

ISTI - CNR

Software per la realizzazione dell'interfaccia "C - Animatronica":

Simone Buono, Giampiero Di Simone, Graziano Bertini

Nota tecnica B4 - luglio 2004

Software per la realizzazione dell'interfaccia "C - Animatronica":

Simone Buono*, Giampiero Di Simone**, Graziano Bertini***,

* Coll. Esterno ISTI
** Coll. NES srl
*** ISTI-CNR

Indice.

Introduzione

1. Note generali sulle funzionalità del sistema e del programma.
2. Descrizione del software per l'utente.
3. Note tecniche sul codice.
4. Funzioni per il trattamento delle relazioni movimenti/audio.
5. Conclusioni.
6. Bibliografia.

Descrizione del sistema

Il sistema prevede due componenti principali (fig.1): un PC su cui è sviluppato un ambiente sw che accetta comandi in ingresso tramite un Joystick ed una scheda esterna basata sul microcontrollore Atmel 128 (particolari in fig.2).

Architettura hardware:

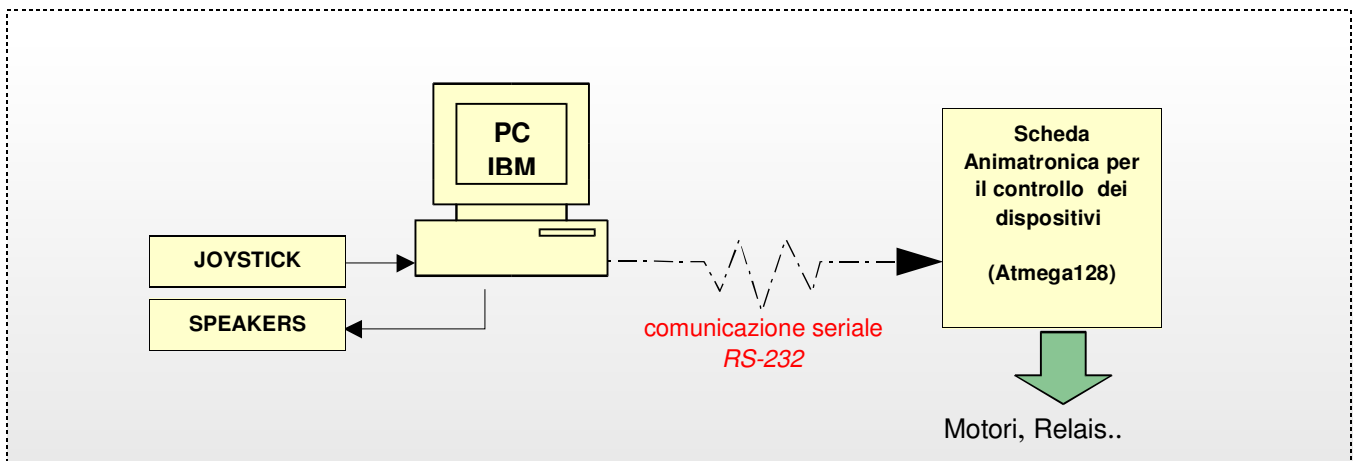


Figura 1. Fase Master-Slave

La scheda Animatronica basata su Atmega128 può controllare a sua volta vari dispositivi elettronici e meccanici tramite 6 porte I/O. Di seguito è riportato lo schema della versione prevista come configurazione finale :

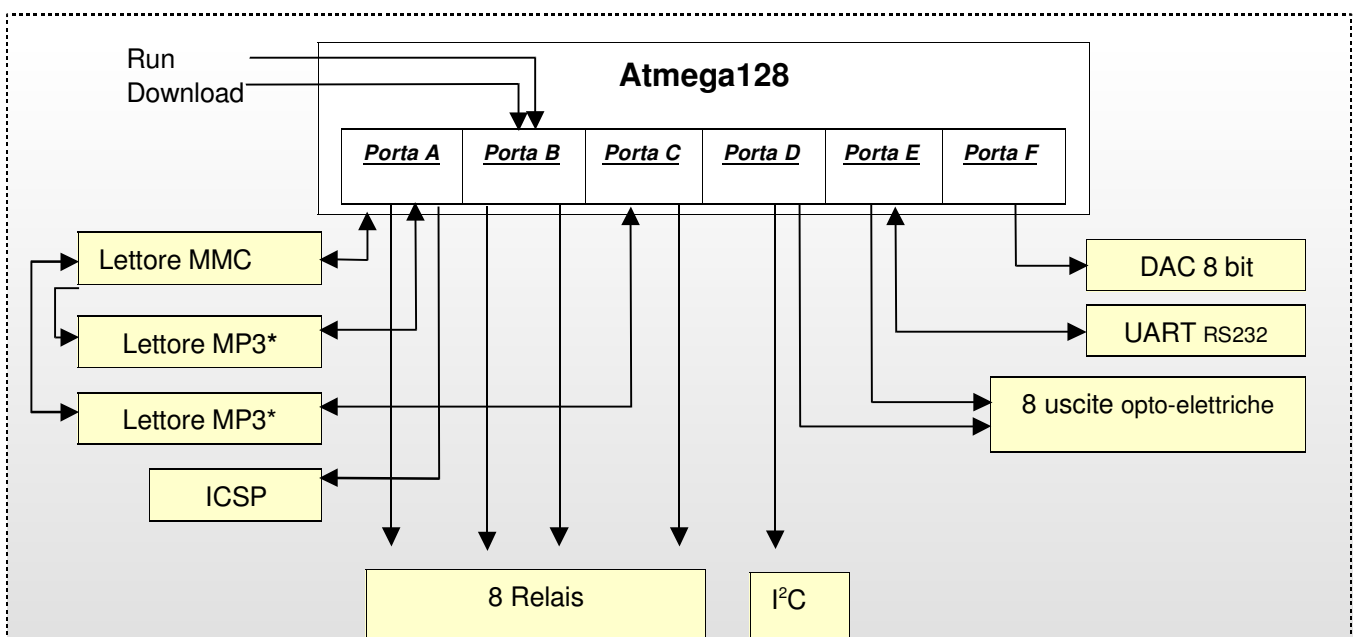


Figura 2. Scheda Animatronica

* Il lettore MP3 è stato realizzato come scheda indipendente basata anch'essa su microcontrollore Atmel32 , lettore Flash MMC e scheda mp3 tipo -VS1001k-:

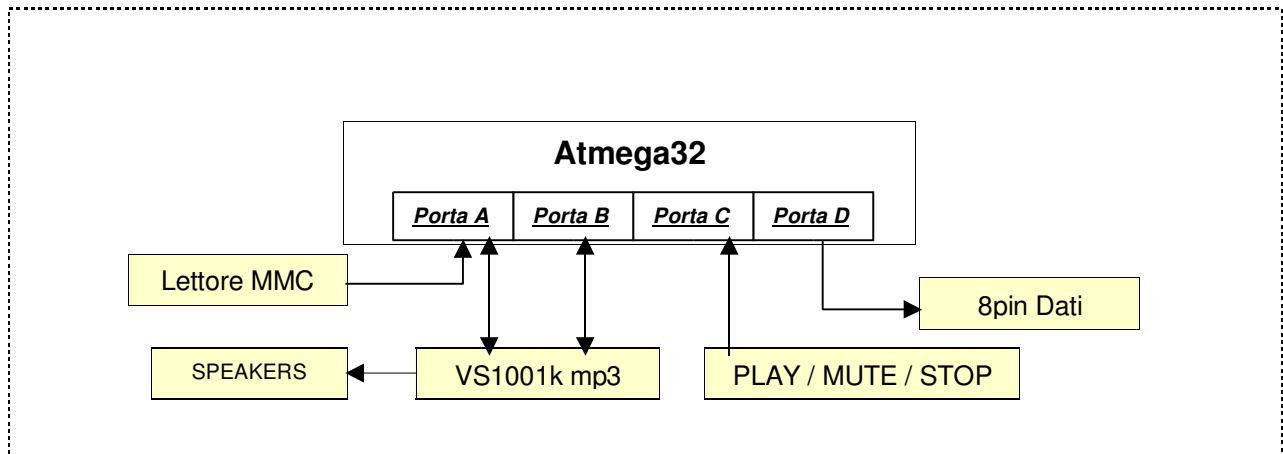


Figura 3. Scheda lettore mp3

Architettura software del sistema:

L'interfaccia utente per l'elaborazione dei dati è implementata in C++ con le classi MFC (Microsoft Foundation Classes ©) e sfrutta il modello di applicazione dialogica, basata su forms di dialogo dai quali è possibile inserire ed elaborare i dati.

Il Firmware contiene delle funzioni C per la lettura dei dati da scheda di memoria flash e invio dati al lettore mp3:



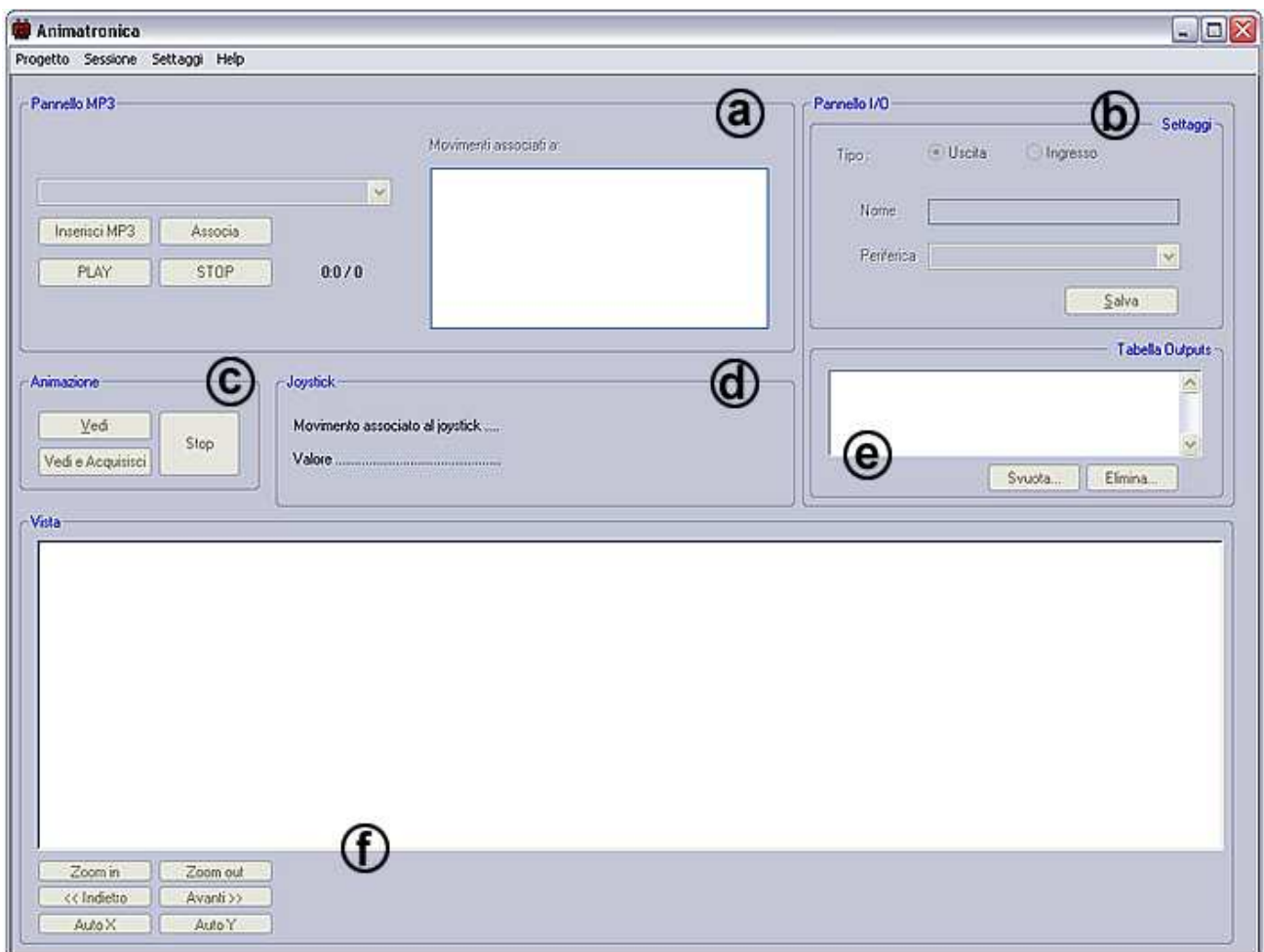


Figura 4. Interfaccia utente, (a) Pannello di gestione dei brani mp3; (b) Pannello per la denominazione dei dispositivi elettrici/meccanici; (c) Pannello di controllo dell'animazione; (d) Pannello di visualizzazione dei valori numerici dell'asse Y del joystick; (e) Lista dei dispositivi output utilizzati nel progetto; (f) grafico in scala tempo/valore dei dati acquisiti da joystick.

Funzionamento del sistema:

Fase master-slave

In modalità master-slave (fig. 1) il PC aziona i dispositivi di controllo dei motori della scheda Animatronica (fig. 2) fornendo i dati dei movimenti tramite trasmissione seriale fra PC e scheda Atmel. In questa fase è possibile azionare i motori passando i dati appena acquisiti da joystick oppure da files.

L'unica eccezione riguarda l'mp3 che viene eseguito da PC e non sulla scheda animatronica.

Fase di download e fase autonoma della scheda

I dati acquisiti e i brani audio vengono riversati dal PC su una memoria di massa di tipo FlashCard. Da questo momento la scheda Animatronica è in grado di lavorare autonomamente senza l'ausilio del PC; tutti i dati vengono gestiti dal programma residente sul microprocessore Atmel.

2) Descrizione del software per l'utente:

Creazione di un progetto di Animatronica:

All'avvio il programma principale genera una *cartella di progetto* che contiene i seguenti files e cartelle:

★ Cartella del progetto:

- cartella *Animazioni* – contiene tutti i files di movimento;

- cartella *Mp3* – contiene tutti i brani mp3 relativi al progetto;
- cartella *Tabelle* – contengono i files con le associazioni come di seguito riportate:
- ★ *Ingressi.txt* - archivio dei dispositivi input associati al progetto;
- ★ *Uscite.txt* - archivio dei dispositivi output associati al progetto;
- ★ *Userver.txt* - archivio delle variabili utente usate in fase di programmazione;

Inoltre sono presenti alcuni file del sistema

- *Atmel.txt.c* - il codice C dell'automazione per il processore atmel;
- *Codice.lst* - archivio delle istruzioni per la programmazione dell'automazione;
- *Progetto.prj* - file che identifica il progetto.

Una volta creata una cartella di progetto, si procede con le altre fasi della progettazione.

La prima fase consiste nell'assegnare un nome ai vari dispositivi meccanici da controllare. Dopodichè si procede con la scelta di un brano mp3 da eseguire in modo sincrono con i movimenti. Una volta scelto il brano, tutti i movimenti che verranno acquisiti saranno legati ad esso anche se sarà sempre possibile gestire le associazioni Animazione/Audio grazie ad un form apposito. Le azioni sono poste nell'ordine seguente:

Fase di assegnazione / inizializzazione della mappa dei dispositivi e dati del sistema:

- ★ Assegnazione di un nome ai dispositivi I/O nell'archivio Output (uscite.txt)
- ★ Inserimento di brani audio mp3 all'interno del progetto;
- ★ Assegnazione dei dispositivi Output ai brani mp3 del progetto;

La seconda fase del lavoro consiste nell'acquisire i dati da joystick da associare ai vari dispositivi:

Fase di acquisizione e anteprima dei dati:

Acquisizione – tramite joystick o altra periferica HID – e anteprima dei movimenti per i dispositivi output sincronizzati con i brani mp3.

La terza fase permette di inviare i dati ai dispositivi meccanici per poter controllare visivamente il risultato delle nostre acquisizioni.

Fase di anteprima dei dati:

Invio – tramite seriale *rs232* - dei movimenti associati al brano mp3.

La quarta fase non è strettamente necessaria, ed è legata all'uso di un modulo sw che permette di gestire l'esecuzione di movimenti e suoni associati ad eventi grazie ad un linguaggio appositamente creato:

Fase di Programmazione dei dati:

Costruzione - tramite uno *script* proprietario – di un automazione che esegue brani audio mp3 e aziona i movimenti dei vari dispositivi output (legati o non legati all'audio) tenendo conto dei dati forniti dai dispositivi input.

3) Note tecniche sul codice:

Il programma è composto da quattro classi principali:

- ★ **CAnimatronicaApp**, la classe principale che avvia l'applicazione;
- ★ **CAnimatronicaDlg***, collegata ad un dialog Form, rappresenta l'interfaccia Animatronica;
- ★ **CAssociaMov**, collegata ad un dialog form secondario, svolge alcune funzioni del programma;
- ★ **CLinguaggioDlg**, collegata ad un dialog form secondario, contiene le funzioni che implementano il linguaggio usato per il controllo ad eventi dei movimenti;

Contiene inoltre cinque funzioni che utilizzano il componente ActiveX *directX*[®] per il controllo del joystick , tre funzioni per il controllo della porta seriale e una libreria per la gestione di files audio in formato mpeg-layer3.

Listato delle funzioni principali: (CAnimatronicaDlg)

- ★ **AggiornaJoy** – aggiorna le coordinate del joystick;
- ★ **AggiornaPlaylist** – aggiorna il contenuto della listbox dei movimenti associati al progetto;

- ★ **CheckPieno** – dato il nome di un file di movimento, controlla se un file di movimento è pieno o vuoto;
- ★ **ErroreSeriale** – notifica un errore sulla porta seriale;
- ★ **FindExactMp3Name** – data una parte del nome di un file mp3 presente in archivio, ne restituisce il nome completo;
- ★ **HandleControls** -abilita/disabilita i componenti nel form principale;
- ★ **ListaUscite** - aggiorna la listbox che contiene i nomi dei dispositivi meccanici inseriti nel progetto;
- ★ **OnAssocia** – (legato all’evento click di un pulsante) avvia il form che si occupa dell’associazione dei movimenti al brano mp3;
- ★ **OnBtnInvia** – (legato all’evento click di un pulsante) salva un nome per un dispositivo meccanico da inserire nel progetto;
- ★ **OnDbClickListmov** – (legato all’evento doubleclick del mouse) avvia il form che si occupa dell’associazione dei movimenti al brano mp3;
- ★ **OnEliminaUscita** - (legato all’evento click di un pulsante) rimuove un dispositivo meccanico dal progetto;
- ★ **OnInitDialog** – funzione di inizializzazione dei dati all’interno del dialog form principale;
- ★ **OnRadio_uscita** – (legato all’evento click di un radio button) specifica il tipo di dato (input o output) per il dispositivo meccanico da inserire nel progetto;
- ★ **OnRadio_ingresso** - (legato all’evento click di un radio button) specifica il tipo di dato (input o output) per il dispositivo meccanico da inserire nel progetto;
- ★ **OnInsClip**– Inserisce un nuovo file mp3 all’interno del progetto;
- ★ **OnPlay** – Esegue un brano mp3;
- ★ **OnPlayButton** – Esegue il file di movimento corrente;
- ★ **OnProgettoAprogetto** – apre un progetto;
- ★ **OnProgettoChiudioprogetto** – chiude il progetto corrente;
- ★ **OnRec**- Salva in un file i dati acquisiti da joystick;
- ★ **OnSessioneProgrammazionesequenza** – voce del menu che visualizza il form che gestisce la programmazione dei movimenti;
- ★ **OnStop** – Ferma l’esecuzione di un mp3;
- ★ **OnStopplay** – Ferma l’esecuzione di un mp3 e l’acquisizione die dati da joystick.

- ★ **OnSvuota** – elimina il contenuto di un file di movimento;
- ★ **OnTimer** – funzione di tipo callback, ha un tempo di chiamata di 20 msec. Controlla l’acquisizione dei dati dal joystick e l’esecuzione del brano audio;
- ★ **OnTuttiPlay** –Esegue tutti i movimenti associati al progetto;
- ★ **PrepareFilesInvioSeriale** – Apre i files di movimento da inviare tramite seriale e prepara i dati per la trasmissione;
- ★ **Pulsanti_mp3** – abilita/disabilita i pulsanti di controllo per l’mp3;
- ★ **RemoveEntry** – elimina ogni relazione fra un file di movimento e un mp3 e rimuove il nome del dispositivo associato al movimento dal progetto corrente;
- ★ **RemoveRelation** - elimina ogni relazione fra un file di movimento e un mp3 senza cancellarlo dal progetto;
- ★ **SalvaPlaylist** – salva in un file tutti i nomi dei file movimento associati all mp3;
- ★ **SetPieno** – imposta l’attributo di un file di movimento;
- ★ **TerminaTrasmissione** – interrompe bruscamente la trasmissione sulla porta seriale e l’esecuzione di tutti i movimenti.

4) Funzioni per il trattamento delle relazioni movimenti/audio:

- (CassociaMov) -

- ★ **BeforeIChange** – gestisce le modifiche alla lista dei movimenti;
- ★ **LeggiUsciteValide** – legge i movimenti disponibili e li aggiunge alla lista;
- ★ **MovLock** – associa un file di movimento ad un mp3 nel file delle uscite;
- ★ **MovUnlock** – dissocia il file di movimento dall’ mp3 nel file delle uscite ;
- ★ **OnAggiungi** –aggiunge un movimento alla lista dei movimenti associati all’mp3;
- ★ **OnApplica** – Applica le modifiche al file dei movimenti;
- ★ **OnRimuovi** – rimuove un movimento dalla lista dei movimenti associati all’mp3;
- ★ **OnSpostaggiu** – sposta il movimento della lista di una posizione verso l’alto;
- ★ **OnSpostasu** - sposta il movimento della lista di una posizione verso il basso;
- ★ **Salvalista** – salva la lista dei movimenti associati all mp3;
- ★ **UnlockMP3** – dissocia l’mp3 da un file di movimento;

★ **Updatelist** - funzione che riempie la lista dei files di movimento associati all mp3.

5) **Conclusioni.**

La presente descrizione è stata redatta come relazione sullo stato di avanzamento dei lavori essendo trascorso circa un anno dall'inizio della collaborazione con la NES su questo progetto. Restano ovviamente da completare alcune parti come ad es. la programmazione del firmware della scheda Animatronica e parte del software per la gestione dei dati su PC.

E inoltre sono possibili diverse soluzioni a seconda delle specifiche applicazioni del sistema.

6) **Bibliografia.**

C++ Primer – S. Lippmann;

AVR Atmega 128/Atmega32 Reference Guide – Atmel corp.;

Visual C++ 6 guida completa – Apogeo.