Tracking the Sun

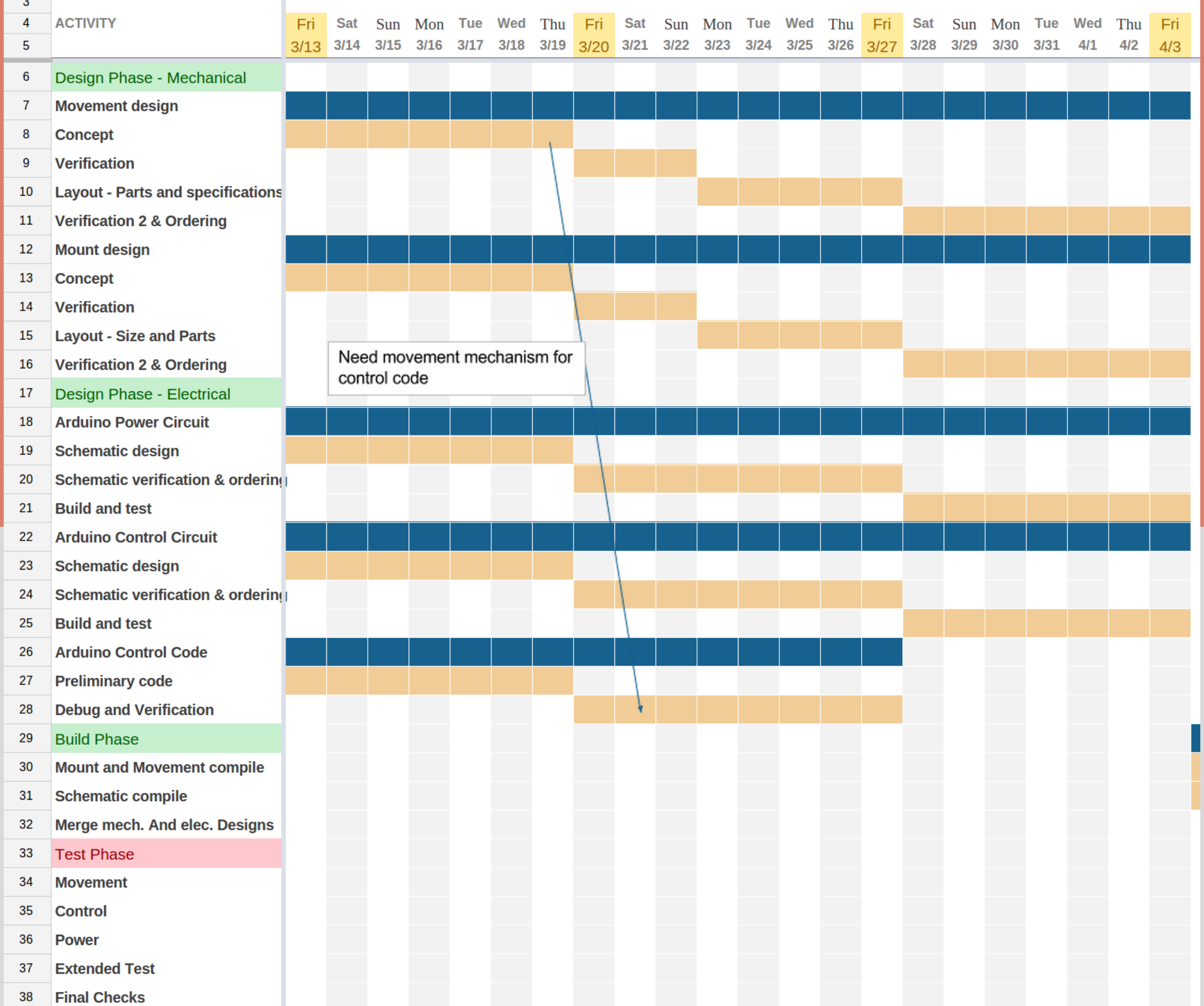
Our process:

- Design Goals
    - Track the sun
    - Low cost:efficiency ratio
    - Reliability
    - Be pretty
- Divide and Conquer
    - Mechanical Design
    - Electronics Design
    - Algorithm Design
- Verification and Testing

Design Goals

1. Tracking the sun
   a. Follow the sun as the day progresses.
2. Low cost:efficiency
   a. Every dollar spent needs to contribute as much or more
   b. Sacrificing good design for low cost is NOT okay
3. Reliability
   a. Don’t break
4. Be pretty
   a. At minimum, contain the capability of being aesthetically pleasing
   b. Get people to use it, many people oppose solar power because it’s ugly.

Source and data available at:
github.com/AnthonyLam/solaaaar



ACTIVITY
Fri 3/13
Sat 3/14
Sun 3/15
Mon 3/16
Tue 3/17
Wed 3/18
Thu 3/19
Fri 3/20
Sat 3/21
Sun 3/22
Mon 3/23
Tue 3/24
Wed 3/25
Thu 3/26
Fri 3/27
Sat 3/28
Sun 3/29
Mon 3/30
Tue 3/31
Wed 4/1
Thu 4/2
Fri 4/3
Design Phase - Mechanical
Movement design
Concept
Verification
Layout - Parts and specifications
Verification 2 & Ordering
Mount design
Concept
Verification
Layout - Size and Parts
Verification 2 & Ordering
Design Phase - Electrical
Arduino Power Circuit
Schematic design
Schematic verification & ordering
Build and test
Arduino Control Circuit
Schematic design
Schematic verification & ordering
Build and test
Arduino Control Code
Preliminary code
Debug and Verification
Build Phase
Mount and Movement compile
Schematic compile
Merge mech. And elec. Designs
Test Phase
Movement
Control
Power
Extended Test
Final Checks
Need movement mechanism for control code

Mechanical Design

Research indicates dual axis tracking is typically worth the trade off in cost versus single axis tracking.[1]

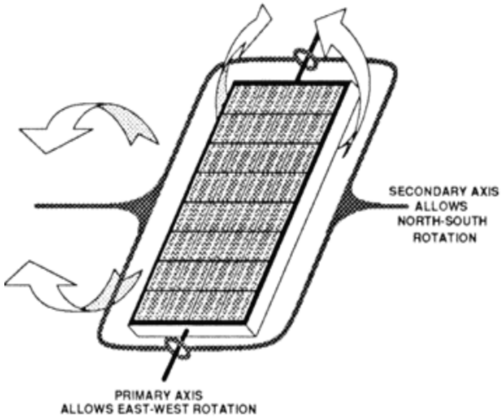


Fig. 5 Dual-Axis Tracker

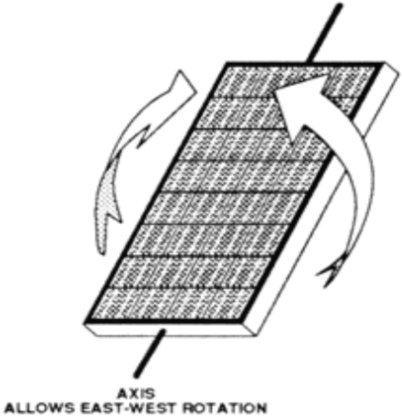


Fig. 1 Single-Axis Tracker

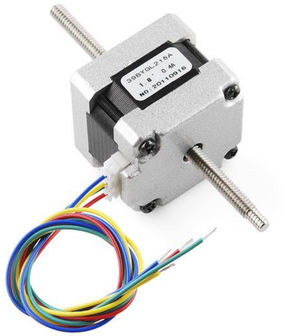
Given our target customer: ease of use and maintenance is a priority over absolute efficiency.

Opted for a single axis design.

[1] [Dual axis vs single axis tracking]()

Mechanical Design

Multiple forms of mechanical motion are available: servos, steppers and actuators.



Threaded Stepper

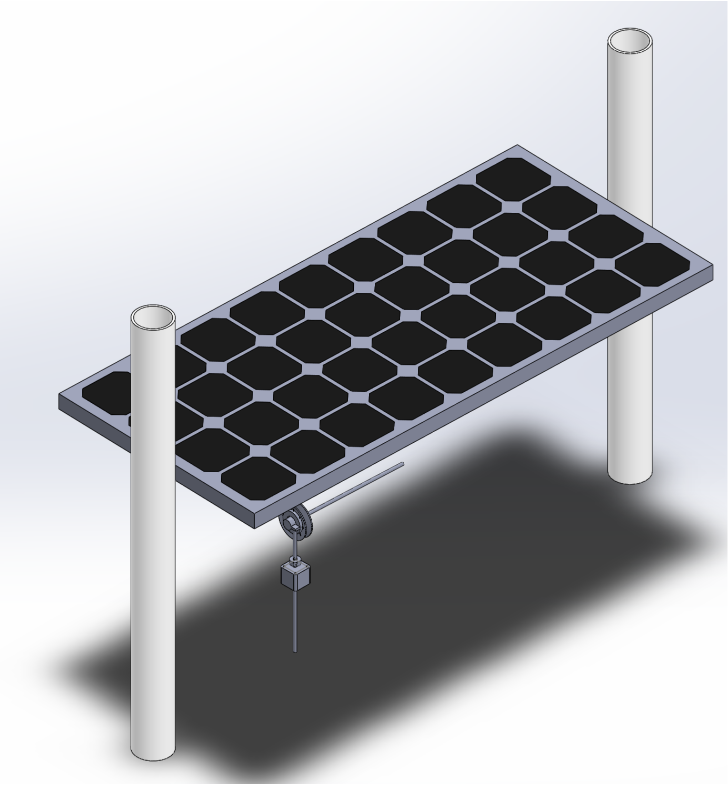
Linear Actuator

Threaded stepper motors yielded the greatest benefit for our system at a lower price point and a smaller form factor with a higher torque rating.

Mechanical Design

Gathered data on various materials that yield the greatest results for the lowest amount of money.

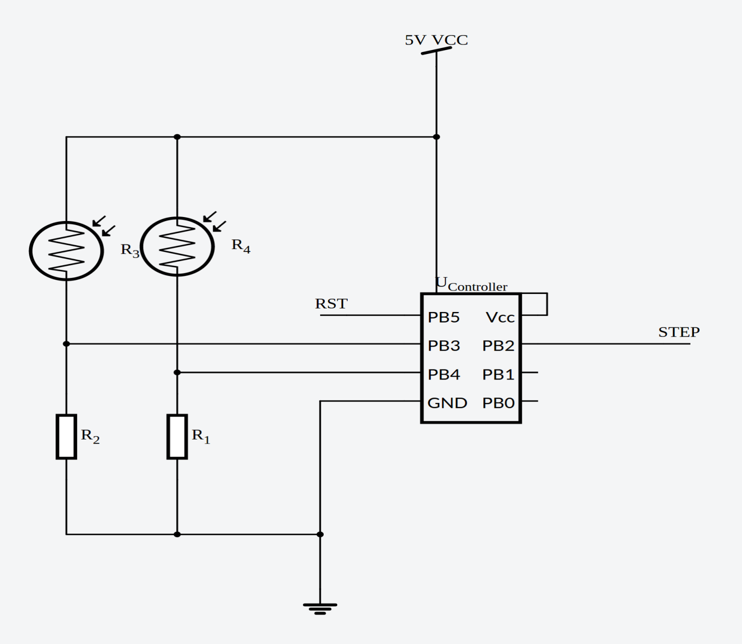
Mechanical Design



Preliminary design

Electronics Design

Simple logic and design means we could potentially do without a complex microcontroller.

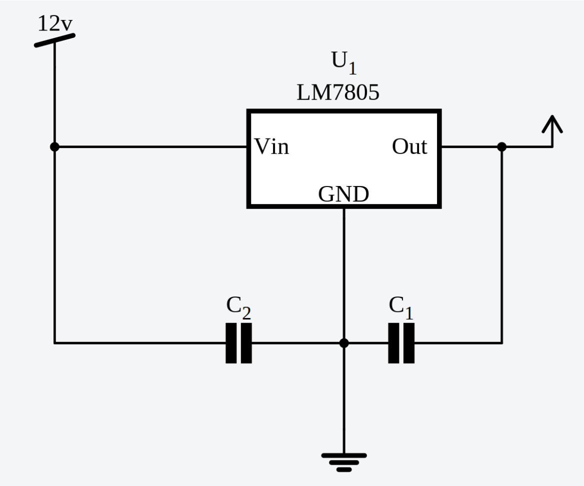


TTL would take too long to implement and a full Arduino is too much. An ATTiny85 will serve our purposes.

- Can be dropped to 1 Mhz clock. Yay power savings
- Small form factor
- Extremely cheap (50c)

Electronics Design

Design for voltage regulation comes down to 2 different designs:

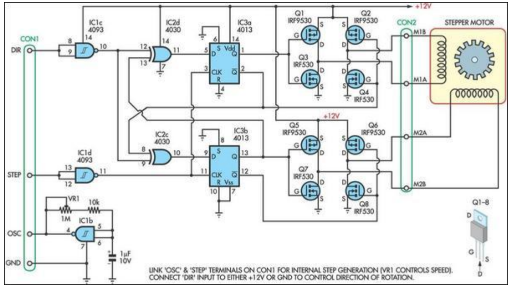


Other circuit wouldn't fit. Just imagine a voltage regulator with lots of resisters, an inductor and a capacitor.

Space constraints are a concern and the power loss is minimal since we deal with low currents.

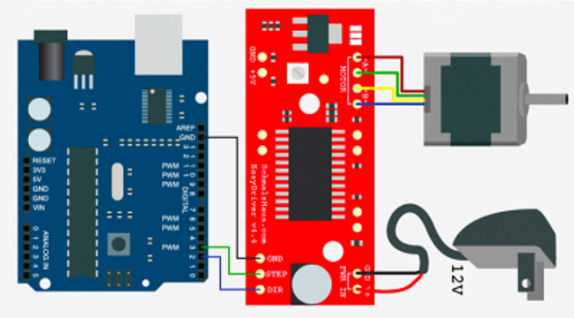
Electronics Design

Motor Control: custom build or use a proven solution?
Use of the ATTiny85 reduces the available pins.



FET circuit for stepper control

EasyDriver



Error prone circuit design or a nicely packaged PCB? It has Ez in the name.

[EasyDriver]

Algorithm Design

Our program can’t be debugged easily when put into the system so it must have fault tolerant behavior and must account for every predictable scenario.

- Derive an event driven system from a finite state machine.
- Minimize impact on runtime: We have very little memory, storage and clock cycles
- Use little power: tune built in clock down to 1 Mhz from 8Mhz

Algorithm Design

LDRw > LDRe = 1
LDRw < LDRe = 0

S1

LDRe+LDRw > threshold = 1
LDRe+LDRw < threshold = 0

S2

Panel @ extreme = 1
Panel inbetween = 0

S3

Sense light = 1
No light = 0

S4

Our algorithm states are defined by a 4-bit value representing various inputs.

Algorithm Design

Action	S1 S2 S3 S4	M
SLEEP	0000	10
PAN	0001	11
SLEEP	0010	10
PAN	0011	11
SLEEP	0100	10
R EAST	0101	00
SLEEP	0110	10
PAN	0111	11
SLEEP	1000	10

Only half of the table is shown but idea should be clear. Each action is represented by a 2-bit binary value and determined by the States: S1 S2 S3 S4.

Algorithm Design

	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	1	1	1
10	1	1	1	1

	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	1	1	1	1
10	0	0	0	0

M1 = S2+!S3+S4

M2=S3S4+S4!S2+S4S1

The previous table was too clunky and would result in inefficiencies. Here we apply a Karnaugh Map to simplify our algorithm down to 2 equations.

Total Cost

Ceramic Capacitor x2	$0.61	PVC x 20 ft	$10.86
Resistors x2	$0.20	Belt	$9.99
ATTiny85	$1.70	Stepper Motor	$32.00
Photoresistors x3	$2.70	Pulley	$4.58
LM7805	$0.67	Misc. Parts est.	$20.00
		Est Cost :	$83.31