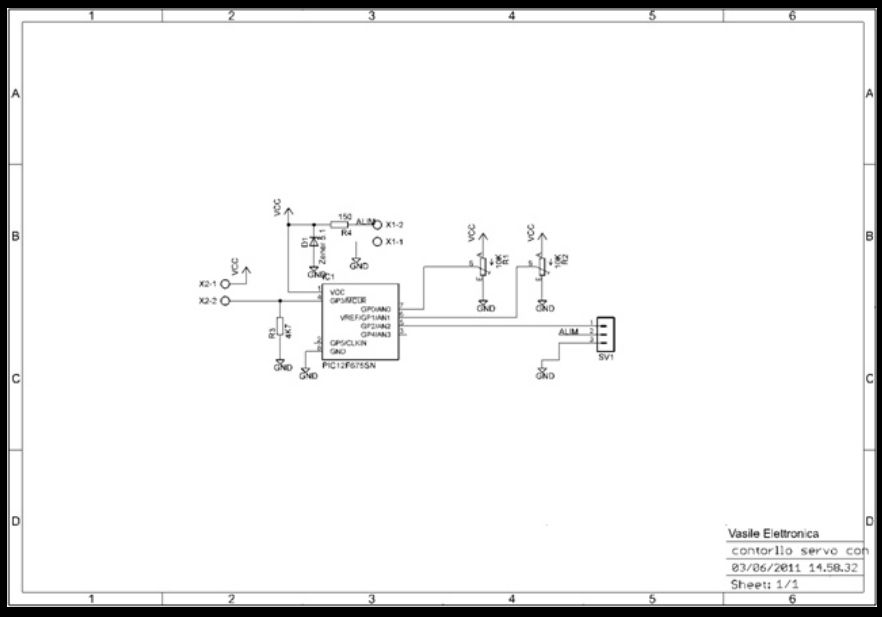
Muovere servo con pulsante

Introduzione:

Un mio amico stava lavorando a un progetto e mi chiese se era possibile ruotare un oggetto (tipo l'apertura di una porta) utilizzando un motore elettrico e un pulsante. Valutando il problema, la soluzione più semplice ma anche la più funzionale era l'utilizzo di un servo; infatti l'angolo da percorrere non superava i 100°.

Era stato anche valutato la possibilità di utilizzare un normale motore in DC, ma decisamente molto più complesso da gestire per avere un risultato soddisfacente.

Descrizione:



Le parti che compongono il circuito sono veramente poche, qualche resistenza, trimmer, zener, morsettiera e il micro. La gestione del movimento è affidata al pic 12F675 di tipo SMD che provvede a indirizzare sull'uscita un "treno" di impulsi necessari al servo per eseguire i movimenti.

L'angolo di lavoro ed il tempo sono settabili attraverso i due trimmer R1-R2 collegati rispettivamente all'ingresso GP0 e GP1 del micro; l'interruttore a pulsante deve essere montato sula morsettiera X2, che fornisce o meno un segnale positivo sull'ingresso GP3; la resistenza R3(pull-down) mantiene a livello logico basso l'ingresso quando il pulsante è aperto.

Il servo deve essere connesso al connettore SV1 da cui preleva l'alimentazione e il segnale di controllo.

Lo zener D1 mantiene stabile la tensione a 5,1 volt indispensabile per il pic, infatti il micro non sempre è compatibile con l'alimentazione dei servo.

Come già detto nell'introduzione, questo circuito è stato ideato per poter ruotare un oggetto con un angolo di 100° utilizzando un interruttore a pulsante. L'angolo di movimento è settabile tramite il potenziometro R1, variabile tra un valore di 0-130°; con il potenziometro R2 è invece possibile settare la velocità dello spostamento.

Firmware:

Sono stati sviluppati due diversi firmware, il primo permette di utilizzare un pulsante normalmente aperto, il secondo invece permette di utilizzare un interruttore a pulsante di tipo on/off.

Il listato è stato sviluppato con il programma Mikroc 8.2 "demo version".

Versione con pulsante normalmente aperto:

```
int tempo, gradi, x, y, valore, gradiRil, i;


void main (void){


  ANSEL  = 1;                          // configurazione input alanogico
  TRISIO = 0b001011;                   // gpio0-1-3 in input - gpio2 in output
  valore = 40;
```

```
  i=0;


while (1){
inizio:


 tempo = ADC_READ(0);
 delay_ms(5);


 gradiRil = ADC_READ(1);
 delay_ms(5);


 tempo = (tempo+50)/50;
 gradiRil = (gradiRil+50)/7;


    for(x=0;x<tempo;x++){
    for(y=0;y<valore;y++){
            gpio.f2 = 1;
}
gpio.f2 = 0;
delay_ms(20);
}


if(gpio.f3)
   i=i^1;
   while(gpio.f3){}



if(i)
   gradi = gradiRil;
  else
   gradi = 40;


if(valore==gradi)
   goto inizio;


 if(valore>gradi)
   valore--;
 else
   valore++;


}}
```

Dopo aver dichiarato le variabili e settate le porte, si procede con le istruzioni contenute all'interno del ciclo infinito while. Utilizzando l'istruzione ADC_read, si va a leggere il valore analogico della tensione presente sull'ingresso GP0 per la velocità di spostamento e GP1 per l'angolo.

Tramite l'istruzione:

```
tempo = (tempo+50)/50;
gradiRil = (gradiRil+50)/7;
```

```
    for(x=0;x<tempo;x++){
    for(y=0;y<valore;y++){
            gpio.f2 = 1;
}
gpio.f2 = 0;
delay_ms(20);
}
```

si dividono i valor1 rilevati dagli ADC in modo da farli assumere un valore compatibile con le istruzioni successive; successivamente i due cicli FOR concatenati provvedono a "generare" sull'uscita GP2 il segnale di controllo per il servo.

L' istruzione:

```
if(gpio.f3)
   i=i^1;
   while(gpio.f3){}



if(i)
   gradi = gradiRil;
  else
   gradi = 40;
```

Permette di attribuire alla variabile "gradi", il valore di riposo(40) oppure il valore rilevato dall'ADC(gradiRil).

Infine,

```
if(valore==gradi)
   goto inizio;

 if(valore>gradi)
   valore--;
 else
   valore++;
```

l'istruzione seguente "ragiona" nel seguente modo: se il valore della variabile "valore" è uguale al valore della variabile "gradi" torna all'inizio del while; altrimenti se il valore della variabile "valore" è maggiore del valore della variabile "gradi", decrementa "valore"; se questa condizione è falsa, incrementa "valore".

Versione con interruttore a pulsante:

```
int tempo, gradi, x, y, valore, gradiRil;


void main (void){


  ANSEL  = 1;                          // configurazione input alanogico
  TRISIO = 0b001011;                   // gpio0-1-3 in input - gpio2 in output
  valore = 40;
```

```
while (1){
inizio:

 tempo = ADC_READ(0);
 delay_ms(5);

 gradiRil = ADC_READ(1);
 delay_ms(5);

tempo = (tempo+50)/50;
gradiRil = (gradiRil+50)/7;

    for(x=0;x<tempo;x++){
    for(y=0;y<valore;y++){
            gpio.f2 = 1;
}
gpio.f2 = 0;
delay_ms(20);
}



if(gpio.f3)
   gradi = gradiRil;
  else
   gradi = 40;

if(valore==gradi)
   goto inizio;

 if(valore>gradi)
   valore--;
 else
   valore++;

}}
```

Il listato è identico al primo, l'unica differenza è nell'istruzione:

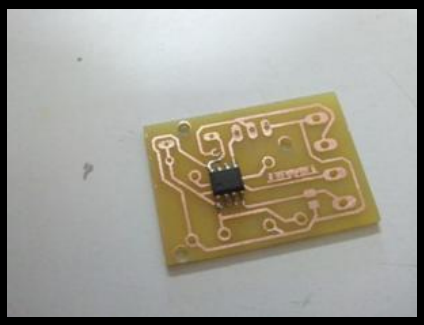
```
if(gpio.f3)
   gradi = gradiRil;
  else
   gradi = 40;
```

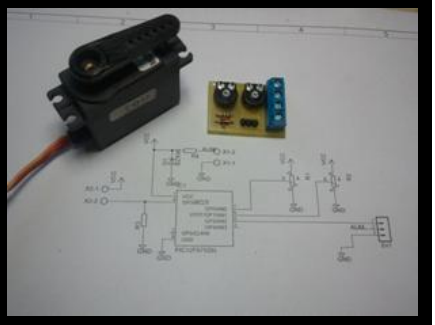
Infatti nel primo listato si utilizzava la variabile "i" per attribuire alla variabile "gradi", il valore di riposo(40) oppure il valore rilevato dall'ADC(gradiRil); qui invece, siccome sull'ingresso gpio3 alla pressione del tasto è sempre presente il valore logico 1,

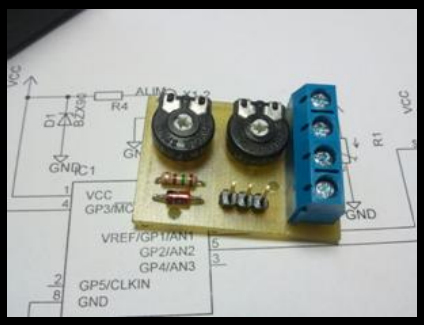
non è necessario utilizzare un'altra istruzione (chiamiamola di mantenimento).

Realizzazione:

Per rendere il circuito più "piccolo", ho utilizzato il PIC in smd con package di tipo SOIC; un altro componente smd(0805) utilizzato è la resistenza R3.







Video:

Download:

schema elettrico;

print bottom;

print serigrafia;

file da caricare nel pic versione .hex

file da caricare nel pic versione .rar

Redatto da Vasile Egidio il giorno 23 luglio 2011 - Photos and videos by Samsung Galaxy S

Commenti: 2

- Marco:

Salve per prima cosa la ringrazio infinitamente perchè ho finalmente trovato quello che cercavo un semplice circuito per comandare un servo.
mastico qulacosina di elettronica ma buio totale sui pic.
sono un appasionato di modellismo ferroviario e il servo incomincia ad essere un importante sviluppo per il movimento degli scambi.
sono in possesso di due modelli di pic il 12c508a e il 12f6751/sn1023
ho gia acquistato un programmatore usb per dispositivi pic flash
e le vorrei porre tre domande
1) quale dei due pic devo usare? anche se non so come programmare il 12f675 visto che sull'espanzione a zoccolo ci posso montare e quindi programmare solo il 12c508
2) come faccio a scaricare il file sul pic ?
3) e per ultimo volevo sapere se è meglio separare le due alimentazione pic/servo ?
mille ma mille grazie
Marco Mazzanti

- Egidio

Salve,
le rispondo subito alle domande:
- il pic da utilizzare deve essere necessariamente con memoria flash, ovvero che abbia la F all'interno del nome, per intenderci: 16F628, 16F84A ecc.; quindi il 12C508A è da escludere. Il pic 12f6751/sn1023 invene non sono riuscito a trovarlo, il nome è scritto correttamente?
- per scrivere il file sul pic è necessario dotarsi di un programmatore, quello che ha acquistato è della microchip?
- io consiglio di separare le due alimentazioni solo in caso di servo molto "potenti" che richiedono una grande quantità di corrente.

per qualsiasi altro dubbio non esiti a contattarmi.

saluti
Egidio

Scrivi un commento:

Vasile Egidio © 2010-2011 - Tutti i diritti riservati - Vietata la riproduzione, anche parziale del sito