

Solar Panel Automatic Adjustment

Course project for Advanced Computer Systems (fall 2017)

Submitted on Mar 27, 2018

By Michal Shagam, 201144607

Abstract:

The project goal is to create an automatically adjusting solar panel. While in most cases, solar panel trackers use separate LDRs to find the maximizing angle of light absorption, here I will use the output from the solar panel itself to find this angle. The use of the CC1350 launchpad requires getting acquainted with the TI API and the launchpad interfaces.

Introduction:

We often see plants, such as sunflowers, readjust their position and angle in order to maximize the sun radiation capture. This inspired the idea to adjust the angle of solar panels instead of keeping the panels immobile.

The photoelectric effect occurs when photons with enough energy are absorbed by material and electrons surpass the binding energy and are released. Solar panels or photovoltaic cells use this effect to produce electricity from solar energy. Their electronic efficiency is usually about 15%. There are several types of solar panels, three of them are common in home use: monolythic, polylythic and thin film. While polylythic crystalline solar cells result in less wasted silicon in the production process, monolythic crystalline solar cells are considered more efficient. Combining several monolythic crystalline solar cells can increase the efficiency even more. Thin film is also mostly less efficient and takes up more space than the monolythic solar panels but it is cheaper to manufacture.

Project Supplies:

- CC1350 launchpad
- Solar panels (3 watt, 7V output)
- Servo motor + separate voltage supply
- Breadboard
- 10 Ω resistors (3 watt)
- Multimeter
- Logic analyzer
- Battery box
- Molex connector or simply cutting off the old connector and soldering new wires.

The logic analyzer helped in the debugging process greatly. It allowed me to check the PWM output when there were problems with the motor. It also helped me validate the pin output configuration.

The Multimeter was used in the measurements of the different parts and in checking for faulty wires and connections.

Process of obtaining supplies:

Choosing the vendor was a very lengthy process. I made several mistakes and ordered parts that took a very long time to arrive. In some cases, the company changed the shipping information which led to extra shipping charges and much anxiety.

The vendor list included:

4Project – a supply shop located in Yehud which allows same day pick-ups and the taxes are included in the price. The variety and the overall experience were excellent. They have excellent customer service.

<https://www.4project.co.il/>

D.A.N. electronics – an electronics store located in Petah Tikva. The variety and the overall experience were excellent but they are closed on Fridays. There is another electronics shop in the nearby building.

<http://www.dash.co.il/index.php?route=information/contact>

digikey – their prices are reasonable, they have a large supply and variety of items and their shipping is fast. When ordering from them, you must use a home address, otherwise they will replace the name of the purchaser which will double the shipping charges and the items might not arrive since they are under a different name. They have horrible customer service.

<https://www.digikey.co.il>

dealextreme – late shipping but the same items that arrived from digikey required taxes as well so if you can wait, this is the cheaper option.

Choice of supplies:

Motor type: A servo motor was chosen since its range of angles (within a sensible price range) is much larger than that of a stepper motor and much easier to control.

The servo motor has three input wires: GND, VCC and SIGNAL. The servo motor is controlled using the pwm peripheral, connected to the signal wire, where the length of the pulse within the period determines the angle. A stepper motor needs to be moved by calculating the correct two coils to power. The servo motor control method is a much more natural method for sweeping through several angles. The typical servo motor is usually perfect for most home projects. They work with a power supply that ranges between 3V and 6V. The speed and torque are affected by this power supply.

Another important aspect of the project was the fit between the motor and the solar panels in terms of strength, speed and torque. It was useful to know that these motors are also commonly used in many

robots and model aircrafts. This gave me freedom when choosing the solar panels in terms of weight and size.

Solar panel type: As explained in the introduction, there are several types of solar panels to choose from. In terms of prices, for panels of about 6" by 4", there isn't a great difference between the different solar panels. The price difference will probably matter more when looking at larger panels.

I purchased monolythic and polylythic solar panels along with a micro photovoltaic cell. I chose to use the monolythic solar panels since their output was closer to the ADC input range and being 3watt panels, were easier to regulate with resistors.

Voltage regulation:

The solar panel used supplied ~5V with a 3 watt limit. The ADC input is limited to 3V based on the documentation. Since this project mainly required comparing the voltage at different angles, the solution of a voltage divider made the most sense. Using the voltage divider to lower the voltage to about half of the original would give a voltage within the ADC range limit. This process does not affect the comparisons performed since the algorithm described below uses the standard deviation to compare two values. This standard deviation is about a 0.1% relative error and would not be affected greatly by the voltage transformation.

The resistance used to regulate the ADC input voltage was chosen based on the solar panel power supply, the inner resistance and the ADC input limit. The solar panel output current required using specific resistors. Typical resistors can handle less than 0.5 watt and therefore won't due for a 3 watt solar panel. I bought 3 watt resistors which are bulkier and can handle a much larger current/power than typical resistors.

Process:

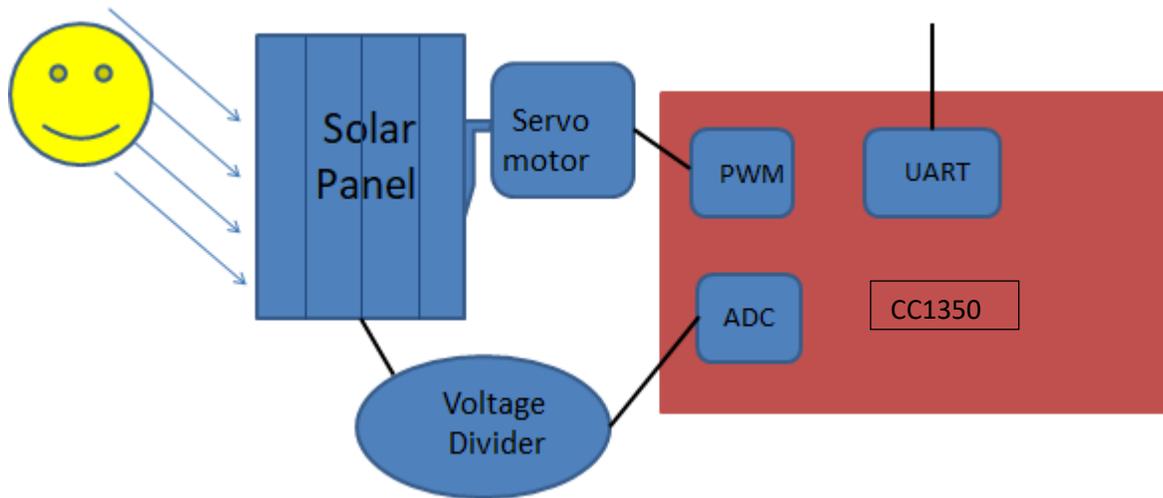
Process Outline:

The process began with understanding exactly what supplies were needed and what the requirements for each part should be. This involved researching typical solar panels and servo motors and understanding which type of voltage regulation I would need and why. During each step, I referred to the launchpad documentation and made sure to understand the peripherals used. This included understanding their limitations.

The next stage was checking that the supplies and parts ordered fit the spec sheet. In some cases, I had to consider whether the differences between the product and the spec sheet would interfere with the actual performance and whether the differences could cause damage to the board as well.

After checking that the peripherals and parts were controlled and working correctly, I combined the separate parts of the code and wrote the algorithm that detects the maximum absorption angle.

Project Sketch:



The Algorithm:

After initiating the ADC, UART and PWM peripherals, the servo motor (attached to the solar panel) will sweep through the different angles. At every angle, the motor will stop and check the output of the solar panel using the ADC. For each angle, many samples are taken and are used to also compute the standard deviation. If the value is much larger than the previous, the new angle is chosen as the maximum. A value is considered much larger than another value if it is larger than the second value by more than the combined error of the two standard deviations. If a maximum angle is not found, the range of the two largest angles is taken as the maximum range and the median angle is chosen as the maximum.

The combined error of the two standard deviations is taken as a combination of two independent errors:

$$\Delta_{comb} = \sqrt{\Delta_1^2 + \Delta_2^2}$$

Supply check:

As explained above, the parts often differ somewhat from their spec sheets. This means that each part must be inspected before connecting it to the board or logic analyzer.

The resistors were expected to have a resistance of 10Ω but the average of the actual resistance measured was 17Ω . The resistors should have had a small deviation (accuracy color - gold). The standard deviation of the measurements was 1.6Ω . Since the difference is much larger than 3 times as much, the likelihood of this deviation is highly unlikely – less than 99.7%.

The multimeter battery was replaced and the multimeter was also checked against other resistors to make sure it was working properly.

If we assume that the multimeter used to measure the resistance has a relative error of two percent for the scale used (200Ω), we would have an error of about 4Ω . Combining the two errors, we get an error of 4.3 and then the deviation is within 3 standard deviations. It is safe to assume that the multimeter error on this small scale is probably larger.

The servo motors receive the input from the board and are therefore unlikely to cause problems. In order to check that the motor was working properly, I connected the motor and supplying it with power for a limited time will move the motor (even when the signal input is not configured). I then wrote a simple sweep program and made sure the two angles were reached. The speed in which the motor moves was inspected and a safer longer period of time was taken as the wait period between ADC measurements in the program allowing the motor to reach the expected angles in time.

The solar panel inspection consisted of first finding the expected voltage ranges. In daylight, the solar panel supplied a little over 5V and surprisingly in indoor lighting only, the solar panel provided an output of about 1V. The maximum voltage in the spec sheet was 7V.

The ADC input was limited to 3V. In order to adapt the solar panel output, I had to adjust the voltage of the solar panel output. In order to do this, I had to measure the inner resistance. Since the photovoltaic cell is itself a voltage supply, one cannot simply measure the resistance with a multimeter as even the slightest of input voltage can affect the measured resistance greatly.

Three methods were used to measure the inner resistance:

The simplest way to calculate the inner resistance would be to base it on the solar panel spec sheet. Using the power and the maximum voltage (or the voltage we can measure), we can compute the inner resistance.

$$P = V \cdot I = \frac{V^2}{R_{inner}} \rightarrow R_{inner} = \frac{V^2}{P}$$

While this method gives a great ballpark, we can't always rely on the spec sheet. We can however calculate this ballpark and, while trying to reach the maximum output voltage, see whether a resistor with a known resistance lowers the voltage as expected to verify the inner resistance we calculated.

The second method was extrapolation. Measuring the voltage and current through several resistors and extrapolating the inner resistance resulted in a much closer value to the expected inner resistance.

The last method was used mainly to show that it is not a dependable method and can't be used. Trying to limit the light in the room until the voltage supply is almost zero and measuring the resistance. I managed to lower the voltage to about 6.3 mA and the resistance measurement was very high, $200k\Omega$. This method clearly did not work out. Trying again to lower this voltage even more, I shifted the

computer screen away from the solar panel which cut the voltage in half. Again the resistance was measured and this time the resistance was about 10 times as much as the expected inner resistance.

The inner resistance was about 14Ω . Using a 10Ω resistor for voltage division would halve the output voltage as we wish.

Debugging and output:

In order to make sure that certain peripherals, such as the ADC, were configured and working correctly, I needed a more definitive form of output and monitoring than the LEDs. Using the UART interface to send back information was the monitoring method chosen.

The output was configured based on the echo UART project (the project notes helped me greatly). When the board is attached to the computer by USB, the device manager recognizes a UART COM port I can connect to in order to read and send bytes (XDS110 Class Application/User UART).

Configuration of the ADC:

The relevant pins and input limitations were found. The input is limited to 0-3.3V. Using the ADC single channel project, I configured the pins. To test whether the converter was configured correctly, the board's ground and the 3V pin were connected to two of the ADC's input pins.

The ADC output values have a variance which was different for the two inputs. These measured values were used to compute both the bias error and the standard deviation error. The bias error was less relevant for the algorithm itself since it mostly used a pairwise comparison. The standard deviation is however very relevant since it can affect the pairwise comparison of two close values.

The standard deviation: $1200\mu V$.

The bias error was computed based on the average of the values measured for 0V and the standard deviation.

While these values can be useful as ballparks and in future projects (where the standard deviation should be measured for the middle of the range), the algorithm simply measured the converted output several times and computed the standard deviation error for each angle. A good reason to use the ballpark standard deviations would be to save time.

Configuration of the PWM peripheral:

This part was very straight forward as we worked with the peripheral in the beginning of the course.

Miscellaneous issues and notes:

Energia is an open source project which implements an Arduino API for the TI boards. Some of the code was written in this API as well as the TI API. While the energia API is much more straight forward and

easier to use, it also takes up much more program memory. A simple angle sweeper (written to make sure the motors were working properly) takes up more than a quarter of the memory.

The TI API version took up about 6kB while the simplest of the programs with the energia API took up about 27kB of the program memory. This is a very large difference and could mean that since the program space is already limited, we would reach this limit very easily with the energia API.

The 5V pin outputs 5V as expected. The current it outputs was measured using a small resistor. This current wasn't strong enough for the servo motor and a separate supply was used. Alternatively, a solar panel would be the perfect replacement in a production version. This was not used here to simplify the ability to work on the project outside daylight hours.

The voltage divider must take into consideration the high current produced. This means that we can't simply use any resistor.

Bibliography:

<https://phys.org/news/2016-05-sunflowers-track-sun.html>

https://en.wikipedia.org/wiki/Photoelectric_effect

<http://energyinformative.org/best-solar-panel-monocrystalline-polycrystalline-thin-film/>

<http://nepsi.com/resources/calculators/motor-torque.htm>

<http://simplemotor.com/calculations/>

<https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>