

Analog Solar Tracker Project

by Mark Harris

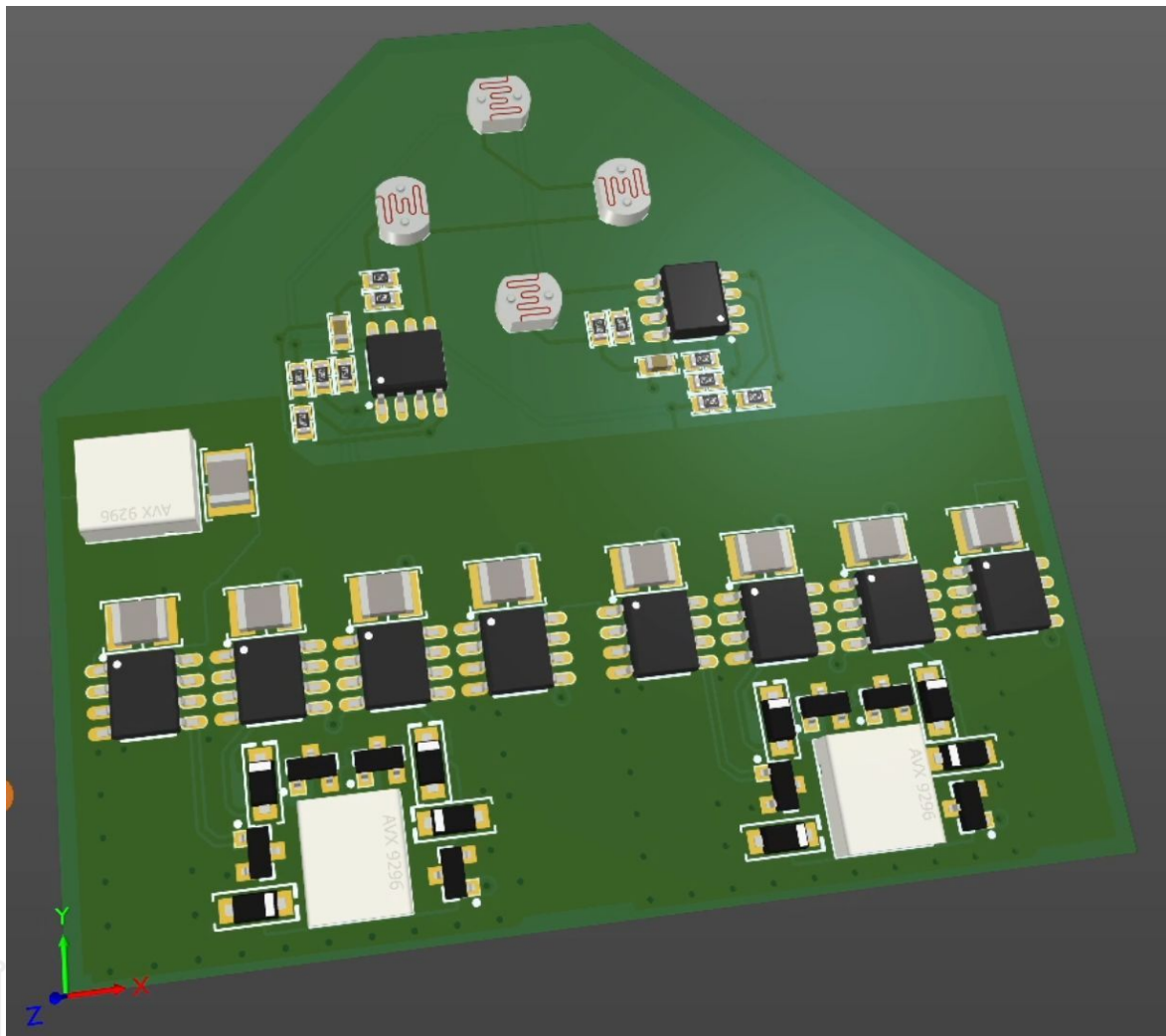


Table Of Contents

What is a Solar Tracker?	4
Sensing Light	6
Window Comparator - Using Sensed Light	12
Driving The Motors	15
Bringing It All Together	18
PCB Layout	21
Adding a Light Shield	33
Further Improvements	35
Conclusion	38

Analog Solar Tracker Project | Altium Designer

In this project we're going to build an analog solar tracker, which will control two DC motors that move a solar panel to keep it facing towards the sun. The solar tracker will be built using a window comparator circuit, which is a very useful circuit even in the age of ultra low cost microcontrollers. The window comparator will be controlling a frankly rather overbuilt H-bridge - I'm only prototyping this circuit with 3V 100mA gear motors, however overbuilding the H-bridge design allows you to scale up the tracker circuit for your own applications which could be using much larger motors, just by switching out the MOSFETs.



Analog Solar Tracker Project | Altium Designer

As always, this project is completely open-source under the MIT license which allows you to freely use any part of the design for any application commercial or otherwise. You can find the Altium Designer project files for the project hosted at [GitHub](#).



01

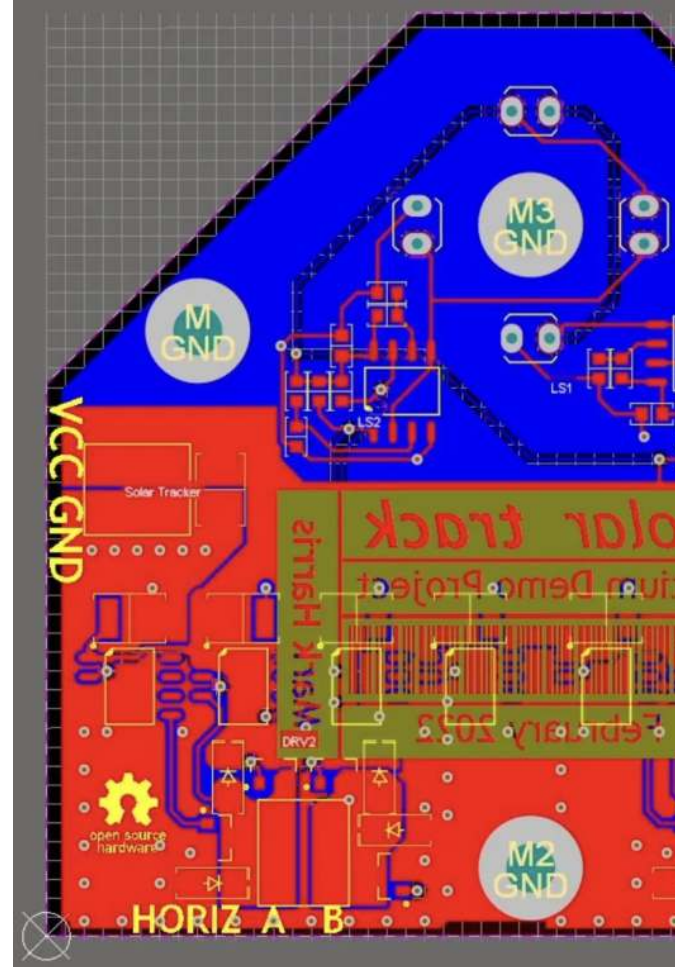
What is a Solar Tracker?



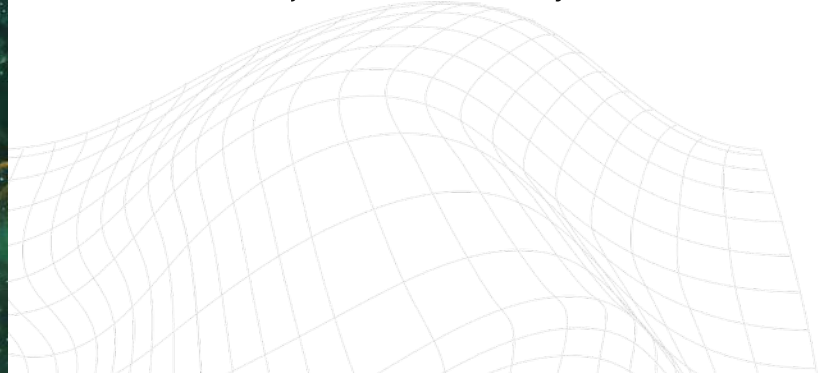
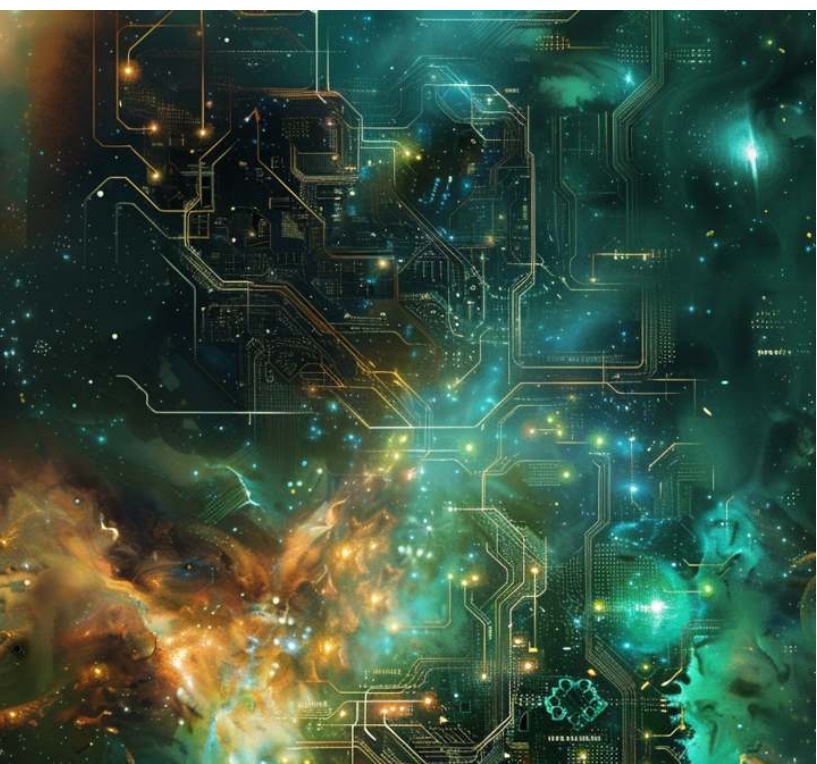
Analog Solar Tracker Project | Altium Designer

A solar tracker detects light and attempts to keep the sensor facing the brightest light source. In this project, we're building a two-axis solar tracker which can track the sun as it moves across the sky. By keeping a solar panel pointing directly at the sun, the maximum amount of energy can be harvested.

A solar panel that tracks the sun is particularly relevant to people living far from the equator. Where I live, in summer the sun rises almost directly north and sets twenty-three hours later just to the north-north-west. In winter, the sun rises in the south-south-east and sets five hours later almost directly south. A fixed solar panel would be facing the wrong direction half the year.



In the real world, solar trackers are only particularly useful for portable, temporary, and very small scale solar applications. For a solar panel or solar farm that won't be moved, a scheduled tracker is more efficient. A scheduled solar tracker calculates the exact position of the sun, which is relatively easy with the exact location of the panel, date and time of day. This ensures the panel continues facing the sun whenever there is adverse weather or even just an overcast day.

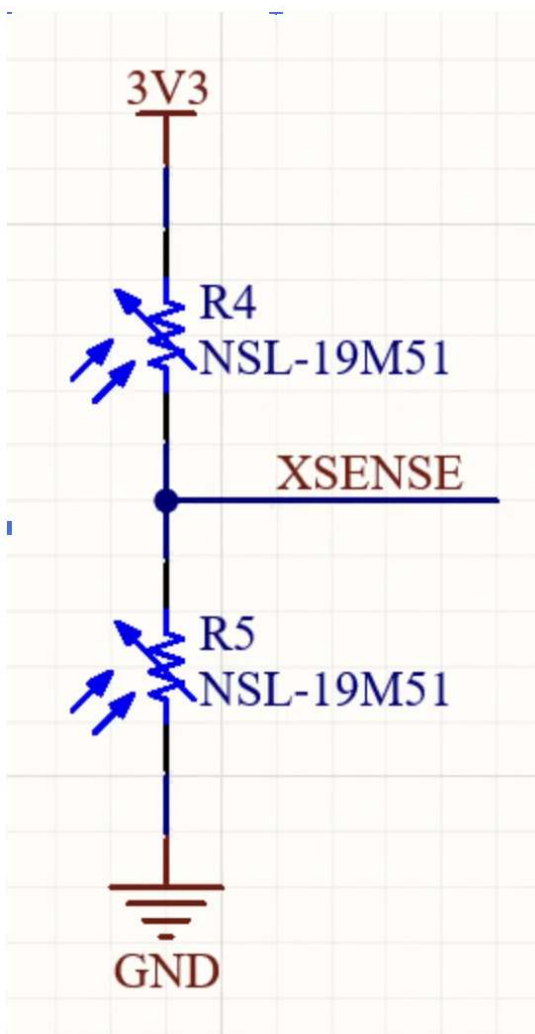


02

Sensing Light



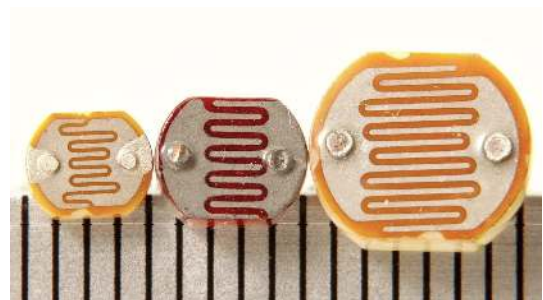
To track the sun, we need to first determine its location. Without computing this from a known location as described above, we can do that by simply detecting its light. During daylight hours, it's almost certainly going to be the brightest light source.



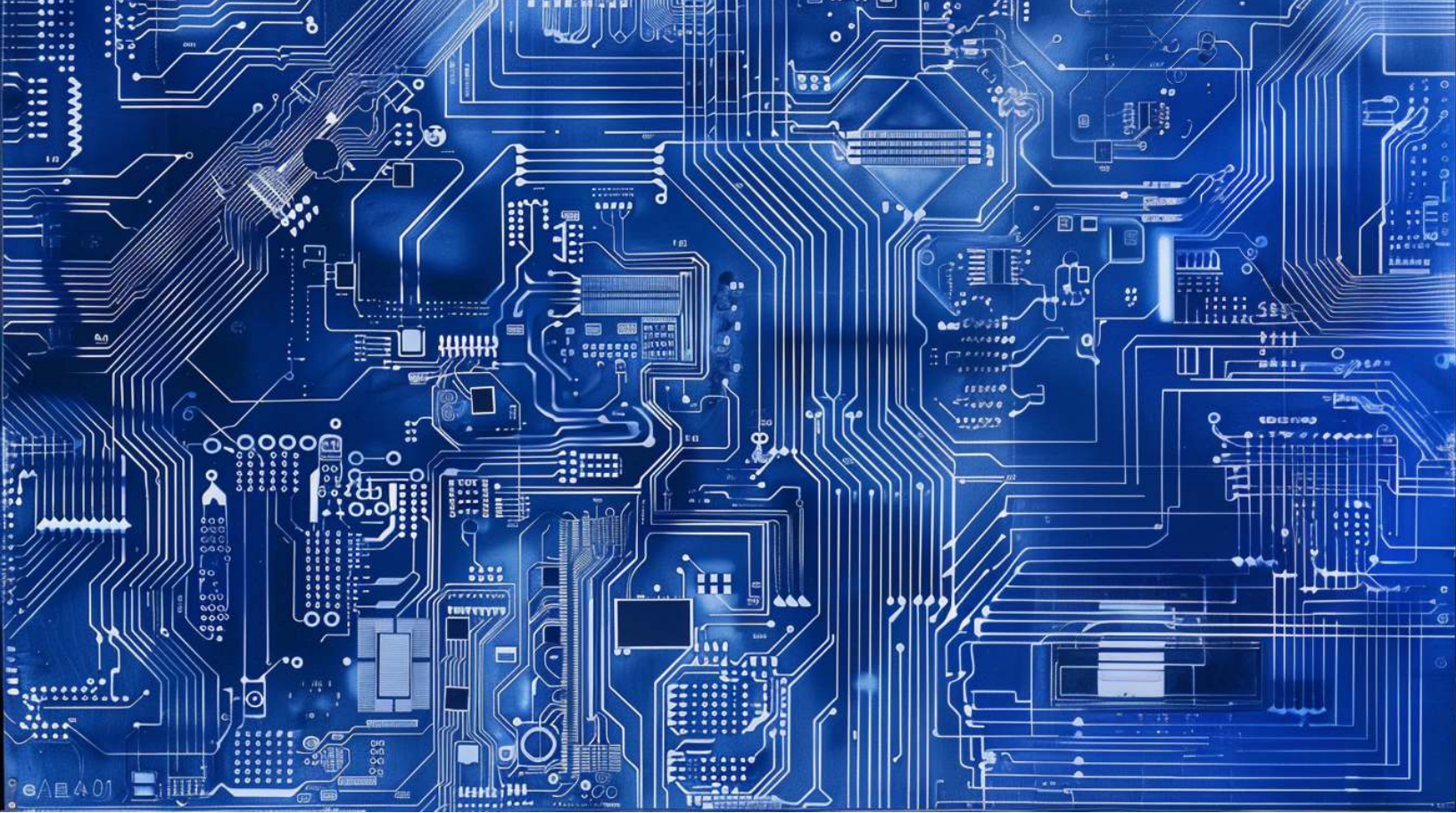
For this project, I'm using the simple photocell which has some ideal properties for a solar tracker such as this, however, it does have some limitations so we'll also discuss alternative methods of light-sensing.

Photoresistor

A photoresistor is a completely passive device, with no polarity just like any other common resistor. As light hits the surface of the photoresistor the resistance falls. We can use this property in our solar tracker to create a voltage divider with a photoresistor for the top (R4 below) and bottom leg (X5 below), with the midpoint (XSENSE below) connected to our windowed comparator circuit.



Source: Junkyardsparkle, Photoresistors - three sizes - mm scale, CC0 1.0



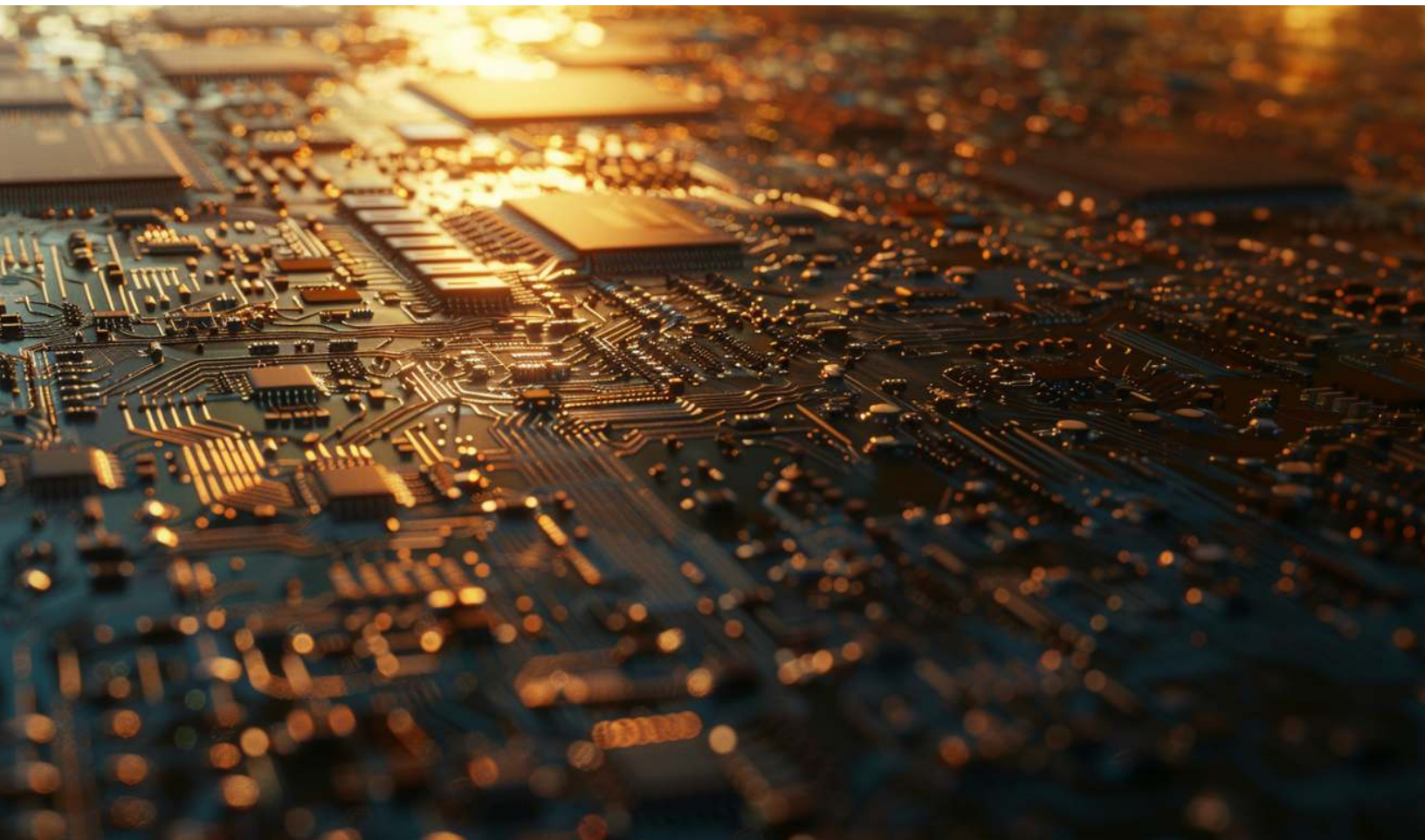
If one photoresistor receives more light than the other, it will skew the voltage of the divider, which the comparator can easily determine which direction the sensor needs to turn to balance the light again.

Another advantage of photoresistors that is particularly useful for this project is that they react to any light hitting their surface, no matter the direction. This allows a photoresistor to sense even very shallow angle light hitting it, which is necessary to bring the solar panel back towards the sun in the morning, or after a period of very heavy cloud cover.

Analog Solar Tracker Project | Altium Designer

The relatively slow response of a photoresistor is typically considered a disadvantage, however this application somewhat of an advantage. Other light sensors can respond in nanoseconds to changes of light intensity, whereas the photoresistor can take milliseconds or longer. This property ensures that we will never have a shoot through on our H-bridge, which is where both MOSFETs on one side of the bridge are enabled at the same time, effectively creating a direct short from the input voltage to ground. Our photo resistors will ensure at least several milliseconds are spent transitioning through the hysteresis zone of the windowed comparator.

The major disadvantage of photoresistors is the materials they are manufactured from. Visible light photoresistors use Cadmium Sulfide for sensing light, which is a banned substance in Europe under ROHS. This makes it virtually impossible to sell a product containing CdS photoresistors in Europe. I had no problem sourcing CdS photoresistors from a major UK component supplier for the prototype of this project however.



Photodiode

One of the most common ways of sensing light is to use a photodiode, which with the right second can easily respond in nanoseconds to changes in light intensity. This makes them perfect for a variety of applications such as communications, optical drives, and optical switches. For this project, however, the implementation adds a lot of complexity to the design compared to a photoresistor. You can read more about building a transimpedance amplifier for photodiodes in my [high speed photogate project article](#). You can also source photodiodes with integrated transimpedance amplifiers, such as the Renesas ISL76671, or with digital output via I²C or SPI which can ease implementation.

For this project, a photodiode also has other disadvantages compared to a photoresistor. A photodiode has a relatively narrow sensing angle which doesn't allow it to respond well from off-axis light, such as you would expect in the morning when the solar array is still pointing at where the sun went down.

Photodiodes respond to a relatively narrow spectrum of light, typically in the infrared range, though visible light sensors are available. This makes them fantastic for sensing a matched emitter's signal while blocking out ambient light, but less ideal when our tracker should be responding to any illumination.

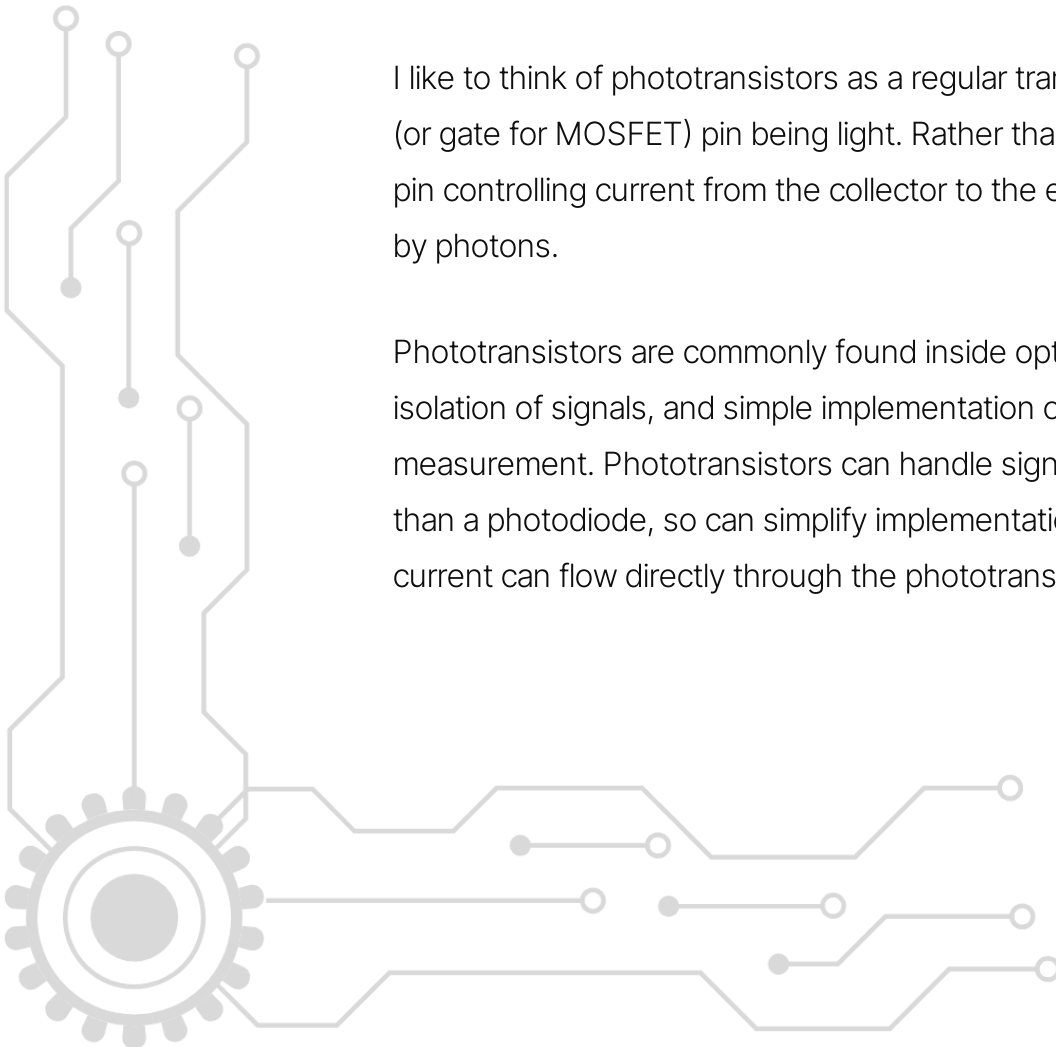


Phototransistor

Where a photodiode is perfect for rapidly responding to light, a phototransistor is ideal for accurately measuring light. The photoresistors used in this project are not appropriate for anything approaching a scientific indication of light level, the tolerances are too poor as is the variance in resistance as temperature changes. A phototransistor can be very consistent for measuring luminous intensity. Phototransistors are available as both bipolar junction transistors (BJTs) and metal oxide semiconductor field effect transistors (MOSFET).

I like to think of phototransistors as a regular transistor with the base (or gate for MOSFET) pin being light. Rather than current on the base pin controlling current from the collector to the emitter, it is controlled by photons.

Phototransistors are commonly found inside optocouplers for optical isolation of signals, and simple implementation of light level measurement. Phototransistors can handle significantly more current than a photodiode, so can simplify implementation in cases where current can flow directly through the phototransistor.

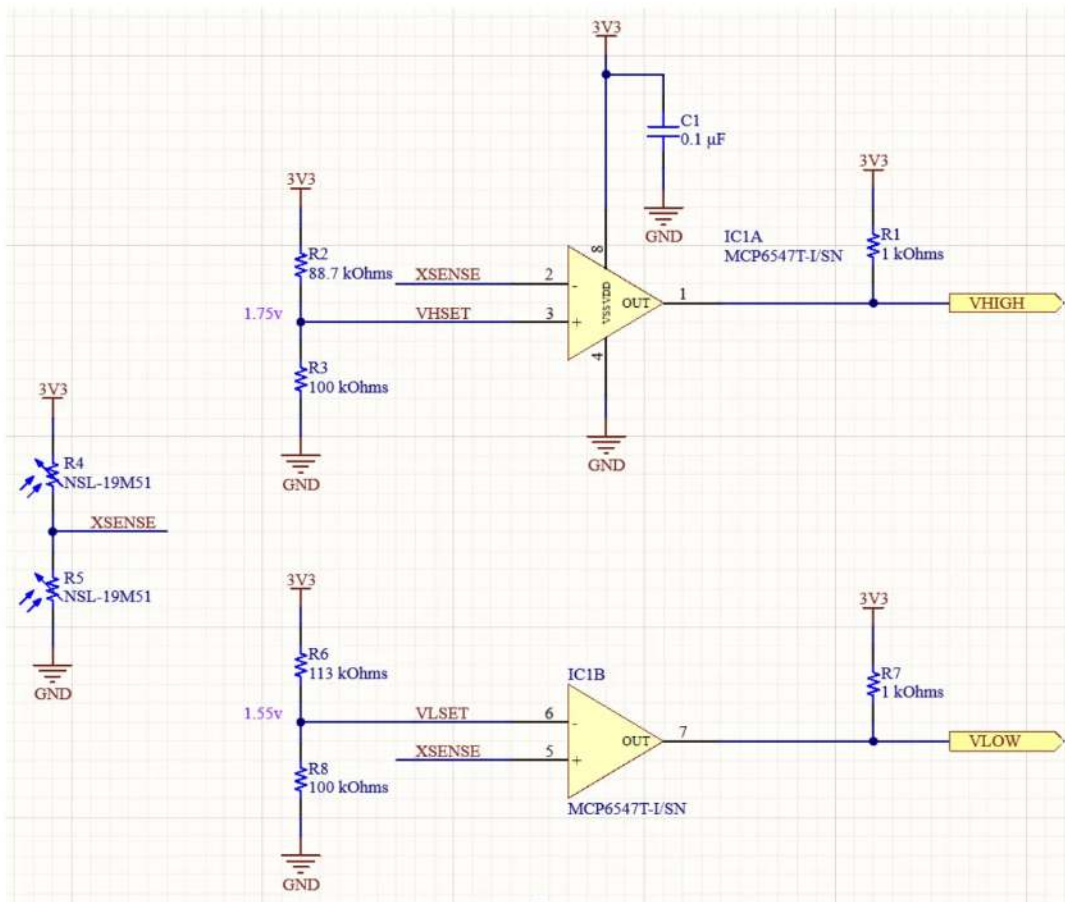


03

Window Comparator - Using Sensed Light



A window comparator is a circuit using two comparators, a sensor voltage (XSENSE below) and two fixed voltages. The fixed voltages provide the upper (VHSET below) and lower (VLSET below) bounds of acceptable value for the signal voltage. If the signal strays outside these bounds, the comparator output will change, allowing it to act as an alarm. While commonly the comparator outputs are tied together to indicate an error state, you can also use each comparator individually to know if the sensor signal strayed too high or too low.



Analog Solar Tracker Project | Altium Designer

Window comparators are an ideal way to monitor a safety critical sensor value, such as temperature. The advantage of doing this with comparators instead of a microcontroller is the ability to ensure a cutoff or shutdown without microcontroller intervention in the event of a microcontroller or firmware failure. A window comparator can be built with ultra-low quiescent current comparators and may consume less power than a microcontroller coming out of a sleep state to take an ADC reading in **low power applications**, with the output of the comparator able to interrupt the microcontroller to bring it out of sleep. A window comparator is generally a highly reliable circuit and can allow offloading of ensuring a sensor voltage stays within a safe range from a microcontroller.

For our solar tracker, we can use this circuit to determine which photoresistor is receiving more illumination and therefore turn the motor towards the light. Assuming the photoresistors are from the same production batch they should exhibit an acceptably similar response to light. The sensors I'm prototyping with give almost exactly 1.65v at the center of the potential divider when under even illumination. Having equal resistance means I don't need any additional resistors or potentiometers to balance the divider.

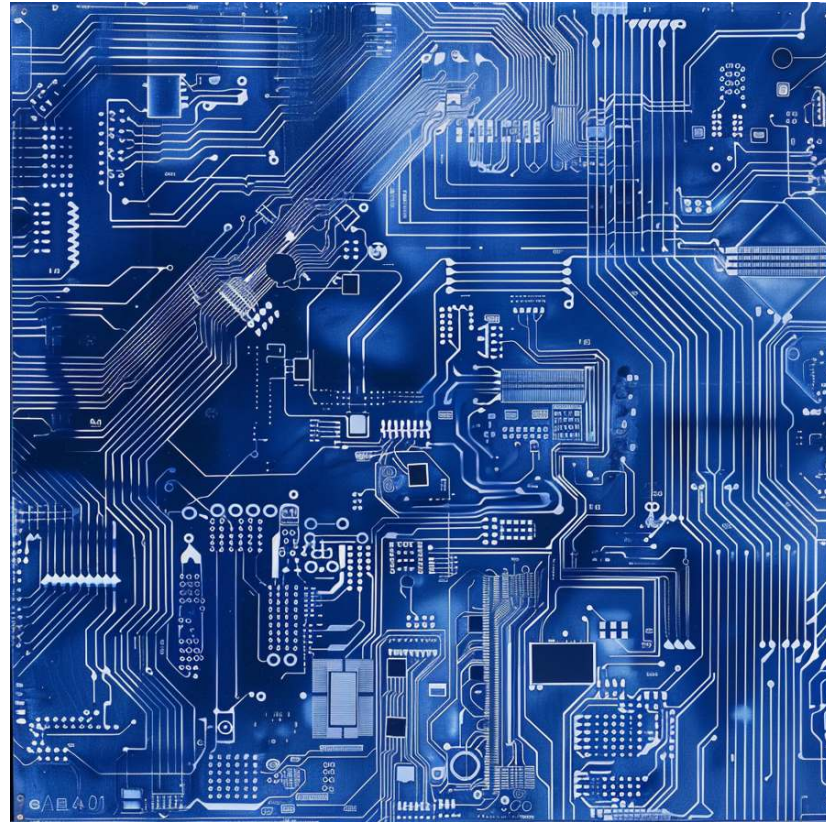
In testing, I found that 0.1v above and below the 1.65v center point of the potential divider worked well for indicating an illumination change worth driving the solar panel toward don't want to have the hysteresis set too tight, or the motors will be constantly jittery trying to find the perfect angle, which can consume a lot of power - defeating the point of optimizing the array direction. Likewise, you don't want the voltage window to be too narrow or the solar array won't be positioned optimally.

04

Driving The Motors



Now that we have a signal which allows us to know which direction we need a motor to move, we can build our motor driver. As mentioned in the introduction, this H-bridge is substantially overbuilt for my prototyping setup but allows you to easily swap out the MOSFETs to drive much more substantial motors. Another reason for the circuit being so overbuilt is the great chip shortage! I was hoping to find a very small 2 or 4 channel high and low side driver with independent inputs but nothing was in stock. This circuit needs independent inputs, whereas most motor driver ICs have an enable/PWM input and direction input which isn't quite as useful with the output from our comparator circuit.



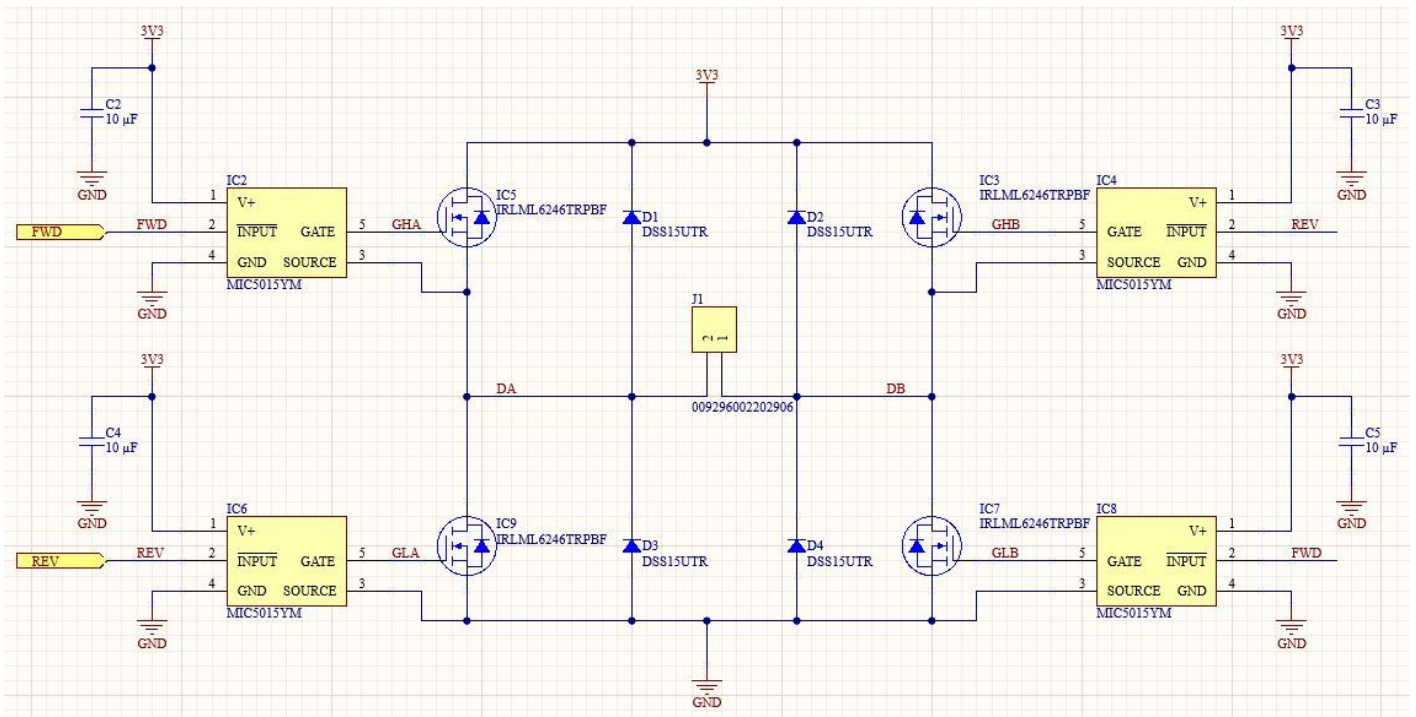
Alternatively, I could have used P-Channel MOSFETs for the high side, or even switched to using PNP and NPN transistors for the high and low side respectively - but that doesn't scale as nicely as using low $R_{ds(on)}$ N-Channel MOSFETs all around. Driving the MOSFET gates directly with the comparator output is fine with very small MOSFETs that have gate threshold voltages, and a low voltage for minimum $R_{ds(on)}$.



However, larger MOSFETs to drive a motor capable of moving several full size solar panels at once will need a more substantial gate voltage to ensure the MOSFET is operating at the lowest resistance possible.

To utilize N-channel MOSFETs for the high side of the h-bridge, we need a driver IC which can generate a higher voltage for the gate than the drain pin. With the chip shortage, I had the choice of 3 different driver ICs at my preferred supplier, so chose the cheapest option.

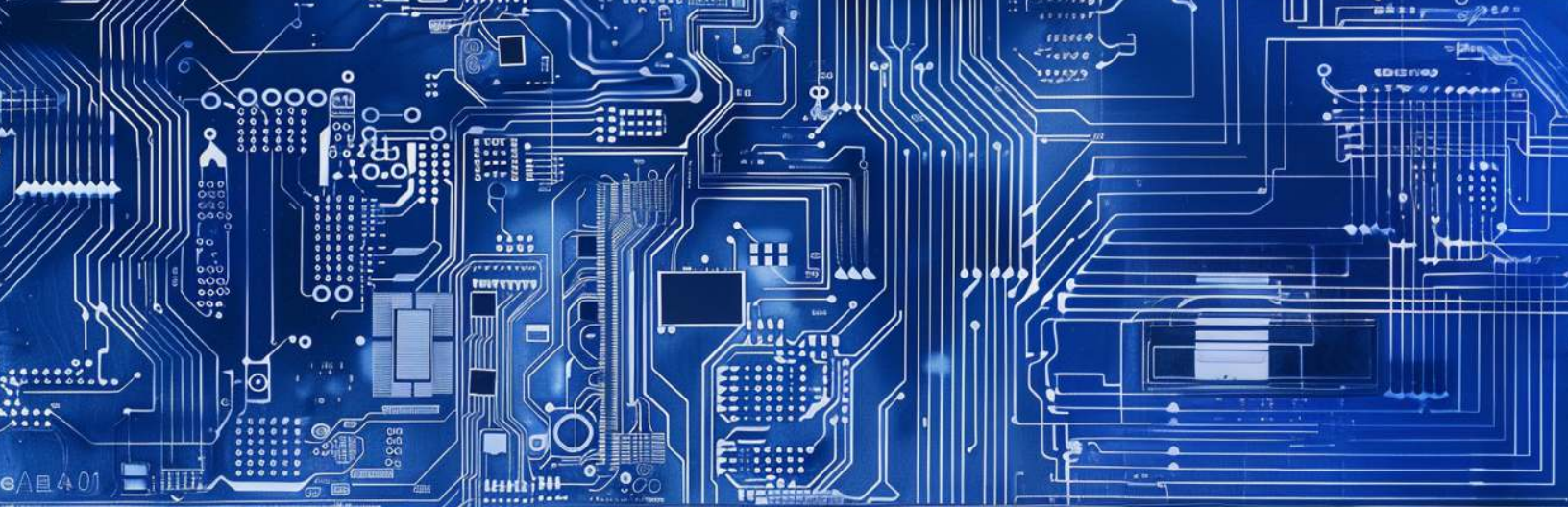
Each MOSFET gate is driven by its own driver. Each MOSFET has a low forward voltage diode paired with it, to ensure the body diode of the MOSFET isn't dealing with flyback voltages from the motor alone. Without the diodes, the MOSFETs are unlikely to survive many drive cycles! You can read more about selecting drivers, MOSFETs and diodes in [my article over on Octopart](#). If you're looking to implement a more substantial H-bridge, or learn more about H-bridges you'll probably enjoy my 100A H-bridge project.



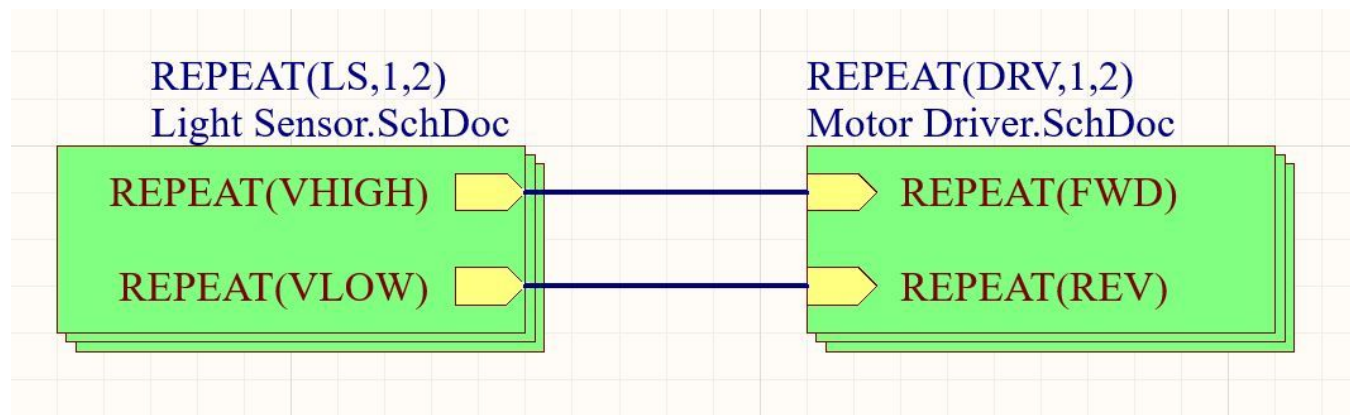
05

Bringing It All Together





To drive our solar tracker on two axes, we will need two light sensors, and two motor drivers. Rather than copy and pasting the schematics, we'll use Altium Designer's **multi-channel functionality** to save a bunch of time. This will allow us to layout and route each schematic block just once, and apply the layout and routing to the other channel - halving the time it takes to design the board. I'm not going to go into depth on creating a multi-channel design, as you can read my full guide to the ins and outs in [my Current Monitor and Controller project](#).



For this project, it's simply two light sensors connected to two motor drivers. A very simple example of multi-channel design, yet still a substantial time saver.

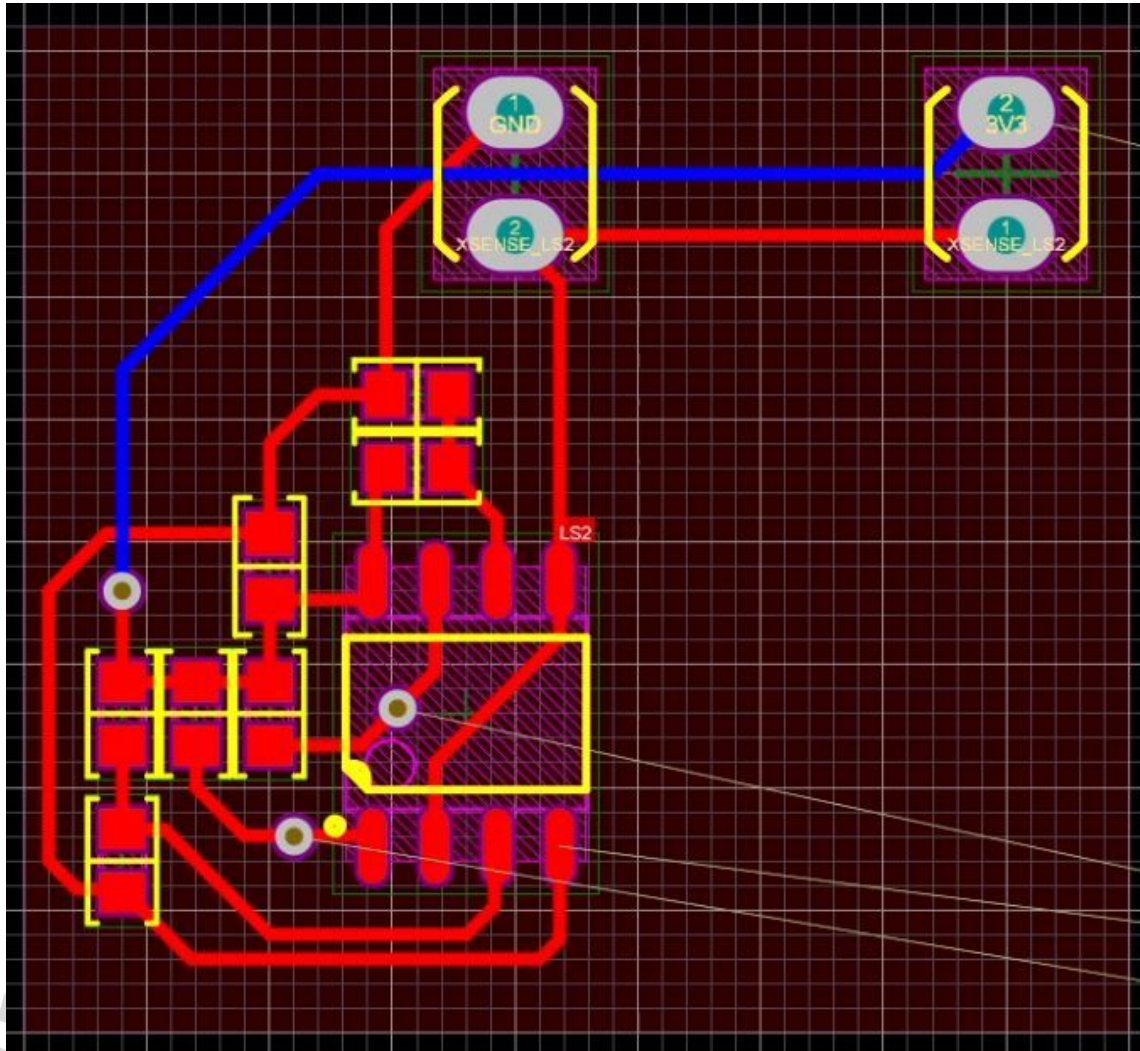
06

PCB Layout



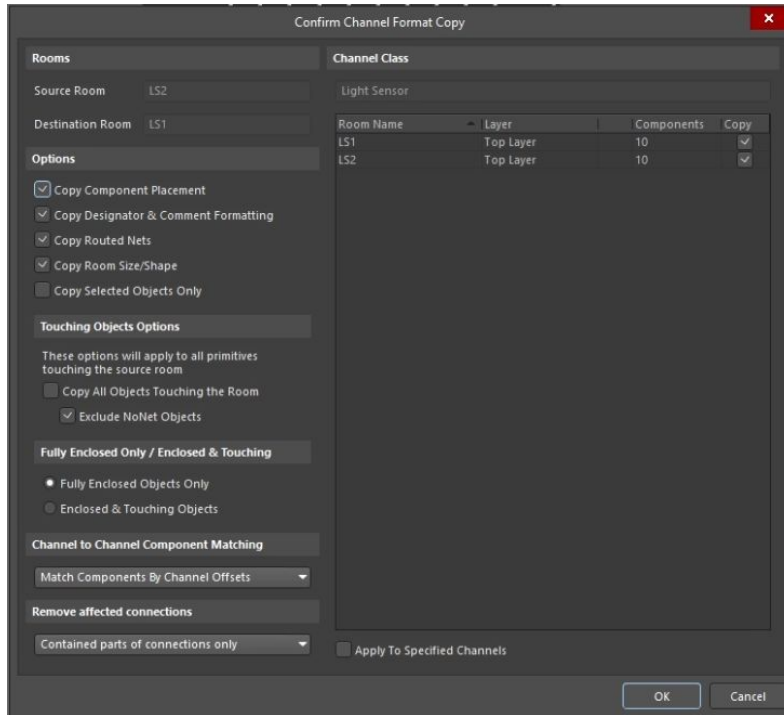
Analog Solar Tracker Project | Altium Designer

The most critical portion of this project is the positioning of the photoresistors. I'm going to position the sensors as a + shape, which will mirror the motor axes. This will allow me to use a x shaped light shield which will help cast shadows. After setting up design rules for the new PCB, the first step is to layout and route a single channel of the light sensor. This layout and routing is not necessarily the final routing, but it will be a good starting point for the two channels.

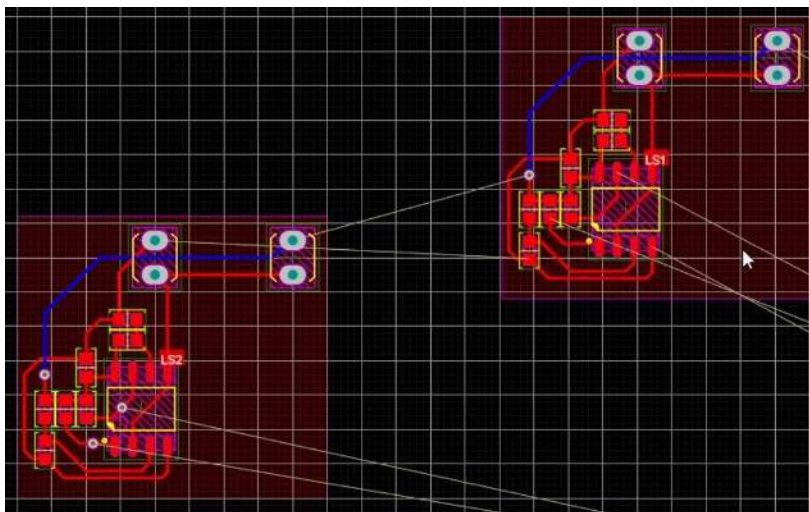


Analog Solar Tracker Project | Altium Designer

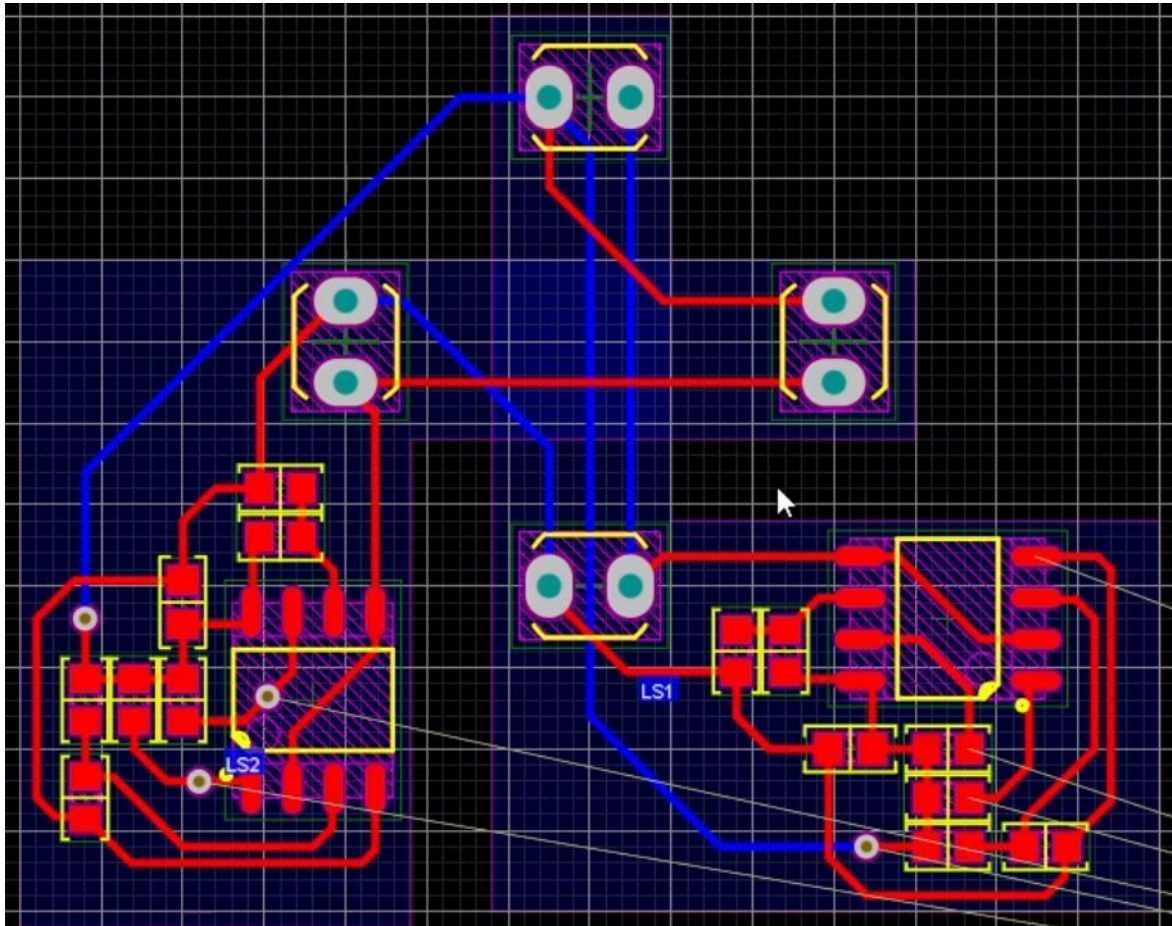
Then we can **copy the room format** from one light sensor channel to the other.



This leaves us with two identical channels we can play with the layout of to position the photoresistors.

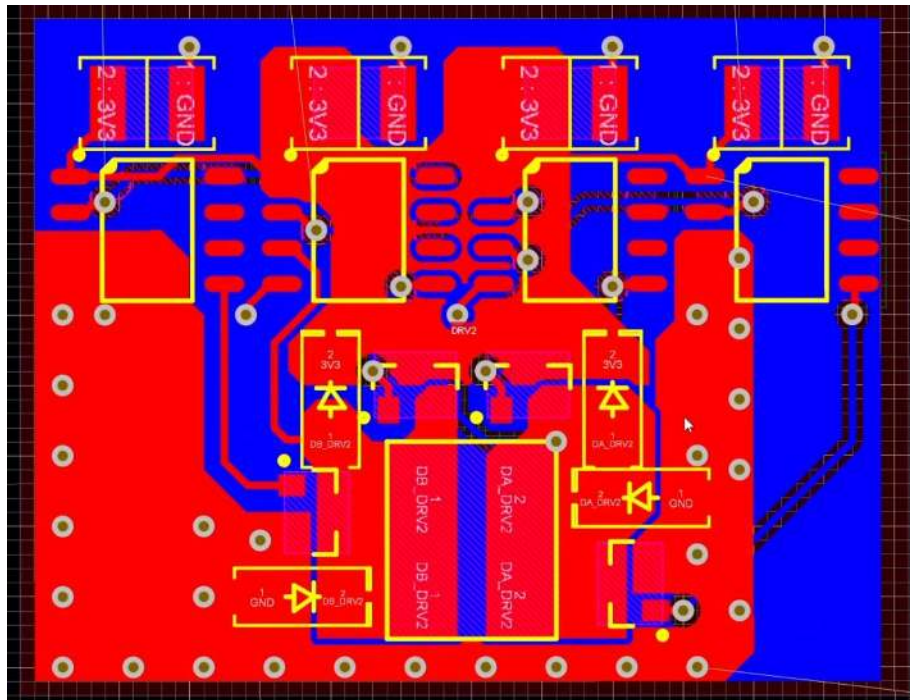


Which gives a layout of the light sensors I'm fairly happy with for the moment. I'll need to make a few changes for mounting holes, and also bring the motor signals out once the drivers are done.

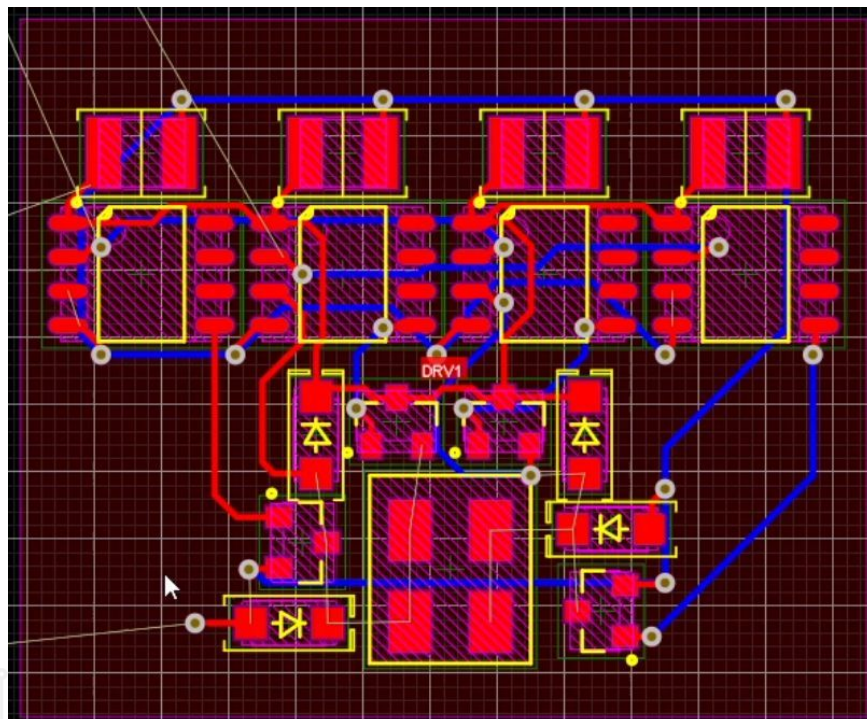


I also took the opportunity to **layout the rooms** as polygons rather than just rectangles, to cover just the items within the room.

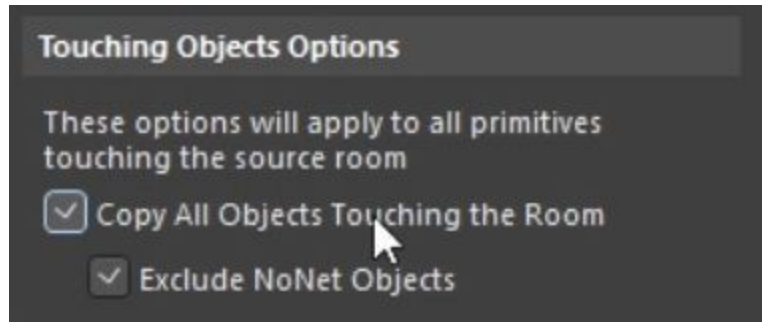
Next up is the motor driver. For my motor this design is absolutely non-critical as the current is so low, however the IRLML6246 MOSFETs are rated for 20V 4A, so I'll be designing around that, rather than my prototyping motors.



After entirely too long playing with component placement, and polygons, I had a layout that I was happy with and could copy to the other channel.

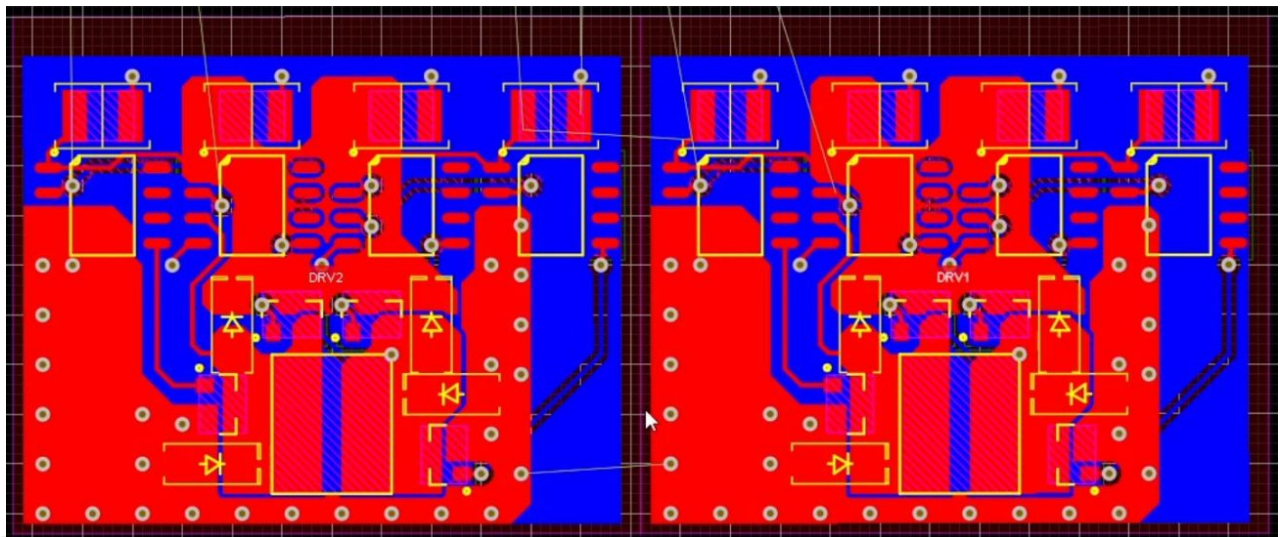


You might notice a few things are missing compared to the previous screenshot. I forgot to check the Copy All Objects Touching Room option on the Confirm Channel Format Copy window!



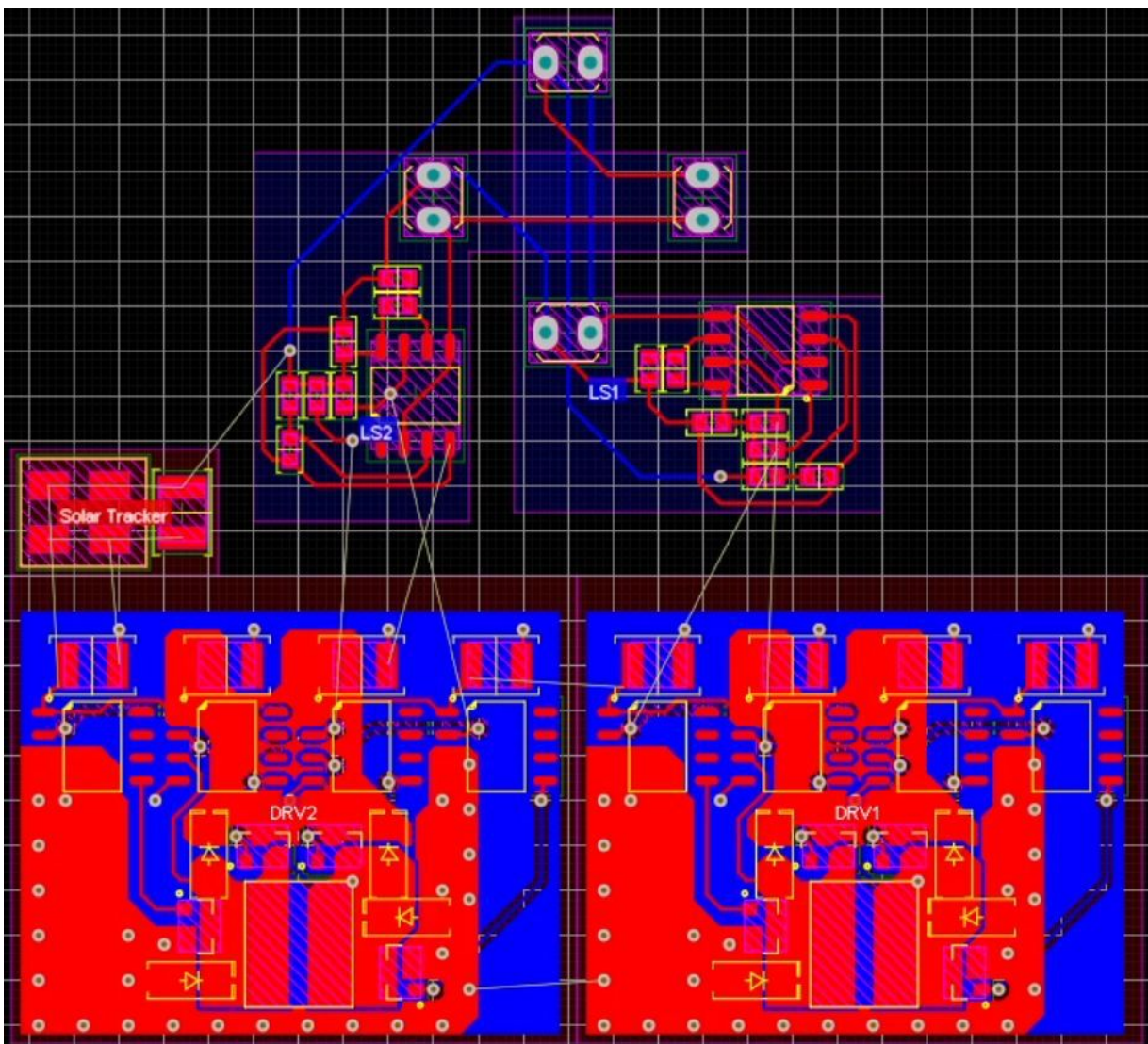
This ensures that the polygons and vias will copy across, as those are not items on the schematic.

This change means the board now has two identical drivers with all the polygons and vias as expected.

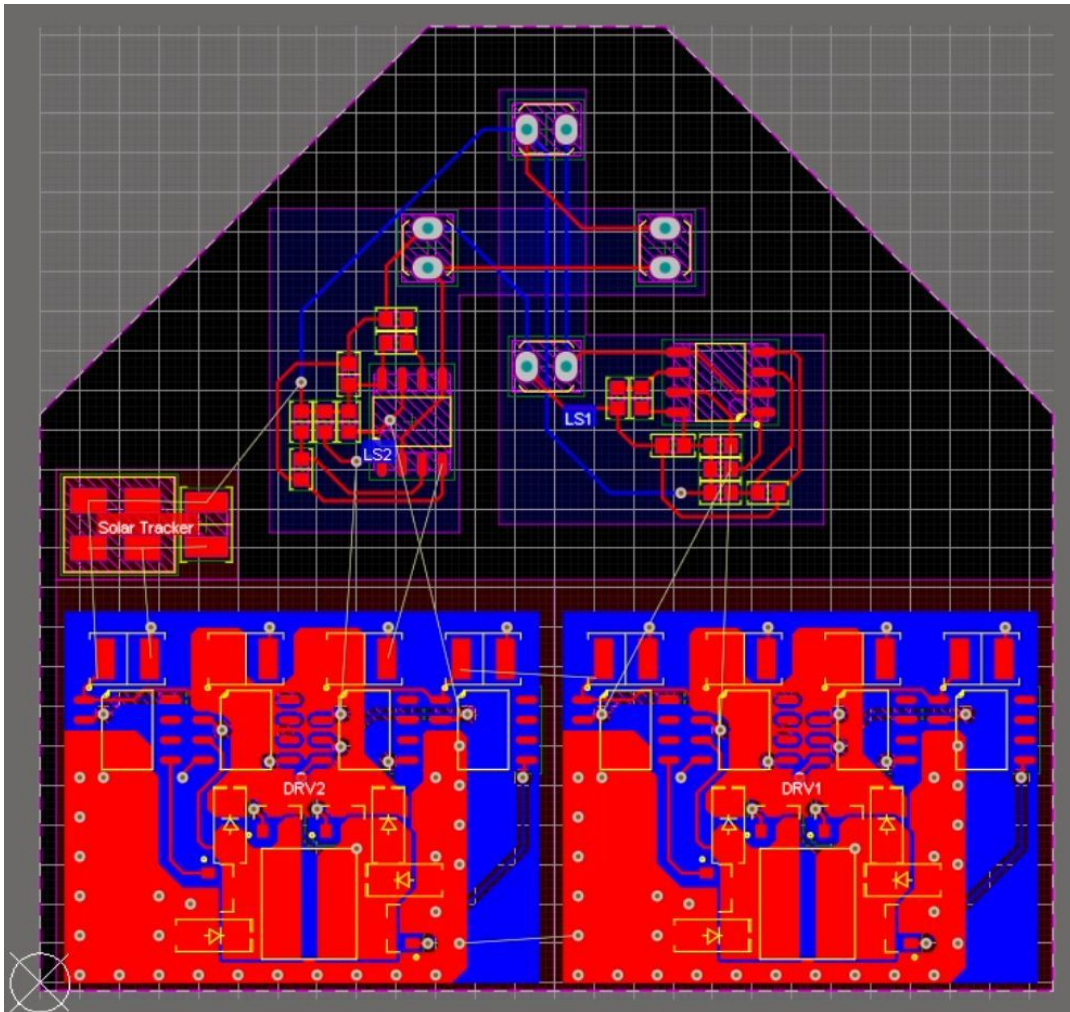


Despite the simplicity of the motor driver circuit, using multi-channel design tools allowed the second motor driver to be completed in just seconds, even with my mistake! Can multi-channel design save you time, and increase your productivity? [Talk to an expert at Altium now](#) to find out how.

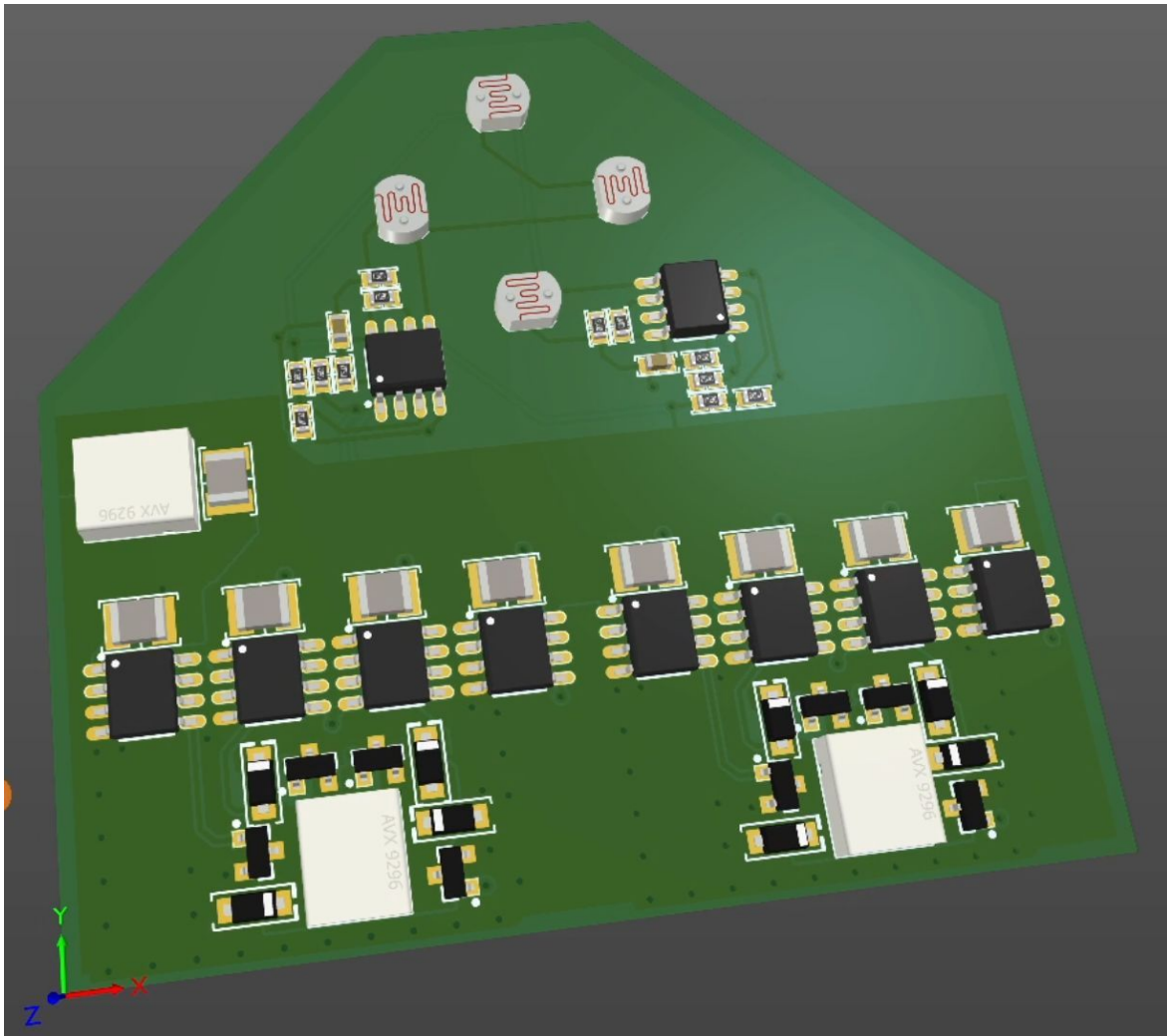
We now have a completed general layout, however, I have just been laying things out without any consideration to the board shape.



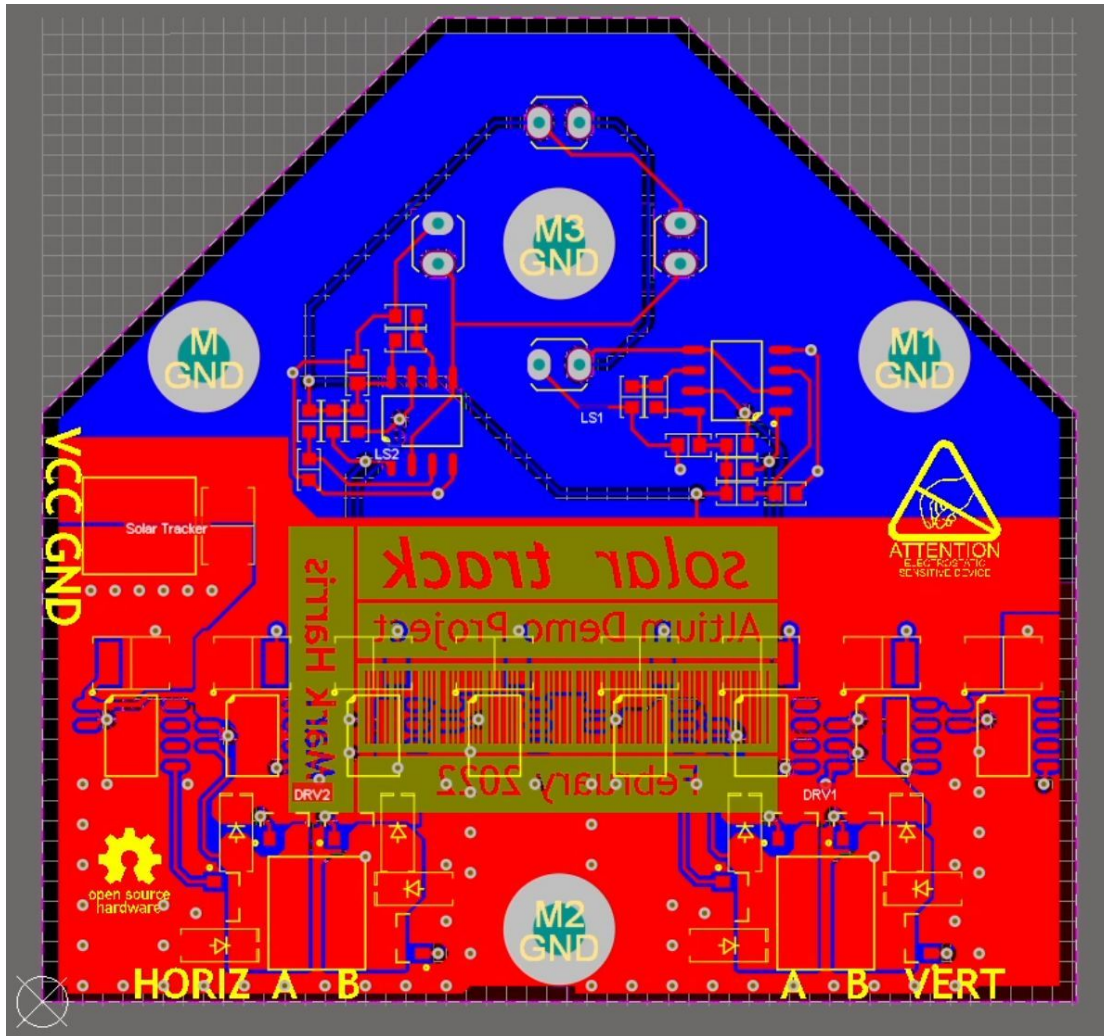
Rather than jump into connecting the **schematic blocks**, I want to first define a board shape that will assist with determining our signal and power paths.



I need to leave a decent amount of room around the photoresistors to allow for the light shield to be added, and as this will need to be mounted to a large solar panel I'm not trying to **save every last square millimeter of board area** possible.



With the final interconnections completed, it's time to think about mounting holes. I want to have sufficient support that a **waterproof enclosure** could be built around it, with secure board mounting and sufficient points of contact with the solar panel array. We're not taking on that aspect of the project in this article, but it's always good to think about the next steps of the design and implementation of the hardware.



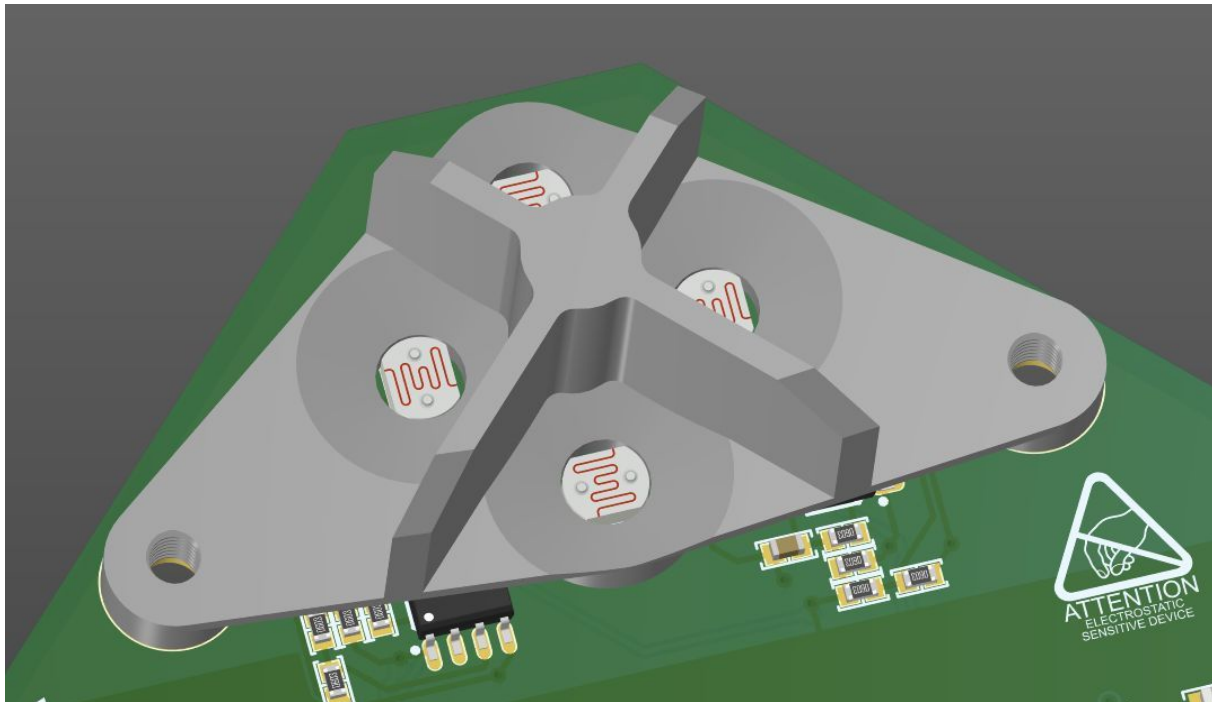
had to make a few minor adjustments to the routing around the photoresistors, but was able to squeeze in all the mounting points I feel that I need without too much difficulty.

07

Adding a Light Shield



I'm using a 3D printable addition to the PCB to improve the gain between the photoresistor pairs. Light at a low angle will cast a shadow on one sensor while also reflecting light onto the other, increasing the difference in resistance between the sensors.



If light is perpendicular to the sensor face, the shield will have little impact on the resistance differential between the sensors. This should improve the solar tracker's performance after a period of darkness such as in the morning or after heavy cloud cover/atmospheric conditions.

The shield uses the mounting holes on the board, allowing the plastic shield to be screwed in from the back of the board.

08

Further Improvements



As with any project, there are always improvements that can be made even when you feel it is feature complete. If you wanted to implement this solar tracker project for your own use, here are some suggestions for other features you might want to consider adding.

CHECKLIST



1. Limit Switches

To prevent a failure causing the solar array to endlessly spin around, destroying its cables in the process, you would likely want to consider some limit switches that would act as stops to limit the total slew range of the solar array.

2. Larger H-bridge

As mentioned a couple of times in this article, the H-bridge gate drivers are ready for larger MOSFETs to build a bigger driver that can handle a larger array. I'm only prototyping this circuit with 100mA/3V gear motors that are perfect for desktop use, but probably a little useless outside.

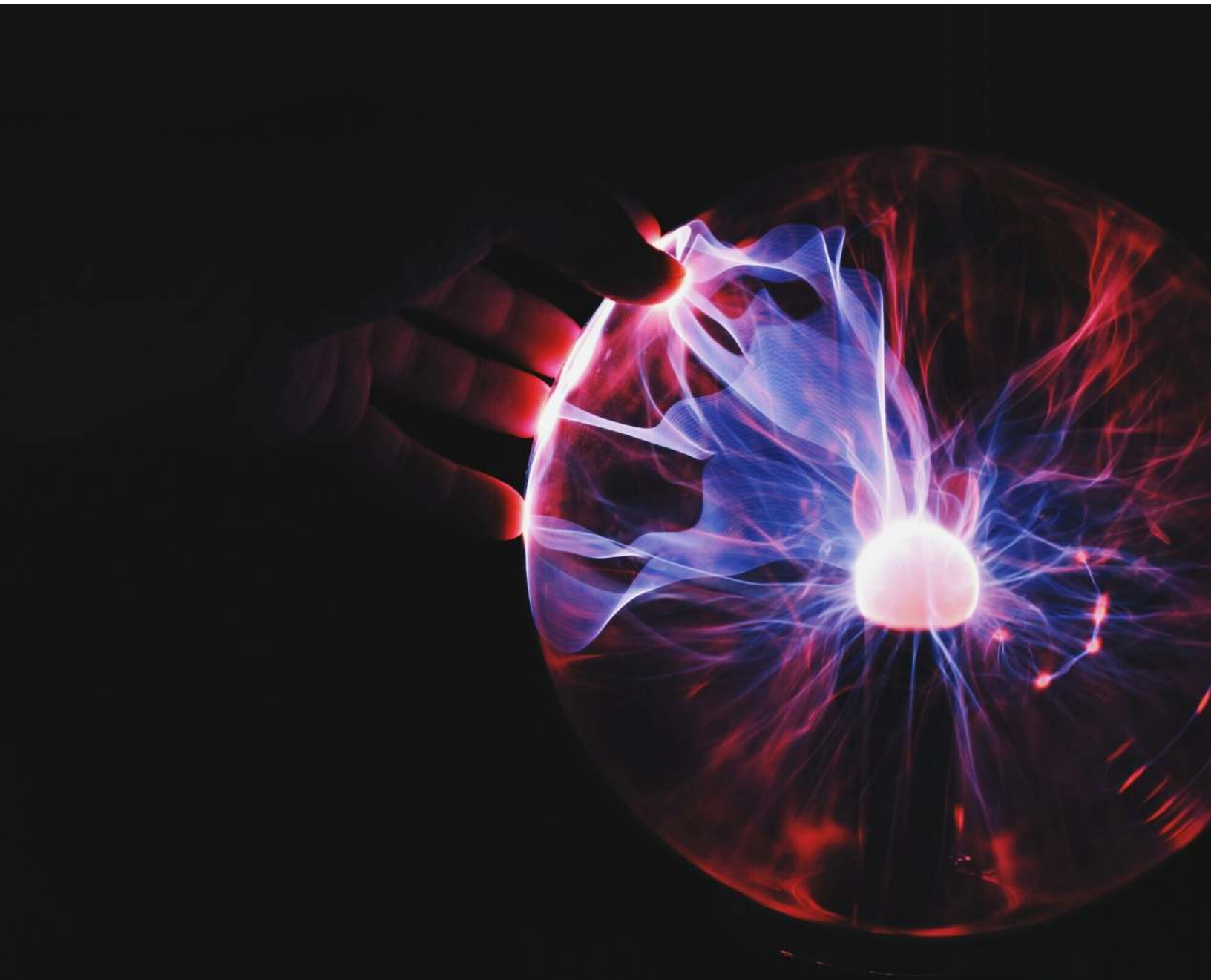
3. Night Detection

To prevent the array from using energy at night, or when atmospheric conditions reduce the light level substantially, another light sensor could be used to disable the motor driver when the ambient light intensity falls below a certain threshold. This will stop the array trying to move because of stray light from automotive headlights for example.

Likewise, during daytime if there is insufficient light intensity to generate meaningful power from the attached solar array, it makes little sense to use power driving the motors.

4. Voltage Regulator

This circuit is designed to run from a 3.3V source, despite many of the components being rated for higher voltages. If the board is expected to run off the array/battery storage voltage, then a regulator would most likely be required onboard.



09

Conclusion



Analog Solar Tracker Project | Altium Designer

In this project we've discussed the range of light sensitive components available and used photoresistors to implement an open solar tracker. You can freely use the Altium project files for your own purposes, either directly or by expanding on the capabilities of the schematics to suit your own requirements. The window comparator circuit which comprises the heart of this project is a fantastic addition to many projects, offering reliable monitoring of signals to ensure they remain within a set voltage window.

Renewable energy is in ever increasing demand, and this solar tracker will allow you to get the most out of a mobile solar array without any need for a microcontroller or programming.

The design tools in [Altium Designer®](#) contain everything you need to keep up with new technology. Sign up for a [free Altium Designer trial](#), or [talk to us today](#) and find out how we can enhance your next PCB Design.

