zeros(1,365);
s(1,[1:44])      = mean1(:,:);
s(1,[45:89])     = mean2(1,[1:45]);%FEB set is 28 day
s(1,[90:135])    = mean3(:,:);
s(1,[136:183]) = mean4(:,:);
s(1,[184:230]) = mean5(:,:);
s(1,[231:275]) = mean6(:,:);
s(1,[276:320]) = mean7(:,:);
s(1,[321:365]) = mean8(:,:);
%Sm given by calculated daily hour from sunrise to sunset(mean)
% Sm =[10.99 11.70 12.45 12.82 12.45 11.70 10.99 10.66] from Exell
Sm = zeros(1,365);
Sm(1,[1:44])     = 10.99;
Sm(1,[45:89])   = 11.70;
Sm(1,[90:135]) = 12.45;
Sm(1,[136:183])= 12.82;
Sm(1,[184:230])= 12.45;
Sm(1,[231:275])= 11.70;
Sm(1,[276:320])= 10.99;
Sm(1,[321:365])= 10.66;
Q1(1,[1:44])   = coefficient(1,1);  Q2(1,[1:44])   = coefficient(1,2);
Q1(1,[45:89])  = coefficient(2,1);     Q2(1,[45:89])  = coefficient(2,2);
Q1(1,[90:135]) = coefficient(3,1);  Q2(1,[90:135]) = coefficient(3,2);
Q1(1,[136:183])= coefficient(4,1);  Q2(1,[136:183])= coefficient(4,2);
Q1(1,[184:230])= coefficient(5,1);    Q2(1,[184:230])= coefficient(5,2);
Q1(1,[231:275])= coefficient(6,1);    Q2(1,[231:275])= coefficient(6,2);
Q1(1,[276:320])= coefficient(7,1);    Q2(1,[276:320])= coefficient(7,2);
Q1(1,[321:365])= coefficient(8,1);    Q2(1,[321:365])= coefficient(8,2);
%Calculated solar insolation on tilt surface
Gtilt = zeros(365,12);
Gd = zeros(365,12);
Id = zeros(365,12);
Dd = zeros(365,12);
rc = zeros(365,12);
r = zeros(365,12);
```

```
for i = 1:365; %
  % clearness factor estimate from duration of sunshine data
  kd(i) = Q1(i) + Q2(i)*((s(i))/Sm(i));%365
  for j = 1:12;
    if Gc(i,j) == 0;
      rc(i,j)= 0;
    else
      rc(i,j) = Dc(i,j)/Gc(i,j);%size(365,12)  Line 70
    end
    r(i,j) = 1 -(0.733*(kd(i).^2))+(0.267*(kd(i).^4))*(1 - rc(i,j));
    Gd(i,j) = kd(i)*Gc(i,j);
    Id(i,j) = (((1 - r(i,j))*Gd(i,j))/cos(z(i,j)*(pi/180)));
    Dd(i,j) = r(i,j)*Gd(i,j);
    if Gd(i,j) == 0;
      F(j)= 0;
    else
      F(j) = 1 -((Dd(i,j)/Gd(i,j)).^2);      % Line 80
    end
    a(j) = (sin((90-z(i,j))*(pi/180))*cos(zeta(i,j)*(pi/180)));
    if a(j) == 0;
      A(j) = 0;
    else
      A(j) = (Gd(i,j)-Dd(i,j))/sin((90-z(i,j))*(pi/180))*cos(zeta(i,j)*(pi/180));
    end
    b(j) = ((1+cos(tilt*(pi/180)))/2);        %Line 90
    if b(j) == 0;
      B(j) = 0;
    else
      B(j) = Dd(i,j)*((1+cos(tilt*(pi/180)))/2);
    end
    C(j) = 1 + F(j)*(((3/4)*sin((tilt/2)*(pi/180)))-((1/4)*sin(3*(tilt/2)*(pi/180))));
    E(j) = 1 + F(j)*((1/2)+(1/2)*(cos(2*zeta(i,j)*(pi/180))))*((3/4)*(sin(z(i,j)*(pi/180)))-...
      ((1/4)*(sin(3*z(i,j)*(pi/180)))));      %Line 100
    Gtilt(i,j) = (A(j)+(B(j))*C(j)*E(j)); %solar insolation on tilt surface size(365,12)
     if Gtilt(i,j) < 0;
       Gtilt(i,j) = 0;
     else
       Gtilt(i,j)=Gtilt(i,j);
     end
  end
end
% save total insolation on tilt surface (watt/sq.metre)
save Gklucher.dat Gtilt -ASCII;%size(365,12)
```

โปรแกรมที่ 6 ประมาณค่าอุณหภูมิแวดล้อมจากข้อมูลสถิติอุณหภูมิ

```
% Program Estimated Abient Temperature by Polynomial Regression Methods
% Name temp_regression.m
format bank;
load data_temp1991.dat; %size (366,26)
load data_temp1992.dat; %size (366,26)
load data_temp1993.dat; %size (366,26)
load data_temp1994.dat; %size (366,26)
load data_temp1996.dat; %size (366,26)
datatemp1 = data_temp1991; %size(366,26)
datatemp2 = data_temp1992;
datatemp3 = data_temp1993;
datatemp4 = data_temp1994;
datatemp5 = data_temp1996;
% Adjust data temperature to size(366,24) and Calculated Mean Temperature
hour = 1:24;
for  i = 1:366;
  temp(i,:) = (datatemp1(i,[2:25])+ datatemp2(i,[2:25])+ datatemp3(i,[2:25])...
    + datatemp4(i,[2:25])+ datatemp5(i,[2:25]))./5;% size(366,24)
end
% Calculate The Coeffecient of Polynomial Regression
for  i = 1:366;
  % Can changeable the number of Coeffecient
  coeff_day(i,:) = polyfit(hour,temp(i,:),10); % (366,No.coeffecient)
end
% Estimate Temperature in the future (Degree Celcius)
for i = 1:366;
  temperature(i,:) = polyval(coeff_day(i,:),hour);%size(366,24)
end
% save data temperature (366,24)
save temperature.dat temperature -ASCII;
```

โปรแกรมที่ 7 คำนวณหาพลังงานไฟฟ้าที่แผงเซลล์แสงอาทิตย์ผลิตได้

```
% Programe Calculation Energy from Module
% Name PV_output.m
load Gklucher.dat; % size (365,12)
load temperature.dat ; % size(366,24)
format bank;
global I_ref;
global V_ref;
% adjust data Gtilt (365,12) is (366,24)
Gt = zeros(366,24);
Gt([1:59],[7:18]) = Gklucher([1:59],[1:12]);
Gt(60,[7:18]) = Gklucher(59,[1:12]);% 28 Feb = 29 Feb
Gt([61:366],[7:18]) = Gklucher([60:365],[1:12]);
% Data Photovoltaic Module
disp ('INPUT SPECIFICATION DATA OF PV-MODULE');
S_ref = input('Reference Solar Radiation [watt/sq.m]  =');
T_ref = input('Reference temperature [Celcius]  =');
Rs = input('Series Resistance of Module [Ohm]  =');
I_ref = input('Rated Current of Module [Amps]  =');
Isc = input('Short circuit current of Module [Amps]  =');
V_ref = input('Rated Voltage of Module [Volts]  =');
coeff_V = input('Coefficience Voltage of Module [Volts/Celcius]  =');
coeff_I = input('Coefficience Current of Module [Amps/Celcius]  =');
% Calculate Energy from Module (Wh/day)
Tc = zeros(366,24);%set matrix cell teperature
D_T = zeros(366,24);%set matrix delta temperature
D_I = zeros(366,24);%set matrix delta module current
D_V = zeros(366,24);%set matrix delta module voltage
Vmodule = zeros(366,24);
Imodule = zeros(366,24);
Wm = zeros(366,24);%set matrix energy from PV-module
for   i = 1:366;
      for j = 7:18;
      Tc(i,j) = temperature(i,j) + ((0.02)*Gt(i,j));
      D_T(i,j) = Tc(i,j) - T_ref;
      D_I(i,j) = (coeff_V*(Gt(i,j)/S_ref)*D_T(i,j))+(((Gt(i,j)/S_ref)-1)*Isc);
      D_V(i,j) = (-(coeff_V)*D_T(i,j))-(Rs*D_I(i,j));
      Vmodule(i,j) = V_ref + D_V(i,j);
      Imodule(i,j) = I_ref + D_I(i,j);
      Wm(i,j) = Vmodule(i,j)*Imodule(i,j);
      if Wm(i,j) <= 0;
        Wm(i,j) = 0;
      end
   end
end

% Save data Energy from Module (Watt-hours) , size(366,24)
save Wm.dat Wm -ASCII;%size(366,24)
```

โปรแกรมที่ 8 จัดข้อมูลโหลดใน 1 ปี

```
% Program Recieve Data Load
% Name dataload.m
disp('INPUT DATA LOAD');
disp('If data load every week is same put data load size (7,24) ');
disp('If data load every 2 week is same put data load size (14,24)');
disp('If data load every 3 week is same put data load size (21,24)');
disp('If data load every 4 week is same put data load size (28,24)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Load data load (ASCII file)
load load_d100n0.m;
load load_d0n100.m;
load load_d20n80.m;
load load_d30n70.m;
load load_d40n60.m;
load load_d60n40.m;
load load_d50n50.m;
load load_d80n20.m;
data_load = load_d20n80;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Wl = zeros(366,24); % case is'n put the load value at this hour data load = 0
[ro,c] = size(data_load);
if ro == [7] & c == [24]; % data load every week is same
  for i = 1:52;
     d1 = [1:7:358];
     d2 = [2:7:359];
     d3 = [3:7:360];
     d4 = [4:7:361];
```

```
    d5 = [5:7:362];
    d6 = [6:7:363];
    d7 = [7:7:364];
    Wl(d1(i),:) = data_load(1,:); % 1 data_load size(1,24)
    Wl(d2(i),:) = data_load(2,:); % 2 data_load size(1,24)
    Wl(d3(i),:) = data_load(3,:); % 3 data_load size(1,24)
    Wl(d4(i),:) = data_load(4,:); % 4 data_load size(1,24)
    Wl(d5(i),:) = data_load(5,:); % 5 data_load size(1,24)
    Wl(d6(i),:) = data_load(6,:); % 6 data_load size(1,24)
    Wl(d7(i),:) = data_load(7,:); % 7 data_load size(1,24)
    Wl(365,:)  = data_load(1,:);
    Wl(366,:)  = data_load(2,:);
  end
elseif ro == [14] & c == [24] % data load every 2 week is same
  for i = 1:26;
    d1 = [1:14:351]; d8 = [8:14:358];
    d2 = [2:14:352]; d9 = [9:14:359];
    d3 = [3:14:353]; d10= [10:14:360];
    d4 = [4:14:354]; d11= [11:14:361];
    d5 = [5:14:355]; d12= [12:14:362];
    d6 = [6:14:356]; d13= [13:14:363];
    d7 = [7:14:357]; d14= [14:14:364];
    Wl(d1(i),:) = data_load(1,:);
    Wl(d2(i),:) = data_load(2,:);
    Wl(d3(i),:) = data_load(3,:);
    Wl(d4(i),:) = data_load(4,:);
    Wl(d5(i),:) = data_load(5,:);
    Wl(d6(i),:) = data_load(6,:);
    Wl(d7(i),:) = data_load(7,:);
    Wl(d8(i),:) = data_load(8,:);
    Wl(d9(i),:) = data_load(9,:);
    Wl(d10(i),:) = data_load(10,:);
    Wl(d11(i),:) = data_load(11,:);
    Wl(d12(i),:) = data_load(12,:);
    Wl(d13(i),:) = data_load(13,:);
    Wl(d14(i),:) = data_load(14,:);
    Wl(365,:)  = data_load(1,:);
    Wl(366,:)  = data_load(2,:);
  end
```

```
elseif ro == [21] & c == [24]; % data load every 3 week is same
  for i = 1:17;
    d1 = [1:21:337]; d8 = [8:14:344];      d15 = [15:21:351];
    d2 = [2:21:338]; d9 = [9:14:345];      d16 = [16:21:352];
    d3 = [3:21:339]; d10= [10:14:346];     d17 = [17:21:353];
    d4 = [4:21:340]; d11= [11:14:347]; d18 = [18:21:354];
    d5 = [5:21:341]; d12= [12:14:348];     d19 = [19:21:355];
    d6 = [6:21:342]; d13= [13:14:349]; d20 = [20:21:356];
    d7 = [7:21:343]; d14= [14:14:350];     d21 = [21:21:357];
    Wl(d1(i),:) = data_load(1,:);
    Wl(d2(i),:) = data_load(2,:);
    Wl(d3(i),:) = data_load(3,:);
    Wl(d4(i),:) = data_load(4,:);
    Wl(d5(i),:) = data_load(5,:);
    Wl(d6(i),:) = data_load(6,:);
    Wl(d7(i),:) = data_load(7,:);
    Wl(d8(i),:) = data_load(8,:);
    Wl(d9(i),:) = data_load(9,:);
    Wl(d10(i),:) = data_load(10,:);
    Wl(d11(i),:) = data_load(11,:);
    Wl(d12(i),:) = data_load(12,:);
    Wl(d13(i),:) = data_load(13,:);
    Wl(d14(i),:) = data_load(14,:);
    Wl(d15(i),:) = data_load(15,:);
    Wl(d16(i),:) = data_load(16,:);
    Wl(d17(i),:) = data_load(17,:);
    Wl(d18(i),:) = data_load(18,:);
    Wl(d19(i),:) = data_load(19,:);
    Wl(d20(i),:) = data_load(20,:);
    Wl(d21(i),:) = data_load(21,:);
    Wl([358:366],:)  = data_load([1:9],:);
   end
  elseif  ro == [28] & c == [24]; % data load every 4 week is same
   Wl([1:28],:) = data_load(:,:);
   Wl([29:56],:) = data_load(:,:);
   Wl([57:84],:) = data_load(:,:);
   Wl([85:112],:) = data_load(:,:);
   Wl([113:140],:) = data_load(:,:);
  Wl([141:168],:) = data_load(:,:);
  Wl([169:196],:) = data_load(:,:);
  Wl([197:224],:) = data_load(:,:);
  Wl([225:252],:) = data_load(:,:);
  Wl([253:280],:) = data_load(:,:);
  Wl([281:308],:) = data_load(:,:);
  Wl([309:336],:) = data_load(:,:);
  Wl([337:364],:) = data_load(:,:);
  Wl([365:366],:) = data_load([1:2],:);
 else  r~= [7] | r~=[14] | r~=[21] | r~=[28] & c ~= [24]
  disp('form data load is worst(not size(7,24)or(14,24)or(21,24)or(28,24)');
 disp('Check form data load agian');
 end
% save data load size (366,24)
save Wl.dat Wl -ASCII
```

โปรแกรมที่ 9 จำลองการทำงานของระบบและวิเคราะห์หาขนาดระบบที่เหมาะสม

```
% Program Calculate optimization
% optimize_size_t2.m
load temperature.dat;%size(366,24)
load Wl.dat;%size(366,24)
load Wm.dat;%size(366,24)
global C10;
global charge_eff;
global Vbend;
global Vbover;
global Tb;
global Emax;
global Emin;
global Nc;
global factor;% set value in program2
disp('INPUT SPECIFICATION DATA OF BATTERY');
C10       = input('Battery capacity at 10 h [Ah] =  ');
Vbend     = input('Battery end of voltage (at 1 Cell)[volts] = ');
Vbover     = input('Battery over voltage(at 1 Cell)[volts] = ');
Tb         = input('Reference temperature of Battery [Celsius] =  ');
Emax        = input('Maximum electrolyte density =  ');
Emin        = input('Minimum electrolyte density =  ');;
Nc           = input('Number cell of battery = ');
```

```
charge_eff      = input('Charger efficiency[0-1] =  ');
LLP = input('Reliability level request[0-1] =  ');
% Adjust data to 1 row
% temperature.dat size(366,24) to size(1,8784)
temp = zeros(1,8784);
for i = 1:366;
   l = [1:24:8761];
   m = [24:24:8784];
  temp(:,[l(i):m(i)]) = temperature(i,:);
end
 %Adjust Wl.dat size(366,24) to size(1,8784)
 WL = zeros(1,8784);
 Wl_daily = zeros(1,366);
for i = 1:366
   l = [1:24:8761];
   m = [24:24:8784];
   WL(:,[l(i):m(i)]) = Wl(i,:); % Adjust = size(1,8784)
   %seperate load day and night[Wh/day or night]
   Wlday(i) = sum(Wl((i),[8:16])); % day load size(1,366)
   Wlnight(i) = sum(Wl((i),[1:7]))+ sum(Wl((i),[17:24]));%night load size(1,366)
   Wl_daily(1,(i)) = sum(Wl((i),:));% total daily load  ,size(1,366)
end
 % Wm.dat size(366,24) to size(1,8784)
WM = zeros(1,8784);
for i = 1:366;
   l = [1:24:8761];
   m = [24:24:8784];
   WM(:,[l(i):m(i)]) = Wm(i,:);%size(1,8784)
 end
 for i = 1:8784;
   if WM(1,i) <= 0; % protect case WM < 0
      WM(1,i) = 0;
   else
      WM(1,i) > 0;
      WM(1,i) = WM(1,i);
   end
 end
 % Estimate energy
 global no_module;
global no_batt;
%global C10_approx;
No_module = mean(WL)/mean(WM);%estimate number of module
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_module = ceil(No_module)+ factor% IMPORTANCE IN ADJUST LLP OF SYSTEMS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WM_approx = WM .*[no_module];%size(1,8784) Energy module use simulation
Pre_module = no_module*V_ref*I_ref  % monitor size PV-module before optimize
disp('[Watt]');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
No_batt = (mean(Winight))/(C10*12);%estimate number of battery
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_batt = ceil(No_batt);%Number Integer(capacity[Wh]=C10*no_batt*12)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C10_approx = (C10*no_batt);%Approximate battery capacity for simulation
Pre_battery_size = C10_approx % monitor battery size before optimize
disp('[Ah]');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ibch = zeros(1,8784);%for set vector protect Ibch start point value  = 0
Wloss = zeros(1,8784);%loss
Wd = zeros(1,8784);%deficit
Wb = zeros(1,8784);
Vbt = zeros(1,8784);
SOC = zeros(1,8784);
Wcharge_batt = zeros(1,8784);
% START
for i = 1:8784;
  I10 = C10_approx/10; % battery current at 1 hour
  if WM_approx(i) > WL(i); % CASE(1) Energy Module over than Load
    W(i) = WM_approx(i) - WL(i);
    Ibch(i) = W(i)/12; % Battery current
    Im(i) = WM_approx(i)/14; % Module Current
    Iload(i) = WL(i)/12; % Load Current
    % set initial FC
    if i == 1;
      FC(1) = 1; % initial state Cef = C10 so set FC value = 1
      Cef(1)= C10_approx;
      SOC(1)= 1;
    % Check Battery Charge voltage
```

```
                E15(1) = Emax; % SOC(1)=1
                Vo(1) = E15(1) + 0.856;
                Vcell(1) = Vo(1)+((Ibch(1)/C10_approx)*0.451);
                Vbt(1) = Vcell(1)*Nc; % Total battery terminal voltage
    else
                FC(i) = (1.79/(1+0.77*((Ibch(i)/I10)^0.726)))*(1+(0.0118*(temp(i)-Tb))-0.00031*((temp(i)-Tb)^2));
% Protect CASE Value = 0
      if FC(i) > 1;
        FC(i) = 1;
      end
      if FC(i) == 0;
        Cef(i) = 0.001;
      else
        Cef(i) = Cef(i-1)*FC(i);
      end
      SOC(i) = SOC(i-1)+charge_eff*((Im(i)-Iload(i))/Cef(i));
      if SOC(i) > 1;
        SOC(i) = 1;
      end
      if SOC(i) == 0;
        E15(i) = Emin;
      else
        E15(i) = (Emax-Emin)*SOC(i)+Emin;
      end
 % end
     % Check Battery Charge voltage
                Vo(i) = E15(i) + 0.856;
                Vcell(i) = Vo(i) +(Ibch(i)/C10_approx)*((0.74/(1.05-SOC(i))+0.451+(0.219*((FC(i)^5.74)-1)));
                Vbt(i) = Vcell(i)*Nc; %  battery terminal voltage
    end
    %  Check Battery state of charge
    if (Vcell(i) >= Vbend) & (Vcell(i) <= Vbover); % Charge battery
      if i == 1;
        Wloss(1) = 0;%
        Wd(1) = 0;
        Wb(1) = C10_approx.*12;
           Wcharge_batt(i) = 0;
      elseif i > 1;
        Wloss(i) = 0; % Energy can storage in battery
```

```
           Wd(i) = 0;
           Wb(i) = Cef(i)*Vbt(i);
           Wcharge_batt(i) = W(i)*charge_eff;%Energy input to battery
        end
    elseif Vcell(i) > Vbover; % Energy loss CASE Over charge
           FC(i) = 1;
           Cef(i) = C10_approx;
           SOC(i) = 1;
           E15(i) = Emax;
           Vo(i) = E15(i)+0.856;%E15 = Emax
           Vcell(i) = Vo(i)+0.451;% Set battery capacity is full
           Vbt(i) = Vcell(i)*Nc;
           Wloss(i) = W(i);% because  Vcell > Vbover ( Overcharge)Energy loss
           Wd(i) = 0;
           Wb(i) = C10_approx*Vbt(i); % battery full
           Wcharge_batt(i) = 0;
     end
  elseif WM_approx(i) == WL(i); % CASE (2) Energy Module balance Load
        Wcharge_batt(i) = 0;
        Wloss(i) = 0;
        Wd(i) = 0;
        Wb(i)  = WCb(i-1);
        FC(i)  = FC(i-1);
        Cef(i) = Cef(i-1);
        SOC(i) = SOC(i-1);
        E15(i) = E15(i-1);
        Vo(i) = Vo(i-1);
        Vcell(i) = Vcell(i-1);
        Vbt(i) = Vbt(i-1);
  else  WM_approx(i) < WL(i); % CASE (3) Module + Battery supply Load
     W(i) = WL(i) - WM_approx(i);
     Ibdis(i) = W(i)/12; % Battery discharge current or load current [Amps]
     Wcharge_batt(i) = 0;
% set initial value
     if i == 1;
        FC(1) = 1;
        Cef(1)= C10_approx;
        SOC(1)= 0;
         E15(1) = Emax; % because SOC(1)
    Vo(1) = E15(1) + 0.856;
    Vcell(1) = Vo(1)-((Ibdis(1)/C10_approx)*0.451);
     Vbt(1) = Vcell(1)*Nc;
   else % i > 1
               FC(i) = (1.79/(1+0.77*((Ibdis(i)/I10)^0.726)))*(1+(0.0118*(temp(i) - Tb))-0.00031*((temp(i) - Tb)^2));
% Protect CASE Value = 0
     if FC(i) > 1;
       FC(i) = 1;
     end
     Cef(i) = Cef(i-1)*FC(i);
     SOC(i) = SOC(i-1)+(-(Ibdis(i))/Cef(i));
     if SOC(i) > 1;
       SOC(i) = SOC(i-1);
     elseif SOC(i) == 0;
       E15(i) = Emin;
     else
       E15(i) = (Emax-Emin)*SOC(i)+ Emin;
     end
 % end
           Vo(i) = E15(i)+0.856;
           Vcell(i) = Vo(i)-(Ibdis(i)/C10_approx)*((0.113/SOC(i))+0.451+(0.219*(FC(i)^5.74)-1));
           Vbt(i) = Vcell(i)*Nc;


    end
    if          Vcell(i) >= Vbend; % CASE battery can supply load
                Wloss(i) = 0;
                Wd(i) = 0;
                Wb(i) = Vbt(i)*Cef(i);
    elseif Vcell(i) < Vbend; % CASE battery can't supply load
         Wloss(i) = 0;
         Wd(i) = W(i);% Energy deficit
         Wb(i) = Vbt(i)*Cef(i);
    end
  end
 end
 %@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
 @@@@@@@@@@
 up1 = sum(Wd);
 up2 = sum(WL);
 Pre_reliabilty = up1/up2 %monitor reliability systems before optimize
 % Adjust Wm and Wb
format bank;
for i = 1:366;
   l = [1:24:8761];
   m = [24:24:8784];
   a = [6:24:8766];%time 06.00-18.00
   b = [18:24:8778];%time 06.00-18.00
   c = [1:24:8761];%time 01.00-05.00
   d = [5:24:8765];%time 01.00-05.00
   e = [19:24:8779];%time 19.00-24.00
   f = [24:24:8784];%time 19.00-24.00

   Wm_daily(i) = sum(WM_approx(:,[l(i):m(i)]));%size(1,366)** total daily energy produce
   Wb_daily(i) = (sum(Wb(:,[l(i):m(i)])))./24;%size(1,366)** daily energy storage in battery
  Wbday(i) = sum(Wb(:,[a(i):b(i)])); %battery energy noon size(1,366)
   Wbnight(i) = sum(Wb(:,[c(i):d(i)]))+ sum(Wb(:,[e(i):f(i)]));%battery energy night size(1,366)
  end
  % Energy of PV-Module
  EPV1 = sum(WM_approx(1,[313:1368]));%size(1,1056)
  EPV2 = sum(WM_approx(1,[1369:2472]));%size(1,1104)
  EPV3 = sum(WM_approx(1,[2473:3576]));%size(1,1104)
  EPV4 = sum(WM_approx(1,[3577:4728]));%size(1,1152)
  EPV5 = sum(WM_approx(1,[4729:5856]));%size(1,1128)
  EPV6 = sum(WM_approx(1,[5857:6936]));%size(1,1080)
  EPV7 = sum(WM_approx(1,[6937:8016]));%size(1,1080)
  EPV([1:312]) = WM_approx(1,[1:312]);%size(1,1080)
  EPV([313:1080]) = WM_approx(1,[8017:8784]);
  EPV8 = sum(EPV);
  % Demand Load
  EL1 = sum(WL(1,[313:1368]));%size(1,1056)
  EL2 = sum(WL(1,[1369:2472]));%size(1,1104)
  EL3 = sum(WL(1,[2473:3576]));%size(1,1104)
  EL4 = sum(WL(1,[3577:4728]));%size(1,1152)
  EL5 = sum(WL(1,[4729:5856]));%size(1,1128)
  EL6 = sum(WL(1,[5857:6936]));%size(1,1080)
  EL7 = sum(WL(1,[6937:8016]));%size(1,1080)
  EL([1:312]) = WL(1,[1:312]);%size(1,1080)
EL([313:1080]) = WL(1,[8017:8784]);
EL8 = sum(EL);
EB1 = sum(Wcharge_batt([313:1368]));
EB2 = sum(Wcharge_batt([1369:2472]));
EB3 = sum(Wcharge_batt([2473:3576]));
EB4 = sum(Wcharge_batt([3577:4728]));
EB5 = sum(Wcharge_batt([4729:5856]));
EB6 = sum(Wcharge_batt([5857:6936]));
EB7 = sum(Wcharge_batt([6937:8016]));
EB([1:312]) = Wcharge_batt([1:312]);
EB([313:1080]) = Wcharge_batt([8017:8784]);
EB8 = sum(EB);
% Energy loss
Eloss1 = sum(Wloss([14:57]));
Eloss2 = sum(Wloss([58:103]));
Eloss3 = sum(Wloss([104:149]));
Eloss4 = sum(Wloss([150:197]));
Eloss5 = sum(Wloss([198:244]));
Eloss6 = sum(Wloss([245:289]));
Eloss7 = sum(Wloss([290:334]));
Eloss([1:32]) = Wloss([335:366]);
Eloss([33:45]) = Wloss([1:13]);
Eloss8 = sum(Eloss);
%##########################################################
global Zmin;
Aeq  = [(EPV1-EB1) (EB1);
        (EPV2-EB2) (EB2);
        (EPV3-EB3) (EB3);
        (EPV4-EB4) (EB4);
        (EPV5-EB5) (EB5);
        (EPV6-EB6) (EB6);
        (EPV7-EB7) (EB7);
       (EPV8-EB8) (EB8)];
  beq = [EL1*(1-LLP);
          EL2*(1-LLP);
          EL3*(1-LLP);
          EL4*(1-LLP);
          EL5*(1-LLP);
          EL6*(1-LLP);
        EL7*(1-LLP);
        EL8*(1-LLP)];
LB = [1 ; 0.1];
UP = [100;100];
f = [(no_module*V_ref*I_ref); ((C10_approx)*12)];
Zmin = linprog(f,[],[],Aeq,beq,LB,UP)
```

โปรแกรมที่ 10 ตรวจสอบความน่าเชื่อถือในการจ่ายกำลังไฟฟ้าของระบบ

```
% Program check answer
% test_klucher2.m
load temperature.dat;
load Wl.dat;
load Wm.dat;
% Adjust data to 1 row 8784 column
% temperature.dat size(366,24) to size(1,8784)
temp = zeros(1,8784);
for i = 1:366;
   l = [1:24:8761];
   m = [24:24:8784];
  temp(:,[l(i):m(i)]) = temperature(i,:);
end
 %Adjust Wl.dat size(366,24) to size(1,8784)
 WL = zeros(1,8784);
 %Wl_daily = zeros(1,366);
for i = 1:366
   l = [1:24:8761];
   m = [24:24:8784];
   WL(:,[l(i):m(i)]) = Wl(i,:); % Adjust = size(1,8784)
end
% Wm.dat size(366,24) to size(1,8784)
WM = zeros(1,8784);
for i = 1:366;
   l = [1:24:8761];
   m = [24:24:8784];
   WM(:,[l(i):m(i)]) = Wm(i,:);%size(1,8784)
 end
 for i = 1:8784;
   if WM(1,i) <= 0; % protect case WM < 0
     WM(1,i) = 0;
  else
    WM(1,i) > 0;
    WM(1,i) = WM(1,i);
  end
end
% Actual Energy module after Optimize
WM_approx = zeros(1,8784);
%++++++++++++++++++++++++++++++++++++++++++
WM_approx = WM .* (no_module*Zmin(1));
%++++++++++++++++++++++++++++++++++++++++++
% Actual Battery capacity after Optimize
%-----------------------------------------
%amp_hour = input('battery capacity = ');
C1 = C10_approx*Zmin(2);
%-----------------------------------------
Ibch = zeros(1,8784);
Iload = zeros(1,8784);
Ibdis = zeros(1,8784);
Im = zeros(1,8784);
Wloss = zeros(1,8784);
Wd = zeros(1,8784);
Wb = zeros(1,8784);
Vbt = zeros(1,8784);
Vcell = zeros(1,8784);
Vo = zeros(1,8784);
SOC = zeros(1,8784);
E15 = zeros(1,8784);
FC = zeros(1,8784);
Cef = zeros(1,8784);
% START
for i = 1:8784;
  I10 = C1/10; % battery current at 1 hour
  if WM_approx(i) > WL(i); % CASE(1) Energy Module over than Load
    W(i) = WM_approx(i) - WL(i);
    Ibch(i) = W(i)/12; % Battery current
    Im(i) = WM_approx(i)/14; % Module Current
    Iload(i) = WL(i)/12; % Load Current

    % set initial FC
    if i == 1;
      FC(1) = 1; % initial state Cef = C10 so set FC value = 1
      Cef(1)= C1;
      SOC(1)= 1;
    % Check Battery Charge voltage
                E15(1) = Emax; % SOC(1)=1
                Vo(1) = E15(1) + 0.856;
                Vcell(1) = Vo(1)+((Ibch(1)/C10_approx)*0.451);
                Vbt(1) = Vcell(1)*Nc; % Total battery terminal voltage
    else
                FC(i) = (1.79/(1+0.77*((Ibch(i)/I10)^0.726)))*(1+(0.0118*(temp(i)-Tb))-0.00031*((temp(i)-Tb)^2));
% Protect CASE Value = 0
      if FC(i) > 1;
        FC(i) = 1;
      end
      if FC(i) == 0;
        Cef(i) = 0.001;
      else
        Cef(i) = Cef(i-1)*FC(i);
      end
      SOC(i) = SOC(i-1)+charge_eff*((Im(i)-Iload(i))/Cef(i));
      if SOC(i) > 1;
        SOC(i) = 1;
      end
      if SOC(i) == 0;
        E15(i) = Emin;
      else
        E15(i) = (Emax-Emin)*SOC(i)+Emin;
      end
 % end
     % Check Battery Charge voltage
                Vo(i) = E15(i) + 0.856;
                Vcell(i) = Vo(i) +(Ibch(i)/C10_approx)*((0.74/(1.05-SOC(i)))+0.451+(0.219*((FC(i)^5.74)-1)));
                Vbt(i) = Vcell(i)*Nc; % battery terminal voltage
    end
    % Check Battery state of charge
    if (Vcell(i) >= Vbend) & (Vcell(i) <= Vbover); % Charge battery
      if i == 1;
        Wloss(1) = 0;%
      Wd(1) = 0;
      Wb(1) = C1.*12;
    elseif i > 1;
      Wloss(i) = 0; % Energy can storage in battery
      Wd(i) = 0;
      Wb(i) = Cef(i)*Vbt(i);
    end
  elseif Vcell(i) > Vbover; % Energy loss CASE Over charge
      FC(i) = 1;
      Cef(i) = C1;
      SOC(i) = 1;
      E15(i) = Emax;
      Vo(i) = E15(i)+0.856;%E15 = Emax
      Vcell(i) = Vo(i)+0.451;% Set battery capacity is full
      Vbt(i) = Vcell(i)*Nc;
      Wloss(i) = W(i);% because  Vcell > Vbover ( Overcharge)Energy loss
      Wd(i) = 0;
      Wb(i) = C1*Vbt(i); % battery full
  end
elseif WM_approx(i) == WL(i); % CASE (2) Energy Module balance Load


    Wloss(i) = 0;
    Wd(i) = 0;
    Wb(i)  = WCb(i-1);
    FC(i)  = FC(i-1);
    Cef(i) = Cef(i-1);
    SOC(i) = SOC(i-1);
    E15(i) = E15(i-1);
    Vo(i) = Vo(i-1);
    Vcell(i) = Vcell(i-1);
    Vbt(i) = Vbt(i-1);
else  WM_approx(i) < WL(i); % CASE (3) Module + Battery supply Load
  W(i) = WL(i) - WM_approx(i);
  Ibdis(i) = W(i)/12; % Battery discharge current or load current [Amps]
  % set initial value
  if i == 1;
    FC(1) = 1;
    Cef(1)= C1;
    SOC(1)= 0;
```

```
                E15(1) = Emax; % because SOC(1)
                Vo(1) = E15(1) + 0.856;
               Vcell(1) = Vo(1)-((Ibdis(1)/C1)*0.451);
              Vbt(1) = Vcell(1)*Nc;
    else % i > 1
                FC(i) = (1.79/(1+0.77*((Ibdis(i)/I10)^0.726)))*(1+(0.0118*(temp(i) - Tb))-0.00031*((temp(i) - Tb)^2));
% Protect CASE Value = 0
      if FC(i) > 1;
        FC(i) = 1;
      end
      Cef(i) = Cef(i-1)*FC(i);
      SOC(i) = SOC(i-1)+(-(Ibdis(i))/Cef(i));
      if SOC(i) > 1;
        SOC(i) = SOC(i-1);
      elseif SOC(i) == 0;
        E15(i) = Emin;
      else
        E15(i) = (Emax-Emin)*SOC(i)+ Emin;
      end
 % end
                Vo(i) = E15(i)+0.856;
           Vcell(i) = Vo(i)-(Ibdis(i)/C1)*((0.113/SOC(i))+0.451+(0.219*(FC(i)^5.74)-1));
      Vbt(i) = Vcell(i)*Nc;
     end
     if          Vcell(i) >= Vbend; % CASE battery can supply load
                Wloss(i) = 0;
                Wd(i) = 0;
                Wb(i) = Vbt(i)*Cef(i);
     elseif Vcell(i) < Vbend; % CASE battery can't supply load
         Wloss(i) = 0;
         Wd(i) = W(i);% Energy deficit
         Wb(i) = Vbt(i)*Cef(i);
     end
   end
 end
%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
% ENERGY DEFICIT
d1 = Wd(:,[313:1368]);% size(1,1056)
d2 = Wd(:,[1369:2472]);% size(1,1104)
d3 = Wd(:,[2473:3576]);% size(1,1104)
d4 = Wd(:,[3577:4728]);% size(1,1152)
d5 = Wd(:,[4729:5856]);% size(1,1128)
d6 = Wd(:,[5857:6936]);% size(1,1080)
d7 = Wd(:,[6937:8016]);% size(1,1080)
d8 = Wd(:,[1:312]);
d8(:,[313:1080])=Wd(:,[8017:8784]);% size(1,1080)
W1 = WL(:,[313:1368]);% size(1,1056)
W2 = WL(:,[1369:2472]);% size(1,1104)
W3 = WL(:,[2473:3576]);% size(1,1104)
W4 = WL(:,[3577:4728]);% size(1,1152)
W5 = WL(:,[4729:5856]);% size(1,1128)
W6 = WL(:,[5857:6936]);% size(1,1080)
W7 = WL(:,[6937:8016]);% size(1,1080)
W8 = WL(:,[1:312]);
W8(:,[313:1080])=WL(:,[8017:8784]);% size(1,1080)

%  MONITOR
    Number_module = no_module*Zmin(1); %no_modult may be fraction ut to factor value
    %disp ('Panel')
  Module_power = (Number_module *V_ref *I_ref)
  disp('[Watt]')%;
  Battery_capacity = [C1]
  disp('[Ah]');
    disp('Reliability system in each duration');
    Reliability_1 = (sum(d1))/(sum(W1))
    Reliability_2 = (sum(d2))/(sum(W2))
    Reliability_3 = (sum(d3))/(sum(W3))
    Reliability_4 = (sum(d4))/(sum(W4))
    Reliability_5 = (sum(d5))/(sum(W5))
    Reliability_6 = (sum(d6))/(sum(W6))
    Reliability_7 = (sum(d7))/(sum(W7))
    Reliability_8 = (sum(d8))/(sum(W8))
   disp('Total Reliability system ');
  Reliability_system = (sum(Wd))/(sum(WL))
```

ภาคผนวก ค.

โปรแกรมควบคุมเครื่องวัดและโหลด

โปรแกรมควบคุมเครื่องวัด

```
VERSION 5.00
Begin VB.Form Form1
  Caption       =  "Form1"
  ClientHeight   =  3195
  ClientLeft     =  60
  ClientTop      =  345
  ClientWidth    =  4680
  LinkTopic      =  "Form1"
  ScaleHeight    =  3195
  ScaleWidth     =  4680
  StartUpPosition =  3  'Windows Default
  Begin VB.CommandButton Command2
    Caption       =  "END"
    Height        =  735
    Left          =  1200
    TabIndex      =  1
    Top           =  2040
    Width         =  2415
  End
  Begin VB.CommandButton Command1
    Caption       =  "RUN Measurement"
    Height        =  855
    Left          =  1200
    TabIndex      =  0
    Top           =  840
    Width         =  2415
  End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Command1_Click()
'%%%%%%%%%%%%%%% DECLARE VARIABLE %%%%%%%%%%%%%%%%%%%
'Data A/D #1 (Volt , Amp)
Dim DATA_A As Single 'A/D Digit 1
Dim DATA_B As Single 'A/D Digit 2
Dim DATA_C As Single 'A/D Digit 3
'Data A/D #2 (Ambient temperature)
Dim DATA_T1 As Single 'A/D Digit 1
Dim DATA_T2 As Single 'A/D Digit 2
Dim DATA_T3 As Single 'A/D Digit 3
Dim VA As Integer 'Defined filename of Volt-Amp
Dim temp As Integer 'Defined filename of Ambient temperature
Dim w4 As Single
Dim i As Long
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%% SET DEFAULT VALUE %%%%%%%%%%%%%%%%%%%%
'date_start = Day(Date)
'date_end = 30
t = 2  ' start time
hh = 1
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%%%%%%% START PROGRAM %%%%%%%%%%%%%%%%%%%
'Do While date_start <= date_end
  loop_hour = 0
  Do While loop_hour <= 23   ' *** START LOOP 24 HOURS ***
    If (t >= hh And t <= (hh + 0.02)) Then
' Set In-Out port
Out &H303, &H9B 'in port (Volt - Amp)
Out &H307, &H9B 'in port (Ambient temperature)
Out &H30B, &H80 'out port
    Print " Start loop hour "
    m = 0
  Do While m <= 7 ' In one hour measure 8 times for one point
    '%%%%%%%%%%%  Start measurment #1 %%%%%%%%%%%%%%%
    Out &H308, &H7 'Set black wire Enable
    'Wait 5 Sec
    w5sec = 1
```

```
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
 'Data input #1
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, " START " ' Start VA.TXT
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%%%%% END #1 %%%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%  Start measurment #2 %%%%%%%%%%%%%%%%
Out &H308, &H6 'Set brown wire Enable
'Wait 5 Sec
w5sec = 1
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
' Data input #2
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%%%%% END #2 %%%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%  Start measurment #3 %%%%%%%%%%%%%%%%
Out &H308, &H5 'Set red wire Enable
'Wait 5 Sec
w5sec = 1
```

```
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
' Data input #3
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%% END #3 %%%%%%%%%%%%%%%%%%%
'%%%%%%%%%  Start measurment #4 %%%%%%%%%%%%%
Out &H308, &H4 'Set orange wire Enable
'Wait 5 Sec
w5sec = 1
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
' Data input #4
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%% END #4 %%%%%%%%%%%%%%%%%%%
'%%%%%%%%%  Start measurment #5 %%%%%%%%%%%%%
Out &H308, &H3 'Set yellow wire Enable
'Wait 5 Sec
w5sec = 1
Do While w5sec < 3000000
```

```
w5sec = w5sec + 1
Loop
' Data input #5
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%%%%% END #5 %%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%  Start measurment #6 %%%%%%%%%%%%%%%
Out &H308, &H2 'Set green wire Enable
'Wait 5 Sec
w5sec = 1
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
' Data input #6
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%%%%% END #6 %%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%  Start measurment #7 %%%%%%%%%%%%%%%
Out &H308, &H1 'Set blue wire Enable
'Wait 5 Sec
w5sec = 1
Do While w5sec < 3000000
w5sec = w5sec + 1
```

```
Loop
' Data input #7
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Close #VA
' %%%%%%%%%%%%%%%% END #7 %%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%  Start measurment #8 %%%%%%%%%%%%%%
Out &H308, &H0 'Set white wire Enable
'Wait 5 Sec.
w5sec = 1
Do While w5sec < 3000000
w5sec = w5sec + 1
Loop
' Data input #8
DATA_A = Inp(&H300)
DATA_B = Inp(&H301)
DATA_C = Inp(&H302)
' Save data VA
VA = FreeFile
Open "C:\data\VA.TXT" For Append As #VA
Print #VA, DATA_A
Print #VA, DATA_B
Print #VA, DATA_C
Print #VA, " END " ' finish VA.TXT
Close #VA
' %%%%%%%%%%%%%%%% END #8 %%%%%%%%%%%%%%%%%%%%
'@@@@@@@@@@@@ Read Ambient temperature @@@@@@@@@@
' Data input #8
DATA_T1 = Inp(&H304)
DATA_T2 = Inp(&H305)
DATA_T3 = Inp(&H306)
' Save data VA
```

```
    temp = FreeFile
    Open "C:\data\temp.TXT" For Append As #temp
    Print #temp, " START " ' Start temp.TXT
    Print #temp, DATA_T1
    Print #temp, DATA_T2
    Print #temp, DATA_T3
    Print #temp, " END " ' finish temp.TXT
    Close #temp
     '@@@@@@@@@@@@@@@@@@@@ END temp@@@@@@@@@@@@@@@@@@@@@
  '$$$$$$$$$$$$$ Wait 5 Minute $$$$$$$$$$$$$
    I = 0
    Do While I <= 47          ' 48 x 5000000 = 5 minute
      For i& = 1 To 5000000 ' 5000000 = 5 Sec
        b = 1 + 1
      Next i&
    I = I + 1
    Loop
  '$$$$$$$$$$$ End wait 5 minute $$$$$$$$$$
    m = m + 1
  Loop ' End loop 8 times
      Print loop_hour
      loop_hour = loop_hour + 1
    Else
      hh = Hour(Time)
      mm = Minute(Time)
      t = hh + (mm / 100)
    End If
  Loop 'END LOOP 24 HOURS
  'date_start = Day(Date)
'Loop
    Print "Finish program"
End Sub
Private Sub Command2_Click()
      End
End Sub
```

โปรแกรมควบคุมโหลดจำลองระบบโทรศัพท์เคลื่อนที่สาธารณะ 470 เมกะเฮิร์ทซ

```
Function form_0() As String
Dim con0 As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'0 time
'*******************************************************
    Out &H288, &HFF ' disconnect
'%%%%%%%%%%%%%%% Wait 1 hour %%%%%%%%%%%%%%%%%%
con0 = 0
Do While con0 <= 580  ' 6 sec x 580 loop = 58 Min
    For x = 1 To 5000000
        A = 1
    Next x
  con0 = con0 + 1
Loop
 ' So that this routine is time leftover 0.5 Min
 '            3 x 19 =  ~ 58 Min
 '*******************************************************
End Function
Function form_1() As String
Dim form_1 As Long
Dim i1 As Long
Dim con1 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'1 time
'*******************************************************
    Out &H288, &HFF ' disconnect
  '%%%%% set delay time after disconnect 27 Min
    i1 = 0
    Do While i1 <= 269  ' 6 Sec x 270 loop = 1620 Sec(27 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i1 = i1 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con1 = 0
    Do While con1 <= 35  ' 12 x 3 = 36
      For x& = 1 To 5000000
        A = 1
      Next x&
    con1 = con1 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF ' disconnect
  '%%%%% set delay time after disconnect 27 Min
    i1 = 0
    Do While i1 <= 269  ' 6 Sec x 270 loop = 1620 Sec(27 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i1 = i1 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 3 Min
'                    (27 + 3 + 27 = 57 Min)
'*****************************************************
End Function
Function form_2() As String
Dim form_2 As Long
Dim i2 As Long
Dim con2 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'2 time
'*****************************************************
    Out &H288, &HFF ' disconnect
form_2 = 0
```

```
Do While form_2 <= 1 '2 loop
'%%%%% set delay time before connect 30 Sec
    i2 = 0
    Do While i2 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i2 = i2 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$$
    con2 = 0
    Do While con2 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con2 = con2 + 1
    Loop
   '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
   form_2 = form_2 + 1
 Loop 'loop form_2
    '%%%%% set delay time after disconnect 51 Min
    i2 = 0
    Do While i2 <= 509  ' 6 Sec x 510 loop = 3060 Sec(51 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i2 = i2 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
  ' So that this routine is time leftover 2 Min
  '              (0.5 x 2)+(3 x 2)+ 51 = 58 Min
  '******************************************************

 End Function
 Function form_3() As String
 Dim form_3 As Long
 Dim i3 As Long
Dim con3 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'3 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_3 = 0
Do While form_3 <= 2 ' 3 loop
'%%%%% set delay time before connect 30 Sec
    i3 = 0
    Do While i3 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i3 = i3 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$
    con3 = 0
    Do While con3 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con3 = con3 + 1
    Loop
  'SSSSSSSSSSS End connect 3 minute SSSSSSSSSS
    Out &H288, &HFF 'disconnect
  form_3 = form_3 + 1
Loop 'loop form_3
    '%%%%% set delay time after disconnect 47 Min
    i3 = 0
    Do While i3 <= 469  ' 6 Sec x 470 loop = 2820 Sec(47 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i3 = i3 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 2.5 Min
'               (0.5 x 3)+(3 x 3)+ 47 = 57.5 Min
'*****************************************************
End Function
Function form_4() As String
Dim form_4 As Long
Dim i4 As Long
Dim con4 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'4 time
'*****************************************************
    Out &H288, &HFF ' disconnect
form_4 = 0
Do While form_4 <= 3 ' 4 loop
'%%%%% set delay time before connect 30 Sec
    i4 = 0
    Do While i4 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i4 = i4 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con4 = 0
    Do While con4 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con4 = con4 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_4 = form_4 + 1
Loop 'loop form_4
    '%%%%% set delay time after disconnect 44 Min
    i4 = 0
    Do While i4 <= 439  ' 6 Sec x 440 loop = 2640 Sec( 44 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i4 = i4 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 2 Min
'                (0.5 x 4)+(3 x 4)+ 44 = 58 Min
'*******************************************************
End Function
Function form_5() As String
Dim form_5 As Long
Dim i5 As Long
Dim con5 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'5 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_5 = 0
Do While fonn_5 <= 4 ' 5 loop
'%%%%% set delay time before connect 30 Sec
    i5 = 0
    Do While i5 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i5 = i5 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
```

```
    con5 = 0
    Do While con5 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con5 = con5 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_5 = form_5 + 1
Loop 'loop form_5
    '%%%%% set delay time after disconnect 40.5 Min
    i5 = 0
    Do While i5 <= 405  ' 6 Sec x 405 loop = 2430 Sec( 40.5 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i5 = i5 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 2 Min
'               (0.5 x 5)+(3 x 5)+ 40.5 = 58 Min
'****************************************************
End Function
Function form_6() As String
Dim form_6 As Long
Dim i6 As Long
Dim con6 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'6 time
'****************************************************
    Out &H288, &HFF ' disconnect
form_6 = 0
Do While form_6 <= 5 ' 6 loop
'%%%%% set delay time before connect 30 Sec
    i6 = 0
    Do While i6 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i6 = i6 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
 '$$$$$$$$$$$$$ Connect 3 Minute SSSSSSSSSSSSS
    con6 = 0
    Do While con6 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con6 = con6 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute SSSSSSSSSS
    Out &H288, &HFF 'disconnect
  form_6 = form_6 + 1
Loop 'loop form_6
    '%%%%% set delay time after disconnect 37 Min
    i6 = 0
    Do While i6 <= 369  ' 6 Sec x 370 loop = 2220 Sec( 37 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i6 = i6 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
 ' So that this routine is time leftover 2 Min
 '                (0.5 x 6)+(3 x 6)+ 37 = 58 Min
 '*******************************************************
End Function
Function form_7() As String
Dim form_7 As Long
Dim i7 As Long
Dim con7 As Long
Dim x As Long
```

```
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'7 time
'*****************************************************
    Out &H288, &HFF ' disconnect
form_7 = 0
Do While form_ <= 6 ' 7 loop
'%%%%% set delay time before connect 30 Sec
    i7 = 0
    Do While i7 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i7 = i7 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con7 = 0
    Do While con7 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con7 = con7 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_7 = form_7 + 1
Loop 'loop form_7
    '%%%%% set delay time after disconnect 33.5 Min
    i7 = 0
    Do While i7 <= 334  ' 6 Sec x 335 loop = 2010 Sec( 33.5 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i7 = i7 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
```

```
' So that this routine is time leftover 2 Min
'                (0.5 x 7)+(3 x 7)+ 33.5 = 58 Min
'*******************************************************
End Function
Function form_8() As String
Dim form_8 As Long
Dim i8 As Long
Dim con8 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'8 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_8 = 0
Do While form_8 <= 7 ' 8 loop
'%%%%% set delay time before connect 30 Sec
    i8 = 0
    Do While i8 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i8 = i8 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con8 = 0
    Do While con11 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con8 = con8 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_8 = form_8 + 1
 Loop 'loop form_8
```

```
    '%%%%% set delay time after disconnect 30 Min
    i8 = 0
    Do While i8 <= 299  ' 6 Sec x 300 loop = 1800 Sec( 30 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i8 = i8 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 2 Min
'               (0.5 x 8)+(3 x 8)+ 30 = 58 Min
'*******************************************************
End Function
Function form_9() As String
Dim form_9 As Long
Dim i9 As Long
Dim con9 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'9 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_9 = 0
Do While form_9 <= 8 ' 9 loop
'%%%%% set delay time before connect 30 Sec
    i9 = 0
    Do While i9 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i9 = i9 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con9 = 0
    Do While con9 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con9 = con9 + 1
    Loop
  '$$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_9 = form_9 + 1
Loop 'loop form_9
    '%%%%% set delay time after disconnect 27 Min
    i9 = 0
    Do While i9 <= 269  ' 6 Sec x 270 loop = 1620 Sec( 27 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i9 = i9 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
 ' So that this routine is time leftover 1.5 Min
 '               (0.5 x 9)+(3 x 9)+ 27 = 58.5 Min
 '*******************************************************
End Function
Function form_10() As String
Dim form_10 As Long
Dim i10 As Long
Dim con10 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'10 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_10 = 0
Do While form_10 <= 9 ' 10 loop
'%%%%% set delay time before connect 30 Sec
    i10 = 0
    Do While i10 <= 4  ' 6 sec x 5 loop = 30 Sec
```

```
      For x = 1 To 5000000
         A = 1
      Next x
    i10 = i10 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
'$$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$$
    con10 = 0
    Do While con10 <= 30  ' 6 sec x 31 loop = 3 Min ***
       For x = 1 To 5000000
          A = 1
       Next x
     con10 = con10 + 1
     Loop
  '$$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$$
     Out &H288, &HFF 'disconnect
  form_10 = form_10 + 1
Loop 'loop form_10
     '%%%%% set delay time after disconnect 23 Min
     i10 = 0
     Do While i10 <= 229  ' 6 Sec x 230 loop = 1380 Sec( 23 Min)
        For x = 1 To 5000000
           A = 1
        Next x
     i10 = i10 + 1
     Loop
     '%%%%%%%%%%%%%%%%%%%%%%
 ' So that this routine is time leftover 0.5 Min
 '                (0.5 x 10)+(3 x 10)+ 23 = 58 Min
 '*****************************************************
 End Function
 Function form_11() As String
 Dim form_11 As Long
 Dim i11 As Long
 Dim con11 As Long
 Dim x As Long
 Out &H283, &H80 ' set card is outport
 Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'11 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_11 = 0
Do While form_11 <= 10 ' 11 loop
'%%%%% set delay time before connect 30 Sec
    i11 = 0
    Do While i11 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i11 = i11 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute SSSSSSSSSSSS$
    con11 = 0
    Do While con11 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con11 = con11 + 1
    Loop
  '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_11 = form_11 + 1
 Loop 'loop form_11
    '%%%%% set delay time after disconnect 19 Min
    i11 = 0
    Do While i11 <= 189  ' 6 Sec x 190 loop = 1140 Sec( 19 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i11 = i11 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%
  ' So that this routine is time leftover 0.5 Min
  '             (0.5 x 11)+(3 x 11)+ 19 = 57.5 Min
```

```
'*******************************************************
End Function
Function form_12() As String
Dim form_12 As Long
Dim i12 As Long
Dim con12 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'12 time
'*******************************************************
      Out &H288, &HFF ' disconnect
form_12 = 0
Do While form_12 <= 11 ' 12 loop
'%%%%% set delay time before connect 30 Sec
      i12 = 0
      Do While i12 <= 4  ' 6 sec x 5 loop = 30 Sec
         For x = 1 To 5000000
            A = 1
         Next x
      i12 = i12 + 1
      Loop
      '%%%%%%%%%%%%%%%%%%%
      Out &H288, &H0  'connect
   '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
      con12 = 0
      Do While con12 <= 30  ' 6 sec x 31 loop = 3 Min ***
         For x = 1 To 5000000
            A = 1
         Next x
      con12 = con12 + 1
      Loop
    '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
      Out &H288, &HFF 'disconnect
    form_12 = form_12 + 1
Loop 'loop form_12
```

```
    '%%%%% set delay time after disconnect 15 Min
    i12 = 0
    Do While i12 <= 149  ' 6 Sec x 150 loop = 900 Sec( 15 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i12 = i12 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 0.5 Min
'                (0.5 x 12)+(3 x 12)+ 15 = 57 Min
'*******************************************************
End Function
Function form_13() As String
Dim form_13 As Long
Dim i13 As Long
Dim con13 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'13 time
'*******************************************************
    Out &H288, &HFF ' disconnect
form_13 = 0
Do While form_13 <= 12 ' 13 loop
'%%%%% set delay time before connect 30 Sec
    i13 = 0
    Do While i13 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i13 = i13 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con13 = 0
    Do While con13 <= 30  ' 6 sec x 31 loop = 3 Min ***
```

```
        For x = 1 To 5000000
          A = 1
        Next x
      con13 = con13 + 1
      Loop
    '$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
      Out &H288, &HFF 'disconnect
    form_13 = form_13 + 1
Loop 'loop form_13
      '%%%%% set delay time after disconnect 7 Min
      i13 = 0
      Do While i13 <= 119  ' 6 Sec x 120 loop = 720 Sec( 12 Min)
        For x = 1 To 5000000
          A = 1
        Next x
      i13 = i13 + 1
      Loop
      '%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 2.5 Min
'                (0.5 x 13)+(3 x 13)+ 12 = 57.5 Min
'*****************************************************
End Function
Function form_14() As String
Dim form_14 As Long
Dim i14 As Long
Dim con14 As Long
Dim x As Long
Out &H283, &H80 ' set card is outport
Out &H287, &H80 ' set card is outport
Out &H28B, &H80 ' set card is outport
'14 time
'*****************************************************
      Out &H288, &HFF ' disconnect
form_14 = 0
Do While form_14 <= 13 ' 14 loop
'%%%%% set delay time before connect 30 Sec
      i14 = 0
      Do While i14 <= 4  ' 6 sec x 5 loop = 30 Sec
        For x = 1 To 5000000
```

```
            A = 1
          Next x
        i14 = i14 + 1
        Loop
        '%%%%%%%%%%%%%%%%%%%%%
        Out &H288, &H0  'connect
     '$$$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$$
        con14 = 0
        Do While con14 <= 30  ' 6 sec x 31 loop = 3 Min ***
          For x = 1 To 5000000
            A = 1
          Next x
        con14 = con14 + 1
        Loop
      '$$$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$$
        Out &H288, &HFF 'disconnect
      form_14 = form_14 + 1
    Loop 'loop form_14
        '%%%%% set delay time after disconnect 9 Min
        i14 = 0
        Do While i14 <= 88  ' 6 Sec x 89 loop = 534 Sec( 8.9 Min)
          For x = 1 To 5000000
            A = 1
          Next x
        i14 = i14 + 1
        Loop
        '%%%%%%%%%%%%%%%%%%%%%
    ' So that this routine is time leftover ~ 2 Min
    '               (0.5 x 14)+(3 x 14)+ 9 = ~ 58 Min
    '**********************************************************
    End Function
    Function form_15() As String
    Dim form_15 As Long
    Dim i15 As Long
    Dim con15 As Long
    Dim x As Long
    Out &H283, &H80 ' set card is outport
    Out &H287, &H80 ' set card is outport
    Out &H28B, &H80 ' set card is outport
```

```
'15 time
'*********************************************************
    Out &H288, &HFF ' disconnect
form_15 = 0
Do While form_15 <= 14 ' 15 loop , usage 15 times
'%%%%% set delay time before connect 30 Sec
    i15 = 0
    Do While i15 <= 4  ' 6 sec x 5 loop = 30 Sec
      For x = 1 To 5000000
        A = 1
      Next x
    i15 = i15 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
    Out &H288, &H0  'connect
  '$$$$$$$$$$$$$$ Connect 3 Minute $$$$$$$$$$$$$
    con15 = 0
    Do While con15 <= 30  ' 6 sec x 31 loop = 3 Min ***
      For x = 1 To 5000000
        A = 1
      Next x
    con15 = con15 + 1
    Loop
  '$$$$$$$$$$$$ End connect 3 minute $$$$$$$$$$
    Out &H288, &HFF 'disconnect
  form_15 = form_15 + 1
Loop 'loop form_15
    '%%%%% set delay time after disconnect 7 Min
    i15 = 0
    Do While i15 <= 69  ' 6 Sec x 70 loop = 420 Sec( 7 Min)
      For x = 1 To 5000000
        A = 1
      Next x
    i15 = i15 + 1
    Loop
    '%%%%%%%%%%%%%%%%%%%%%
' So that this routine is time leftover 0.5 Min
'              (0.5 x 15)+(3 x 15)+ 7 = 59.5 Min
```

```
'*******************************************************
End Function
Private Sub Command1_Click()
Dim L55(1 To 120) As Integer
Dim tt As Integer
Dim h As Integer
Dim mt As Integer
Dim x As Integer
x = 1 ' set index start of table load
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
' Table load type day 50%  night 50%
L55(1) = 4 '
L55(2) = 4
.
. ( ข้อมูลโหลดแต่ละแบบ)
.
L55(119) = 5
L55(120) = 5
'%%%%%%%%%%%%%%% End Table load%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%% SET DEFAULT VALUE %%%%%%%%%%%%%%%%%%%%%
day_ini = 1
day_end = 5  ' set end of day
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'%%%%%%%%%%%%%%%% START PROGRAM %%%%%%%%%%%%%%%%%%%
' Set Out port @ I/O Card 8255 #2
Out &H283, &H80 'out port
Out &H287, &H80 'out port
Out &H28B, &H80 'out port
Do While day_ini <= day_end
        h = Hour(Time)
        mt = Minute(Time)
        tt = h + (mt / 100)
   loop_24 = 0
   Do While loop_24 <= 23   ' *** START LOOP 24 HOURS ***
      If (tt >= h And tt <= (h + 0.02)) Then
' Condition If, it's to start at beginning of hour,and neglect any time
' so that if program start user is get approximate time by self
```

```
      If L55(x) = 1 Then
        form_1
      ElseIf L55(x) = 2 Then
        form_2
      ElseIf L55(x) = 3 Then
        form_3
      ElseIf L55(x) = 4 Then
        form_4
      ElseIf L55(x) = 5 Then
        form_5
      Else:  L55(x) = 6
        form_6
      End If
      x = x + 1 ' set index next hour for load
      loop_24 = loop_24 + 1 ' set index next hour for counter
    Else
      h = Hour(Time)
      mt = Minute(Time)
      tt = h + (mt / 100)
    End If
  Loop 'END LOOP 24 HOURS
  day_ini = day_ini + 1
Loop
    Print "Finish program"
End Sub
Private Sub Command2_Click()
End
End Sub
```

ภาคผนวก ง.

ข้อมูลเทคนิคอุปกรณ์ที่ใช้ในการทดลอง

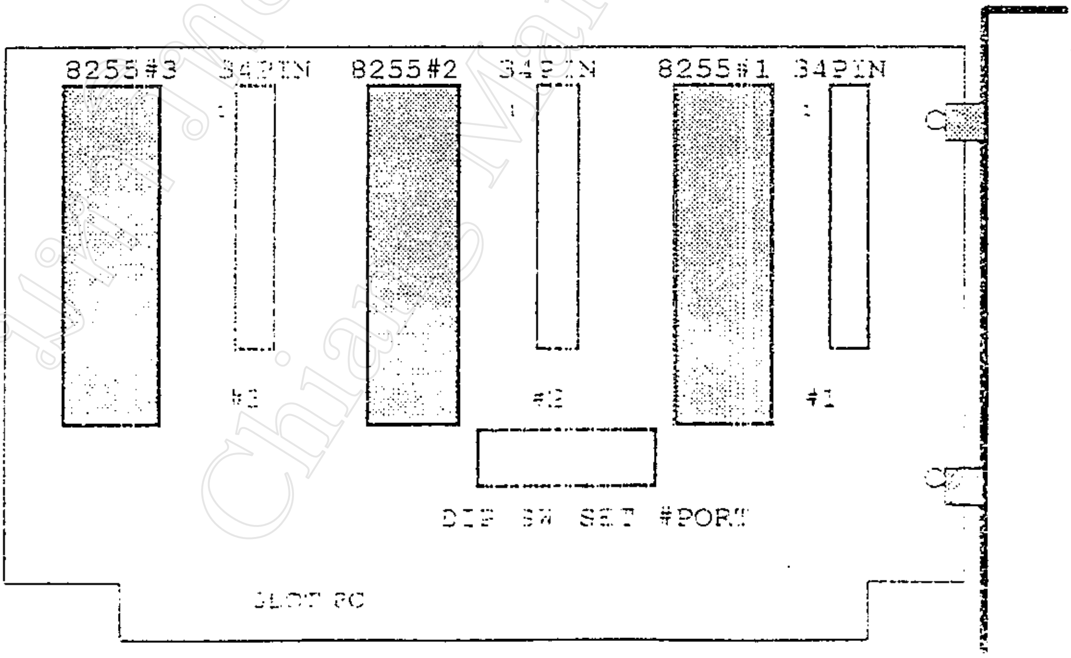
CARD ET-PC 8255

การทำงานของ ET-PC 8255

Card ET-PC 8255 ประกอบไปด้วย 2 ส่วนใหญ่ ๆ คือ ส่วน IC 8255 ซึ่งเป็น IC ทำหน้าที่เป็น อินพุทพอร์ต หรือ เอาท์พุทพอร์ต และส่วนของวงจร IC DECODE (เลือกตำแหน่งของ PORT 8255) คือ IC 74LS688 , 74LS139 และ DIP SW.

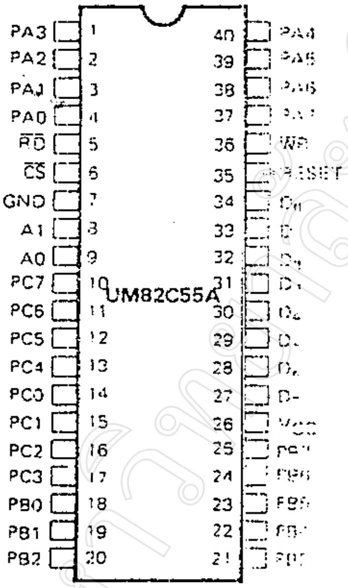
การใช้งาน IC 8255

IC 8255 นี้จะเป็น IC ซึ่งประกอบด้วยพอร์ตใช้งาน 3 พอร์ต และอีก 1 พอร์ต ควบคุม ก่อนที่เราใช้งาน 8255 เราจะต้องส่งข้อมูลไปให้ยังพอร์ตควบคุมก่อนว่าจะให้พอร์ตทั้ง 3 พอร์ต ของ 8255 ที่เหลือนั้นทำหน้าที่อะไรเป็น อินพุทพอร์ต หรือ เอาท์พุทพอร์ต เราจะต้องเป็นผู้กำหนด Control Code port ควบคุมดังรูป

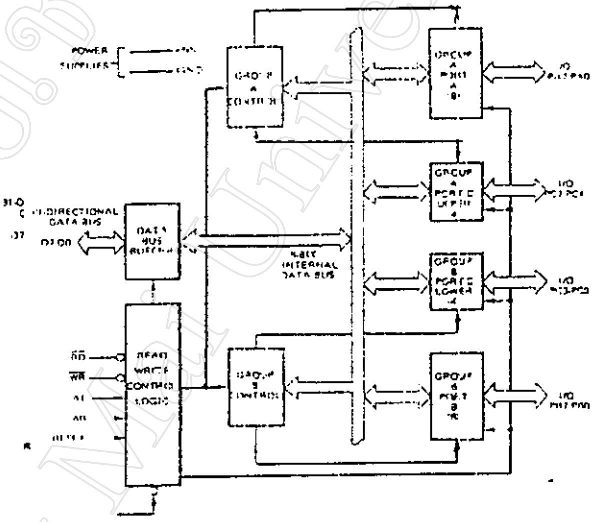


สถาปัตยกรรมของ IC 8255

Pin Configuration



Block Diagram



PIN NAMES

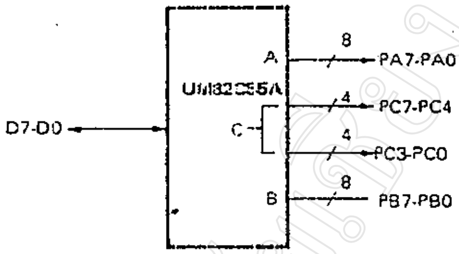
D7-D0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
VCC	+5 VOLTS
GND	0 VOLTS

ข้อมูลควบคุมพอร์ตต่าง ๆ

Mode 0 Configurations

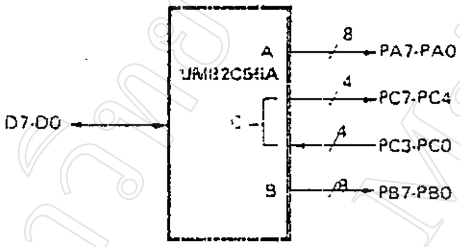
CONTROL WORD #0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0



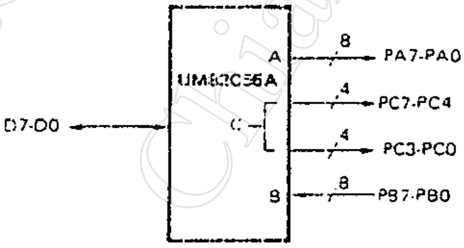
CONTROL WORD #1

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	1



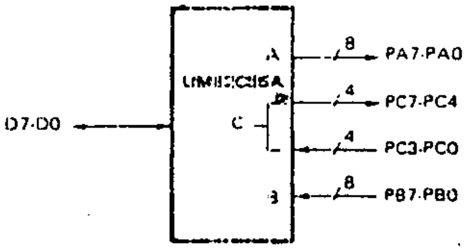
CONTROL WORD #2

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0



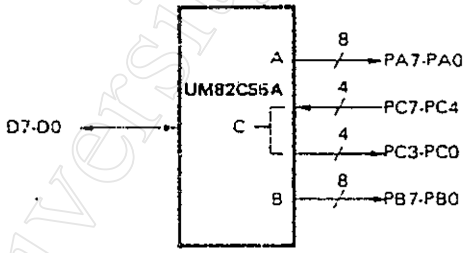
CONTROL WORD #3

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	1



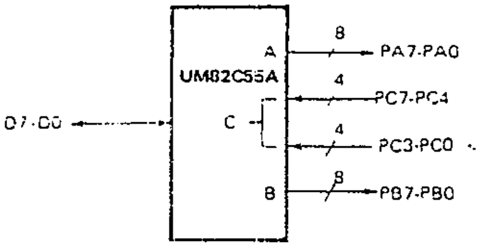
CONTROL WORD #4

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	0



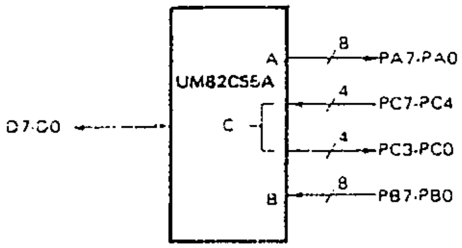
CONTROL WORD #5

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	1



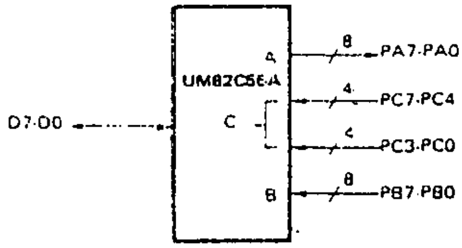
CONTROL WORD #6

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0



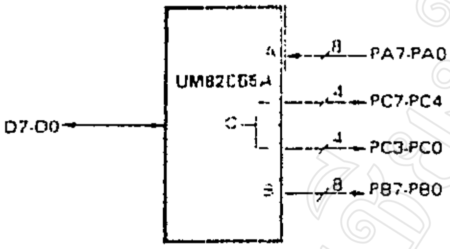
CONTROL WORD #7

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	1



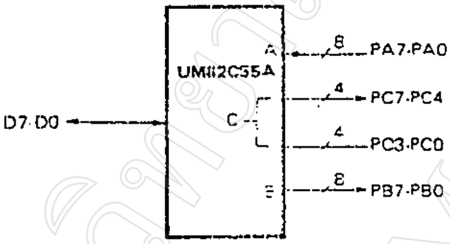
CONTROL WORD #8

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0



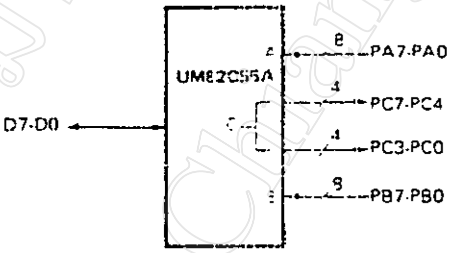
CONTROL WORD #9

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1



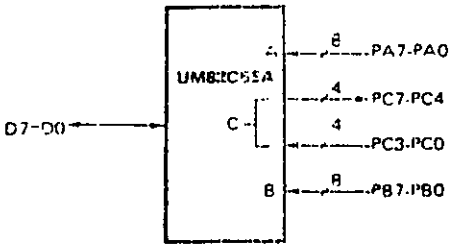
CONTROL WORD #10

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



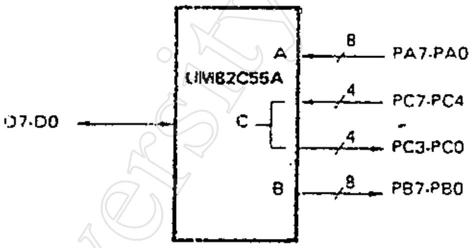
CONTROL WORD #11

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1



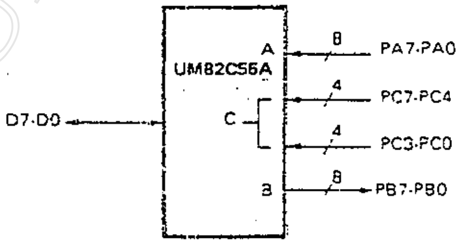
CONTROL WORD #12

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0



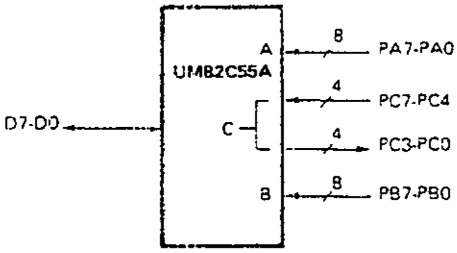
CONTROL WORD #13

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	1



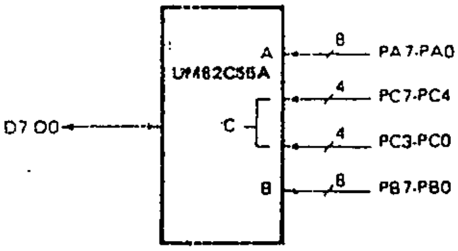
CONTROL WORD #14

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	0



CONTROL WORD #15

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	1



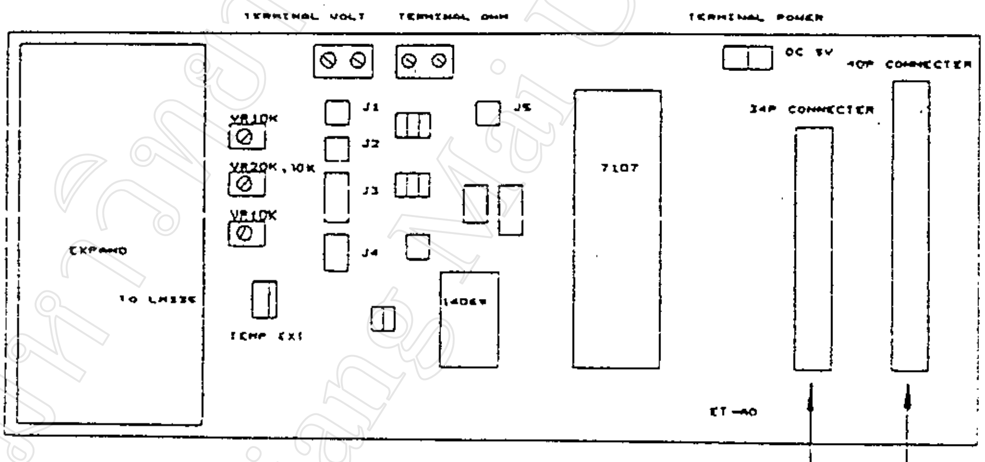
CARD ET- AD

การทำงานของ CARD ET-AD

ET-AD เป็นการ์ดประเภทเปลี่ยนแปลงสัญญานจากแอแนลอกเป็นดิจิตอล (Analog to Digital) ซึ่งใช้ IC 7107 เป็น A to D ขนาด 3 ½ Digit หรือ 0 – 1999 ระดับ หรือเทียบได้ขนาด 10 บิท สามารถที่จะประยุกต์วัดค่าต่าง ๆ ได้ เช่น

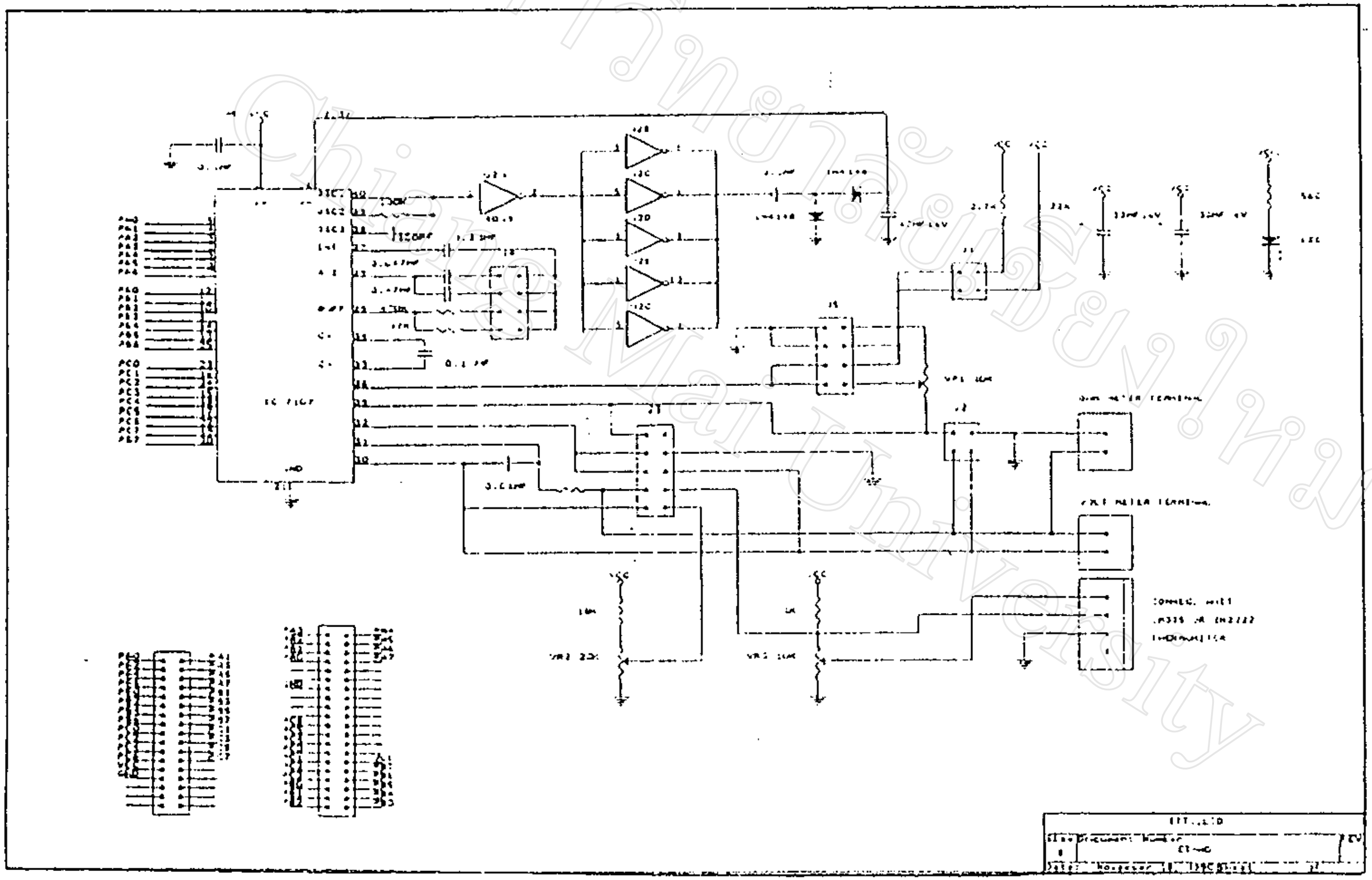
- □ เครื่องวัดความต้านทาน
- □ เครื่องวัดอุณหภูมิ
- □ เครื่องวัดแรงดัน

รูป CARD ET-AD



ต่อกับ I/O PORT ของ CP32BASIC
หรือ 7210 Z80

ต่อกับ 8255 PORT ของ
ET-BOARD V 3.0


OSC1
OSC2
OSC3
100K
100PF
0.1uF
0.47uF
47K
IC 7107
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
GND
VR1 10K
J1
J2
J3
OHM METER TERMINAL
VOLT METER TERMINAL
CONNECT WITH
LM335 OR 2N2222
THERMOMETER
VR2 20K
VR3 10K
VCC
Document Number
Date:
Sheet

HARRIS
SEMICONDUCTOR

ICL7106, ICL7107

$3^1/_2$ Digit LCD/LED Display A/D Converter

January 1994

Features

- Guaranteed Zero Reading for 0V Input on All Scales
- True Polarity at Zero for Precise Null Detection
- 1pA Typical Input Current
- True Differential Input and Reference, Direct Display Drive
  - LCD ICL7106
  - LED ICL7107
- Low Noise - Less Than 15μVp-p
- On Chip Clock and Reference
- Low Power Dissipation - Typically Less Than 10mW
- No Additional Active Circuits Required
- New Small Outline Surface Mount Package Available

Ordering Information

PART NUMBER	TEMPERATURE RANGE	PACKAGE
ICL7106CPL	0°C to +70°C	40 Lead Plastic DIP
ICL7106RCPL	0°C to +70°C	40 Lead Plastic DIP (Note 1)
ICL7106CM44	0°C to +70°C	44 Lead Metric Plastic Quad Flatpack
ICL7107CPL	0°C to +70°C	40 Lead Plastic DIP
ICL7107RCPL	0°C to +70°C	40 Lead Plastic DIP (Note 1)
ICL7107CM44	0°C to +70°C	44 Lead Metric Plastic Quad Flatpack

NOTE. 1 "R" indicates device with reversed leads.

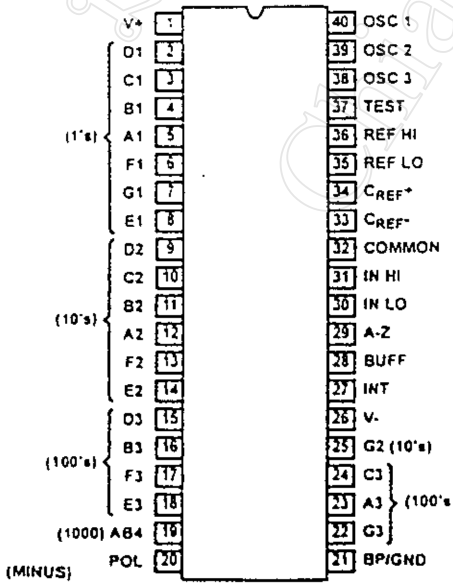
Description

The Harris ICL7106 and ICL7107 are high performance, low power $3^1/_2$ digit A/D converters. Included are seven segment decoders, display drivers, a reference, and a clock. The ICL7106 is designed to interface with a liquid crystal display (LCD) and includes a multiplexed backplane drive; the ICL7107 will directly drive an instrument size light emitting diode (LED) display.

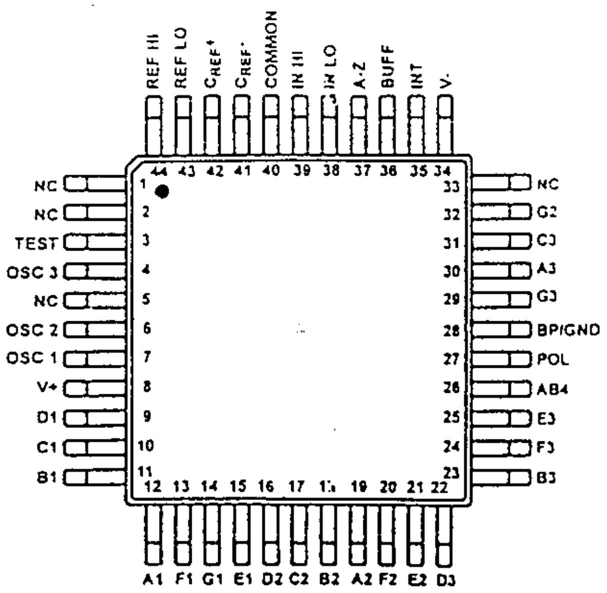
The ICL7106 and ICL7107 bring together a combination of high accuracy, versatility, and true economy. It features auto-zero to less than 10μV, zero drift of less than 1μV/°C, input bias current of 10pA max., and rollover error of less than one count. True differential inputs and reference are useful in all systems, but give the designer an uncommon advantage when measuring load cells, strain gauges and other bridge type transducers. Finally, the true economy of single power supply operation (ICL7106), enables a high performance panel meter to be built with the addition of only 10 passive components and a display

Pinouts

ICL7106, ICL7107
(PDIP)
TOP VIEW



ICL7106, ICL7107
(MQFP)
TOP VIEW



CAUTION These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures

Specifications ICL7106, ICL7107

Absolute Maximum Ratings

Supply Voltage	
ICL7106, V+ to V-	15V
ICL7107, V+ to GND	6V
ICL7107, V- to GND	-9V
Analog Input Voltage (Either Input) (Note 1)	V+ to V-
Reference Input Voltage (Either Input)	V+ to V-
Clock Input	
ICL7106	TEST to V+
ICL7107	GND to V+

Thermal Information

Thermal Resistance (MAX, See Note 1)	θ_{JA}
40 Pin Plastic Package	50°C/W
44 Pin MQFP Package	80°C/W
Maximum Power Dissipation	
ICL7106	1.0W
ICL7107	1.2W
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering 10s Max)	+265°C
Junction Temperature	+150°C

CAUTION. Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Electrical Specifications (Note 3)

PARAMETERS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SYSTEM PERFORMANCE					
Zero Input Reading	V_{IN} = 0.0V, Full-Scale = 200mV	-000.0	±000.0	+000.0	Digital Reading
Ratiometric Reading	$V_{IN} = V_{REF}$, V_{REF} = 100mV	999	999/ 1000	1000	Digital Reading
Rollover Error	$-V_{IN} = +V_{IN} \cong$ 200mV Difference in Reading for Equal Positive and Negative Inputs Near Full-Scale	-	±0.2	±1	Counts
Linearity	Full-Scale = 200mV or Full-Scale = 2V Maximum Deviation from Best Straight Line Fit (Note 5)	-	±0.2	±1	Counts
Common Mode Rejection Ratio	V_{CM} = 1V, V_{IN} = 0V, Full-Scale = 200mV (Note 5)	-	50	-	µV/V
Noise	V_{IN} = 0V, Full-Scale = 200mV (Pk-Pk Value Not Exceeded 95% of Time)	-	15	-	µV
Leakage Current Input	V_{IN} = 0 (Note 5)	-	1	10	pA
Zero Reading Drift	V_{IN} = 0, 0° < T_A < +70°C (Note 5)	-	0.2	1	µV/°C
Scale Factor Temperature Coefficient	V_{IN} = 199mV, 0° < T_A < +70°C, (Ext. Ref. 0ppm/°C) (Note 5)	-	1	5	ppm/°C
End Power Supply Character V- Supply Current	V_{IN} = 0 (Does Not Include LED Current for ICL7107)	-	0.8	1.8	mA
End Power Supply Character V- Supply Current	ICL7107 Only	-	0.6	1.8	mA
COMMON Pin Analog Common Voltage	25kΩ Between Common and Positive Supply (With Respect to + Supply)	2.4	2.8	3.2	V
Temperature Coefficient of Analog Common	25kΩ Between Common and Positive Supply (With Respect to + Supply)	-	80	-	ppm/°C
DISPLAY DRIVER ICL7106 ONLY					
Pk-Pk Segment Drive Voltage Pk-Pk Backplane Drive Voltage	V+= to V- = 9V, (Note 4)	4	5	6	V

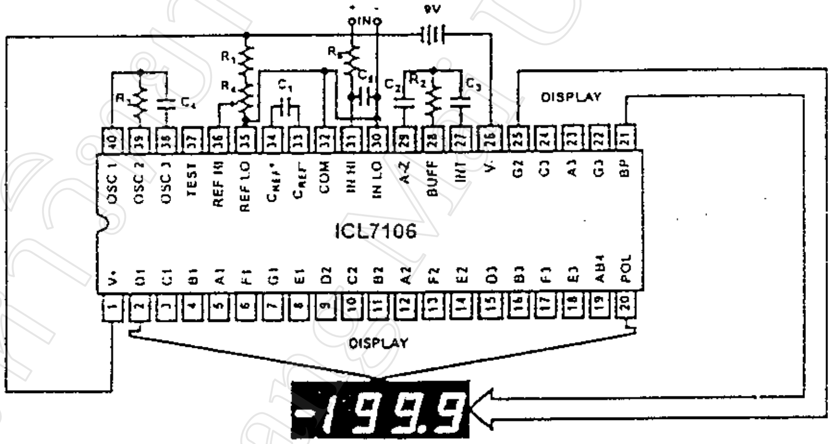
Electrical Specifications (Note 3) (Continued)

PARAMETERS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
ICL7107 ONLY					
Segment Sinking Current (Except Pin 19 and 20)	V+ = 5V, Segment Voltage = 3V	5	8	-	mA
Pin 19 Only		10	16	-	mA
Pin 20 Only		4	7	-	mA

NOTES.

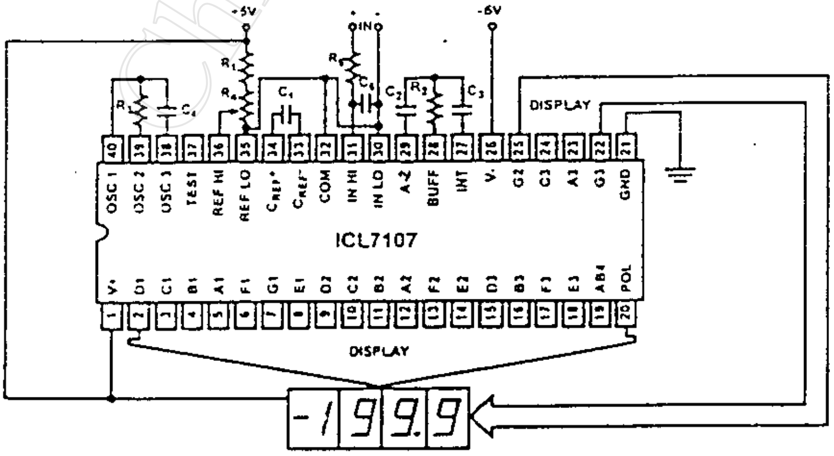
1. Input voltages may exceed the supply voltages provided the input current is limited to ±100μA.
2. Dissipation rating assumes device is mounted with all leads soldered to printed circuit board.
3. Unless otherwise noted, specifications apply to both the ICL7106 and ICL7107 at T_A = +25°C, f_{CLOCK} = 48kHz. ICL7106 is tested in the circuit of Figure 1. ICL7107 is tested in the circuit of Figure 2.
4. Back plane drive is in phase with segment drive for 'off' segment, 180° out of phase for 'on' segment. Frequency is 20 times conversion rate. Average DC component is less than 50mV.
5. Not tested, guaranteed by design.

Typical Applications and Test Circuits



C_1 = 0.1μF
C_2 = 0.47μF
C_3 = 0.22μF
C_4 = 100pF
C_5 = 0.02μF
R_1 = 24kΩ
R_2 = 47kΩ
R_3 = 100kΩ
R_4 = 1kΩ
R_5 = 1MΩ

FIGURE 1: ICL7106 TEST CIRCUIT AND TYPICAL APPLICATION WITH LCD DISPLAY COMPONENTS SELECTED FOR 200mV FULL-SCALE

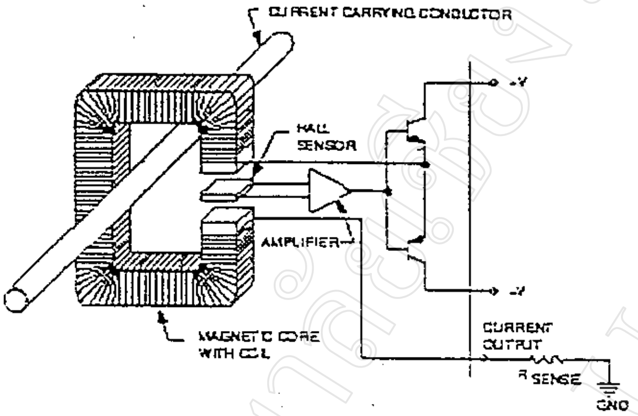


C_1 = 0.1μF
C_2 = 0.47μF
C_3 = 0.22μF
C_4 = 100pF
C_5 = 0.02μF
R_1 = 24kΩ
R_2 = 47kΩ
R_3 = 100kΩ
R_4 = 1kΩ
R_5 = 1MΩ

FIGURE 2. ICL7107 TEST CIRCUIT AND TYPICAL APPLICATION WITH LED DISPLAY COMPONENTS SELECTED FOR 200mV FULL-SCALE

Closed Loop Current Sensors

A closed loop current sensor consists of a Hall sensor mounted in an air gap of a magnetic core, a coil wound around the core and a current amplifier. The current carrying conductor placed through the aperture of the sensor produces a magnetic field that is proportionate to the current. This field is concentrated by the core and sensed by the Hall sensor. The Hall sensor is connected to the input of the current amplifier, which drives the coil. The current through the coil produces an opposing field to that provided by the current through the aperture, thus the flux in the core is constantly driven to zero. The coil connects to the output of the sensor. Therefore the output is a current proportional to the aperture current, multiplied by the number of turns on the coil. A sensor with a 1000 turn coil provides an output of 1mA per ampere. The current output is converted to a voltage by connecting a resistor between the output of the sensor and ground. The output is scaled by selecting the resistor value. Closed loop sensors measure DC and AC currents and provide electrical isolation. They offer fast response, high linearity and low temperature drift. The current output of the closed loop sensor is relatively immune to electrical noise. They are the sensor of choice when high accuracy is essential.



• Fast Response • Excellent linearity • Wide bandwidth • Low temperature drift

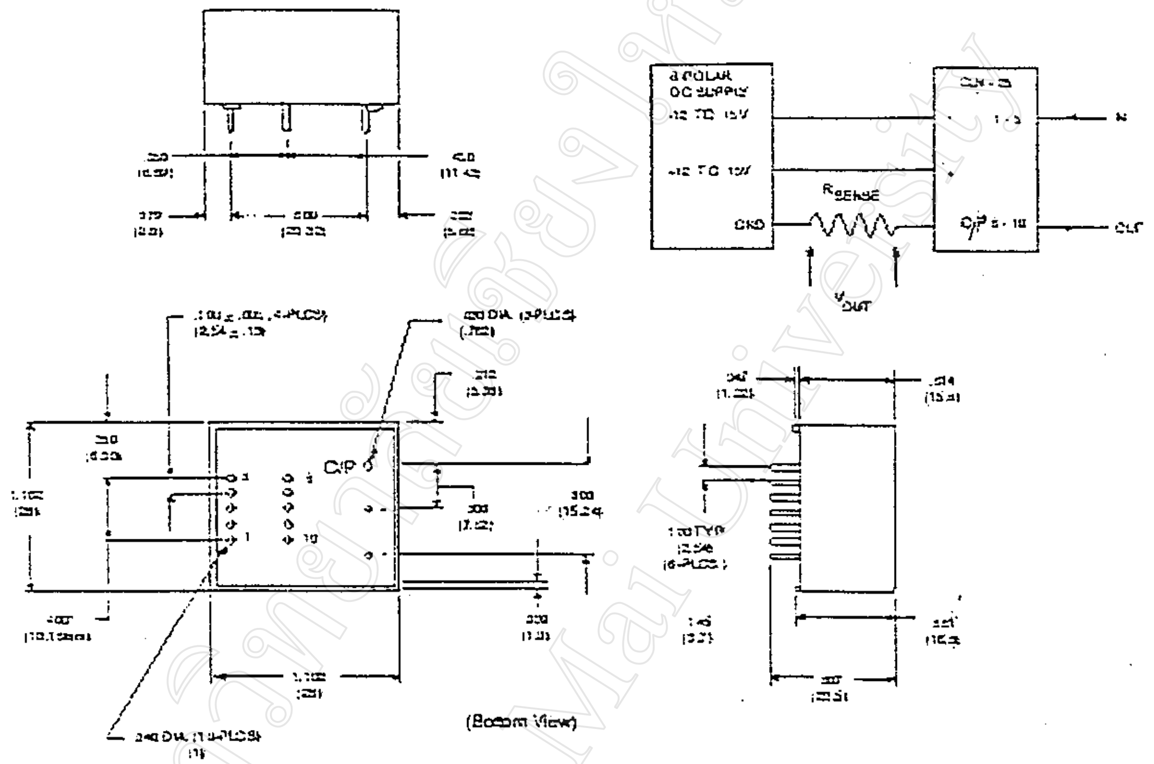
Model	Nominal Current (IN)	Linearity (%)	Frequency	Output (mA at (IN))	Aperature Inches (mm)	Page(s)
CLS-25	25	0.1	dc to 150 kHz	50	PCB Connection	30-31
CLN-25	25	0.2	dc to 150 kHz	25	PCB Connection	32-33
CLN-50	50	0.1	dc to 150 kHz	50	0.83 x 0.39	34-35
CLN-50SP1	50	0.1	dc to 150 kHz	50	PCB Connection	36-37
CLN-100	100	0.1	dc to 150 kHz	100	0.53 x 0.39	34-35
CLN-100SP1	25	0.2	dc to 150 kHz	100	PCB Connection	36-37
CLN-200	200	0.1	dc to 150 kHz	100	0.53 x 0.39	[illegible]
CLN-300	300	0.1	dc to 150 kHz	160	0.9	38-39
CLN-300SP1	150	0.1	dc to 150 kHz	160	Busbar	40-41
CLN-500	500	0.1	dc to 150 kHz	100	1.02	42-43
CLN-500SP1	100	0.1	dc to 100 kHz	100	Busbar	40-41
CLN-1000	1000	0.1	dc to 150 kHz	100	1.02	42-43
CLO-300	300	0.1	dc to 150 kHz	160	0.905	44-45
CLO-500	500	0.1	dc to 100 kHz	100	1.025	44-45

Note: Due to continuous process improvement, all specifications listed in this catalog subject to change without notice.

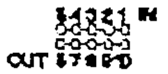
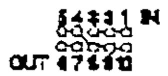
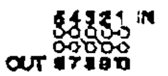
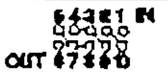
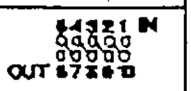
Model CLN-25

Mechanical Dimensions

All dimensions are in inches (millimeters)



Measuring Range Table

Number of Turns of I_N	I_N(A)	Peak (A)	Normal output Current (mA)	Turn Ratio	Insertion Loss Resistance (mΩ)	Insertion Loss Inductance (µH)	Recommended Connections
1	25	38	25	1/1000	0.3	0.023	 5 4 3 2 1 IN OUT 6 7 8 9 10
2	12	18	24	2/1000	1.1	0.09	 5 4 3 2 1 IN OUT 6 7 8 9 10
3	8	12	24	3/1000	2.5	0.21	 5 4 3 2 1 IN OUT 6 7 8 9 10
4	6	9	24	4/1000	4.4	0.37	 5 4 3 2 1 IN OUT 6 7 8 9 10
5	5	7	25	5/1000	6.3	0.56	 5 4 3 2 1 IN OUT 6 7 8 9 10

Note: Due to continuous process improvement, all specifications listed in this catalog subject to change without notice.

Model CLN-25

Description

The Model CLN-25 is a closed loop Hall effect current sensor that accurately measures dc and ac currents and provides electrical isolation between the current carrying conductor and the output of the sensor.

Electrical Specifications

Nominal current (I_N)	25 Ampere turns rms	
Measuring range *	0 to 36 Ampere turns (A.t.)	
Sense resistor	R. min.	R. max.
with ± 12 V at 25 A.t. peak	100 ohms	200 ohms
at 36 A.t. peak	100 ohms	140 ohms
with ± 15 V at 25 A.t. peak	100 ohms	320 ohms
at 36 A.t. peak	100 ohms	190 ohms
Nominal analog output current	25 mA	
Turns ratio	1-2-3-4-5:1000	
Overall accuracy at 25 °C and ±12 V	± 0.7% of I_N	
Overall accuracy at 25 °C and ± 15 V	± 0.5% of I_N	
Supply voltage (Vdc)	± 12 to ± 15 (± 5%)	
Dielectric strength	between the current carrying conductor and the output of the sensor: 5 kV rms/50 Hz/1 min.	

Accuracy-Dynamic Performance

	Typical	Max.
Zero current offset at 25 °C (± 15V)	± 0.05 mA	± 0.15 mA
Residual current offset after an overload of 3 x I_N	± 0.05 mA	± 0.15 mA
Offset current temperature drift (± 15V)		
(between 0 °C and +25 °C)	± 0.06 mA	± 0.25 mA
(between +25 °C and +70 °C)	± 0.1 mA	± 0.35 mA
Linearity	better than ±0.2%	
Response time	less than 1 μs	
di/dt accurately followed	better than 50 A/μs	
Bandwidth	0 to 150 kHz (- 1 dB)	

General Information

Operating temperature	-40 °C to +85 °C
Storage temperature	-40 °C to +90 °C
Current drain	10 mA (at ± 15 V) plus output current
Coil resistance	110 ohms (at 70 °C)
Package	Flame retarded plastic case
Weight	16 grams
Mounting	Designed to mount directly on PCB via through hole connection pins
Output reference	To obtain a positive output on the terminal marked "O/P", current must flow from terminals 1,2,3,4 and 5 to terminals 10,9,8,7 and 6 (conventional flow)

Notes

1. The CLN-25 offers a choice of 5 measuring ranges (see table)
2. Due to continuous process improvement, all specifications listed in this catalog subject to change without notice.

ประวัติผู้เขียน

ชื่อ	นายศุภกิจ แสงทอง
วัน เดือน ปี เกิด	19 ตุลาคม 2510
ประวัติการศึกษา	สำเร็จการศึกษาระดับปริญญาตรี ครุศาสตร์อุตสาหกรรม (สาขาวิศวกรรมไฟฟ้า) จากสถาบันเทคโนโลยีราชมงคล จังหวัดเชียงใหม่ เมื่อปีการศึกษา 2535
ประสบการณ์ในการทำงาน	สายงานวิศวกร แผนกอาคารและการกำลัง เขตโทรศัพท์ภาคเหนือที่ 1 องค์การโทรศัพท์แห่งประเทศไทย ตั้งแต่ ปี พ.ศ. 2535 - ปัจจุบัน