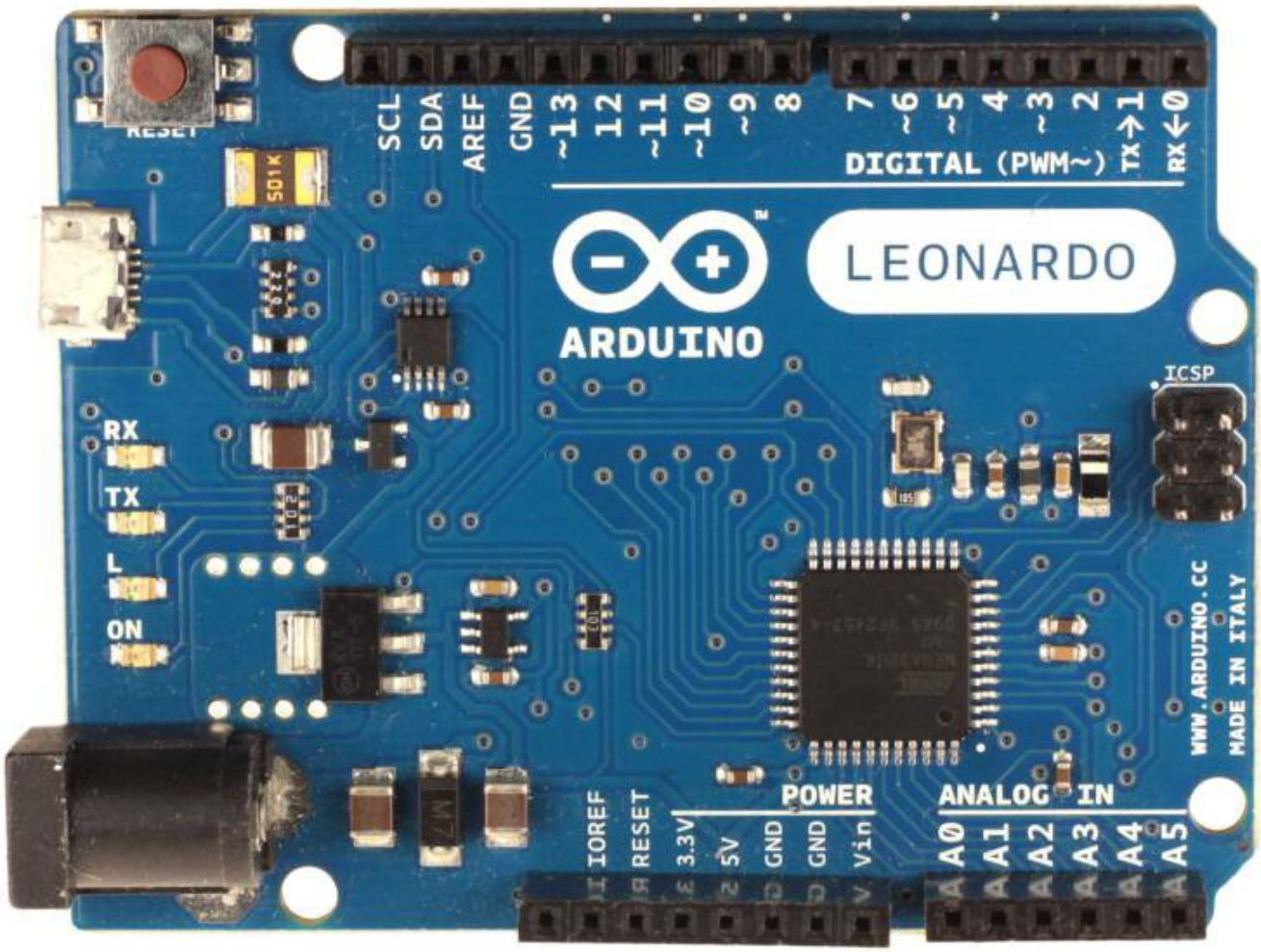
Sencillo seguidor de luz en Arduino

galideas.wordpress.com/2014/01/04/sencillo-seguidor-de-luz-en-arduino/



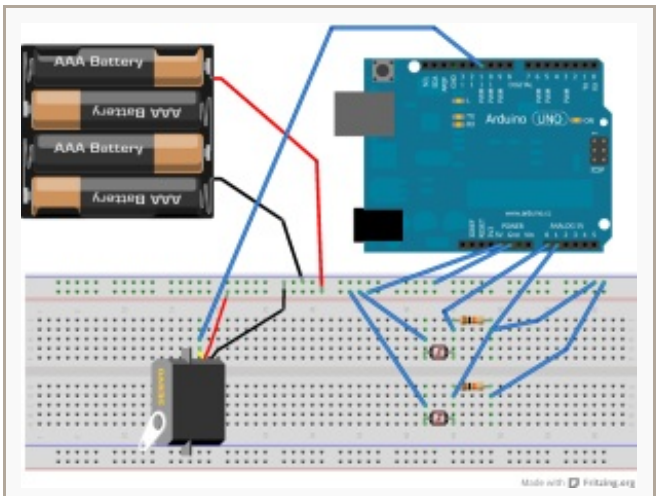
Tengo por intención construir un pequeño seguidor solar para instalarle una pequeña placa fotovoltaica y ver qué rendimientos se pueden obtener en cuanto a obtención de energía empleando seguidores solares o programando la posición solar sin necesidad sensores externos.
Evidentemente todo esto será a nivel “casero” ya que mi seguidor solar tendrá una sensibilidad muy limitada: la que pueda conseguir con dos fotorresistencias y ajustando lo más posible mi programación en Arduino.

He realizado unas primeras pruebas con un motor paso a paso (stepper motor) y dos fotorresistencias instaladas en la placa (para que quedara más bonito habría que instalarlas en el brazo de motor y “aislar” una de otra quizá con un cartón entre medio de ellas).
El montaje es algo como así:

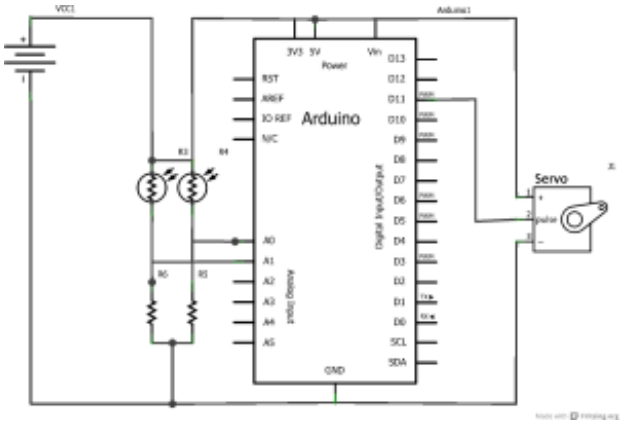
Breve explicación: el montaje es sencillo, las fotorresistencias están conectadas entre los 5V y una resistencia que hace las veces de divisor de tensión. La placa Arduino lee en sus entradas analógicas 0 y 1 el valor de la tensión después de la fotorresistencia, valor que será variable en función de la cantidad de luz que ésta reciba y que nos servirá para orientar el motor. Mi intención es que la cantidad de luz que reciban ambas resistencias esté

equilibrada lo cual significará que las dos están orientadas hacia la fuente de luz.

Para los que prefieran la vista en modo esquema:



Primera prueba de un seguidor de luz



Y a continuación el código empleado:

```
#include <Servo.h>

int const leftSensorPin = A0;
int const rightSensorPin = A1;

Servo myServo;

int angle = 90;
int leftSensorValue = 0;
int rightSensorValue = 0;
// Es casi imposible que left y right sean exactamente iguales por lo que el motor
// no se detendra, le voy a poner un colchon en el que considere left = right para
// que se detenga cuando esté en ese rango. Este valor se puede ir ajustando para
lograr
// la precisión deseada sin que perjudique el funcionamiento del sistema
int colchon = 50;

void setup(){
  myServo.attach(11);
  Serial.begin(9600);
  // Ponemos inicialmente el motor en 2°, no usar 0, puede dar vibraciones en el
motor
  // si este no consigue alcanzar los 0°
  myServo.write(angle);
}

void loop(){
  leftSensorValue = analogRead(leftSensorPin);
  delay(10);
  rightSensorValue = analogRead(rightSensorPin);
  delay(10);
  Serial.print("Left Sensor Value: ");
  Serial.print(leftSensorValue);
  Serial.print(" - Right Sensor Value: ");
  Serial.print(rightSensorValue);
  Serial.print(" - Angle: ");
  Serial.println(angle);

  if (rightSensorValue > leftSensorValue && angle > 5 && rightSensorValue-
leftSensorValue > colchon){
    angle = angle - 5;
    myServo.write(angle);
  }
  if (rightSensorValue < leftSensorValue && angle < 175 && leftSensorValue-
rightSensorValue > colchon){
    angle = angle + 5;
    myServo.write(angle);
  }
  else
    //Este equilibrado, no mover
  delay(15); // Ponemos un delay para que la placa pueda procesar los valores
}
```

El código es sencillo pero al ser el primero se explicará paso a paso:

```
#include <Servo.h>

int const leftSensorPin = A0;
int const rightSensorPin = A1;

Servo myServo;

int angle = 90;
int leftSensorValue = 0;
int rightSensorValue = 0;
// left y right nunca seran iguales asi que el motor siempre gira hasta el tope
// le voy a poner un colchon en el que considere left = right para que se
detenga
int colchon = 50;
```

Aquí se realiza la declaración de variables, para poder usar los motores paso a paso importamos la librería servo.h que viene con el IDE de Arduino de serie. Creamos una objeto myServo que será la que apunte a nuestro motor. El resto de variables son las ordinarias de cualquier programa: el pin del sensor izquierdo (A0), el pin del sensor derecho (A1), y los valores iniciales de mis variables del programa.

En los pin A1 y A0 se lee la tensión en el divisor de tensión que forman la fotorresistencia y la resistencia que lo conecta a tierra. Esta tensión será proporcional a la luz que incida sobre la fotorresistencia.

```
void setup(){
myServo.attach(11);
Serial.begin(9600);
//Ponemos inicialmente el motor en 2°, no usar 0, puede dar
problemas
myServo.write(angle);
}
```

En el código de inicialización se indica que myServo se encuentra en el pin 11, que es a dónde conectamos el cable de datos de nuestro servo. Abrimos un puerto serial de comunicación para imprimir en pantalla y poder controlar los datos. Finalmente ponemos el motor en su posición inicial con la función ***write*** del objeto myServo (incluida por defecto en la librería servo.h).

Por último, el bucle:

```
void loop(){
leftSensorValue = analogRead(leftSensorPin);
delay(10);
rightSensorValue = analogRead(rightSensorPin);
delay(10);
Serial.print("Left Sensor Value: ");
Serial.print(leftSensorValue);
Serial.print(" - Right Sensor Value: ");
Serial.print(rightSensorValue);
Serial.print(" - Angle: ");
Serial.println(angle);

if (rightSensorValue > leftSensorValue && angle > 5 && rightSensorValue-
leftSensorValue > colchon){
angle = angle - 5;
myServo.write(angle);
}
if (rightSensorValue < leftSensorValue && angle < 175 && leftSensorValue-
rightSensorValue > colchon){
angle = angle + 5;
myServo.write(angle);
}
else
//Este equilibrado, no mover
delay(15);
}
```

Lee el valor del sensor derecho y del izquierdo (con un delay para dar tiempo a la lectura de la placa, que no es inmediata). Imprime por pantalla los valores que leen los sensores y el ángulo en el que se encuentra el servo.

Finalmente se realiza la comparación para comprobar si es necesario mover el motor con dos ***if***. Explicaré el primero de ellos (el segundo es análogo). **Comprueba si** la lectura del sensor derecho es mayor que la del izquierdo **Y** si el ángulo es mayor de 5º (mi motor vibra si trato de rotarlo más hacia 0º) **Y** si la diferencia de lecturas (valor del sensor derecho menos izquierdo) es mayor que el colchón; si se cumplen **simultáneamente** las tres condiciones rota el motor hacia la derecha 5º (angle = angle – 5º; si hubiese usado 2º de rotación, en el if hubiera usado 2º en lugar de 5º también. Esto es para no llevar el motor hasta los 0º dónde se producen vibraciones en mi motor) rotando el motor de nuevo con la función ***write***.

Realizo una comprobación similar para ver si tengo que rotar a la izquierda.

Si no se cumplen ninguno de los dos condicionales, el motor está equilibrado (la diferencia de lecturas será inferior al colchón o estará en su tope de movimiento) con lo cual no haré nada.

De momento está funcionando bastante bien. Iré ajustando el colchón cuando haya incorporado la placa fotovoltaica al sistema para tratar de que sea lo más sensible posible.

También incluiremos un condensador de desacoplamiento cuando el motor tenga que mover la placa fotovoltaica para evitar las caídas de tensión en la placa Arduino durante el arranque del motor (ya que este absorbe más corriente en el arranque que cuando se está ya moviendo).

Ya os iré contando cómo va el proceso.

Un saludo.

Edición: Podéis consultar la segunda parte aquí.

Anuncios