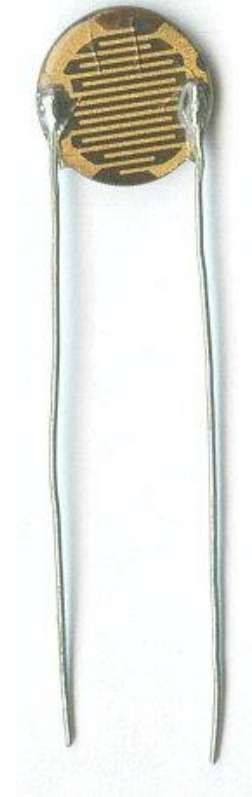
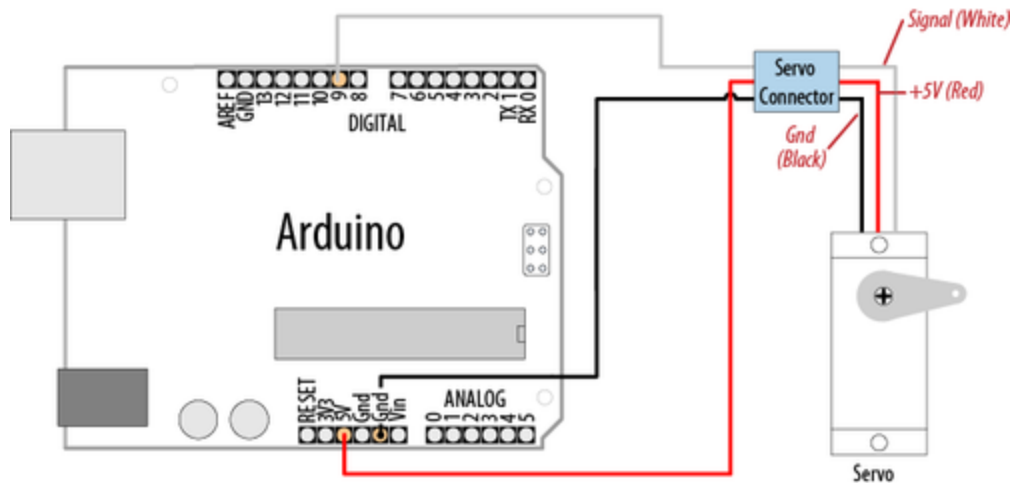


# INSEGUITORE SOLARE CON ARDUINO

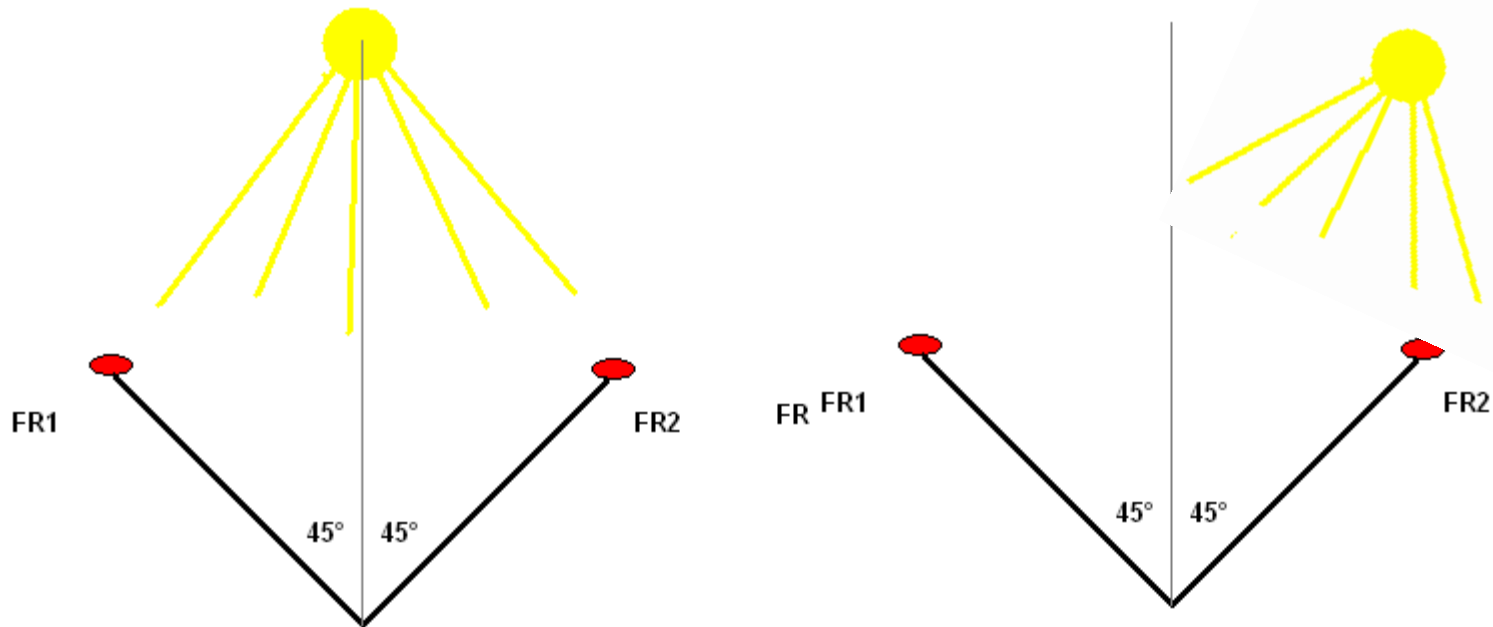
Dispositivo che insegue il movimento solare realizzato con:

1. Arduino
2. 2 foto resistenze
3. 1 servomotore



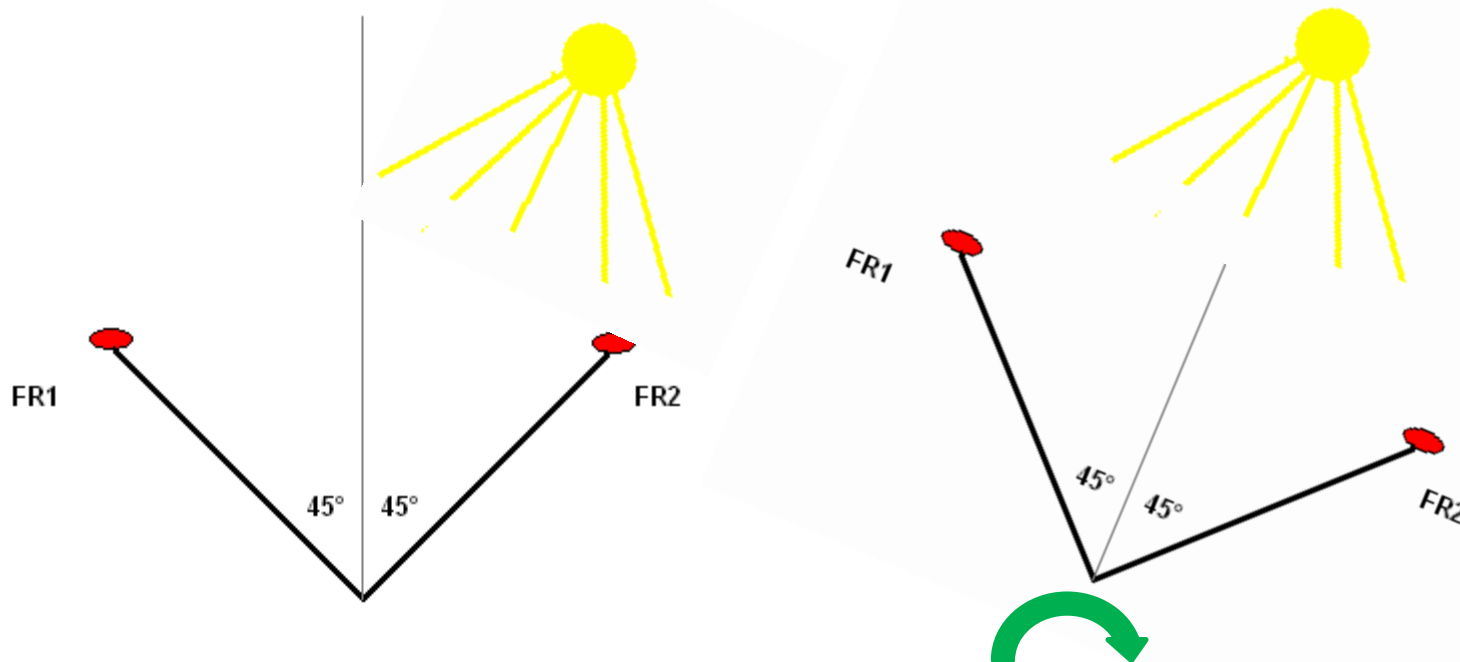
# PRINCIPIO DI FUNZIONAMENTO

Le due fotoresistenze devono essere posizionate in modo da risultare a  $90^\circ$  l'una rispetto all'altra in modo che quando l'illuminazione è verticale presentino lo stesso valore di resistenza, mentre i valori di resistenza si sbilanciano allo spostamento della fonte di luce.

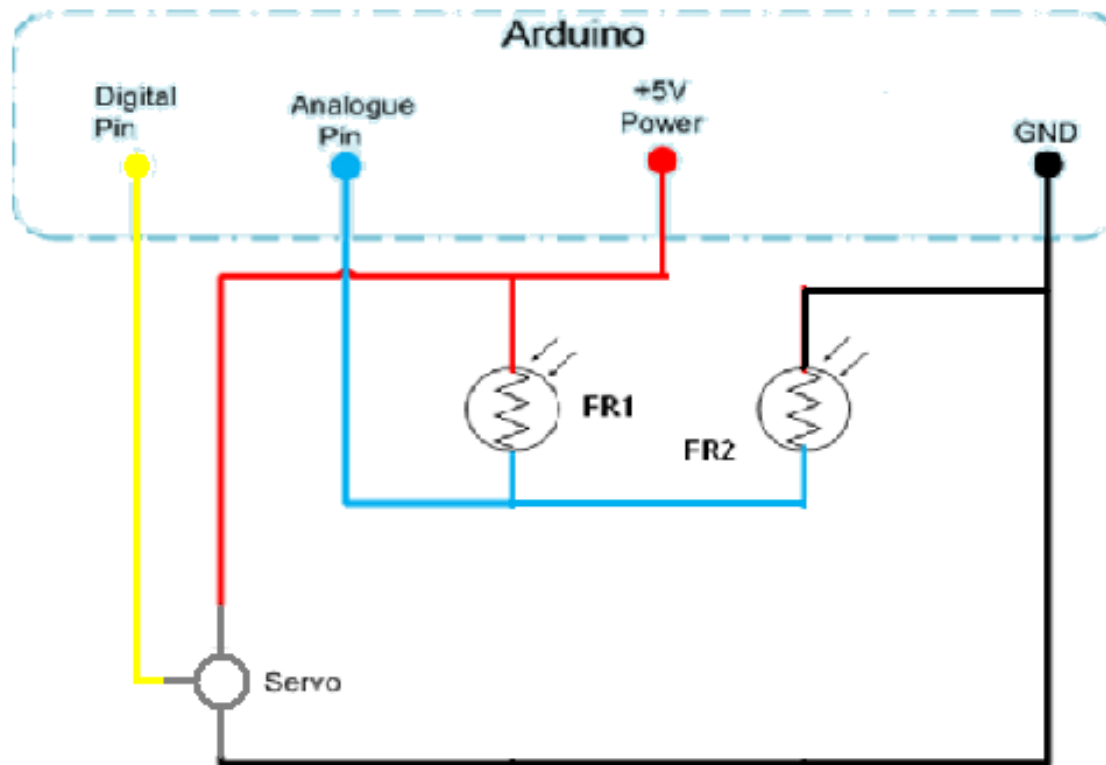


# PRINCIPIO DI FUNZIONAMENTO

Quando la fonte di luce si muove e di conseguenza il valore delle fotoresistenze varia, Arduino deve rilevare il cambiamento rispetto alla condizione iniziale e far muovere il servo nella direzione della resistenza che riceve più luce, finché la luminosità rilevata non sarà ancora circa uguale.



# IL CIRCUITO



# STRUTTURA DEL PROGRAMMA

Attivazione della libreria

```
#include <Servo.h>
```

## Dichiarazione delle variabili

1. variabile di tipo *integer* a cui assegnare il pin a cui sono collegati i sensori
2. variabile di tipo *integer* a cui assegnare il pin a cui è collegato il servomotore
3. variabile di tipo *integer* a cui si assegnerà il valore letto dai sensori
4. variabile di tipo *integer* per indicare la posizione, in gradi, del servo, allo start (90°)
5. variabile di tipo *integer* a cui assegnare una tolleranza entro la quale il servo non deve ruotare anche se il sensore rileva variazioni dal valore centrale

Inizializzare l'oggetto Servo assegnandolo all'istanza *myservo*

```
Servo myservo;
```

# STRUTTURA DEL PROGRAMMA

```
void setup() {
```

- Impostare il pin a cui è connesso il sensore in modalità INPUT
- Usando *attach(pin)* per l'oggetto *myservo* indicare a quale pin è collegato il servomotore
- Usando *write(grad)* impostare i gradi a cui il servo deve posizionarsi allo start

# STRUTTURA DEL PROGRAMMA

```
void loop() {
```

Assegnare alla variabile che indica il sensore, il valore letto sul corrispondente pin

**SE**

Valore letto < di 512 – la tolleranza → incrementare la posizione in gradi del servo nel senso opportuno, un grado alla volta , controllando di non superare il limite estremo (se)

Valore letto >di 512 +la tolleranza → incrementare la posizione in gradi del servo nel senso opposto al precedente un grado alla volta senza superare l'altro limite estremo

Con *write( gradi )* applicare al servo lo spostamento determinato

Inserire un ritardo di 100 ms

# LIBRERIA SERVO.h

Per l'azionamento di un servo si deve far riferimento alla libreria standard *Servo.h*, che si compone delle seguenti istruzioni.

## **Attach()**

Associa la variabile servo ad uno specifico pin.

Sintassi *servo.attach(pin)*

Parametri servo: variabile di tipo Servo. Pin: numero del pin hardware utilizzato

## **Detach()**

Dissocia la variabile servo al pin specificato.

Sintassi: *servo.detach()*.

Parametri: servo= variabile di tipo servo.



# LIBRERIA SERVO.h

## Read()

Legge l'attuale posizione del servo corrispondente all'ultima posizione passata con l'istruzione `write()`.

Sintassi: `servo.read()`. Parametri servo: variabile di tipo servo.

Ritorno: l'angolo del servo da 0 a 180 gradi.

## Write()

Invia il valore in gradi relativo alla posizione del perno del servo.

Un valore 0 corrisponde alla massima rotazione a sinistra, mentre 180 equivale alla massima rotazione a destra; il valore 90 indica la posizione centrale.

Sintassi: `servo.write(angle)`. Parametri servo: variabile di tipo servo.

Angle: valore corrispondente alla rotazione in gradi.

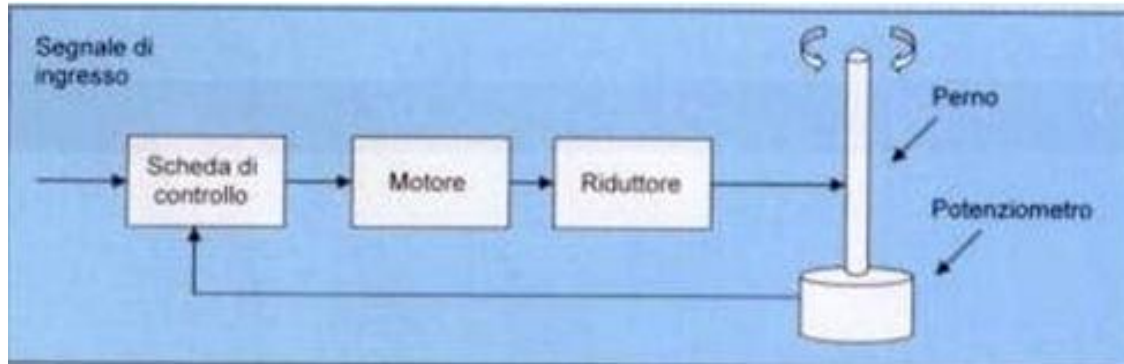
## WriteMicroseconds()

Imposta la posizione del servo come valore relativo alla durata dell'impulso espressa in microsecondi. Normalmente un valore 1000 corrisponde alla massima rotazione a sinistra, 2000 alla massima rotazione a destra ed il valore 1500 corrisponde alla posizione centrale (neutro)

Sintassi `writeMicroseconds(μS)` Parametri servo: variabile di tipo servo.

μS: valore in microsecondi relativo alla posizione del servo.

# SERVOMOTORI



**Un servomotore è costituito da un piccolo motore in corrente continua che, mediante un sistema di ingranaggi, fa ruotare un perno sul quale è calettato un piccolo potenziometro: la lettura del valore resistivo di questo potenziometro fornisce la posizione esatta del perno.**

**Un circuito elettronico realizzato in tecnologia SMD (surface mounting device) provvede al controllo bidirezionale del motore ed al corretto posizionamento del perno in relazione al segnale elettrico di comando.**

**Tale posizionamento avviene confrontando il valore in tensione fornito dal potenziometro con quello ricavato dal segnale di ingresso e ruotando di conseguenza il motore sino a quando questi due valori non coincidono perfettamente.**

**In questo modo si ottiene un controllo di posizione molto veloce e preciso, comandabile con segnali elettrici.**

# SERVOMOTORI

Il cavetto di collegamento è composto da un filo di riferimento (GND), un filo per l'alimentazione (da 4,8 a 6 volt) ed un filo per il segnale di comando. La rotazione del perno è di  $\pm 90$  gradi rispetto alla posizione centrale.

colore	funzione
Nero o marrone	Negativo di alimentazione (GND)
rosso	Positivo di alimentazione (+Vcc)
Giallo o bianco	Segnale di comando (ingresso)



# SERVOMOTORI

Il segnale di controllo è di tipo PWM (Pulse Wide Modulation) formato da impulsi ad onda rettangolare ripetuti ogni 20 ms, la cui "larghezza" permette di impostare la posizione del perno del servo.

La posizione centrale si ottiene quando gli impulsi hanno una durata di 1.5 ms. Questo tipo di segnale digitale si presta benissimo ad essere generato da una logica programmabile, quindi i servo possono essere comodamente gestiti dai microcontrollori.

