Gestione di un dispositivo Pan & Tilt con la scheda Arduino Esplora

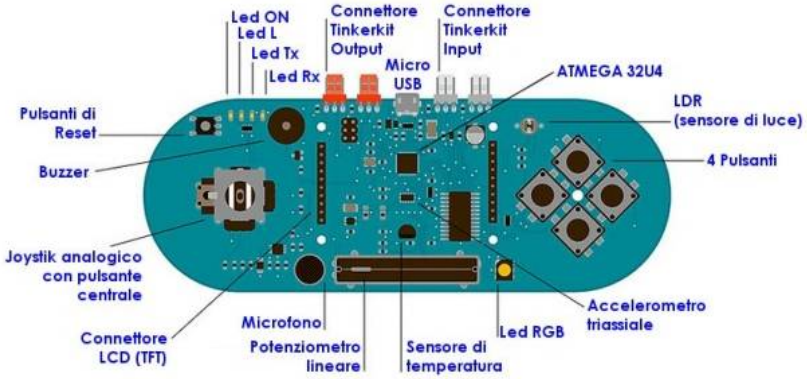
Posted By *Adriano Gandolfo* On 20 gennaio 2015 @ 7:00 In Arduino,LED Lighting & Opto | 51 Comments

Lavorare con servomotori ed una scheda Arduino UNO è una cosa piuttosto semplice, basta collegare l'alimentazione e il cavo del segnale del servo ad una delle tante porte digitali presenti. La gestione è facilitata dall'opportuna libreria già presente tra quelle standard. Ma cosa succede se come nel caso della scheda Arduino Esplora le porte digitali disponibili sono solamente due e i servo da comandare sono molti di più? La soluzione più semplice è quella presentata in questo articolo, che mostrerà come utilizzare dei semplici moduli appositamente creati per comandare i servomotori: possono essere gestiti tramite una linea seriale per la ricezione dei comandi e che permettono di comandare un minimo di 6 servomotori.

Scheda Arduino Esplora

In questo articolo spiegheremo come collegare dei servomotori alla **scheda Arduino Esplora**, della quale si è parlato più diffusamente nell'articolo **Scopriamo la nuova scheda Arduino Esplora** [1], in cui ne trovate una descrizione completa.
La scheda Arduino Esplora ha le seguenti caratteristiche:

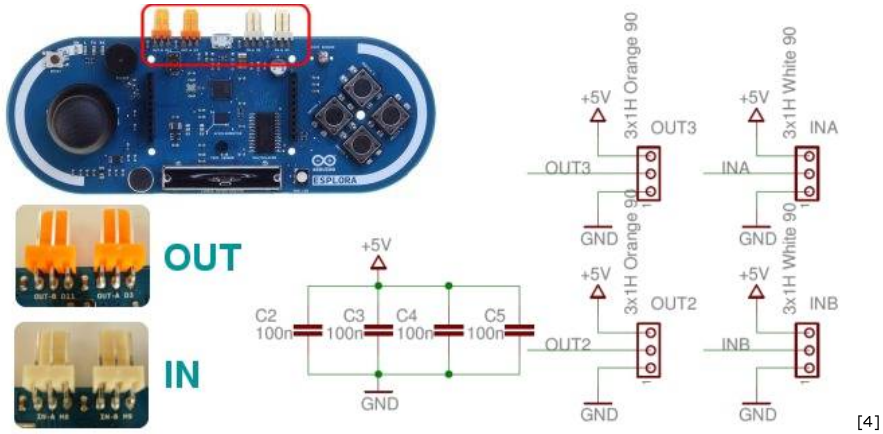
Processore **ATMEGA32U4** [2] con bus a 8 bit prodotto dalla Atmel con architettura di tipo RISC, Velocità di clock 16 MHz, memoria Flash da 32 kB di cui 4 kB utilizzati dal bootloader, memoria EEPROM da 1 kB, 20 porte Digital pin I / O, 12 Canali di ingresso analogici, 7 canali PWM, tensione di funzionamento 5V, 1 porta USB Full speed
Un joystick analogico a due assi (X e Y) con pulsante centrale
4 pulsanti disposti a rombo
Un potenziometro lineare a cursore
Un microfono per rilevare il rumore ambientale.
Un sensore di luce per la misurazione dell'intensità luminosa
Un sensore per la misurazione della temperatura ambiente
Un accelerometro triassiale (X, Y e Z)
Un buzzer
Un LED luminoso a LED tipo RGB con elementi Rosso Verde e Blu.
2 Ingressi per collegare i moduli sensore della serie Tinkerkit.
2 uscite per collegare i moduli attuatori della serie Tinkerkit.
Un connettore per l'inserimento del display TFT a colori, dotato di uno slot per scheda SD



[3]

Ingressi e uscite Tinkerkit

Come abbiamo visto, la scheda Arduino Esplora ha quattro connettori compatibili con il sistema Tinkerkit, due ingressi di colore bianco e due di uscita di colore arancio.
Questi hanno tre terminali, due per l'alimentazione (+5V e GND) e uno per l'ingresso o l'uscita del segnale.
Per il nostro progetto utilizzeremo i due connettori di uscita colorati in Arancione, connessi alle porte D3 e D11.

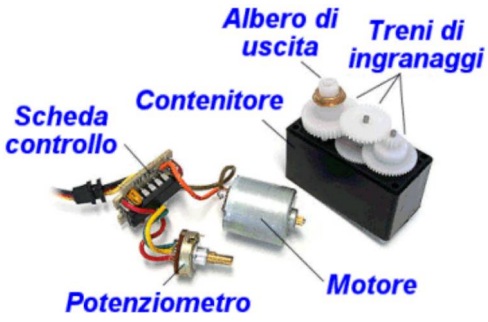


[4]

Qualche cenno inerente i servomotori.

Il servomotore è un piccolo dispositivo che incorpora un motore DC, un treno di ingranaggi, un potenziometro, un circuito integrato e un albero di uscita.

Dei tre fili che sporgono dal corpo motore, uno è per l'alimentazione, uno è per la massa e uno è una linea di ingresso di controllo. L'albero del servo può essere posto a determinate posizioni angolari inviando un segnale codificato. Finché esiste il segnale codificato sulla linea di ingresso, il servo manterrà la posizione angolare dell'albero; se il segnale codificato cambia, si modifica la posizione angolare dell'albero.



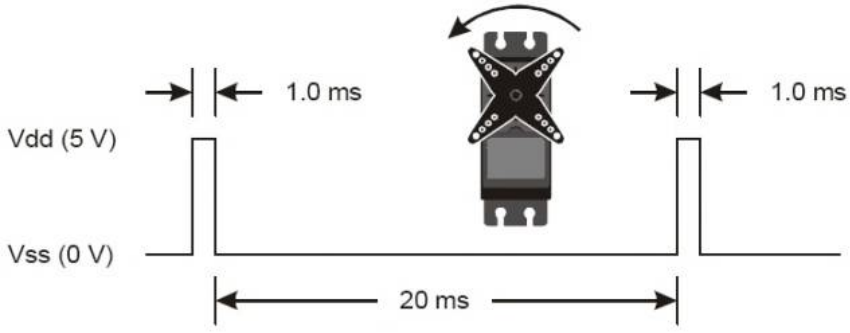
[5]

Il circuito di controllo ha il compito di generare gli impulsi per controllare il servo.

Possono essere generati impulsi compresi tra 0.25 ms e 2.75 ms che coprono la richiesta della maggioranza dei tipi di servo per rotazioni oltre i 180 gradi.

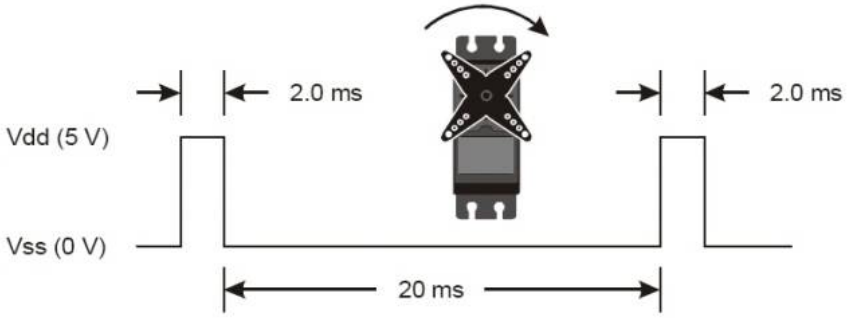
Generalmente con un impulso di durata pari a 1.5 ms, il perno del servomotore si posiziona esattamente al centro del suo intervallo di rotazione; da questo punto il perno può ruotare fino a -90 gradi (senso antiorario) se l'impulso fornito ha una durata inferiore a 1.5 ms e fino +90 gradi (senso orario) se l'impulso fornito ha durata superiore a 1.5 ms. Il rapporto esatto tra la rotazione del perno e la larghezza dell'impulso fornito può variare tra i vari modelli di servo.

Se questi valori sono ripetuti con un intervallo non superiore a 20 ms, la posizione raggiunta sarà mantenuta.



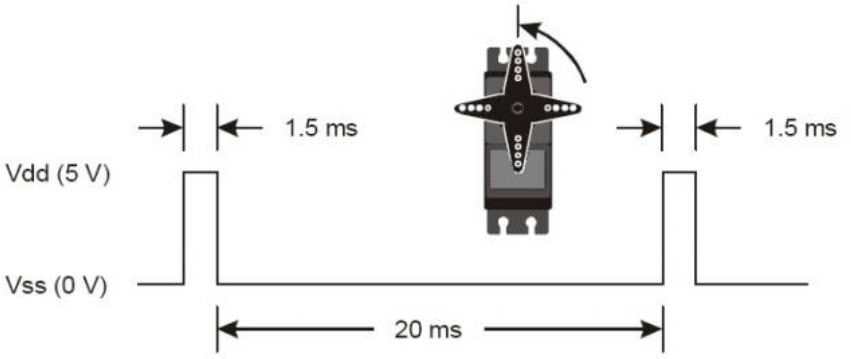
[6]

Diagramma temporizzazione per rotazione antioraria.



[7]

Diagramma temporizzazione per rotazione oraria.



[8]

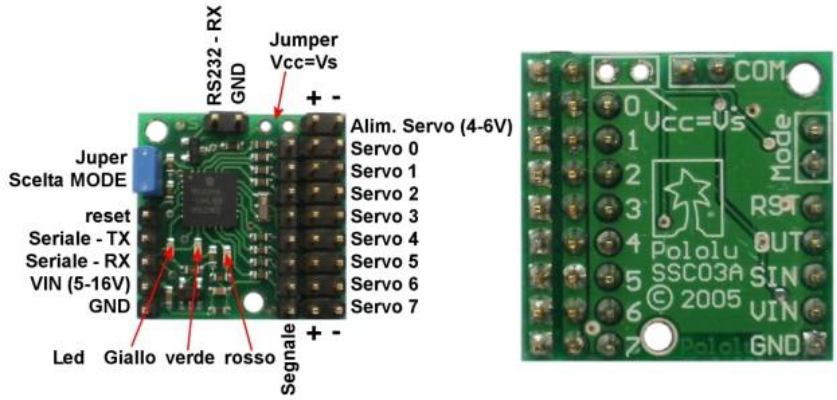
Diagramma temporizzazione per posizionamento al centro.

Micro Serial Servo Controller.

La prima scheda di controllo servo che analizziamo è denominata **Micro Servo Serial Controller** [9] ed è prodotta dalla **Pololu** [10]: si presenta come uno stampato di piccole dimensioni di poco più di due centimetri di lato.

[11]

Su di essa trovano posto, oltre al processore a 8 bit con memoria flash tipo **PIC16F628A** [12] della **Microchip**, il risuonatore per il clock, un regolatore di tensione, tre LED di stato e vari connettori con funzione di alimentazione di interfaccia e per il collegamento dei servomotori. La scheda può controllare fino a 8 servocomandi con trasmissione tramite una linea seriale.
Nonostante le sue minuscole dimensioni, è ricca di funzioni: può controllare la velocità e la posizione di ognuno degli 8 servi indipendentemente, la velocità della porta seriale è rilevata automaticamente nel range da 1200 a 38400 Baud ed ha 3 LED di stato.
I collegamenti del Servo Controller sono visibili nella foto di seguito, la maggior parte dei pin sono identificati sul retro della scheda del Servo Controller. Tutte le piazzole connesse a massa hanno forma quadra.



[13]

Specifiche Tecniche:

Dimensioni stampato	~ 23x23 mm
Numero di porte per servo	8
Range larghezza impulso	0.25-2.75 ms
Risoluzione	0.5 µs (~0.05 °)
Tensione di alimentazione	5-16V

Tensione I/O	0 e 5 V
Velocità seriale	1200 - 38400 (autodetect)
Consumo	5 mA (valore medio)

Alimentazione.

I servo sono alimentati con una tensione compresa tra i 4.8 ed i 6 V. Questa alimentazione può essere fornita tramite il connettore posto in alto a destra.
L'alimentatore deve essere in grado di garantire la corrente richiesta dai servo che, nel caso siano tutti collegati e vengano mossi simultaneamente, può essere molto elevata (vicina a 10 A). Il processore necessita di una sua alimentazione separata che può essere compresa tra i 5 e i 16 V, fornita tramite il connettore in basso a sinistra (PIN segnati con VIN e GND).

Segnale di controllo.

Il Servo controller è controllato tramite una logica seriale TTL (0-5 V) fornita al PIN "logic-level serial input".
Esistono poi i pin "reset" e "logic-level serial input" che nella maggioranza delle applicazioni, possono essere lasciati sconnessi.

Opzioni di interfaccia

Si può comunicare con il Servo controller con due diversi protocolli di comunicazione.
La scelta avviene utilizzando un cavallotto che è letto all'atto dell'accensione della scheda. Per cambiare il protocollo occorre resettare la scheda.

Modo Pololu: è la modalità di funzionamento predefinita con cavallotto non inserito. In questa maniera, il Servo Controller può essere connesso con altre apparecchiature seriali.
Questo modo permette anche l'accesso a tutte le caratteristiche speciali come settaggio velocità, range e settaggio posizione neutra.

Modo Mini SSC II: Questa modalità operativa è scelta inserendo il cavallotto sull'apposito spinotto. In questo modo il Servo Controller risponde al protocollo usato dal controllore **Mini SSC II Servo** [14] realizzato dalla **Scott Edwards Elettronics.**
Questo protocollo è più semplice e permette solamente di specificare il numero del servo e la sua posizione.

Micro Maestro.

La seconda scheda che analizziamo e che utilizzeremo per le prove, è la **Micro Maestro** [15] prodotta sempre dalla **Pololu**,

[16]

questa scheda fa parte di una serie di altri 4 modelli che differiscono principalmente per il numero di servomotori che possono comandare che va da 6 a 24.

[17]

Sulla scheda troviamo un processore **PIC18F14K50-I/SS** [18] a 8 bit, 48MHz, memoria 16 kB, memoria dati da 256 bytes, con implementati un'interfaccia USB V2.0, un risuonatore per il clock, un regolatore di tensione, tre LED di stato e vari connettori con funzione di alimentazione di interfaccia e per il collegamento dei servomotori.
Supporta tre metodi di controllo: USB per una connessione diretta con il computer, seriale TTL per l'utilizzo con sistemi embedded, e internal

scripting per applicazioni integrate, senza bisogno di un controllore esterno.
I canali possono essere configurati come uscite servo per utilizzarli con i normali servo RC da modellismo o con gli ESC (Electronic Speed Control), come uscite digitali, oppure ancora come ingressi analogici.
Con il modulo è possibile il controllo integrato di velocità ed accelerazione su ciascun canale così da ottenere movimentazioni morbide e senza scatti.
Ogni scheda può essere collegata in cascata con altri controllers Pololu su una singola linea seriale. Il firmware della scheda è aggiornabile.
Il modulo Micro Maestro ha tre LED indicatori di tre colori diversi: il LED verde indica lo stato del dispositivo USB, il LED rosso error/user di solito indica un errore, il LED giallo indica lo stato di controllo.

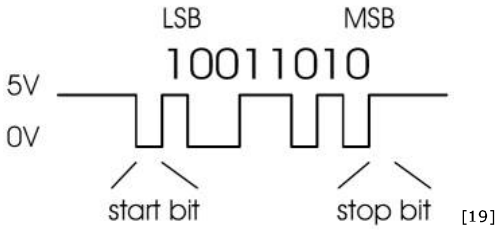
Specifiche Tecniche :

Dimensioni stampato	~ 21mm×30mm
Numero di porte per servo	6
Range larghezza impulso	64 e 3280 µs
Risoluzione	0,25 µs (~0.025 °)
Tensione di alimentazione	5-16V
Tensione I/O	0 e 5 V
Velocità seriale	300 - 115200 bps
Consumo	30 mA (valore medio)

Alimentazione.
L'alimentazione del processore può essere fornita tramite la presa USB o da un alimentatore esterno che abbia un valore compreso tra i 5 e i 16V collegato agli ingressi VIN e GND. I servo sono alimentati da un connettore presente in alto a destra nella scheda.
L'alimentazione ai servi è fornita direttamente, senza passare attraverso un regolatore, in modo che le uniche restrizioni siano la potenza dell'alimentatore e che sia in grado di fornire la corrente necessaria; approssimativamente l'assorbimento di corrente di un servo di media grandezza è di 1 A. È presente un'uscita 5V che consente di alimentare dispositivi esterni, la corrente è però limitata a 50mA.

Segnale di controll0.
Il Servo controller è controllato tramite una logica seriale **TTL (0-5 V)** fornita al PIN **TX/RX**. Esiste poi il pin **RST** che può anche essere lasciato sconnesso.

La comunicazione seriale.
I comandi seriali inviati ai moduli controller devono essere in blocchi di 8 bit, senza parità e con un bit di stop, oppure abbreviato **8N1.**



[19]

Il livello logico deve essere non-invertito, considerando che "0" è inviato come 0V, mentre "1" è inviato come 5V.

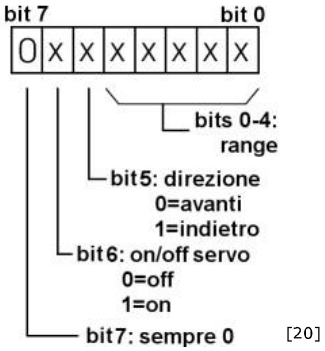
Comandi della scheda Micro Serial Servo Controller - Modo Pololu.
In questa modalità sono possibili molti comandi del movimento del servomotore.
Baud Rate: la serie di velocità di trasmissione disponibili in questa maniera è approssimativamente tra 2000 e 40000 baud.
Protocollo: per comunicare con il Servo Controller occorre inviare una sequenza di 5 o 6 byte.

Il primo byte è di sincronizzazione e deve essere sempre 0x80.
Il secondo byte identifica l'apparecchiatura Pololu nel nostro caso 0x01.
Il terzo byte è uno di sei valori per i diversi comandi (vedere sotto).
Il quarto byte è il numero del servo che si vuole comandare.
Il 5 o 6 byte sono i parametri per il comando impartito.
I valori dei byte da 2 a 6 sono compresi tra 0-0x7F (0-127).

Start byte = 0x80	Numero ID = 0x01	Comando	Dato 1	Dato 2

Comando 0: Impostazione parametri (1 byte di dati)

Il byte 7 è sempre 0;
Il byte 6 specifica se un servo è attivo oppure no; 1 attiva il servo, 0 (default) lo disattiva;
Il byte 5 stabilisce la direzione di rotazione; 0 (default) avanti, 1 indietro;
I byte 0-4 stabiliscono il range del movimento.



[20]

Comando 1: Impostazione velocità (1 byte di dati)
Questo comando permette di variare la velocità con cui si muove il servomotore.
Di default la velocità è posta a 0, in questo modo il servo si sposterà immediatamente alla posizione fissata.
Se il valore di velocità è diverso da 0, il servo si sposterà progressivamente dalla vecchia alla nuova posizione. Con una velocità di 1, gli impulsi cambiano di 50 µs per secondo; la massima velocità è di 6.35 ms quando la velocità è posta a 127.

Comando 2: Impostazione posizione, 7-bit (1 byte di dati)
Quando questo comando è inviato, il valore del dato è moltiplicato per il range impostato per il servomotore corrispondente e corretto per il settaggio neutrale.
Questo comando può essere utile nell'andare velocemente ad una posizione in quanto si utilizzano solamente 5 byte per impostare la posizione. Assegnando una posizione, automaticamente il servo sarà attivato.

Comando 3: Impostazione posizione, 8-bit (2 byte di dati)
Questo comando è simile a quello a 7-bit eccetto per il fatto che devono essere inviati due byte.

Comando 4:Impostazione Posizione, Assoluto (2 byte di dati)
Permette il controllo diretto della posizione del servo. Range Neutrale e impostazione direzione non hanno effetto in questo comando. La serie di valori validi è compresa tra 500 e 5500. Assegnando una posizione automaticamente il servo sarà attivato.

Comando 5: Impostazione Neutrale (2 byte di dati)
L'impostazione neutrale è applicabile solamente a comandi a 7 e 8 bit. Il valore neutrale imposta il valore medio del range e corrisponde per il bit 7 a 63.5 e per il bit 8 a 127.5. La posizione neutrale è una posizione assoluta simile al comando 4, e impostando la posizione neutrale, il servo si sposterà a quella posizione. Il valore predefinito è 3000. Può essere utile cambiare la posizione neutrale per calibrare un sistema in modo da correggere tolleranze dei meccanismi meccanici connessi ai servomotori.

Comandi della scheda Micro Maestro - Modo Pololu.

I comandi per la gestione di servomotori disponibili per il **Micro Maestro** in modo Pololu sono:

Set Target – Impostazione della posizione del servo;
Set Speed- Imposta la velocità con cui il servo raggiunge la posizione assegnata;
Set Acceleration - Questo comando imposta l'accelerazione del servomotore;
Get Position - Questo comando consente di ricevere il valore di posizione di un servomotore;
Get Moving State - Questo comando è utilizzato per determinare se le uscite servo hanno raggiunto la loro posizione finale o stanno ancora cambiando;
Get Errors - Utilizzato per esaminare gli errori che il Maestro ha rilevato;
Go Home - Questo comando porta tutti i servi alle loro posizioni iniziali.
Esaminiamo in dettaglio alcuni comandi:
Set Target
Protocollo : 0xAA, numero del dispositivo, 0x04, numero servo, target low bit, target high bit.
Si utilizza questo comando per ruotare il servo di un determinato angolo.
Nel nostro caso il target rappresenta la larghezza dell'impulso da trasmettere in unità di quarti di microsecondi.
Un valore di 0 indica alla scheda di interrompere l'invio di impulsi al servo. Ad esempio: se si vuole inviare sul canale 2 un impulso di 1500 ms (1500 × 4 = 6000 = 01011101110000 in binario), è possibile inviare la seguente sequenza di byte:

in binario: 10000100, 00000010, 01110000, 00101110
in esadecimale: 0x84, 0x02, 0x70, 0x2E
in decimali: 132, 2, 112, 46
Set Speed
Protocollo: 0xAA, numero del dispositivo, 0x07, numero servo, accelerazione bit bassi, accelerazione bit alti.
Questo comando limita la velocità con cui cambia il valore di uscita di un canale servo. Il limite di velocità è espresso in unità di (0,25 ms) / (10 ms). Una velocità di 0 (valore di default) rende la velocità illimitata, in modo che l'impostazione della destinazione influenzi immediatamente la posizione. Si noti che la velocità reale alla quale si muove il servo è limitata dalle caratteristiche del servo stesso, la tensione di alimentazione e carichi meccanici.

Set Acceleration
Protocollo: 0xAA, numero de dispositivo, 0x09, numero servo, accelerazione bit bassi, accelerazione bit alti.
Questo comando imposta l'accelerazione del servomotore. Il limite di accelerazione è un valore compreso tra 0 a 255 in unità di (0,25 ms) / (10 ms) / (80 ms). Il valore 0 corrisponde a nessun limite di accelerazione. Il limite di accelerazione determina la velocità che impiega il servo ad

arrivare alla velocità massima; il tempo impiegato in decelerazione provoca un movimento relativamente fluido da un punto ad un altro. Con l'impostazione di minima accelerazione 1, l'uscita del servo impiega circa 3 secondi per passare agevolmente da un valore di 1 ms ad un valore di 2 ms.

Get Position

Protocollo: 0xAA, numero di dispositivo, 0x10, numero del canale.

Questo comando consente al dispositivo di comunicare con la scheda di controllo per ottenere il valore di posizione di un servo. La posizione è inviata come risposta a due byte immediatamente dopo che il comando viene ricevuto. Il valore rappresenta la larghezza d'impulso che la scheda sta trasmettendo al servo.

Get Moving State

Protocollo: 0xAA, numero di dispositivo, 0x13

Risposta: 0x00 se i servo non si muovono, 0x01 se i servo si stanno muovendo. Questo comando è utilizzato per determinare tramite le uscite se i servo hanno raggiunto la loro posizione o stanno ancora ruotando. Il valore sarà 1 purché vi sia almeno un servo che sia ancora in movimento. L'utilizzo di questo comando è necessario per avviare diversi movimenti servo e attendere che tutti i movimenti per finire prima di passare alla fase successiva del programma.

Get Errors

Protocollo: 0xAA, numero di dispositivo, 0x21

Risposta: bit di errore 0-7, errore di bit 8-15. Si può utilizzare questo comando per esaminare gli errori che la scheda controllo ha rilevato. Sul manuale, alla Sezione 4.b, sono elencati gli errori specifici che possono essere rilevati dal Maestro. Il registro di errore viene inviato come risposta a due byte immediatamente dopo che il comando è stato ricevuto, allora tutti i bit di errore vengono cancellati.

Go Home

Protocollo: 0xAA, numero di dispositivo, 0x22

Questo comando invia a tutti i servi la loro posizioni iniziale, proprio come se si fosse verificato un errore. Per i servi e le uscite impostate su "Ignora", la posizione sarà invariata.

Uso della libreria per la gestione del modulo Micro Maestro.

Come abbiamo visto, i comandi per la gestione dei servo sono molti, e realizzare un programma di gestione da utilizzare con la scheda Arduino potrebbe essere complesso.

Per questo motivo è molto più comodo utilizzare un'apposita libreria che si occupi di inviare i giusti comandi.

Ricercando in rete si è trovata una libreria già pronta, questa denominata **PMCtrl** realizzata da **Graham Sortino** [21]

Per utilizzare la libreria sarà sufficiente accedere alla pagina e scaricare l'ultima versione disponibile, questa dovrà essere salvata all'interno della cartella Library dell'IDE di Arduino.

Funzioni di assegnazione e inizializzazione

Per utilizzarla, occorre prima di tutto includerla nel programma con il comando.

```
#include <PMCtrl.h>
```

Quindi occorre definire il nome con cui si chiamerà il modulo, a quali pin sarà connesso per il dialogo seriale e a quale velocità si utilizzerà per la comunicazione; il comando da utilizzare è

```
PMCtrl servoCtrl (11, 3, 9600); //RX, TX, Baud
```

Comandi disponibili nella libreria.

A questo punto si potranno utilizzare i vari comandi disponibili nella libreria, questi sono:

servoCtrl.setServoSpeed (servoSpeed, channel, 12)

Il comando imposta la velocità servoSpeed con cui il servo si muove, channel è il servo che di cui vogliamo variare la velocità e deviceID per il Micro Maestro è il numero 12.

servoCtrl.setTarget (pos, channel, 12);

Il comando imposta la posizione pos che vogliamo far assumere al servo, channel è il servo che di cui vogliamo variare la posizione e deviceID è sempre il numero 12.

servoCtrl.goHome(12)

Il comando sposta tutti i servi collegati alla loro posizione di riposo

servoCtrl.getPosition(channel,l12)

Il comando restituisce la posizione attuale di un servo

getErrors (channel, 12)

Il comando restituisce le informazioni riguardanti lo stato di errore del servo connesso a canne, per i codici vedere la **relativa pagina** [22].

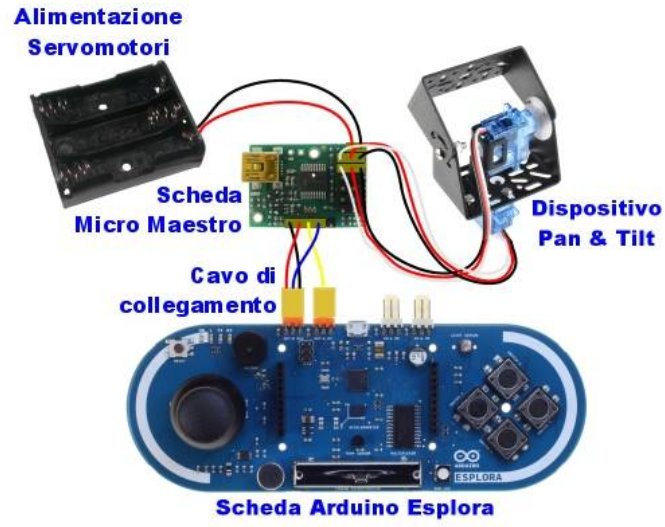
Circuito di prova

Per testare il modulo con la scheda Arduino Esplora, si utilizzerà il circuito di prova mostrato di seguito che è molto semplice e prevede i seguenti elementi.

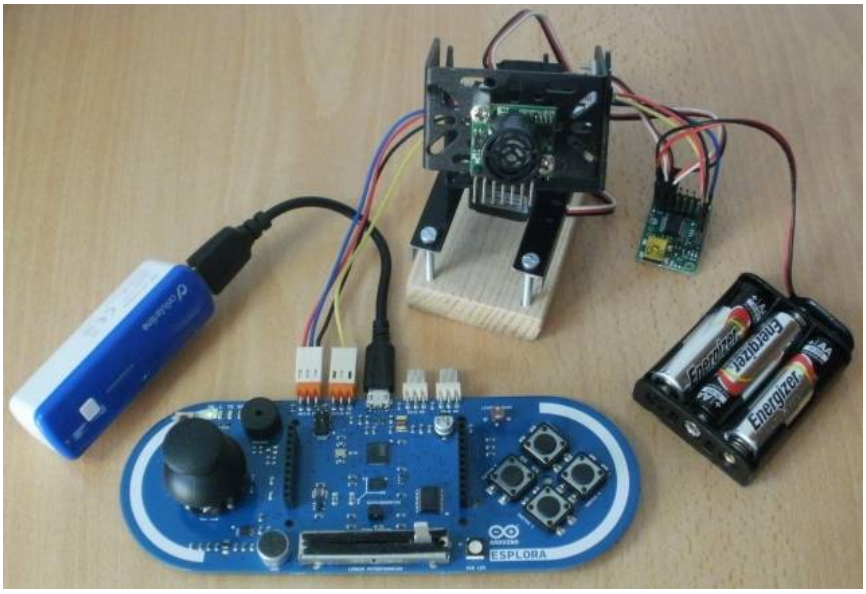
Scheda Arduino Esplora;
Scheda Micro Maestro;
Due servomotori per esempio connessi ad un semplice dispositivo pan & Tilt;
Cavo di collegamento seriale tra scheda Arduino Esplora e Micro Maestro;

Fonte di alimentazione per i servomotori, per esempio porta batterie e 3 batterie tipo AA 1,5V;

Fonte di alimentazione per la scheda Arduino Esplora, per esempio una batteria/caricabatteria [23] che fornisca 5V.



[24]

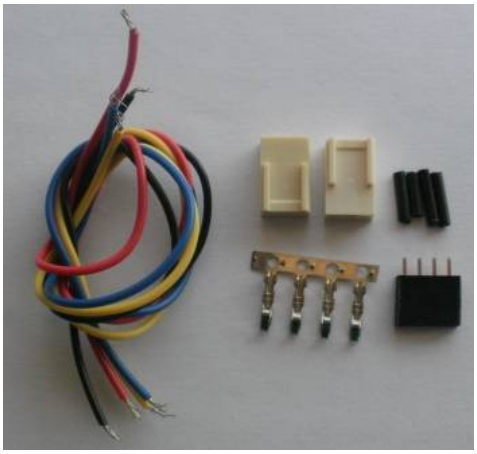


[25]

Realizzazione cavi di collegamento tra scheda Arduino Esplora e Micro Maestro

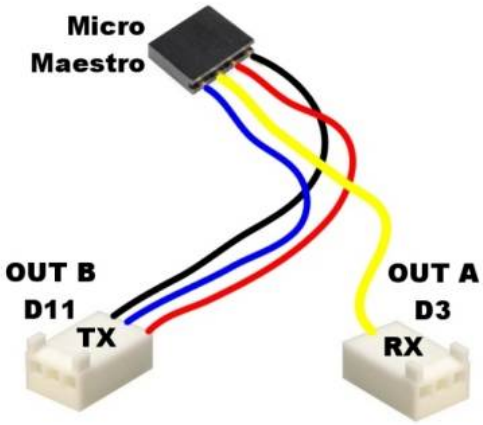
Per realizzare il cavo di collegamento tra scheda Arduino Esplora e Micro Maestro sono necessari:

N° 2 connettori a tre pin femmina passo 2,54 mm tipo Molex 22-1-3037 (codice **RS 679-5375P** [26]);
N° 1 connettore pin strip femmina 1X4 passo 2.5 mm;
N° 4 spezzoni di cavi lunghezza a piacere di colori: nero, rosso, blu, giallo.
N° 4 spezzoni di guaina termoretraibile.



[27]

Nella figura è riportato come devono essere collegati i 4 cavi che sono Positivo e Negativo, alimentazione, TX e RX.



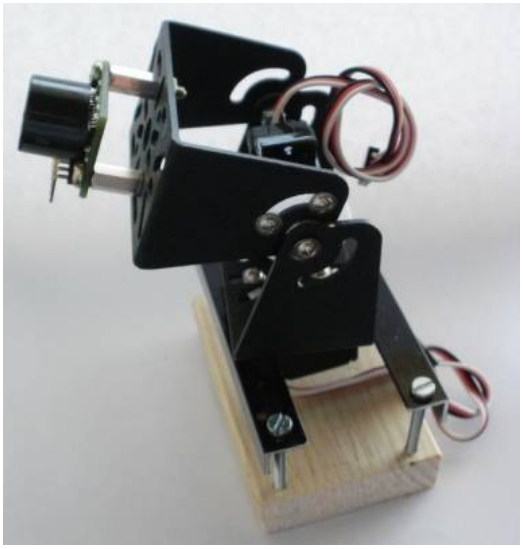
[28]

Dispositivo Pan & tilt

Per il test si è utilizzato un piccolo dispositivo Pan & Tilt, questo è venduto dalla **SparkFun codice ROB-10335** [29].
Nel kit sono presenti i componenti meccanici cioè due staffe e tutto l'hardware necessario per il fissaggio, i servomotori devono essere presi a parte, quelli consigliati sono dei servo compatibili con il modello Hitec HS-55 codice **Sparkfun ROB-09065** [30]

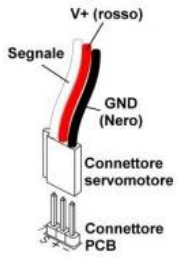

[31]

Per le istruzioni di montaggio si potrà fare riferimento alla **seguente pagina** [32].


[33]

Collegamento dei Servomotori

Quando si collegano i servomotori occorre prestare attenzione poiché un errato collegamento può danneggiarli. Assicurarsi che il filo del segnale (di solito bianco o giallo) sia posto verso l'interno della scheda mentre il cavo nero verso l'orlo esterno.



[34]Programma di test

Il programma di test (**Arduino_esplora_servo** [35]) è molto semplice, è utilizzato il Joystick presente sulla scheda Arduino Esplora per comandare un dispositivo Pan & Tilt dotato di due servomotori.
Il programma utilizza le seguenti librerie:
Gestione Arduino Esplora
#include <Esplora.h>
Gestione della Micro Maestro
include <PMCtrl.h>
che a sua volta utilizza la libreria gestion seriale
#include <SoftwareSerial.h> [36]

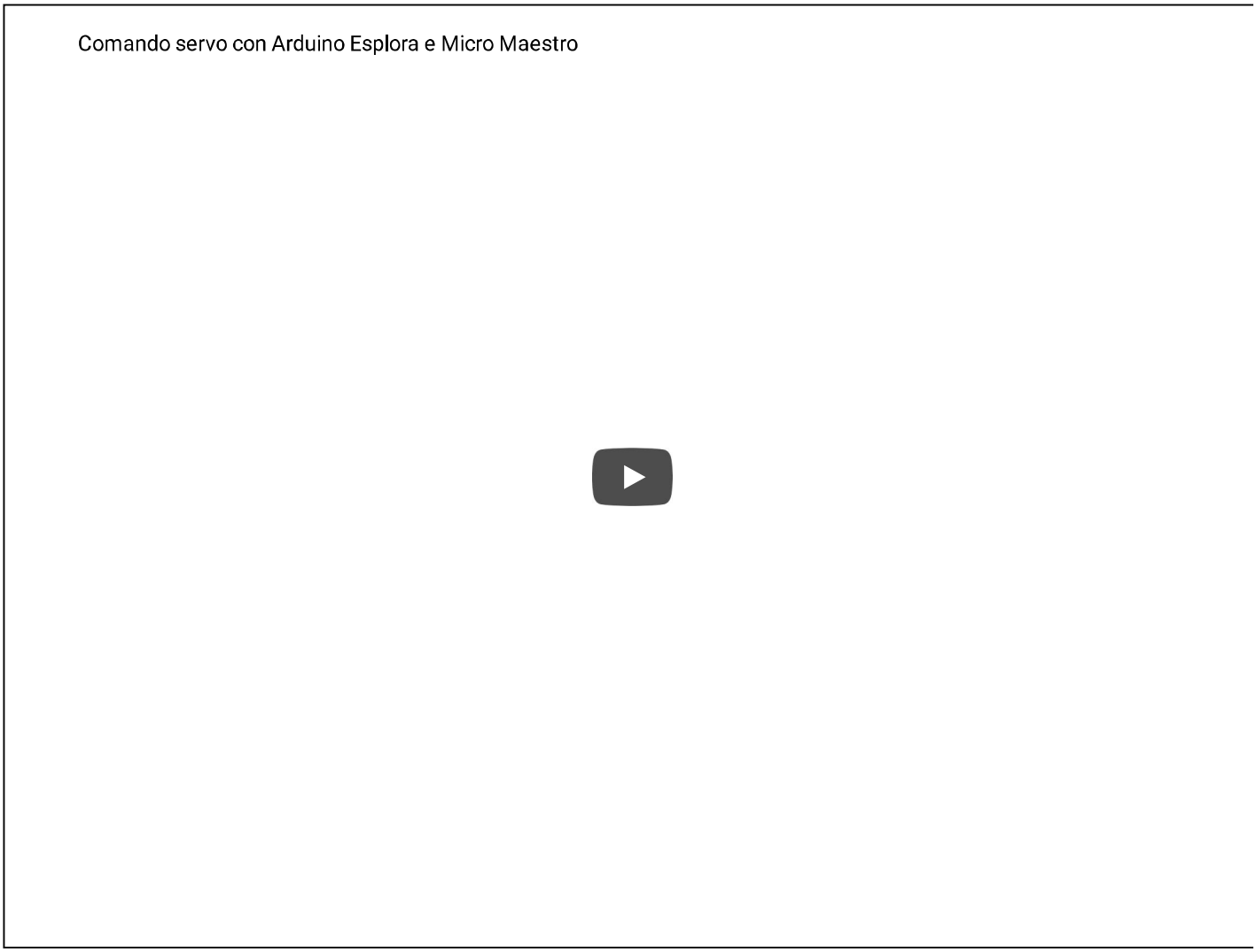
Il programma è sufficientemente commentato e non dovrebbe presentare problemi di comprensione.
Per quanto riguarda il comando map, questo è utilizzato per adattare i valori letti dal Joystick ai valori e nei limiti necessari alla gestione dei servomotori, si potrà far riferimento alla guida presente sul sito Arduino [37].

```
//Gestione Arduino Esplora
#include <Esplora.h>:
//Gestione Micro Mestro
#include <PMCtrl.h>
#include <SoftwareSerial.h>
//Definizione pin gestione Micro Maestro
 #define rxPin 11
 #define txPin 3
 //Impostazione Parametri di collegamento con Micro Mestro
 PMCtrl servoCtrl (rxPin, txPin, 9600); //RX, TX, Baud
 //Definizione delle varibili utilizzate
 //Assegnazione canali servomotori
 const int servo0 = 0;
 const int servo1 = 1;
 //Definizione delle posizioni minime e massime
 // per i due assi
 const int MAX_W = 2224;
 const int MIN_W = 598;
 const int MAX_H = 1808;
 const int MIN_H = 495;
 //Definizione delle variabili della posizione
 //assunta dal Joystick
 int xPos = 0;
 int yPos = 0;
 void setup ()
 {
 //Imposta la velocità dei due servomotori
 servoCtrl.setServoSpeed (100, servo0, 12);
 servoCtrl.setServoSpeed (100, servo1, 12);
 //Sposta i servomotori nella posizione di riposo
servoCtrl.goHome(12);
 }
 void loop ()
 {
 //Assegna a XValue e YValue i valori letti dal Joystick
 int xValue = Esplora.readJoystickX();
 int yValue = Esplora.readJoystickY();
 //Utilizzo del comando map http://arduino.cc/en/reference/map
 //per adattare i valori letti nel range dei servomotori
 xPos = map( xValue, 512, -512, MIN_W,MAX_W);
 yPos = map( yValue, 512, -512, MIN_H,MAX_H);
 //Invia alla scheda MicroMestro le posizioni dei due servomotori
 servoCtrl.setTarget (xPos, servo0, 12);
 servoCtrl.setTarget (yPos, servo1, 12);
 delay (50); //Pausa tra gli aggiornamenti dei valori letti
```

}

Filmato illustrativo

Nel filmato, potrete vedere i vari componenti utilizzati e l'esecuzione del programma.



Conclusioni

Questo è solo un esempio di quello che è possibile fare con la **Micro Mastro** in unione con la **Arduino Esplora.**

In questo caso, la gestione era effettuata tramite cavo ma, come abbiamo visto in un precedente articolo, "**Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth** [38]", è possibile interfacciare la scheda con un altro dispositivo dotato anch'esso di un modulo Bluetooth.

In questo caso occorrerà utilizzare magari un **Arduino Micro** [39] che funga da interfaccia tra il modulo Bluetooth [40]e la scheda **Micro Maestro.** L'unico limite ai progetti realizzabili è dato dai sei servomotori che possono essere pilotati, limite che può essere superato utilizzando una delle altre schede della serie Maestro che, nel modello "maggiore", arriva a ben 24 unità.

URLs in this post:

[1] **Scopriamo la nuova scheda Arduino Esplora**: **https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora**
[2] **ATMEGA32U4**: **http://www.atmel.com/devices/atmega32u4.aspx**
[3] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Arduino-esplora.jpg**
[4] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_esplora_connettori.jpg**
[5] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Servo_Breakdown.gif**
[6] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Servo_antiorario.jpg**
[7] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Servo_orario.jpg**
[8] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_servo_centro.jpg**
[9] **Micro Servo Serial Controller**: **http://www.pololu.com/product/207**
[10] **Pololu**: **http://www.pololu.com/**
[11] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Scheda_ssc03a.jpg**
[12] **PIC16F628A**: **http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010210**
[13] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_ssc03a.jpg**
[14] **Mini SSC II Servo**: **http://www.seetron.com/sscasd2.html**
[15] **Micro Maestro**: **http://www.pololu.com/product/1350**
[16] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_scheda.jpg**
[17] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_modelli.jpg**
[18] **PIC18F14K50-I/SS**: **http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en533924**

[19] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_serial-input.jpg**
[20] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_command-0.jpg**
[21] **Graham Sortino** : **https://github.com/gNSortino/PMCtrl**
[22] **relativa pagina**: **http://www.pololu.com/docs/0J40/all#4.b**
[23] batteria/caricabatteria: **http://www.cellularline.com/catalog/it/product/usb_pocket_charger_smart**
[24] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Schema_collegamento.jpg**
[25] Image: **https://it.emcelettronica.com/wp-content/uploads/2015/01/Arduino_esplora_Micro_Maestro_foto_test.jpg**
[26] **RS 679-5375P**: **http://it.rs-online.com/web/p/custodie-per-connettori-da-pcb/6795375P/**
[27] Image: **https://it.emcelettronica.com/wp-content/uploads/2015/01/Arduino_esplora_Micro_Maestro_cavo_kit.jpg**
[28] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Schema_cavo_collegamento.jpg**
[29] **SparkFun codice ROB-10335**: **https://www.sparkfun.com/products/10335**
[30] Sparkfun ROB-09065: **https://www.sparkfun.com/products/9065**
[31] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_kit-pan_tilt.jpg**
[32] **seguente pagina**: **http://www.adrirobot.it/pan&tilt/Pan_Tilt-Bracket.htm**
[33] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_Pan_Tilt-Bracket-_dettaglio.jpg**
[34] Image: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_Micro_Maestro_connettore_servo.jpg**
[35] Arduino_esplora_servo: **https://it.emcelettronica.com/wp-content/uploads/2014/12/Arduino_esplora_servo.zip**
[36] #include <SoftwareSerial.h>: **http://arduino.cc/en/Reference/SoftwareSerial**
[37] guida presente sul sito Arduino: **http://arduino.cc/en/reference/map**
[38] **Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth**: **https://it.emcelettronica.com/dotiamo-larduino-esplora-dellinterfaccia-bluetooth**
[39] **Arduino Micro**: **http://arduino.cc/en/Main/arduinoBoardMicro**
[40] Bluetooth : **https://it.emcelettronica.com/?s=bluetooth**