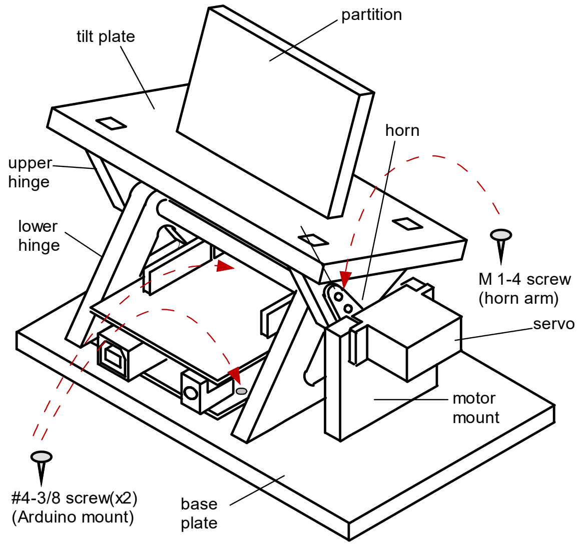
SOLAR TRACKER PROJECT

ENGR 133 –Mechatronics I

Project completion & testing (week 16)



SUMMARY

In this project you will design and fabricate a 1-axis solar tracker. The tracker will be driven using an Arduino micro-controller. The mechanical components can be fabricated using laser cutting of wood. Students will present their project to the instructor.

MATERIALS LIST

The following materials will be needed. Other items may also be used.

1. Arduino UNO kit (STUDENTS PROVIDE) that includes:
   - Arduino UNO R3 or Arduino Mega 2560 microcontroller
   - Breadboard shield
   - micro-servo SG-90 (180 deg, 4.8 - 6v Vin; current (no load): 100 mA, current idle 5 mA)
   - screws for micro-servo (2 to mount, 1 to mount horn to servo)
   - photo-resistors (or phototransistors (850 nm T1 3/4, $1.49) (x2) (parallax.com/product/350-0029)
     - detects light in the 450 – 1050 nm range (responds to visible light thru low IR range)
   - resistors (2k Ohm, 1/4 W) (x2)
   - jumper or dupont wire (some red & black)
2. Wood (Home Depot, 1/4" thick x 3.5" x 24" Poplar) (STUDENTS PROVIDE)
3. Dupont wire (M-F) - to attach sensors to Arduino board

(Packet given to students below)

4. Shaft (3/16 dia x 4 inches), wood
5. Screws, wood, #4 - 3/8 (to mount Arduino board) (x2)
6. Screws, wood, M1 x 4 (mini, to mount horn to hinge) (use 1.0 flat head jeweler's screwdriver)

OPTIONAL ITEMS

1. Solar panel (optional), PRT-13781 from Sparkfun, 2W, 5.3 x 4.4 x 0.2 in, $29;
   Vpeak 6.0V, Ipeak 378 mA, Ppeak 2.27W (www.sparkfun.com/products/13781) (not really needed)
2. Power supply (motor shield, DC power supply or 5 AA batteries & holder) (to operate w/o computer)

PROJECT REQUIREMENTS

1. Complete design & fabrication of mechanical components (Solidworks, laser cutting)
2. Complete circuitry & connections (Arduino, breadboard, wiring, sensors, motors)
3. Arduino programs perform the required tasks.
4. Demonstrate system functionality to instructor
5. NO purchased systems allowed.

CONSTRUCTION

You may do construction at home but you are encouraged to fabricate mechanical components using the fabrication equipment at SAC (laser cutter). Construction may also involve the use of hand tools, power tools (such as a hand drill), sander, etc. It is important that work carefully, think about what you are doing, and NOT GET HURT.
Note – Arduino is powered with 6-12VDC. The hobby servo motors typically require 4.8 - 6V power.

MECHANICAL ASSEMBLY

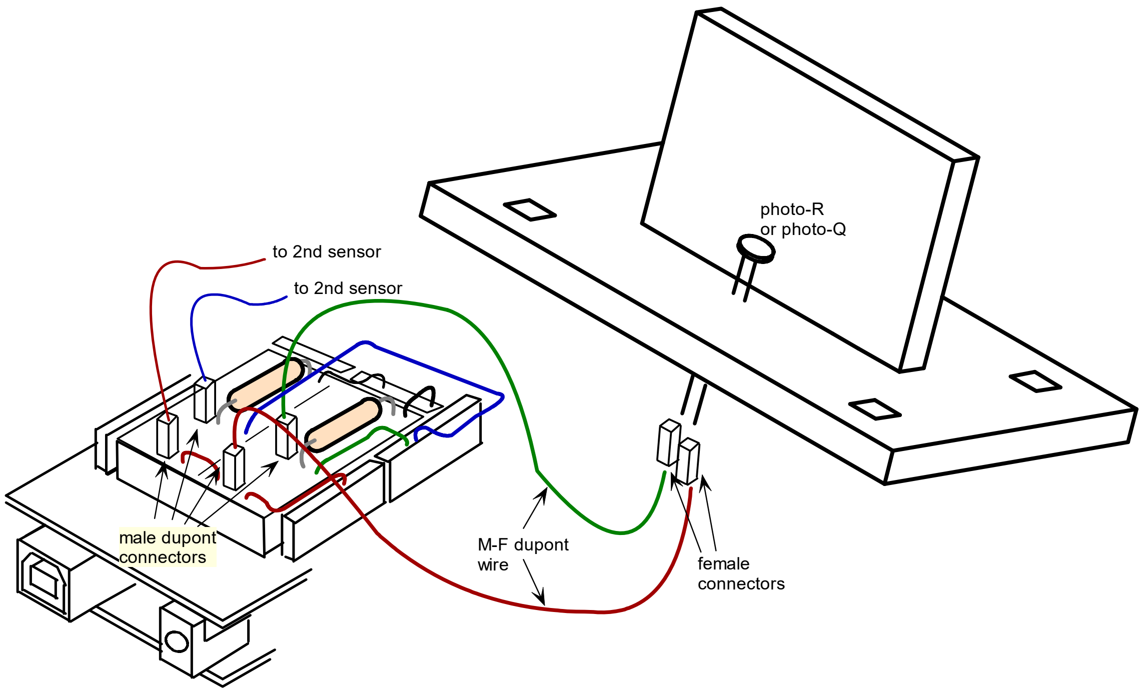
Complete laser-cutting all of the parts first. Then assemble components without glue to verify fit. Then assemble everything with glue. If gluing in class be careful not to get glue on the desks or counters (use some scrap cardboard or silicone pads). Move the hinge by hand to ensure it can move smoothly. If it gets stuck or bind, you must fix that first before mounting the servo and horn.

Additional laser requirement - add some kind of raster design to engrave on either the partition or the tilt plate.

ELECTRICAL ASSEMBLY

Create the same circuit as used in the light-sensing lab. For the project, most of the wire used should be hook-up (or spool) wire. Cut wire to length and strip. The wire should be FLUSH with the breadboard surface. Cut the resistor leads so the resistors are FLUSH with the breadboard surface. Use red wire for 5V supply. Use black wire for ground (GND). The other wires can be any color but not red or black. Use male-female (M-F) dupont wire to connect to the photoresistors (or phototransistors).

The leads of the photoresistor (phototransistor) will insert into the female connector of the M-F dupont cable. This connection may not be really secure. Any pulling will cause the photoresistor to pull from the connector. So use dupont cable with some extra length. If the sensor leads are slipping out too easily from the female connectors, you can use needle nose pliers to put "wiggly" bends in the sensor leads. This may help achieve a tighter fit. Let the instructor know if the connection is not working. We MAY have to create a solder connection (talk to the instructor before trying to solder in lab).



CENTERING SERVO

Command the servo (whose range is 0 to 180 degrees) to 90 degrees - its center position. Remove power from motor & assemble it to the platform (screw the horn to the upper hinge) so that the platform is close to LEVEL when the servo is at 90 degrees.

PROGRAMMING

The code will be the same as for the "Servo PID" lab, except that the CONSTRAIN range will be more limited (20 degrees on either side of 90 degrees, so 70 to 110). This is necessary because, for some reason, the motor will jump to an extreme position upon start up. This is a problem now that the motor is tied to a mechanism. The narrow constraint range helps to solve this problem.

```
// solar_tracker_PD_volts.ino --------------------
// sense light (2 analog voltage light detectors)
// ctl variable is DIFF in output of 2 light circuits
// filter the control u (exponFilter)
// include deadband

#include <Servo.h>
Servo myservo;

int servoPin = 10;
int sensorPin1 = A1;
int sensorPin2 = A3;

void setup() // -------------------------------------------
{
```

```
  Serial.begin(9600);
  myservo.attach(servoPin);
}

void loop() // --------------------------------------------
{
  //int servoAngle;

  int sensor1 = analogRead(A1);       //0 - 1023
  int sensor2 = analogRead(A3);
  int x = sensor2 - sensor1;          // x = diff
  //int xexpon = exponFilterCT(x);
  float u;
  static float servoAngle = 0;
  float servoAngleFiltered;

  u = computePID(x);

  servoAngle += u;                                 // add control to servoAngle
  servoAngleFiltered = exponFilterCT(servoAngle);   // filter servoAngle

  servoAngleFiltered = constrain(servoAngleFiltered, 70, 110); // constrain value

  myservo.write((int)servoAngleFiltered);         // command motor
  //myservo.write((int)servoAngle);               //option: command motor unfiltered

  float x1 = x;                                   // output to SM to debug
  float x2 = u;
  Serial.print(x1);   Serial.print("  ");
  Serial.println(x2);

  delay(50);
}

float computePID (int x) //--------------------------------
{
  float Kp = .01;             //P gain - PLAY WITH THIS
  float Kd = 0.02;            //D gain - PLAY WITH THIS
  float v, u;
  float xd = 0;
  float vd = 0;
  unsigned long t;
  static unsigned long to = 0;
  static float xo = 0;
  int minerror = 50;        // min error for deadband

  t = millis();
  //v = (float) (x - xo) / (t - to);      // compute veloc
  v = 0.0;
  //Serial.println(v);
  xo = x;               //reset old val for next loop
  to = t;
  u = Kp * (xd - x) + Kd * (vd - v);

  //add dead band code here if nec
  if (abs(x - xd) < minerror)
  {
    u = 0.0;
  }

  return u;
}
```

```
int exponFilterCT(int rawData) //------------------------------------
{
  static int lastFilteredData;
  float w = .05;
  int y;
  y = w * rawData + (1 - w) * lastFilteredData;
  lastFilteredData = y;
  return y;
}


//end ---------------------------------------------------------------
```

TESTING & EVALUATION

Tracker tilt plate should follow the light of a flashlight. Do not move the flashlight quickly. A solar tracker does not need to have fast response. It is okay for the platform may lag light movement. The platform should not oscillate.

Adjust the Kp and Kd gains in the program to obtain: proper tracking, smoothness of movement, and no excessive oscillation.

GRADING RUBRICS

80% Performance (hardware completed, tracker works properly), demo to instructor on time

-

10% Project planning (achieving week-to-week progress)
10% Effort, attitude, safe lab practices, contribution to project (if using teams)

Deductions for:
1. Performance issues (using measures above), plus deduction for oscillations,
2. Not being present on testing day
3. Exhibiting poor laboratory practices (dangerous behavior, not cleaning up, etc.)