Arduino Based Blind Solar Tracking Controller

Ahmed J. Abid

***Abstract*— There is no doubt that energy production of photovoltaic systems is a matter of the sun light's incident angle of the solar panel. The proposed system is offering a dual axis tracking controller, that track the sun based on astronomical equations which used time, date and location to locate the sun position. The presented system is using an Arduino Mega 2560 to find the instant sun position based on a PSA algorithm and with the aid of a real-time controller RTC and the global positioning system GPS. This system is direct the panel to the specific location using two linear actuators, the first for azimuth angle, and the other for elevation angle. Finally, the system periodically measured panel location using an accelerometer to adjust any error between the measured and the calculated angles.**

Keywords—component; solar tracker; arduino application; embedded system; astronomical equations; sun position.

I. INTRODUCTION

Single and dual solar tracking system are highly recommended for the countries that have a low solar radiation [1]. There are many designs for these solar tracking systems, but most of them tracking the sun based the light sensors that follow the sun based on the light and shadow method which are not reliable for many reasons which will describe later.

Fixed, tracking, or concentrating photovoltaic systems have a different outcome depend on the system locations around the world. As a rule of thumb, dual axis tracker is more efficient than single axis tracker and both of them are more efficient than fixed solar panel. There are many articles compared the efficiencies of the fixed flat plate systems and 1-axis and 2-axis tracking flat plate systems [2], [3].

Design of a solar tracking system must adopt some important features like; accuracy, durability, reliability, easy to maintain and cost efficient. As a literature, many tracker designed to locate the sunlight based on light sensor using dual LDRs [4], [5], but this kind of trackers is facing many problems because light sensor's measurements are affected by dust, bird droppings, clouds partially shading, or reflected light by the glass of the surrounding windows of buildings [6]. To solve such problem, designers present systems that located the sun position using microcontrollers that solve astronomical equations to locate the sun position. These systems periodically read, time and date from a real time controller, and detect location by GPS[7] , [8].

These systems detect the sun's position with high precision and did not have the light sensor problem, but it still needs a feedback sensor to detect if the solar panel at an exact location. To solve this problem some designers, used an analogue angular sensor as a feedback sensor which attached to the solar panel on both axis. This sensor has moving parts and this is a problem in a matter of durability. Other designers use the built-in encoder of the DC motor inside the linear actuators. This system counts the feedback pulses from the encoder to calculate the current location of the panel, but there is also a problem with this technique that when need to reset the linear actuator every time we reset the tracker because there is no direct way to know the panel current location.

II. ARDUINO DUTIES

The system used Arduino Mega 2560 as a main controller for this solar tracker. This microcontroller has specified duties:

A. Time and Date Update

Time and date information is so important to the system to locate the sun position for that Arduino update time and date periodically using DS1307 serial real-time clock (RTC) as shown in Fig. 1, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM which will be used later to save the tracker location. Address and data are transferred to Arduino serially through an I^2C, bidirectional bus.

B. Gathering Latitude and Longitude

According to the astronomical equations, sun position at a specific time and date is a function of location too according to the PSA algorithm. So, the system has to gather the tracker location information manually or automatically using GPS module as in Fig. 2. Arduino communicates with GPS module serially via RS232.

C. Executing the PSA Algorithm

To locate the sun position, the system adopts the PSA algorithm which calculates the sun position with high accuracy. The input to the PSA Algorithm is time, date and location. The time for the instant under consideration is given as the date (year, month, and day) and the Universal Time (hours, minutes and seconds). The location is given as the longitude and latitude of the observer in degrees[9].

The output of this algorithm consists of the main sun position angles which are the zenith and the azimuth angle. The chart in Fig. 3 shows the sequences of the PSA algorithm equations calculations. The PSA equations are briefly described by Blanco-Muriel in [9], but a brief list of the equations are presented from Eq.1 to Eq.13.

Ahmed J. Abid Author is with the Electronic Engineering Department, Middle Technical University, Zafaranya, Baghdad, Iraq (e-mail: dr.ahmedjabbar@yahoo.com).

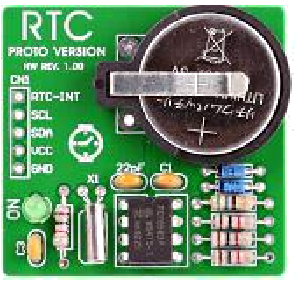


Fig. 1 DS1307 serial real-time clock

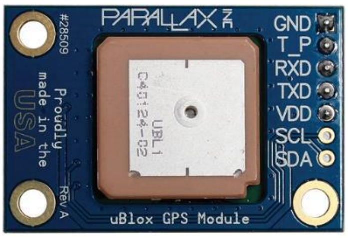


Fig. 2 GPS module

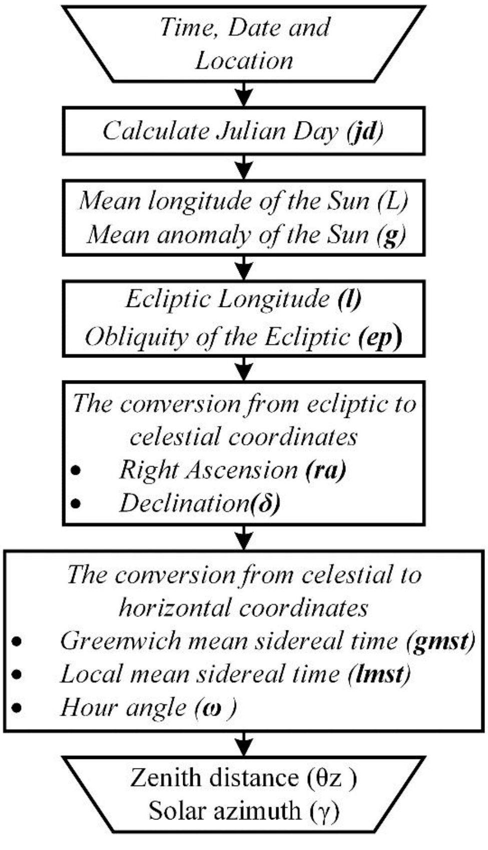

Time, Date and Location
Calculate Julian Day (jd)
Mean longitude of the Sun (L)
Mean anomaly of the Sun (g)
Ecliptic Longitude (l)
Obliquity of the Ecliptic (ep)
The conversion from ecliptic to celestial coordinates
Right Ascension (ra)
Declination(δ)
The conversion from celestial to horizontal coordinates
Greenwich mean sidereal time (gmst)
Local mean sidereal time (lmst)
Hour angle (ω)
Zenith distance (θz)
Solar azimuth (γ)

Fig. 3 Sun Position Calculation Flowchart

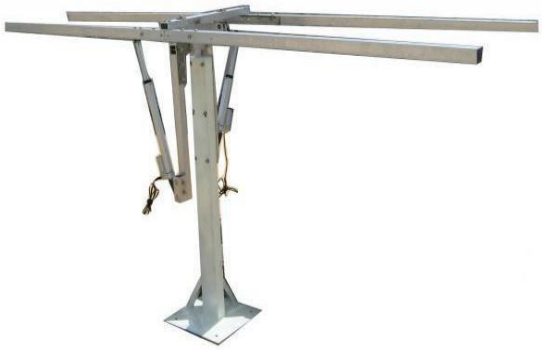


Fig. 4 Solar Tracker

Julian Day equation:

$$Jd=(1461 \times (y+4800+(m-14)/12))/4+(367\times(m-2-12\times((m-14)/12)))/12-(3\times((y+4900+(m-14)/12)/100))/4+d-32075-0.5+hour/24.0 \qquad (1)$$

Hour of the day equation:

$$n=jd-2451545.0 \qquad (2)$$

Mean longitude of the Sun Equation:

$$L=4.8950630+0.017202791698\times n \qquad (3)$$

Mean Anomaly of the Sun Equation:

$$g=6.2400600+0.0172019699\times n \qquad (4)$$

Ecliptic Longitude Equation:

$$l=L+0.033416073\times \sin(g)+0.00034894\times \sin(2g)-0.0001134-0.0000203\times \sin(\Omega) \qquad (5)$$

Where $\Omega=2.1429-0.0010394594\times n$

Obliquity of the Ecliptic Equation:

$$ep=0.4090928-6.2140\times10^{-9}\times n+0.0000396\times \cos(\Omega) \qquad (6)$$

Right Ascension Equation:

$$ra=\tan^{-1}\left[\frac{\cos(ep)\times \sin(l)}{\cos l}\right] \qquad (7)$$

Declination Equation:

$$\delta=\sin^{-1}[\sin(ep)\times\sin(l)] \qquad (8)$$

Greenwich Mean Sidereal Time Equation:

$$gmst=6.6974243242+0.0657098283\times n+hour \qquad (9)$$

Local mean sidereal time Equation:

$$lmst=(gmst\times15+Long)\times\frac{\pi}{180} \qquad (10)$$

Hour Angle

$$\omega=lmst-ra \qquad (11)$$

Zenith distance

$$\theta_z = \cos^{-1} [\cos\phi \cos\omega \cos\delta + \sin\delta \sin\phi] \tag{12}$$

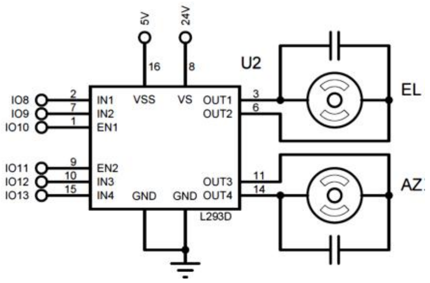
Solar Azimuth Equation:

$$\gamma = \tan^{-1} \left(\frac{-\sin\omega}{\tan\delta \cos\phi - \sin\phi \cos\omega} \right) \tag{13}$$

D. Directed the Solar Panel

The adopted tracker directed the panel via two linear actuators, one for elevation angle and the other for zenith angle by Eco-Worthy company as shown in Fig. 4. Arduino control driving the two actuators' speed and directions using the L293D model for two reasons, the first reason, it can move the motor in both directions which it required to bring the panel back in the next morning.

The second reason is that speed can be controlled easy which will be used to increase the speed of moving the panel in fast if the difference between the current and desired angle is too wide. Linear actuator size and power is a matter of the static and dynamic loads. Static load is represented by the panel weight and the dynamic load represented by wind speed. L293D drive model is shown in Fig. 5a and the circuit diagram in Fig. 5b.



(a) Driver module

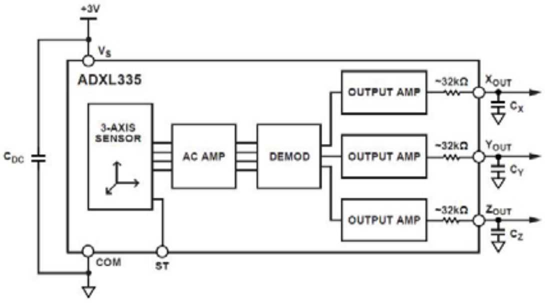
(b) Driver connection

Fig. 5 Driving circuit

E. Check the Panel Position

While the panel moving to face the sun, Arduino checks the panel position continually using an accelerometer. The microcontroller compared the current and required position of the panel then correct the panel angles using the actuators. The used sensor is ADXL335 which is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. It can measure the static acceleration of gravity in tilt-sensing applications. Three ADC pins used to measure XOUT, YOUT and ZOUT. ADXL335 block diagram and its model are shown in Fig. 6a and Fig. 6b respectively.

(a) Accelerometer Model



(b) Block diagram

Fig. 6 ADXL335 Sensor

III. System Management

Arduino manages the entire system with specific repeated steps as listed below:

RTC Unit

This unit responsible on update time and date using an integrated circuit (DS1307), and manage and save them even with losing power because it has a power backup (3V battery). At the starting of the tracker, system will display the current time and date and offer update as required. If the user select "N" the system will not update the time, but in case if select "Y" the system will ask for the time and date as shown in Fig. 7.

GPS Unit

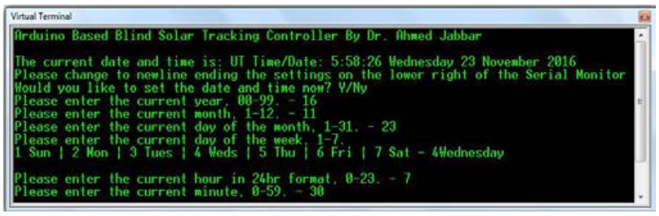
This unit responsible on gather location latitude and longitude with three options.

1) Option 1: Keep the default location which can be saved according to the specific location for now its Iraq, Baghdad (Latitude :33.312806N, Longitude= 44.361488E) as shown in Fig. 8.
2) Option 2: Change the default location manually by entering the new latitude and longitude as shown in Fig. 9.
3) Option 3: Using GPS to gather the current location as shown in Fig. 10.

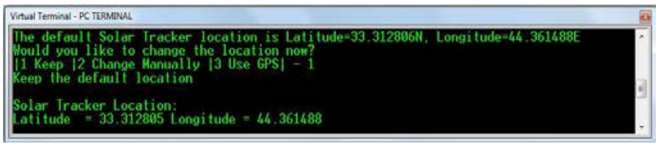
IV. Angles Calculations

Arduino calculates the sun position angles (Elevation and Azimuth) based on the PSA algorithm with help of RTC and

GPS. After that it reads the panel angles (Phi and Theta) using accelerometer, then calculates different between the two angles (deltaEle & deltaAz). Finally, its ready to drive the panel to correct location using the actuators and based on the measured different value and sign. All angles will send to PC as shown in Fig. 11.



```
Arduino Based Blind Solar Tracking Controller By Dr. Ahmed Jabbar

The current date and time is: UT Time/Date: 5:58:26 Wednesday 23 November 2016
Please change to newline ending the settings on the lower right of the Serial Monitor
Would you like to set the date and time now? Y/Ny
Please enter the current year, 00-99. - 16
Please enter the current month, 1-12. - 11
Please enter the current day of the month, 1-31. - 23
Please enter the current day of the week, 1-7.
1 Sun | 2 Mon | 3 Tues | 4 Weds | 5 Thu | 6 Fri | 7 Sat - 4Wednesday

Please enter the current hour in 24hr format, 0-23. - 7
Please enter the current minute, 0-59. - 30
```



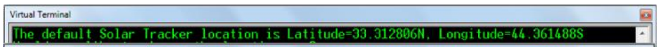
```
The default Solar Tracker location is Latitude=33.312806N, Longitude=44.361488E
Would you like to change the location now?
|1 Keep |2 Change Manually |3 Use GPS| - 1
Keep the default location

Solar Tracker Location:
Latitude  = 33.312805 Longitude = 44.361488
```

Fig. 7 Offer to update time and date

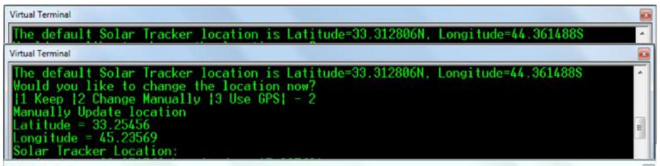
Fig. 8 GPS option 1

Fig. 9 GPS option 2

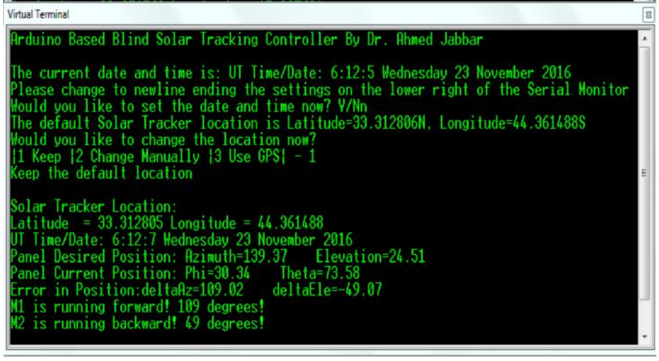
Fig. 10 GPS Option 3



```
The default Solar Tracker location is Latitude=33.312806N, Longitude=44.361488S
```



```
The default Solar Tracker location is Latitude=33.312806N, Longitude=44.361488S
Would you like to change the location now?
|1 Keep |2 Change Manually |3 Use GPS| - 2
Manually Update location
Latitude = 33.25456
Longitude = 45.23569
Solar Tracker Location:
```



```
Arduino Based Blind Solar Tracking Controller By Dr. Ahmed Jabbar

The current date and time is: UT Time/Date: 6:12:5 Wednesday 23 November 2016
Please change to newline ending the settings on the lower right of the Serial Monitor
Would you like to set the date and time now? Y/Nn
The default Solar Tracker location is Latitude=33.312806N, Longitude=44.361488S
Would you like to change the location now?
|1 Keep |2 Change Manually |3 Use GPS| - 1
Keep the default location

Solar Tracker Location:
Latitude  = 33.312805 Longitude = 44.361488
UT Time/Date: 6:12:7 Wednesday 23 November 2016
Panel Desired Position: Azimuth=139.37    Elevation=24.51
Panel Current Position: Phi=30.34    Theta=73.58
Error in Position:deltaAz=109.02    deltaEle=-49.07
M1 is running forward! 109 degrees!
M2 is running backward! 49 degrees!
```

Fig. 11 Angles measurements

V. Circuit Design and Simulation

The system is mainly controlled by Arduino Mega 2560 which has a ATMEGA2560 microcontroller. This microcontroller has a greater ability to manage a highly precision number and trigonometric functions, but the main reason to choose this type is the high endurance non-volatile memory segments, 4Kbytes EEPROM and 8Kbytes Internal SRAM which can manage the presented software. Fig. 12 shows the controller circuitry, Arduino receive the data from the GPS via RS232, communicate with Real Time Controller via I2C bus, reads accelerometer signals (XOUT, YOUT, ZOUT) as three analog signals, and finally drives two linear actuators using L293D to direct the solar panel into the required azimuth and elevation angles.

VI. Result and Discussion

The system tested by simulating at many locations, times and dates. Test 1 adopts a specific location (Baghdad 33.312806N, 44.361488E), specific date (21st Mar 2016), but at different time of day (one hour step) as presented in Table I. Panel location is a matter of time between adjustments, sun move 1 degree /4 mins. So, with two degrees of accuracy, solar tracker automatically adjusted every 8 mins.

TABLE I Elevation and Azimuth Angles for Test 1.

Local Time[a]	Desired location (Deg.)		Panel Location (Deg.)	
	Elevation	Azimuth	Phi	Theta
6:00 AM	<5		Stop Tracking	
7:00 AM	10.98	96.86	10	96
8:00 AM	23.08	105.72	22	104
9:00 AM	34.77	116.47	34	116
10:00 AM	45.46	130.90	44	130
11:00 AM	53.38	150.30	52	150
12:00 AM	57.09	175.91	56	174
13:00 AM	55.07	202.47	54	202
14:00 AM	48.05	224.32	48	224
15:00 AM	38.22	239.75	38	238
16.00 AM	26.71	251.44	26	250
17:00 AM	14.76	260.72	14	260
18:00 AM	<5		Stop Tracking	

[a] Location (Baghdad 33.312806N, 44.361488E), date (21st Mar 2016)

Test 2 adopts a specific Time and Date (12:00 PM GMT, 21st Mar 2016) for different cities around the world as listed in *Table II*.

TABLE II Elevation and Azimuth Angles for Test 2.

City	Local Time[b]	Tracker Location (Deg.)		Sun Position (Deg.)	
		Latitude	Longitude	Elevation	Azimuth
Hawaii	2:00 AM	+19.89676	-155.58278	<0	
Santiago	8:00 AM	-33.43783	-70.65044	14.55	79.48
Salvador	9:00 AM	+13.79418	-88.89653	3.04	90.21
Azores	11:00 AM	+37.74124	-25.67559	46.69	143.55
London	12:00 AM	+51.50735	-0.127758	39.01	181.63
Paris	1:00 PM	+48.85661	+2.35222	41.59	184.34
Cairo	2:00 PM	+30.04442	+31.23571	47.76	231.59
Baghdad	3:00 PM	+33.31280	+44.36148	38.22	239.75
Baku	4:00 PM	+40.40926	+49.86709	30.34	241.02
Islamabad	5:00 PM	+33.72938	+73.09314	14.72	260.55
Sydney	10:00 PM	-33.86882	+151.20929	<0	

[b] At 12:00 PM GMT, 21st Mar 2016

Test 3 adopts a specific local time (12:00 PM) on 21st Mar 2016 for different cities around the world as listed in *Table III*.

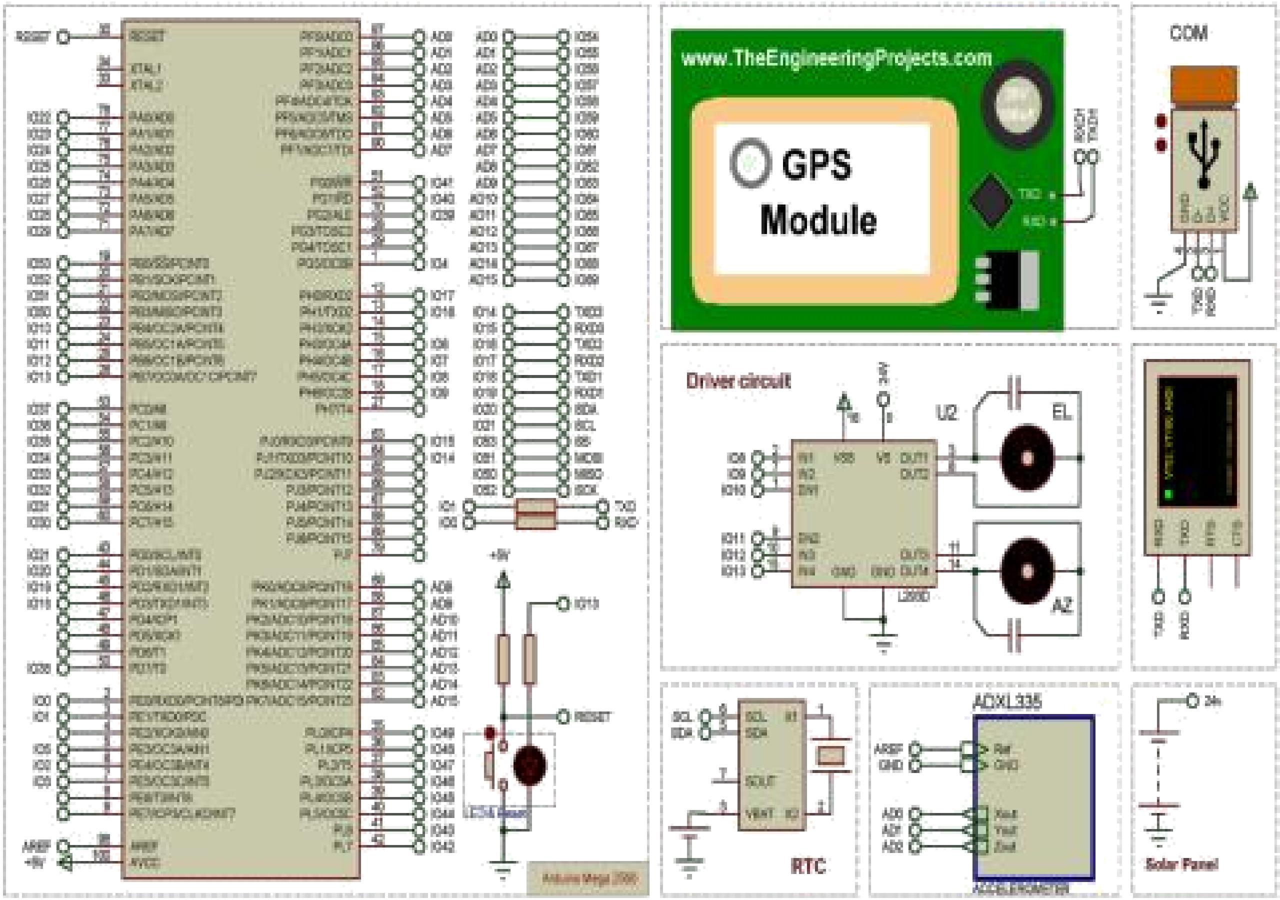


Fig. 12 Controller System Circuitry

TABLE III ELEVATION AND AZIMUTH ANGLES FOR TEST 3.

City	Universal Time[c]	Tracker Location (Deg.)		Sun Position (Deg.)	
		Latitude	Longitude	Elevation	Azimuth
Hawaii	10:00 PM	+19.89676	-155.582	69.56	159.14
Santiago	4:00 PM	-33.43783	-70.6504	54.07	21.27
Salvador	3:00 PM	+13.79418	-88.8965	43.23	102.52
Azores	1:00 PM	+37.74124	-25.6755	51.10	160.12
London	12:00 AM	+51.50735	-0.12775	39.00	178.12
Paris	11:00 AM	+48.85661	+2.3522	40.12	161.24
Cairo	10:00 AM	+30.04442	+31.2357	60.44	179.23
Baghdad	9:00 AM	+33.31280	+44.3614	57.09	175.91
Baku	8:00 AM	+40.40926	+49.8670	48.66	162.19
Islamabad	7:00 AM	+33.72938	+73.0931	56.55	173.65
Sydney	2:00 AM	-33.86882	+151.209	55.78	0.58

[c.] Local time (12:00 PM) on 21st Mar 2016

VII. CONCLUSION

The presented blind solar tracking controller offers a reliable, precise and efficient system. It's designed to work with different climate and at any location around the world. The offered feedback system gave the controller a stability and high accuracy in directing the solar panel to the sun location which increase the produced energy. System follows a strategy to manage the controller can be described briefly by Fig. 13.

Table I shows a precise data to position the sun in Baghdad on 21st Mar 2016. It's also shows the panel locations with one hour step sample. *Table II* presents the calculated sun position at difference cites. The selected cities chosen to cover NE (Baghdad), SE (Sydney), GMT (London), NW (Hawaii) and SW (Santiago) of the world at a specific date and a specific time (12:00 PM) and the results shows how the system manages different cities at specific time.

Finally, *Table III* shows the sun position for the same selected cities, but at a specific local time (12:00 PM). All the presented data show an accurate system to locate the sun at different regions on the earth.

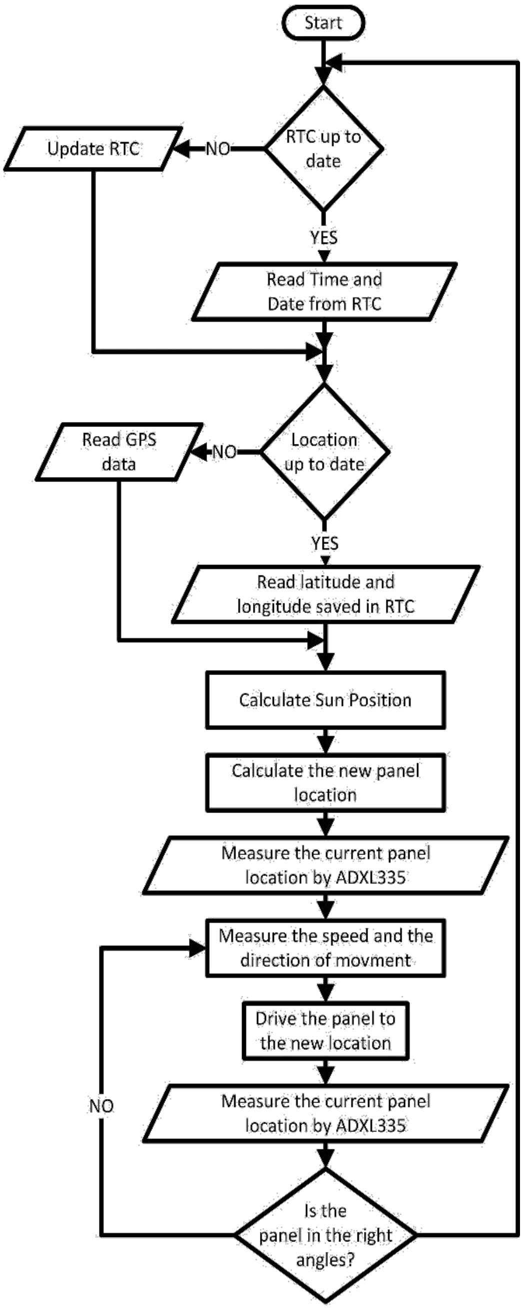

Start
RTC up to date
NO
Update RTC
YES
Read Time and Date from RTC
Location up to date
NO
Read GPS data
YES
Read latitude and longitude saved in RTC
Calculate Sun Position
Calculate the new panel location
Measure the current panel location by ADXL335
Measure the speed and the direction of movment
Drive the panel to the new location
Measure the current panel location by ADXL335
NO
Is the panel in the right angles?

Fig. 13 Tracking strategy

REFERENCES

[1] 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,” Int. J. Photoenergy, vol. 2013, 2013.

[2] F. J. Gómez-Gil, X. Wang, and A. Barnett, “Energy production of photovoltaic systems: Fixed, tracking,

and concentrating,” Renew. Sustain. Energy Rev., vol. 16, no. 1, pp. 306–313, 2012.

[3] R. Dhanabal, V. Bharathi, R. Ranjitha, A. Ponni, S. Deepthi, and P. Mageshkannan, “Comparison of efficiencies of solar tracker systems with static panel single-axis tracking system and dual-axis tracking system with fixed mount,” Int. J. Eng. Technol., vol. 5, no. 2, pp. 1925–1933, 2013.

[4] J. M. Wang and C. L. Lu, “Design and implementation of a sun tracker with a dual-axis single motor for an optical sensor-based photovoltaic system,” Sensors (Switzerland), vol. 13, no. 3, pp. 3157–3168, 2013.

[5] A. Ponniran, A. Hashim, and H. Ali Munir, “A design of single axis sun tracking system,” in 2011 5th International Power Engineering and Optimization Conference, PEOCO 2011 - Program and Abstracts, 2011, pp. 107–110.

[6] F. M. Al-Naima, R. S. Ali, and A. J. Abid, “Solar Tracking System: Design based on GPS and Astronomical Equations,” IT-DREPS Conf. Exhib., pp. 1–6, 2013.

[7] F. M. Al-naima, R. S. Ali, and A. J. Abid, “Design and Implementation of a Smart Dual Axis Sun Tracker Based on Astronomical Equations,” vol. 1, pp. 1–6, 2012.

[8] Y. Rizal, S. H. Wibowo, and Feriyadi, “Application of solar position algorithm for sun-tracking system,” in Energy Procedia, 2013, vol. 32, pp. 160–165.

[9] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, and M. Lara-Coira, “Computing the solar vector,” Sol. Energy, vol. 70, no. 5, pp. 431–441, 2001.