

Introduzione

Un pan-tilt è un meccanismo utilizzato per orientare o posizionare un dispositivo in due direzioni, orizzontale (pan) e verticale (tilt). Questo tipo di sistema consente di controllare con precisione l'angolo di visualizzazione o l'orientamento di una telecamera, un sensore, un laser o qualsiasi altro dispositivo montato su di esso.

L'integrazione di un laser in un sistema pan-tilt può aggiungere diverse funzionalità e capacità al dispositivo, ampliandone le applicazioni e le prestazioni. Nel mio caso integra il puntamento di un punto preciso presente nella base su cui esso è montato.

Pan-Tilt: Concetti di Base

Un pan-tilt è composto da due parti principali: il pan, che consente il movimento orizzontale, e il tilt, che permette il movimento verticale. Il funzionamento di un pan-tilt coinvolge motori e meccanismi di controllo per regolare la posizione e l'orientamento del dispositivo montato su di esso.

-Motori per il pan e il tilt: Ogni asse del pan-tilt è azionato da un motore dedicato.

Ho utilizzato motori servo per fornire un movimento preciso.

-Potenziometro: Per controllare con precisione la posizione del pan-tilt, ho utilizzato un potenziometro collegato ai motori. Questo sensore fornisce feedback sulla posizione attuale dell'asse, consentendo al sistema di regolare con precisione il movimento.

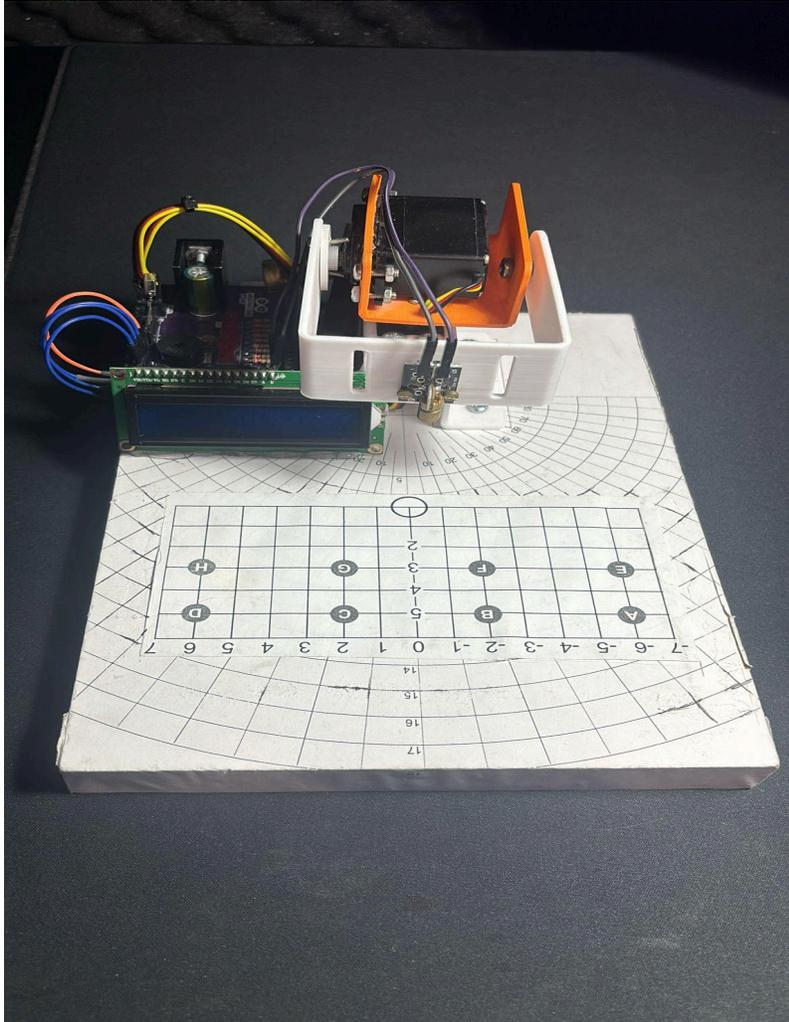
-Interfaccia utente: Serve per consentire all'utente di controllare il pan-tilt, in questo caso ho utilizzato l'applicazione Arduino ID, utilizzato per scrivere e caricare il codice (in linguaggio "C").

-Implementazione con telecomando infrarossi e sensore: Per rendere il tutto più facile e comprensibile ho utilizzato un telecomando infrarossi, permette di spostare il pan-tilt nel punto desiderato, grazie anche al sensore che riceve il segnale dato dal telecomando.

-Schermo lcd: Lo schermo lcd serve in questo caso a far vedere le coordinate esatte del punto preciso scelto dall'utente.

Conclusione

In conclusione, il pan-tilt utilizza motori, meccanismi di trasmissione e feedback di posizione per controllare con precisione l'orientamento del laser montato su di esso, mentre l'interfaccia utente permette all'utente di interagire con il sistema.



Codice Arduino:

```
/* Librerie necessarie */
#include <IRremote.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

/* Configurazione Tasti */
/* HEX          Tasto
   0xBA45FF00   ON/OFF
```

0xB946FF00	Vol +
0xB847FF00	FUNC/STOP
0xBB44FF00	Precedente
0xBF40FF00	Pausa
0xBC43FF00	Successivo
0xF807FF00	Giù
0xEA15FF00	Vol -
0xF609FF00	Su
0xE916FF00	0
0xE619FF00	EQ
0xF20DFF00	ST/REPT
0xF30CFF00	1
0xE718FF00	2
0xA15EFF00	3
0xF708FF00	4
0xE31CFF00	5
0xA55AFF00	6
0xBD42FF00	7
0xAD52FF00	8
0xB54AFF00	9

*/

/* Pin Sensore Infrarossi */

```
int IRSENS = 11;  
IRrecv ricevitore(IRSENS);  
decode_results results;
```

/* Pin Laser */

```
int laser = A3;  
int statolaser;
```

/* Dichiarazione Servo */

```
Servo tilt;  
Servo pan;
```

/* LCD */

```
LiquidCrystal_I2C lcd(0x27, 16, 2);  
int tempo = 1000;  
int dd = 300;
```

```

/* Variabili x y z */
int x, y, z;

void setup() {
  Serial.begin(9600);
  ricevitore.enableIRIn();
  /* Laser */
  pinMode(laser, OUTPUT);

  /* Servomotori */
  tilt.attach(10);
  pan.attach(9);
  pan.writeMicroseconds(1466.7);
  tilt.writeMicroseconds(2000);

  /* LCD */
  lcd.init();
  lcd.backlight();
  delay(250);
  lcd.noBacklight();
  delay(250);
  lcd.backlight();
  delay(250);
  lcd.noBacklight();
  delay(250);
  lcd.backlight();
  Schermo();
  lcd.setCursor(4, 1);
  lcd.print("Laser OFF");
}

void loop() {
  if (ricevitore.decode()) {
    uint32_t code = (ricevitore.decodedIRData.decodedRawData);

    switch (code) {
      case 0xB946FF00: //premuto il tasto Vol +
        x--;
        if (x == 0 || x <= 10 || x >= -10) {
          lcd.clear();

```

```
}  
Schermo();  
vai();  
break;
```

```
case 0xB847FF00: //premuto il tasto Func/Stop  
  x++;  
  if (x == 0 || x <= 10 || x >= -10) {  
    lcd.clear();  
  }  
  Schermo();  
  vai();  
  break;
```

```
case 0xBF40FF00: //premuto il tasto Pausa  
  y--;  
  if (y == 0 || y <= 10 || y >= -10) {  
    lcd.clear();  
  }  
  Schermo();  
  vai();  
  break;
```

```
case 0xBC43FF00: //premuto il tasto Successivo  
  y++;  
  if (y == 0 || y <= 10 || y >= -10) {  
    lcd.clear();  
  }  
  Schermo();  
  vai();  
  break;
```

```
case 0xEA15FF00: //premuto il tasto Vol -  
  z--;  
  if (z == 0 || z <= 10 || z >= -10) {  
    lcd.clear();  
  }  
  Schermo();  
  vai();  
  break;
```

```
case 0xF609FF00: //premuto il tasto Su
  z++;
  if (z == 0 || z <= 10 || z >= -10) {
    lcd.clear();
  }
  Schermo();
  vai();
  break;
```

```
case 0xBA45FF00: //premuto il tasto ON/OFF
  statolaser = digitalRead(laser);
  statolaser = !statolaser;
  digitalWrite(laser, statolaser);
  Schermo();
  break;
```

```
case 0xE916FF00: //premuto il tasto 0
  x = 0;
  y = 0;
  z = 0;

  Schermo();
  vai();
  break;
```

```
case 0xF30CFF00: //premuto il tasto 1
  x = -6.5;
  y = 5.5;

  Schermo();
  vai();
  break;
```

```
case 0xE718FF00: //premuto il tasto 2
  x = -2.3;
  y = 5;

  Schermo();
  vai();
```

```
break;
```

```
case 0xA15EFF00: //premuto il tasto 3
```

```
x = 2;
```

```
y = 5;
```

```
Schermo();
```

```
vai();
```

```
break;
```

```
case 0xF708FF00: //premuto il tasto 4
```

```
x = 5,5;
```

```
y = 5.3;
```

```
Schermo();
```

```
vai();
```

```
break;
```

```
case 0xE31CFF00: //premuto il tasto 5
```

```
x = -6;
```

```
y = 3;
```

```
Schermo();
```

```
vai();
```

```
break;
```

```
case 0xA55AFF00: //premuto il tasto 6
```

```
x = -2;
```

```
y = 3;
```

```
Schermo();
```

```
vai();
```

```
break;
```

```
case 0xBD42FF00: //premuto il tasto 7
```

```
x = 2;
```

```
y = 3;
```

```
Schermo();
```

```
vai();
```

```

        break;

    case 0xAD52FF00: //premuto il tasto 8
        x = 5.5;
        y = 3;

        Schermo();
        vai();
        break;
    }
    ricevitore.resume();
}
}

```

```

void Schermo() {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("x=");
    lcd.print(x);
    lcd.setCursor(6, 0);
    lcd.print("y=");
    lcd.print(y);
    lcd.setCursor(11, 0);
    lcd.print("z=");
    lcd.print(z);

    lcd.setCursor(4, 1);
    if (statolaser == HIGH) {
        lcd.print("Laser ON");
    } else if (statolaser == LOW) {
        lcd.print("Laser OFF");
    }
}
}
/* Variabili Utilizzate */

```

```

/* q (dai grafici) */
double qpan = 1466.7; //ins
double qtilt = 2000; //ins

```

```

/* m (dai grafici) */

```

```

double mpan = 9.722222; //ins
double mtilt = -9.444444; //ins

/* distanza tra scacchiera e centro base (cm) */
double distanzabase = 6;

/* Link */
double L1; //il laser (cambia con ogni posizione)
double L2 = 6; //distanza tra vite tilt e laser (cm)
//ins

/* Triangolo Rettangolo che si forma tra il pan e il target */
double d; //((base) distanza del target dal centro del pan
double a = 7; //(altezza) altezza nodo tilt dal piano
//ins
double c; //(ipotenusa)

void vai() {
    double tgfi, tgalfa, tggamma;
    double firad, alfarad, gammarad;
    double figradi, alfagradi, gammagradi;

    double impgamma, impfi;

    /* calcolo posizione pan rispetto al target */
    tgfi = x / (y + distanzabase); //calcolo tgfi
    firad = atan(tgfi); //calcolo arctgfi (rad)
    figradi = ((180 * firad) / 3.14); //conversione firad to figradi

    impfi = mpan * figradi + qpan; //impulso per comando base (y=mx+q)
    pan.writeMicroseconds(impfi);

    /* calcolo posizione tilt rispetto al target */
    d = sqrt((x * x) + ((y + distanzabase) * (y + distanzabase)));
    tgalfa = a / d;
    alfarad = atan(tgalfa);
    alfagradi = ((180 * alfarad) / 3.14);

    c = sqrt((d * d) + (a * a));

```

```
L1 = sqrt((c * c) - (L2 * L2));
```

```
tggamma = L1 / L2;
```

```
gammarad = atan(tggamma);
```

```
gammagradi = (((180 * gammarad) / 3.14) - alfagradi);
```

```
impgamma = mtilt * gammagradi + qtilt;
```

```
tilt.writeMicroseconds(impgamma);
```

```
/* calcolo posizione tilt rispetto al target con aggiunta di asse z */
```

```
double f;
```

```
double alfaPgradi, alfaPrad;
```

```
f = a - z;
```

```
c = sqrt((d * d) + (f * f));
```

```
L1 = sqrt((c * c) - (L2 * L2));
```

```
gammarad = atan(L1 / L2);
```

```
gammagradi = ((180 * gammarad) / 3.14);
```

```
alfaPrad = atan(f / d);
```

```
alfaPgradi = ((180 * alfaPrad) / 3.14);
```

```
gammagradi = gammagradi - alfaPgradi;
```

```
impgamma = mtilt * gammagradi + qtilt;
```

```
tilt.writeMicroseconds(impgamma);
```

```
}
```

[Link sito web](#)