ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

Development of Automated Solar Tracker for High Efficiency Solar Panels

Benny Cherian[1], Lakshmi P[2], Linda Joseph[3], , Madhu PM[4], ,Minu John[5]

Professor, Dept. of EEE, Mar Athanasius College of Engineering, Kothamangalam[1]
UG Student, Dept. of EEE, Mar Athanasius College of Engineering, Kothamangalam, India[2,3,4,5]

ABSTRACT: A Solar Tracker is a device onto which solar panels are fitted which tracks the motion of the sun ensuring that maximum amount of sunlight strikes the panels throughout the day. Power output from a solar cell will be maximum when it is facing the sun i.e. the angle between its surface and sun rays is 90 degree. The components used for its construction are servo motor, microcontroller chip Atmega 328 and LDR. The active sensors constantly monitors the sunlight and rotate the panel towards the direction where the intensity of sunlight is maximum.

Keywords*:* Solar Tracker, Light Detecting Resistor, Arduino, Atmega 328, Servo Motor

I. INTRODUCTION

Solar energy is rapidly gaining importance as an energy resource due to its versatility in terms of renewability, cost effectiveness, improved operating efficiency and its unlimited reserve, the sun. Solar panels, being the modern tech of alternative power source, aims at absorbing maximum amount of sunlight and converting into useable electricity and thus achieving increasing popularity since the realization of fossil fuel shortcomings.

Our project includes design and construction of a microcontroller based single axis solar tracker. When light falls on the LDRs, ,its resistance varies and a potential divider circuit is used to obtain corresponding voltage value from the resistance of LDR. This voltage signal is send to the microcontroller. Based on the voltage signal, a corresponding PWM signal is send to the servo motor which causes it to rotate and finally attains a position where intensity of light falls on the solar panel is maximum.

II. CIRCUIT DESIGN

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

Features:

Microcontroller	Atmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 Ma
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

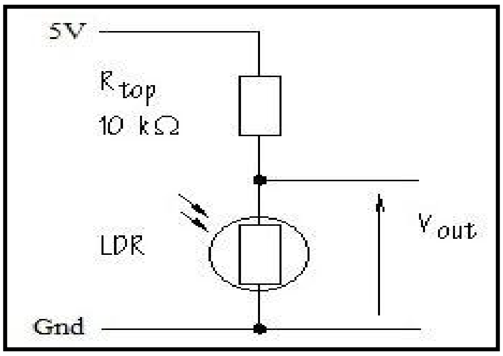
ATMEGA 328 : Atmega 328 is a high-performance Atmel 8-bit AVR RISC-based microcontroller. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic and Logic Unit(ALU) , allowing two independent registers to be accessed in one single instruction executed in one single clock cycle.

Atmega 328 provides the following features : 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timers/Counters, USART, 2-wire Serial interface, SPI port and interrupt system to continue functioning. The Power Down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip ISP Flash allows the program memory to be reprogrammed In- system through an SPI serial interface, by a conventional non-volatile memory programmer or by an on-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self Programmable Flash on a monolithic chip, Atmega328 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

Servo Motor : A servomotor is a rotary actuator that allows for precise control of angular position. It uses a feedback mechanism to maintain its shaft position with minimum error. The servo motor has three wires. Among those 3 wires, 2 wires provide power supply to the motor and one is used to send signals to the motor for required shaft positions. These signals are PWM signal of varying duty cycle. Servo shaft is rotated by an angle depending on the ON-Time and Off-time of the pulse.

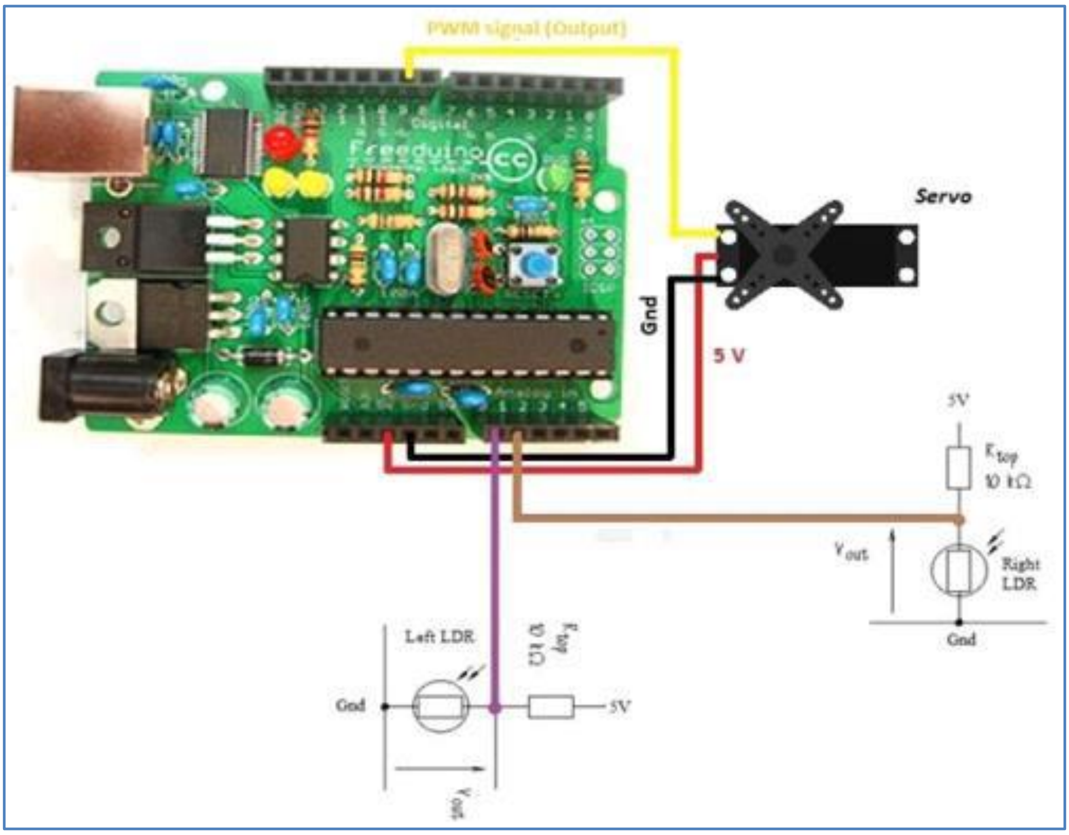
Sensor : LDRs(Light Dependent Resistor) are used as sensors. LDR is a variable resistor whose resistance is inversely proportional to the intensity of the incident light. As it is a passive transducer, a potential divider circuit is used to obtain the corresponding voltage value from the LDRs. The higher the intensity of light, lower the LDR resistance and hence lower the Output voltage (Vout) and vice versa.



INPUT(ADC): Arduino has an inbuilt 10-bit Analog to Digital converter(ADC), hence it can provide Digital values from 0-1023.

OUTPUT(PWM) : Arduino has a 8-bit PWM generator, which generates the PWM signal based on the required servo angle, that controls the rotation of the servo. First an initial angle is set for servo arm and then it is incremented or decremented according to the output from LDR.

III. IMPLEMENTATION OF THE CIRCUIT



In this project the angles are calculated by finding which LDR is in shadow. For eg: If light source is right w.r.t. to the platform then Right LDR will receive maximum light and left LDR will be in shadow. So we'll use If statement to see which LDR is in shadow and rotate the platform in opposite direction so that it's again facing the source. In above example, the platform will be rotated in right direction.

The program has two parts namely Setup part and Loop part. Setup part is executed only once and loop part is executed again and again. So first the sensors are calibrated for light and dark values. Since calibration is required to be done only once the calibration code is put in the setup part.

Void Setup :

1. Initialize the servo and write 90 to it. When shaft turns to 90 degrees then immediately attach the movable platform to servo and platform should be perfect horizontal and facing the light source, now during tracking, the platform can move
+/-90^0 on both sides. Then write 750 to the servo, the platform will turn through -150 from normal. At this point the values of both sensors are read and saved in two separate variables(x1& y1). When platform is 150 from normal (or 750 from horizontal) then sensor 1 is in shadow and other one is receiving maximum light.

2. Then write 1050 to the servo, now the servo will go +150 from normal and sensor 2 is in shadow and 1st one is receiving max light. Again the values are sampled and saved in another set of variables(x2&y2).

3. Now the variables x1, x2 & y1, y2 will hold values of light and shadow for both sensors and using that values averages are calculated and it is stored in another set of variables.
X=(x1+x2)/2.
Y=(y1+y2)/2.

4. X and Y are declared as global variables i.e. they will be initialized before void_setup part and can be used in both setup and loop part.

- Void_Loop :

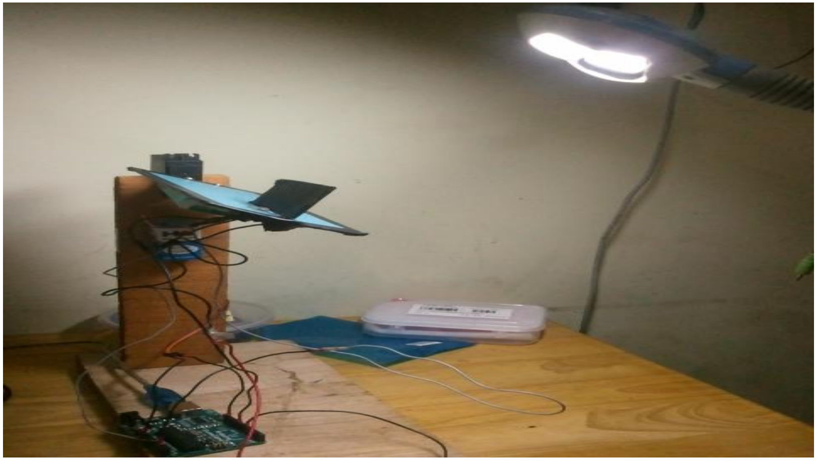
1. The sensors are read .

2. The values are compared with the average value.

3. If value read from a sensor is greater than the average value then it means that sensor is in shadow and we will increment servo angles until the value being read becomes less than average value.

4. Same goes for other sensor, if it is in shadow then the servo angles are decremented until sensor value becomes less than average value.

5. Finally write the corrected angle values to servo.



IV. CONCLUSION

As of now, the use of solar energy is increasing rapidly due to rising fuel prices and electricity crisis. A single axis solar tracker (prototype) for obtaining maximum efficiency from solar energy has been made. But there will be slight variation in the desired alignment of the solar tracker in the presence of multiple light sources. Still this will not affect the smooth working of the solar tracker in applications. We hope our prototype device will be helpful for more improved versions

ACKNOWLEDGMENT

We hereby express our sincere gratitude to the Head of the Department of Electrical And Electronics Engineering, Prof. K. Radhakrishnan for providing us with the necessary arrangement for the completion of our project. We also thank our faculty advisor Prof. Elizabeth Sebastian for her valuable guidance. We express our sincere gratitude to our guide Ms. Sini Paul, Department of Electrical And Electronics Engineering, without whose valuable guidance and support the project would not have been a success. We thank her for the good will and encouragement extended to us.

REFERENCES

[1] Brian Evans, “Beginning Arduino Programming”, Apress Publishers, 2011
[2] Benjamin Reh, “An Introduction to programming an Atmega microcontroller”, November 14, 2013