## 5.1 CONCLUSION

A solar tracker has been designed and simulated in order to track the sun position according to the user choice (Dual-Axis, One-Axis, Annual) using four LDR to find the sun position, a Atmega32 microcontroller was then used to drive two stepper motors to change the panel angle through the day.

The PV panel ALT50-12P50 WATTS, 12 VoLTS was simulated under Proteus. And The PV tilte angle, zenith angle, voltage, current and power plus the solar radiation and the surrounding temperature the has been measured to allow the user Know the success and status of the tracker. A graphical User interface has been built using the Matlab environment to allow the user to remotely monitor and control the tracker, Database was also constructed in order to monitor the system performance through its life time and the obtained simulation results are quite encouraging; they show the efficiency and the simplicity of the proposed design.

## 5.2 RECOMMENDATIONS

The project gives a clear idea about the Solar Tracking Mechanism. for further work I recommend to

- Use different Maximum Power Point Tracking MPPT method and modified algorithms for increasing efficiency like Fuzzy Control or Neural Control.
- Making Prototype for the proposed design to be tested, compared with current design and optimized.
- Calibrating of the sensors in real environment for doubtlessness.
- Finding the best tele-controlling medium suits the system purpose.
- Designing Android GUI instate of the PC based GUI.
- The optimum measuring time to be found.
- Annual control method could be every day adjustment rather than every month

# REFRENCESS


[1] V. P. M. Libra, "New bifacial solar trackers and tracking concentrators .", Czech University of Agriculture in Prague, 2007.

[2] A. Balabel1, A. A. Mahfouz, and F. A.Salem, "Design and Performance of Solar Tracking Photo-Voltaic System; Research and Education," *International journal of Control, Automation and Systems .* vol. 1, 2013.

[3] A. K. a. Hossein Mousazadeh a , Arzhang Javadi b , Hossein Mobli a , Karen Abrinia c , Ahmad Sharifi, "A review of principle and sun-tracking methods for maximizing solar systems output," *Renewable and Sustainable Energy Reviews,* 13 (2009) 1800-1818.

[4] J. Rizk and Y. Chaiko, "Solar tracking system: more efficient use of solar panels," in *Proceedings of World Academy of Science, Engineering and Technology .* 2008.

[5] Y. J. Huang, T. C. Kuo, C. Y. Chen, C. H. Chang, P. C. Wu, and T. H. Wu, "The Design and Implementation of a Solar Tracking Generating Power System by " *Engineering Letters .* 19 November 2009 2009.

[6] C. Alexandru and C. Pozna, "Simulation of a dual-axis solar tracker for improving the performance of a photovoltaic panel," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy,* vol. 224, pp. 797-811, 2010.

[7] N. Barsoum and P. Vasant, "Simplified solar tracking prototype," *Global Journal of Technology and Optimization GJTO,* vol. 1, pp. 38-45, 2010.

[8] A. Catarius and M. Christiner, "Azimuth-Altitude Dual Axis Solar Tracker," Bachelor of Science, Worcester Polytechnic Institute . 2010.

[9] C. Chin, A. Babu, and W. McBride, "Design, modeling and testing of a standalone single axis active solar tracker using MATLAB/Simulink," *Renewable Energy,* vol. 36, pp. 3075-3090, 2011.

[10] M. Usta, O. Akyaszi, and I. Atlas, "Design and performance of solar tracking system with fuzzy logic Controller," in *Sixth International Advanced Technologies Symposium (IATS'11)* . , Elazig Turkey, 2011, pp. 16-18.

[11] S. Kivrak, M. Gunduzalp, and F. Dincer, "Theoretical and experimental performance investigation of a two-axis solar tracker under the climatic condition of Denizli, Turkey," *Przeglqd Elektrotechniczny,* vol. 88, pp. 332-336, 2012.

[12] S. K. Ray, M. A. Bashar, M. Ahmad, and F. B. Sayed, "Two ways of rotating freedom solar tracker by using ADC of microcontroller," *Global Journal of Researches In Engineering,* vol. 12, 2013.

[13] J. F. Lee, N. A. Rahim, and Y. A. Al-Turki, "Performance of dual-axis solar tracker versus static solar system by segmented clearness index in Malaysia," *International Journal of Photoenergy,* vol. 2013, 2013.

[14] F. A. Salem, "Mechatronics Design of Solar Tracking System," *International Journal of Current Engineering and Technology,* vol. 3, June 2013 2013.

[15] H. B. Nirmal and S. A. Naveed, "MICROCONTROLLER BASED AUTOMATIC SOLAR POWER TRACKING SYSTEM," *International Journal of Electrical Engineering and Technology (I JEET)* vol. 4, January- February (2013) 2013.

[16] V. P. More and V. Kulkarni, "Design and Implementation of Microcontroller Based Automatic Solar Radiation Tracker," *International Journal of Current Engineering and Technology* . vol. 2, pp. 230-234, 2014.

[17] R. A. Ferdaus, M. A. Mohammed, S. Rahman, S. Salehin, and M. A. Mannan, "Energy Efficient Hybrid Dual Axis Solar Tracking System," *Journal of Renewable Energy,* p. 12, 2014.

[18] G. Prinsloo, "Solar Tracking, Sun Tracking, Sun Tracker, Solar Tracker, Follow Sun, Sun Position," JUNE 2014 2014.

[19] K. Jäger, O. Isabella, A. H. M. Smets, R. A. C. M. M. v. Swaaij, and M. Zeman, *Solar Energy Fundamentals , Technology and Systems*: Delft University of Technology, 2014.

[20] S. Baidar, N. Kille, I. Ortega, R. Sinreich, D. Thomson, J. Hannigan*, et al.*, "Development of a digital mobile solar tracker," *Development,* vol. 8, pp. 11401-11427, 2015.

[21] K. Maharaja, R. J. Xavier, L. J. Amla, and P. P. Balaji, "Intensity Based Dual Axis Solar Tracking System," *International Journal of Applied Engineering Research,* vol. 10, pp. 19457-19465, 2015.

[22] M. Prodhan, M. Hamid, D. Hussain, and M. Huq, "Design, Construction and Performance Evaluation of an Automatic Solar Tracker," *Journal of Scientific Research,* vol. 8, pp. 1-12, 2016.

[23] N. Barsoum, "Fabrication of Dual-Axis Solar Tracking Controller Project " *Intelligent Control and Automation, 2011, 2, 57-68,* Online May 2011.

[24] M. A. Ionită and C. Alexandru, "Optimal design of the mechanical device for a photovoltaic tracking mechanism," in *Applied Mechanics and Materials* 2012, pp. 114-123.

[25] C. Alexandru, "A novel open-loop tracking strategy for photovoltaic systems," *The Scientific World Journal,* vol. 2013, 2013.

[26] C. Alexandru, "A comparative analysis between the tracking solutions implemented on a photovoltaic string," *Journal of Renewable and Sustainable Energy,* vol. 6, p. 053130, 2014.

[27] R. Patel, "Design and Implementation of an Isolated Solar Photovoltaic Power Generation System," Master of Technology in Electrical Engineering, Department of Electrical Engineering, National Institute of Technology, Rourkela 2014.

[28] H. Bourdoucen and A. Gastli, "Analytical Modelling and Simulation of Photovoltaic Panels and Arrays," *The Journal of Engineering Research* . vol. 41, 2007.

[29] A. zagaoui, P. Petit, M. Aillerie, J. P. Sawicki, A. W. Belarbi, M. D. Krachi*, et al.*, "photopholtaic Cell Panel Array Characterizations and Modeling Considering Both Reverse and Direct Modes," *Elsevier,* vol. 6, pp. 695-703, 2011.

[30] H. K. Abd-Elmaged, "Passive Solar Tracking System," M.Sc Department of physics, University of Khartoum, 2005.

[31] S. A. Kalogirou, "Solar thermal collectors and applications," *Progress in energy and combustion science,* vol. 30, pp. 231-295, 2004.

[32] D. Calcutt, F. Cowan, and H. Parchizadeh, *8051 Microcontroller: An Applications Based Introduction*: Newnes, 2003.

[33] P. Alley, "Introductory Microcontroller Programming," Worcester Polytechnic Institute, 2011.

[34] G. Gridling and B. Weiss, "February 26, 2007," *Introduction to Microcontrollers*.

[35] R. Condit and D. W. Jones, "Stepping motors fundamentals," *Microchip Inc. Publication AN907,* pp. 1-22, 2004.

# APPENDICES:

## A. MATLAB PROGRAM:

```
function varargout = DAST(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @DAST_OpeningFcn, ...
                   'gui_OutputFcn',  @DAST_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before DAST is made visible.
function DAST_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = DAST_OutputFcn(hObject, eventdata, handles)
%clc
% Get default command line output from handles structure
varargout{1} = handles.output;

for i=0:1:inf
  ti =datestr(now, 'HH:MM:SS PM');
  set(handles.time,'string',ti);
```

```
  datt =datestr(now, 'dd/mmmm/yyyy');
  set(handles.date,'string',datt);

wait=str2num(get(handles.time_interval,'string'));
t =str2num(datestr(now, 'MM'));
wt=rem(t,wait);
s =str2num(datestr(now, 'SS'));

pause (0.5)

if ((wait < 60)&&( wt ==0)&&( s ==0))
 load('E:\my master project\DASTGUI\readdata.mat');
s = serial('COM4','BaudRate',9600);
fopen(s);
fprintf(s,'%i',17);

l=fgetl(s);
set(handles.current,'string',l)
readdata.l=l;
readdata.current(length(readdata.current)+1)={l};
n=fgetl(s);
set(handles.volt,'string',n)
readdata.n=n;
readdata.volt(length(readdata.volt)+1)={n};
tt=fgetl(s);
readdata.tt=tt;
readdata.temp(length(readdata.temp)+1)={tt};
o=fgetl(s);
readdata.o=o;
readdata.tilte(length(readdata.tilte)+1)={o};
p=fgetl(s);
readdata.p=p;
readdata.zen(length(readdata.zen)+1)={p};
fclose(s);
%close all
set(handles.temp,'string',tt)
set(handles.tilte,'string',o)
set(handles.zen,'string',p)
```

```
q1=str2num(get(handles.current,'string'));
q2=str2num(get(handles.volt,'string'));
q=q2*q1;
r=q/(0.6*0.5);
set(handles.power,'string',q)
set(handles.radiation,'string',r)

readdata.power(length(readdata.power)+1)={q};
readdata.radiation(length(readdata.radiation)+1)={r};

ti =datestr(now, 'HH:MM:SS PM');
readdata.rtime(length(readdata.rtime)+1)={ti};

datt =datestr(now, 'dd/mmmm/yyyy');
readdata.rdate(length(readdata.rdate)+1)={datt};

save('E:\my master project\DASTGUI\readdata.mat')

%pause (2)
end
end
% --------------------------------------------------------------------
function Start_Callback(hObject, eventdata, handles)
% --------------------------------------------------------------------
function login_Callback(hObject, eventdata, handles)
  set(handles.text1,'visible','off')
  set(handles.text2,'visible','off')
  set(handles.text3,'visible','off')
  set(handles.text4,'visible','off')
  set(handles.text5,'visible','off')
  set(handles.text6,'visible','off')
  set(handles.text7,'visible','off')
  set(handles.text8,'visible','on')
  set(handles.text9,'visible','on')
  set(handles.name,'visible','on')
  set(handles.password,'string','')
  set(handles.name,'string','')
  set(handles.password,'visible','on')
```

```
  set(handles.log_in,'visible','on')
  set(handles.uipanel2,'visible','off')
  set(handles.logd,'visible','on')
  set(handles.logt,'visible','on')
  set(handles.log,'visible','on')
  set(handles.serial_interval,'visible','off')
 m =importdata('E:\my master project\DASTGUI\logdata.mat')  ;
set(handles.logd,'string',m.dat)
set(handles.logt,'string',m.time)
function log_in_Callback(hObject, eventdata, handles)

n=char('p');
p=1;
name=char(get(handles.name,'string'));
password=str2num(get(handles.password,'string'));
if (isequal(name,n)  && isequal(password,p))
  set(handles.text8,'visible','off')
  set(handles.text9,'visible','off')
  set(handles.name,'visible','off')
  set(handles.password,'visible','off')
  set(handles.log_in,'visible','off')
  set(handles.uipanel2,'visible','on')
  set(handles.datetime,'visible','on')
  set(handles.current_reading,'visible','on')
  set(handles.Maximum_Reading,'visible','on')
  set(handles.logt,'visible','off')
  set(handles.logd,'visible','off')
  set(handles.log,'visible','off')
  set(handles.notation,'visible','off')
  set(handles.serial_interval,'visible','on')
  set(handles.uitoolbar2,'visible','on')

else
   set(handles.notation,'visible','on')
end
 m =importdata('E:\my master project\DASTGUI\readdata.mat') ;
set(handles.current,'string',m.readdata.current(length(m.readdata.current)));
set(handles.volt,'string',m.readdata.volt(length(m.readdata.volt)));
```

```
set(handles.power,'string',m.readdata.power(length(m.readdata.power)));
set(handles.radiation,'string',m.readdata.radiation(length(m.readdata.radiation)));
set(handles.temp,'string',m.readdata.temp(length(m.readdata.temp)));
set(handles.zen,'string',m.readdata.zen(length(m.readdata.zen)));
set(handles.tilte,'string',m.readdata.tilte(length(m.readdata.tilte)));
% --------------------------------------------------------------------
function close_Callback(hObject, eventdata, handles)
load ('E:\my master project\DASTGUI\logdata.mat')
dat =datestr(now, 'dd/mm/yyyy');
time =datestr(now, 'HH:MM:SS PM');
logdata.time(length(logdata.time)+1)={time};
logdata.date(length(logdata.date)+1)={dat};
save('E:\my master project\DASTGUI\logdata.mat')
close all
% --- Executes on button press in up.
function up_Callback(hObject, eventdata, handles)
time_interval=str2num(get(handles.time_interval,'String'))
if (time_interval<60)
time_interval=time_interval+10;
else
     errordlg('Maximum  Value  ',' Error');
end
set(handles.time_interval,'string',time_interval)

% --- Executes on button press in down.
function down_Callback(hObject, eventdata, handles)
time_interval=str2num(get(handles.time_interval,'String'))
if (time_interval<=60)
time_interval=time_interval-10;
end
if (time_interval<10)
    errordlg('Minimum  Value  ',' Error');
else
set(handles.time_interval,'string',time_interval)
end
% --- Executes on button press in update.
function update_Callback(hObject, eventdata, handles)
l=importdata('E:\my master project\DASTGUI\readdata.mat');
kk1=l.readdata.power;
f1=cell2mat(kk1);
```

```
v1=max(f1)
set(handles.mpower,'string',v1)

kk2=l.readdata.current;
f2=sort(kk2);
v2=f2(length(f2))
set(handles.mcurrent,'string',v2);

kk3=l.readdata.volt;
f3=sort(kk3);
v3=f3(length(f3));
set(handles.mvolt,'string',v3);

kk4=l.readdata.radiation;
f4=cell2mat(kk4);
v4=max(f4)
set(handles.mradiation,'string',v4);
kk5=l.readdata.temp;
f5=sort(kk5);
v5=f5(length(f5))
set(handles.mtemp,'string',v5);
% --- Executes on button press in clear.
function clear_Callback(hObject, eventdata, handles)
hold off
plot(0)
% --- Executes on button press in plot.
function plot_Callback(hObject, eventdata, handles)
l=importdata('E:\my master project\DASTGUI\readdata.mat');

kk1=l.readdata.power;
f=cell2mat(kk1);

kk2=l.readdata.current;
f2=cell2mat(kk2);
f2=sscanf(f2,'%g');
kk3=l.readdata.volt;
f3=cell2mat(kk3);
f3=sscanf(f3,'%g');
kk4=l.readdata.temp;
```

```
f4=cell2mat(kk4);
f4=sscanf(f4,'%g');
grid on
hold on
if (get(handles.tpower,'Value') == get(handles.tpower,'Max'))
    plot(f,'y')
end
if (get(handles.tvolt,'Value') == get(handles.tvolt,'Max'))
    plot(f3,'k')
end
if (get(handles.tcurrent,'Value') == get(handles.tcurrent,'Max'))
plot(f2,'g')
end
if (get(handles.ttemp,'Value') == get(handles.ttemp,'Max'))
plot(f4,'b')
end
function oneaxeis_Callback(hObject, eventdata, handles)
m = serial('COM4','BaudRate',9600);
fopen(m);
fprintf(m,'%i',4);
fclose(m)
% --- Executes on button press in annual.
function annual_Callback(hObject, eventdata, handles)
if get(handles.annual,'value') == get(handles.annual,'Max')
      ti =datestr(now, 'HH:MM:SS');
      datt =datestr(now, 'dd');
  if ((ti=='12:00:00')&&(dat=='1'))
      m = serial('COM4','BaudRate',9600);
      fopen(m);
      fprintf(m,'%i',3);
      fclose(m);
end
end
% --- Executes on button press in auto.
function auto_Callback(hObject, eventdata, handles)
m = serial('COM4','BaudRate',9600);
fopen(m);
fprintf(m,'%i',2);
fclose(m);
```

## B. MEASURING MICROCONTROLLER PROGRAM:

```
$regfile = "m32def.dat"
$crystal = 1000000
$baud = 9600
$hwstack = 32
$swstack = 10
$framesize = 40

'-----------------------------------------------------------------
'              Sudan University of Science & Technology
'                Collage of Graduate Studies
'              MSc. Program In Mechatronics Engineering

' Design a Remotely Monitored and Controlled Dual Axis Solar Tracker (DAST)
'       Prepared by:
'        Rawah Adam Idriss Eldilideam
'-----------------------------------------------------------------
'     // CONFIGRATIONS

Config Graphlcd = 128 * 128 , Dataport = Portb , Controlport = Portc , Ce = 2 , Cd
= 3 , Wr = 0 , Rd = 1 , Reset = 4 , Mode = 8 , Fs = 5
Config Clock = Soft
  Config Date = Dmy , Separator = .
  Date$ = "01.09.09"
  Time$ = "23:59:52"
  Cursor Off
  Enable Interrupts
Config Adc = Single , Prescaler = Auto , Reference = Avcc
  Start Adc
```

```
Open "comd.3:9600,8,n,1" For Output As #1
Open "comd.2:9600,8,n,1" For Input As #2
Config Serialin = Buffered , Size = 20
'.            ... USED CONISTANSES
  ' ;;;;;;;;;; for sensors readings
Dim W As Word , Current As Single , Voltage As Single , Temp As Single
Dim Angle1 As Single , Angle2 As Single
Dim Current1 As String * 7 , Voltage1 As String * 6 , Temp1 As String * 6
Dim Angle11 As String * 6 , Angle21 As String * 6
Dim Buff As Byte , D As Byte
'              ... SUB configrations
Declare Sub Sensors
' ……THE PROGRAME
   Locate 2 , 1 : Lcd "IN THE NAME OF "
     Showpic 0 , 30 , Plaatje
     Wait 1
     Cls
  Line(1 , 1) -(1 , 128) , 200
  Locate 2 , 2 : Lcd "MSc.Program In Mechatronics Engineering"
    Waitms 2
  Locate 10 , 2 : Lcd "BY: Rawah Adam"
    Waitms 200
  Cls
  Locate 3 , 8 : Lcd "DAST"
    Showpic 5 , 20 , Logo
    Wait 1
   Cls Text
Cls Graph
          Wait 1
 Do
```

```
    Locate 1 , 8 : Lcd Date$
    Locate 2 , 8 : Lcd Time$
    Locate 3 , 1 : Lcd "----------"
    Gosub Sensors
   Locate 5 , 1 : Lcd "PV I =" ; Current1 ;" " ; "AMP "
   Locate 7 , 1 : Lcd "PV v =" ; Voltage1 ;" " ; "VOLT "
   Locate 9 , 1 : Lcd "TEMP = " ; Temp1 ; " " ; "C"
   Locate 11 , 1 : Lcd "TILTE = " ; Angle11 ; " DEG "
   Locate 12 , 1 : Lcd "ZEIN = " ; Angle21 ; " DEG "
Buff = Ischarwaiting()
   If Buff = 1 Then
      D = Waitkey()
     'lcd d
Clear Serialin
     If D = 17 Then
       Print #1 , Current1
       Print #1 , Voltage1
       Print #1 , Temp1
       Print #1 , Angle11
       Print #1 , Angle21
     End If
   End If
Loop
'/////////////////This label holds the Image data///////////////
 Plaatje:
$bgf "allah.bgf"
Logo:
$bgf "LOGO.BGF"
'/////////////////This label holds the Sub Fun///////////////
Sensors:
```

```
'..... THE CURRENT
  Portd.6 = 0
W = Getadc(0)
Current = W / 1024
Current = Current * 5
Current = Current - 2.5
Current = Current / .066
Current = Current + 0.000686
Current1 = Fusing(current , "#.&&&&")
Portd.6 = 1
'..... THE VOLTAGE
W = Getadc(1)
Voltage = W * 5
Voltage = Voltage / 1024
Voltage = Voltage * 5
Voltage = Voltage + 0.024045
Voltage1 = Fusing(voltage , "#.&&&&")
'..... THE TEMPERATURE
W = Getadc(2)
Temp = W * 5
Temp = Temp / 1024
Temp = Temp * 100
Temp1 = Fusing(temp , "#.&")
'..... THE ANGLE1
W = Getadc(3)
Angle1 = W * 5
Angle1 = Angle1 / 1024
Angle1 = Angle1 * 360
Angle1 = Angle1 / 5
```

```
Angle11 = Fusing(angle1 , "#.&")

'..... THE ANGLE2
W = Getadc(4)
Angle2 = W * 5
Angle2 = Angle2 / 1024
Angle2 = Angle2 * 360
Angle2 = Angle2 / 5
Angle21 = Fusing(angle2 , "#.&")
Return
```

**C. MOTORS CONTROLLING MICRO PROGRAM:**

```
$regfile = "m32def.dat"
$crystal = 1000000
 $baud = 9600
$hwstack =
$swstack = 10
$framesize = 40
'  ------------------------------------------------------------------------
Open "comd.2:9600,8,n,1" For Input As #2
Open "comd.3:9600,8,n,1" For Output As #1
Config Serialin = Buffered , Size = 20
 Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
 Enable Interrupts
Dim W As Word , S1 As Single , S2 As Single , S3 As Single , S4 As Single
Dim Buff As Byte , D As Byte , L As Byte , S5 As Single
Declare Sub Auto
Declare Sub Oneaxies
Portb.0 = 1
Portb.1 = 1
Portb.2 = 1
Portb.3 = 1
Do
If Portc.1 = 1 Then
Set Portb.6
End If
Buff = Ischarwaiting()
  If Buff = 1 Then
       D = Waitkey()
```

```
        Clear Serialin
        L = D
   End If
 If L = 18 Then
 Gosub Auto
 Set Portb.5
 Reset Portb.6
 Reset Portb.7
 End If
 If L = 19 Then
 Gosub Auto
 Reset Portb.5
 Set Portb.6
 Reset Portb.7
  L = 0
 End If

 If L = 20 Then
 Gosub Oneaxies
 Reset Portb.5
 Reset Portb.6
 Set Portb.7
 End If
 Loop
'/////////////////This label holds the Sub Fun///////////////
Auto:
 W = Getadc(0)
 S1 = W * 5
 S1 = S1 / 1024
```

```
W = Getadc(1)
S2 = W * 5
S2 = S2 / 1024
 W = Getadc(2)
S3 = W * 5
S3 = S3 / 1024
W = Getadc(3)
S4 = W * 5
S4 = S4 / 1024

If S1 < S2 Then

 Portb.0 = 0
 Portb.1 = 0
End If
If S1 > S2 Then
 Portb.0 = 0
 Portb.1 = 1
End If
If S1 = S2 Then
 Portb.0 = 1
 Portb.1 = 1
End If
 If S3 < S4 Then

 Portb.2 = 0
 Portb.3 = 0
End If
If S3 > S4 Then
```

```
 Portb.2 = 0
 Portb.3 = 1

End If
If S3 = S4 Then
 Portb.2 = 1
 Portb.3 = 1
End If
Return
Oneaxies:
W = Getadc(0)
S1 = W * 5
S1 = S1 / 1024
W = Getadc(1)
S2 = W * 5
S2 = S2 / 1024
If S1 < S2 Then
 Portb.0 = 0
 Portb.1 = 0
End If
If S1 > S2 Then
 Portb.0 = 0
 Portb.1 = 1
End If
If S1 = S2 Then
 Portb.0 = 1
 Portb.1 = 1
End If
Return
```