Università degli Studi di Bologna

FACOLTÁ DI INGEGNERIA

Geometria e Algebra

RESA TATTILE DEL COLORE NELL'APPARATO VIDET PER VIDEOLESI

Tesi di Laurea di:

LUCA CAPPELLETTI

Relatore:

Prof. MASSIMO FERRI

Correlatori:

Prof. Ing. TULLIO SALMON CINOTTI

Ing. LUIGI DI STEFANO

Anno Accademico 1995-96

PAROLE CHIAVE:

Cecità

Tatto

Vibrazioni

Segnale RGB

Protesi per non vedenti

Alma Mater Studiorum
Sæcularia Nona

INDICE

Alma Mater Studiorum
Sæcularia Nona

INTRODUZIONE

Le tecnologie moderne offrono nuove speranze nell'ambito dell'handicap e in particolare della cecità. VIDET è un progetto sviluppato presso l'Università di Bologna, volto a fornire, mediante robotica e realtà virtuale, una nuova possibilità di esplorazione dell'ambiente ad un cieco. Il soggetto avrà a disposizione un modello virtuale in scala dell'ambiente circostante, che esplorerà con un dito. È proprio da osservazioni di non vedenti coinvolti nel progetto che è nata l'esigenza di aggiungere al modello virtuale l'informazione cromatica.

A questa tesi è affidato il compito di studiare le prime soluzioni per una resa del colore; la scelta è caduta su una resa completamente tattile. Il lavoro di tesi si è esteso dalla scelta fra diverse alternative alla realizzazione dell'hardware e del software, fino alla vera e propria sperimentazione su soggetti vedenti e non vedenti. I risultati sono parsi subito del tutto soddisfacenti, superando le più ottimistiche aspettative.

La tesi inizia con un capitolo dedicato alle premesse sul progetto VIDET e sulle caratteristiche fisiologiche della mano come organo di senso. Segue una valutazione delle soluzioni che si sono presentate inizialmente, alla luce della natura del segnale RGB. Il terzo capitolo riporta il progetto della scheda impiegata nella trasduzione. Il capitolo 4 descrive il programma "colori.exe" utilizzato per gestire la scheda. La tesi si conclude con un capitolo dedicato ad esperimenti realizzati con la collaborazione di soggetti normovedenti e videolesi.

PREMESSE

1.1 Rassegna degli ausili e delle protesi già esistenti

La soluzione più significativa al problema della lettura per non vedenti è costituita dall'Optacon, che comprende: a) una testina di lettura che deve scorrere sul testo da leggere, nella quale è collocato un insieme di fotodiodi posti secondo una matrice di 6 colonne e 24 righe pari a 144 fotodiodi; b) un pannello di "punti" vibranti sul quale appoggia il pollice del cieco. Collegando ogni punto con un fotodiodo, si ottiene il carattere in rilievo. L'addestramento consiste nell'insegnare a riconoscere i singoli caratteri e nel far scorrere la testina di lettura sul testo. A tale scopo esistono opportuni nastri che scorrono in modo automatico sotto il pollice del soggetto.

Per quanto riguarda il problema della mobilità, il più antico strumento d'interazione con l'ambiente per un non vedente, cioè il bastone, sembra passibile di una modernizzazione di ridotto impatto fisico, di costo sostenibile e di ampia diffusione. Una delle apparecchiature di maggior successo è il bastone a raggi-eco munito di alcuni generatori a raggi laser o ad ultrasuoni. Questi raggi, quando incontrano un ostacolo, danno luogo a un'onda riflessa che viene captata come un'eco da opportuni ricevitori posti sul bastone; i ricevitori a loro volta danno luogo ad un'uscita sonora oppure tattile per figurare il raggio riflesso. Nel caso di tre raggi aventi diversa inclinazione, quello superiore

serve per informare il cieco di eventuali ostacoli posti circa all'altezza della sua testa, quello intermedio per localizzare ostacoli quali tavoli o mobili, mentre quello inferiore per informare il cieco di variazioni del piano su cui cammina, cioè buchi, gradini e così via. Pur essendo da tempo costruiti ed anche commercializzati, ausili di questo tipo non sembrano soddisfacenti per i ciechi che preferiscono mezzi più tradizionali quale l'usuale bastone e il cane opportunamente addestrato. Una ragione di questo è senz'altro data dall'invasività delle informazioni trasmesse: infatti il cieco utilizza l'udito per percepire i suoni provenienti dall'ambiente circostante e questo tipo di informazione può distrarlo invece che aiutarlo; inoltre va rilevata l'impossibilità di ottenere informazioni sulla giacitura dei piani tangenti se non attraverso derivate direzionali della profondità.

Negli USA sono allo studio delle protesi elettroniche per l'occhio, ma il carattere invasivo e costoso dell'impianto ed il necessario e non facile addestramento della corteccia visiva (in particolare per la visione stereoscopica) ne renderanno forse limitato l'impiego.

1.2 Videt

L'apparato visivo è, fra gli apparati di senso, quello che convoglia più informazione sull'ambiente circostante: la vista di un oggetto permette di valutarne la posizione nello spazio con sufficiente accuratezza anche se è a distanza notevole, ed anche se non produce esso stesso luce. In questi aspetti differisce sostanzialmente dall'apparato tattile, che agisce a breve distanza, e da quello uditivo, che consente una utile individuazione quasi solo di oggetti che producono suoni.

Perciò una lesione grave dell'apparato visivo ha come principale conseguenza una drammatica riduzione della capacità del soggetto di conoscere l'ambiente circostante e quindi di interagire con esso.

Il VIDET è un sistema robotico per videolesi, basato sulla resa

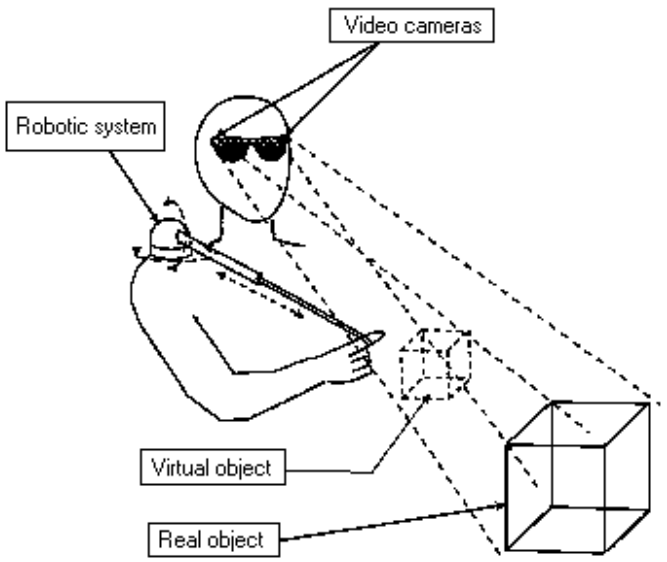


figura 1.1 - Schema del funzionamento del VIDET.

della profondità mediante ritorno di forza, atto a presentare ad un soggetto non vedente una superficie virtuale rappresentante, in forma tattile e in scala, una porzione dell'ambiente circostante.

Esso consta dei seguenti componenti (figura 1.1):

- una coppia di microtelecamere poste su un'intelaiatura che verrà montata, presumibilmente, sul capo o sulle spalle del soggetto;
- un elaboratore portatile, eventualmente dedicato;
- una struttura robotica da fissarsi al corpo del soggetto, dotata di un elemento terminale in cui alloggiare uno o più dita di una mano.

Il funzionamento è organizzato nelle seguenti fasi:

- acquisizione ed eventuale pre-elaborazione di una coppia di immagini;
- calcolo della profondità di una griglia di punti delle immagini mediante disparità;
- interpolazione dei punti quotati con una superficie virtuale e sua riduzione in scala;
- determinazione dei vettori di posizione e di velocità dell'elemento terminale della struttura robotica;
- confronto dei vettori di posizione e di velocità con la superficie virtuale;
- nel caso di impatto, applicazione all'elemento terminale di una forza normale alla superficie tale da impedirne l'attraversamento.

Oltre alle informazioni relative alla posizione e al riconoscimento dell'oggetto, il VIDET prevede la possibilità di fornire informazione tattile sul colore dell'oggetto.

1.3 Risposta tattile della mano

Prima di ogni possibile considerazione su quale tipo di segnale sia meglio utilizzare, è opportuno studiare quale parte del corpo umano sia la più indicata per recepire e distinguere la suddetta informazione.

Se tocchiamo con il polpastrello il foglio che stiamo leggendo, sapremo immediatamente che il foglio è liscio, piuttosto morbido e a temperatura ambiente. Ovviamente questo ci indica che l'organo di senso del tatto, della sensibilità dolorifica e della

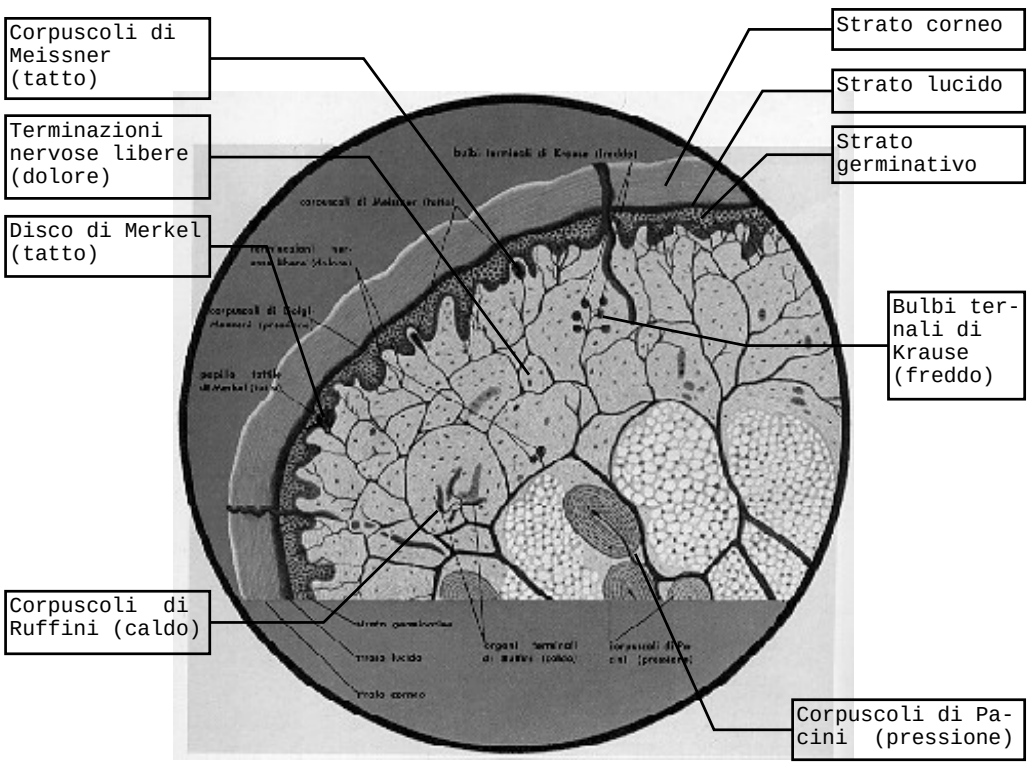


figura 1.2 - Sezione dell'epidermide e degli strati sottostanti, nella quale si possono osservare diversi organi recettori per le sensazioni cutanee.

sensibilità termica sta nella pelle.

La pelle è composta da diversi strati: i due più esterni, lo strato corneo e lo stato lucido (figura 1.2), sono praticamente solo strati di rivestimento, privi di sensibilità; sotto si trovano gli altri strati dell'epidermide e del derma, i quali sono per così dire strati "vivi", perché in essi vi sono le terminazioni nervose destinate a raccogliere le diverse sensazioni.

Per mezzo della pelle l'uomo avverte quattro diverse sensazioni: di contatto e pressione, di freddo, di caldo, di dolore. Ciascuna di queste viene avvertita non indistintamente sulla pelle, ma in diversi punti di essa, distanti tra loro alcuni millimetri. La distanza minima alla quale due stimoli puntiformi possono essere percepiti separatamente è di 1 mm sulla punta della lingua, di 2 mm sui polpastrelli delle dita delle mani, 4÷5 mm sulle labbra, 31÷32 mm sul dorso della mano e 60÷70 mm sulla schiena.

Quando la pelle viene toccata o compressa, oppure viene a contatto con corpi caldi o freddi, oppure subisce tagli, punture, sfregamenti o bruciature, le terminazioni nervose interessate registrano la sensazione ricevuta e la trasmettono immediatamente al cervello. Questo significa che esistono diverse categorie di terminazioni nervose, ognuna delle quali è preposta ad un

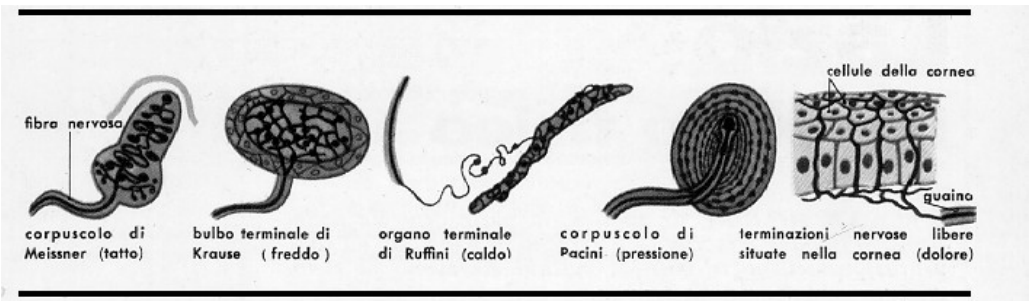


figura 1.3 - Le diverse forme di alcuni recettori della pelle

determinato compito. In figura 1.3 si possono osservare le diverse forme di alcuni recettori della pelle.

Le mani, più precisamente i polpastrelli delle dita, sono fra le parti del corpo che presentano la sensibilità più elevata. Per questo motivo, e tenendo conto del fatto che la struttura robotica impegna comunque uno o più dita di una mano, si limita lo studio della sensibilità al solo ambito della mano.

La caratterizzazione delle principali funzioni di tipo meccanico proprie del senso del tatto dipende dalle risposte dei recettori sensoriali periferici di stimoli pressori, i meccanocettori. Nonostante una gran parte degli aspetti della trasduzione meccano-neurale sia ancora sconosciuta, sono state individuate e studiate dal punto di vista anatomico e funzionale alcune terminazioni nervose specializzate. I principali tipi di meccanocettori presenti nell'insieme epidermide-derma, con particolare riferimento ai polpastrelli delle dita, che ne possiedono la maggior densità, sono:

- *i corpuscoli di Meissner*, terminazioni nervose allungate ed incapsulate presenti soprattutto nei polpastrelli delle dita e in altre zone sensibili della pelle. Sono situati nei peduncoli tra derma ed epidermide e vengono considerati sensori di velocità essendo particolarmente sensibili al movimento di oggetti sulla pelle oltre che a vibrazioni a bassa frequenza. Costituiscono circa la metà delle unità tattili e rispondono alle pressioni normali sviluppate dal contatto;
- *i dischi di Merkel*, che grazie alla loro adattabilità molto lenta forniscono l'informazione relativa alla permanenza del contatto di un oggetto sulla pelle, rilevando quindi i segnali statici di pressione; costituiscono circa il 25% del totale dei recettori della mano;
- *i corpuscoli di Ruffini*, strutture fusiformi incapsulate e ramificate nel derma e negli strati più profondi della pelle che costituiscono circa il 19% del totali dei recettori della mano: a causa della loro struttura formata da una fitta rete di

fibre di collagene, sono in grado di misurare le funzioni di taglio; inoltre sono i responsabili delle sensazioni di caldo;

- *i bulbi terminali di Krause*, situati più in superficie e più numerosi rispetto ai corpuscoli di Ruffini, sono i responsabili delle sensazioni di freddo;
- *i corpuscoli di Pacini*, recettori situati più in profondità all'interno del derma con dimensioni che vanno da 1 a 4 mm di lunghezza e da circa 0,5 a 1 mm di diametro; sono stimolati solo da movimenti molto rapidi del tessuto e si adattano allo stimolo in pochi centesimi di secondo: grazie a queste caratteristiche sono preposti alla detezione delle vibrazioni all'interno del tessuto;
- *terminazioni nervose libere*, localizzate ovunque, con diametro variabile tra 0,5 e 2,5 mm, con funzioni di termorecettori, recettori di dolore e di stimoli tattili pressori.

Grazie alle caratteristiche di veloce adattabilità allo stimolo applicato, i corpuscoli di Pacini sono i recettori che meglio si prestano ad essere utilizzati per recepire una vibrazione. È proprio attraverso questi recettori sensoriali che avverrà la percezione del segnale che porta l'informazione che si vuole far

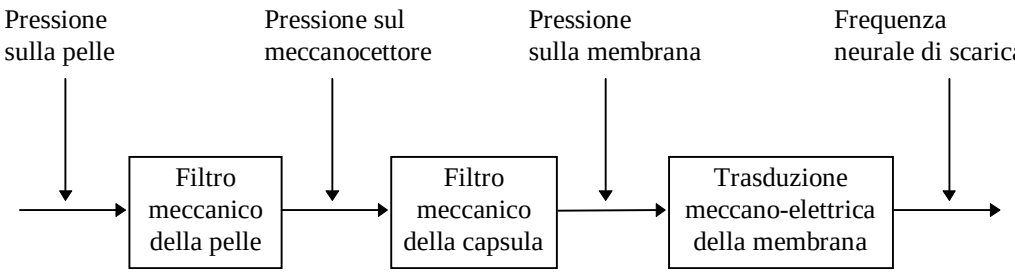


figura 1.4 - Schema delle componenti fondamentali che influenzano la risposta temporale di un meccanocettore.

giungere al non vedente.

Poiché i corpuscoli di Pacini sono quelli situati più in profondità nella pelle, bisogna considerare l'effetto filtrante sia dell'epidermide che del derma (figura 1.4). Quando si applica uno stimolo sulla superficie della pelle, la pressione effettivamente applicata alla parte sensibile del corpuscolo non dipende solo dall'intensità dello stimolo, ma anche dalla legge temporale di applicazione. Infatti la pelle si comporta come un mezzo visco-elastico non lineare. Nel caso di piccole deformazioni, le caratteristiche dei parametri della pelle fanno sì che essa abbia un comportamento passa-banda (figura 1.5). Gli effetti filtranti a bassa frequenza sono dovuti principalmente ai fluidi contenuti nei tessuti (effetti viscosi), mentre l'attenuazione ad alta frequenza è dovuta agli effetti inerziali della massa della pelle.

Oltre all'effetto passa-banda della pelle, bisogna tener conto della capsula dentro la quale è protetto il corpuscolo di Pacini, che attenua ulteriormente lo stimolo nella banda di frequenze non

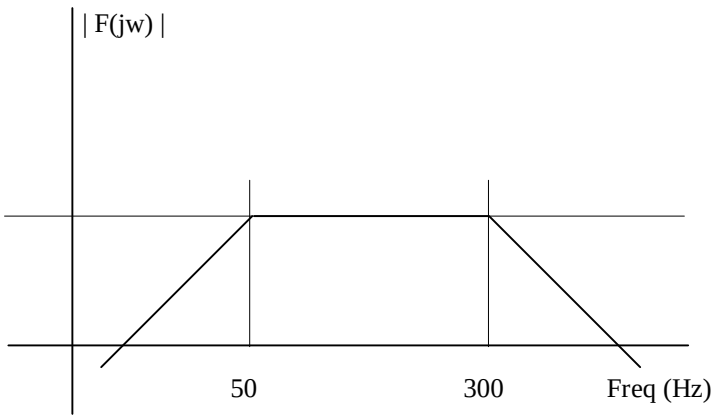


figura 1.5 - Risposta frequenziale della funzione di trasferimento tra la pressione applicata alla pelle e la pressione sul recettore.

comprese tra i 50 e i 300 Hertz. Volendo quindi utilizzare la mano per trasmettere al soggetto cieco l'informazione concernente il colore, è opportuno considerare l'applicazione di segnali con frequenze strettamente comprese nel suddetto intervallo.

ORGANIZZAZIONE TEORICA

2.1 Fondamenti fisiologici della visione

Il sistema visivo umano è parte del sistema nervoso, il quale è senza dubbio la più complicata rete di comunicazione esistente ed è controllata dal più potente computer mai progettato: il cervello.

L'informazione visiva è recepita dall'uomo attraverso la via ottica composta dai mezzi rifrangenti, dalla retina (che insieme formano il bulbo oculare) e dal sistema nervoso associato al sistema di visione (figura 2.1). I mezzi rifrangenti (cornea, umor acqueo, cristallino, corpo vitreo) hanno lo scopo di formare sulla retina una rappresentazione bidimensionale della radiazioni luminose presenti nel campo visivo. La luce entra nell'occhio attraverso la cornea; quest'ultima è una membrana trasparente di curvatura superiore a quella del bulbo oculare per garantire una rifrazione costante dei raggi luminosi provenienti da una qualunque direzione del campo visivo. L'intensità luminosa che entra nell'occhio è regolata dalla pupilla, un diaframma di diametro variabile da 2 a 9 mm. Il cristallino è una lente di convessità variabile, allo scopo di focalizzare sulla retina l'immagine indipendentemente dalla distanza. A meno di difetti geometrici del bulbo (ipermetropia, miopia, presbiopia, astigmatismo), questo perfetto meccanismo focalizza sulla retina una rappresentazione bidimensionale che riproduce l'immagine presente nel campo visivo. In realtà, a causa dell'effetto di schermo della pupilla, anche un occhio senza difetti subisce gli

effetti dell'aberrazione sferica. Questo fenomeno si può rappresentare come un filtraggio passa-basso nello spazio.

Nella retina le radiazioni luminose sono convertite in segnali bioelettrici. La retina è principalmente costituita da fotorecettori, neuroni specializzati in grado di indurre variazioni nel potenziale di membrana se colpiti da radiazioni elettromagnetiche che cadono nello spettro del visibile. Poi l'informazione visiva viaggia sotto forma di segnale bioelettrico tra i neuroni. Infine, il nervo ottico provvede alla comunicazione con il cervello.

2.2 Il colore come segnale RGB

La visione dell'ambiente circostante che hanno gli animali e gli

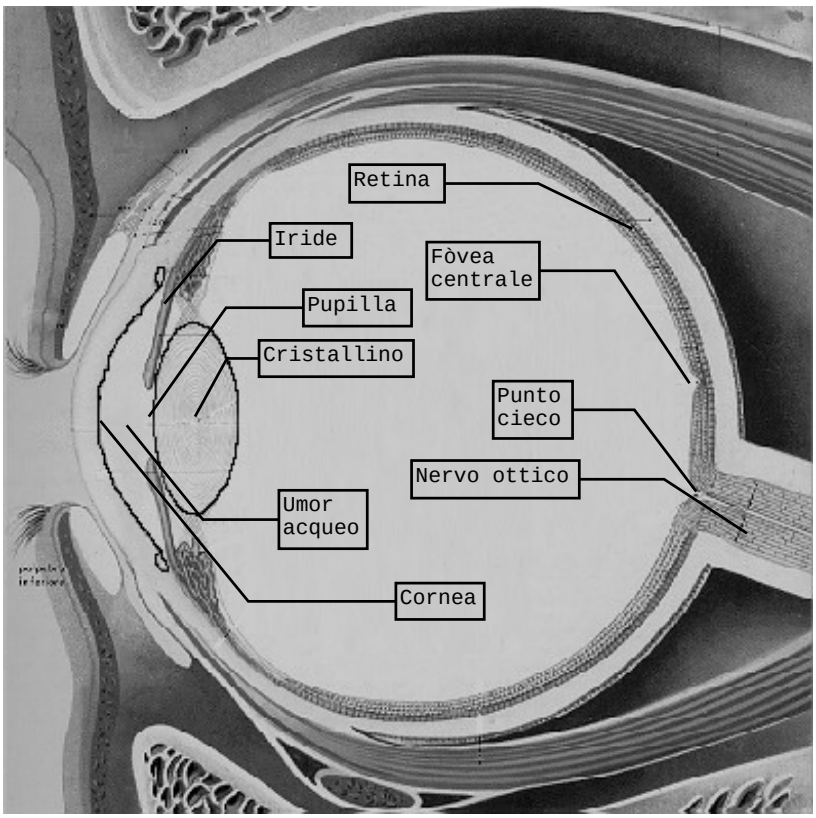


figura 2.1 Sezione ingrandita dell'orbita oculare

esseri umani è resa possibile dalla sensibilità dell'occhio a vibrazioni elettromagnetiche di lunghezza d'onda determinata. Non tutte le lunghezze d'onda possono essere recepite dall'uomo: i raggi infrarossi e ultravioletti non riescono a stimolare l'occhio umano, mentre possono essere percepiti dalle api (ultravioletti) e dai crostacei (infrarossi).

Nella retina sono distribuiti in modo caratteristico due diversi tipi di recettori:

- i coni, che predominano nella fòvea[1] centrale, ove l'acutezza visiva è massima (in questo punto ognuna delle cellule sensitive è collegata ai centri ottici del sistema nervoso centrale);
- i bastoncelli, che predominano alla periferia della retina.

In ogni occhio sono presenti circa 6 milioni di coni e 120 milioni di bastoncelli.

La *teoria della specificità* suppone, nella sua espressione classica, che i coni servano alla visione del colore e i bastoncelli alla percezione dell'intensità luminosa; essa si basa su numerose osservazioni fatte sui vertebrati:

- riduzione della percezione dei colori, per esempio nei pesci, quando i coni si trovano nello strato pigmentato;
- gli animali che hanno solo bastoncelli (geco[2], pipistrello, gatto) non distinguono i colori, ma solo l'intensità luminosa.

Tale teoria non può tuttavia essere considerata generale, in quanto negli invertebrati non esistono due tipi diversi di fotorecettori.

[1] La fòvea è un leggero affossamento nell'area centrale della retina dei vertebrati, dove sono maggiormente addensati e assottigliati gli elementi sensibili terminali (coni e bastoncelli) e che rappresenta il centro della più acuta visione dell'occhio. In molte specie di uccelli (nei rapaci diurni), anziché un'unica fovea centrale, si ha una fovea doppia (una centrale, l'altra laterale).
[2] Il geco è il nome comune di alcuni rettili della famiglia dei Geconidi, comuni in Italia, e in particolare della tarantola o stellione, abbondante in tutta l'Italia peninsulare, e del tarantolino, caratteristico della Sardegna e della Corsica.

La *teoria dei tre componenti*, (T. Young[3] e H. Helmholtz[4]) si basa sulla constatazione che tutte le percezioni cromatiche si possono ottenere con differenti combinazioni di tre colori. Se ne deduce che la retina contiene tre diversi sistemi di recettori con diversi fotopigmenti, il cui massimo di eccitamento è determinato da una lunghezza d'onda specifica. Questo equivale ad affermare che gli uomini hanno una visione a tre colori della realtà.

Un famoso chimico inglese, John Dalton[5], oltre a lasciare una famosa teoria dell'atomo, nel 1794 scoprì su se stesso il fenomeno dell'insensibilità ad alcuni colori, che da allora prende il nome di daltonismo. Egli lasciò alla scienza i suoi occhi perché questo difetto visivo fosse studiato e approfondito. Nei daltonici i coni sono meno sensibili ad alcune lunghezze d'onda: chi non vede il rosso è affetto da "protanopia", chi non vede il verde da "deuteranopia" e chi non vede il blu (raro) da "tritanopia". Ci sono circa otto persone su cento che hanno uno di questi tre difetti, e una su centomila che invece, essendo insensibile a due colori, vede praticamente solo in bianco e nero.

Una volta che l'informazione è stata raccolta, più o meno correttamente, dai coni, il cervello mischia le tre stimolazioni e ricava quella globale dando all'uomo la sensazione di colore che più gli è familiare.

Una analogia alla rappresentazione dei colori dell'occhio umano è costituita dalla modalità di funzionamento di un

[3] Thomas Young, nato a Milverton nel 1773 e morto a Londra nel 1829, medico, fisico ed egittologo. Fra le sue numerose scoperte, nel 1801 descrisse il difetto visivo noto oggi come astigmatismo, e pochi anni dopo formulò la teoria della percezione dei colori, ripresa e approfondita da H. Helmholtz, secondo la quale la percezione dipende dalla presenza sulla retina di tre strutture nervose, eccitate rispettivamente dal rosso, dal verde e dal violetto.

[4] Hermann Ludwig Ferdinand von Helmholtz, nato a Potsdam nel 1821 e morto a Berlino nel 1894, fisiologo, matematico e fisico.

[5] John Dalton, nato a Eaglesfield nel 1766 e morto a Manchester nel 1844, chimico, matematico e fisico. La legge di Dalton stabilisce l'additività delle pressioni parziali dei componenti di una miscela di gas. Altre ricerche gli permisero di stabilire che la solubilità dei diversi gas componenti un miscuglio dipende dalla pressione parziale dei componenti nella miscela. La *legge delle proporzioni* multiple e l'*ipotesi atomica* enunciata poi dal Dalton per spiegare tale legge, lo pongono fra i fondatori della concezione atomica della materia. Nel 1794 studiò su sé stesso quel difetto della vista che fu chiamato poi *daltonismo*.

televisore a colori.

Il principio base utilizzato per la televisione a colori (spesso indicata con la sigla TVC) è praticamente lo stesso della stampa in tricromia. Tralasciando le modalità e l'analisi del tipo di segnale (compatibile con la ricezione in bianco e nero) che parte dal ricevitore e viene captato dalle antenne dei televisori, è interessante vedere come viene riprodotta sullo schermo l'immagine a colori, secondo la tecnologia di qualche anno fa.

Appositi circuiti estraggono dal segnale in arrivo le componenti di crominanza, che opportunamente combinate tra loro e con la componente di luminanza, vanno a pilotare i tre cannoni posti nel collo del cinescopio; sullo schermo sono disposte numerosissime terne di fosfori di tipo diverso, secondo geometrie opportune, per i tre colori fondamentali. In prossimità dello schermo è posta una sottile superficie metallica dorata, detta maschera d'ombra, che provvede a fare in modo che i singoli fosfori siano eccitati solo dal fascetto elettronico emesso dal cannone di colore corrispondente. I tre fascetti elettronici, emessi dai cannoni e deviati da un unico giogo di deflessione, si incontrano sulla maschera; gli angoli di incidenza sono tali che, attraverso i fori di questa, ognuno di essi colpisce il solo fosforo del colore corrispondente. La struttura del cinescopio è completata da opportuni magneti posti sul collo del tubo, necessari per ottenere una sufficiente purezza dei colori e un'appropriata convergenza dei tre fasci su tutti i punti dello schermo, e da uno schermo metallico, contro l'interferenza di campi elettromagnetici esterni.

Per quanto riguarda i personal computer, ci sono diversi modi per definire i colori che appaiono sul monitor. I due più comuni sono la rappresentazione RGB (Red, Green, Blu) e HLS (Hue, Saturation, Luminance). In quest'ultimo caso, le tre componenti, che si possono tradurre con tinta, saturazione, luminosità, hanno il seguente significato:

- *hue* (tinta) descrive la tonalità del colore ed è misurata su

uno spettro circolare che parte dal rosso e passando per il verde e il blu torna al rosso;

- *saturation* (saturazione) descrive la purezza della tinta. Un colore con il 100% di saturazione è luminoso e vivido, mentre uno con lo 0% è una tonalità di grigio;
- *luminance* (luminosità) descrive la vivacità del colore. Un colore con il 100% di luminosità è sempre bianco, mentre uno con lo 0% è sempre nero.

Tra i due tipi di rappresentazione appena descritti esiste una corrispondenza, come si può vedere dalla tabella 2.1.

tabella 2.1

Colore	Formato RGB			Formato HSL		
	Red	Green	Blu	Hue	Sat	Lum
rosso	255	0	0	0	240	120
arancione	255	128	0	20	240	120
giallo	255	255	0	40	240	120
verde	0	255	0	80	240	120
azzurro	0	255	255	120	240	120
blu	0	0	255	160	240	120
viola	255	0	255	200	240	120

2.2 Prima soluzione: un unico segnale

La prima soluzione da sperimentare è quella di rappresentare i colori così come sono nella realtà. Ciò significa preparare, per ognuno dei tre colori fondamentali, un segnale di una certa lunghezza d'onda. I tre segnali risultanti, essendo di lunghezze d'onda diverse, possono essere riuniti in uno unico, che attraverso un attuatore viene trasmesso al cieco.

Per sperimentare la sensibilità del dito ad un segnale di frequenza variabile, si prova, appoggiando una mano ad un altoparlante e non ascoltando il suono, a riconoscere diverse note musicali, tutte emesse alla stessa intensità sonora.

Dopo alcune sedute di sperimentazione, si nota che i risultati ottenuti sono scadenti, sia per risoluzione (cinque o sei note riconoscibili), sia per problemi di offset (ad esempio, errori di un livello su tutte le note suonate).

Una prova ulteriore riguarda il riconoscimento di un accordo di tre note ad intensità sonora variabile generate da un computer. I tre segnali così prodotti sono inviati tutti ad uno stesso altoparlante a cui è appoggiato il dito. Se ne ricavano solamente accordi che si confondono tutti tra loro, o meglio, si nota che quello che si distingue è l'ampiezza del segnale, indipendentemente dalle frequenze da cui è composto.

Il risultato di queste prime sperimentazioni è deludente, ma porta a considerare che il punto debole di questa scelta risiede nell'utilizzo di un solo attuatore. Benché sia opportuno impegnare la mano del cieco nel modo più discreto possibile, non è possibile riunire i tre segnali in uno unico. Nel prossimo paragrafo si analizza una soluzione che prevede l'uso di tre attuatori.

2.3 Seconda soluzione: tre oscillazioni monofrequenza

Dalla precedente esperienza si deduce, oltre alla riconsiderazione sulla scelta del numero di attuatori, che è molto più facile distinguere due segnali alla stessa frequenza, ma con ampiezza diversa, piuttosto che due segnali a frequenza diversa, ma con la stessa ampiezza. Da qui deriva la necessità di applicare tre diversi attuatori, uno che trasduca un segnale proporzionale al rosso, uno al verde e uno al blu. I tre attuatori possono essere applicati a tre falangi distali di tre dita della stessa mano, ma sarebbe più opportuno limitare il numero di dita impegnate nell'acquisizione dati utilizzando le tre falangi di un unico dito, magari quello già impegnato dalla struttura robotica del VIDET.

La disposizione degli attuatori potrebbe essere questa:

- attuatore del rosso: falange distale, dal lato del palmo della mano;
- attuatore del verde: falange mediale, dal lato dorsale della mano;
- attuatore del blu: falange prossimale, dal lato del palmo della mano;
- struttura robotica: falange mediale, dal lato del palmo della mano.

In questo modo si riconoscono tutte le configurazioni di colore a disposizione, anche se si nota la difficoltà di distinguere il segnale della falange prossimale quando alle falangi distale e mediale sono applicati due segnali alla massima ampiezza. Il problema non si risolve nemmeno cambiando la frequenza del solo segnale applicato alla falange prossimale, come si poteva prevedere dopo le prove descritte nel paragrafo precedente. Ad ogni modo, come si vedrà nel capitolo dedicato al software,

questo non costituisce un problema a causa della codifica scelta per rappresentare i colori.

Una volta preparata la codifica del segnale, il software e l'hardware, si procederà ai test, che saranno condotti su persone precedentemente addestrare a riconoscere i segnali.

L'addestramento consiste nell'associare il colore visto sul video all'insieme dei tre segnali applicati alla mano; una volta che la persona abbia imparato a riconoscere i tre colori fondamentali, si procede a mischiarli tra loro per ottenere tutti gli altri.

Fra le persone utilizzate per la realizzazione dei test sono compresi alcuni soggetti videolesi, alcuni fin dalla nascita, altri che in passato erano dotati di una vista normale: dunque i soggetti che in gioventù erano normovedenti conoscono il significato dei colori e riescono a visualizzarli mentalmente.

Tutti i test effettuati sono riportati nel capitolo 5, intitolato appunto "Sperimentazione".

HARDWARE

3.1 Schema generale

Il progetto hardware (figura 3.1) prevede l'utilizzo di un personal computer con software appositamente preparato, la cui descrizione è tema del prossimo capitolo. Il computer prepara sulla porta parallela i segnali da mandare al circuito elettronico illustrato in questo capitolo.

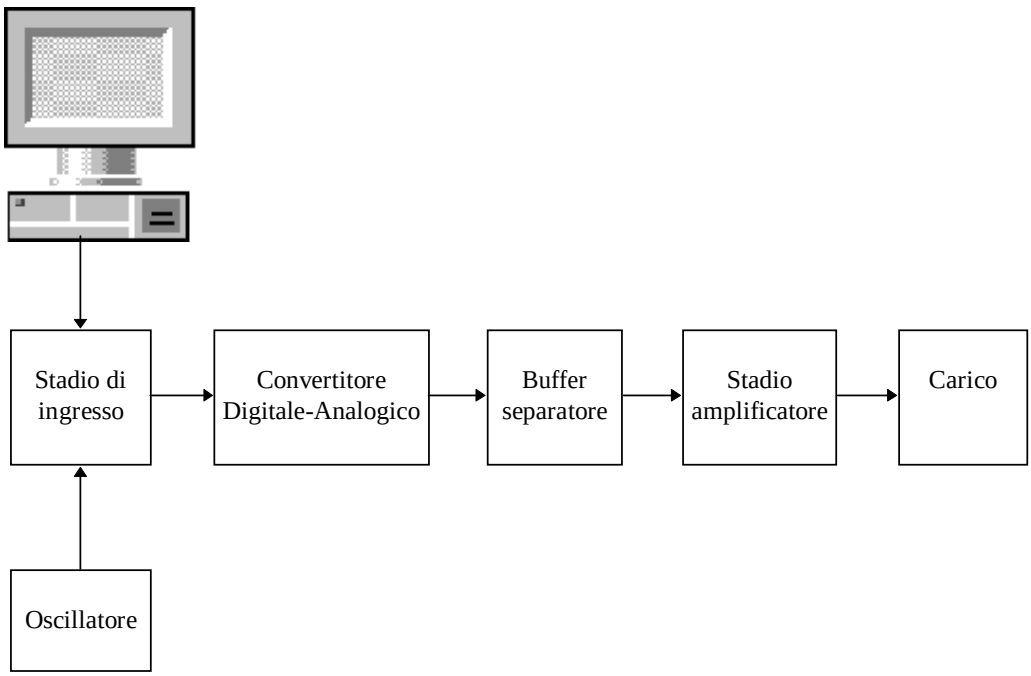


figura 3.1 - Schema a blocchi del circuito

Ognuno dei tre colori fondamentali deve essere quantizzato via software in tre livelli (minimo, intermedio e massimo) e pertanto è rappresentabile con due bit.

La porta parallela dispone di quattro bit di ingresso e otto bit di uscita, quindi è possibile far uscire contemporaneamente e

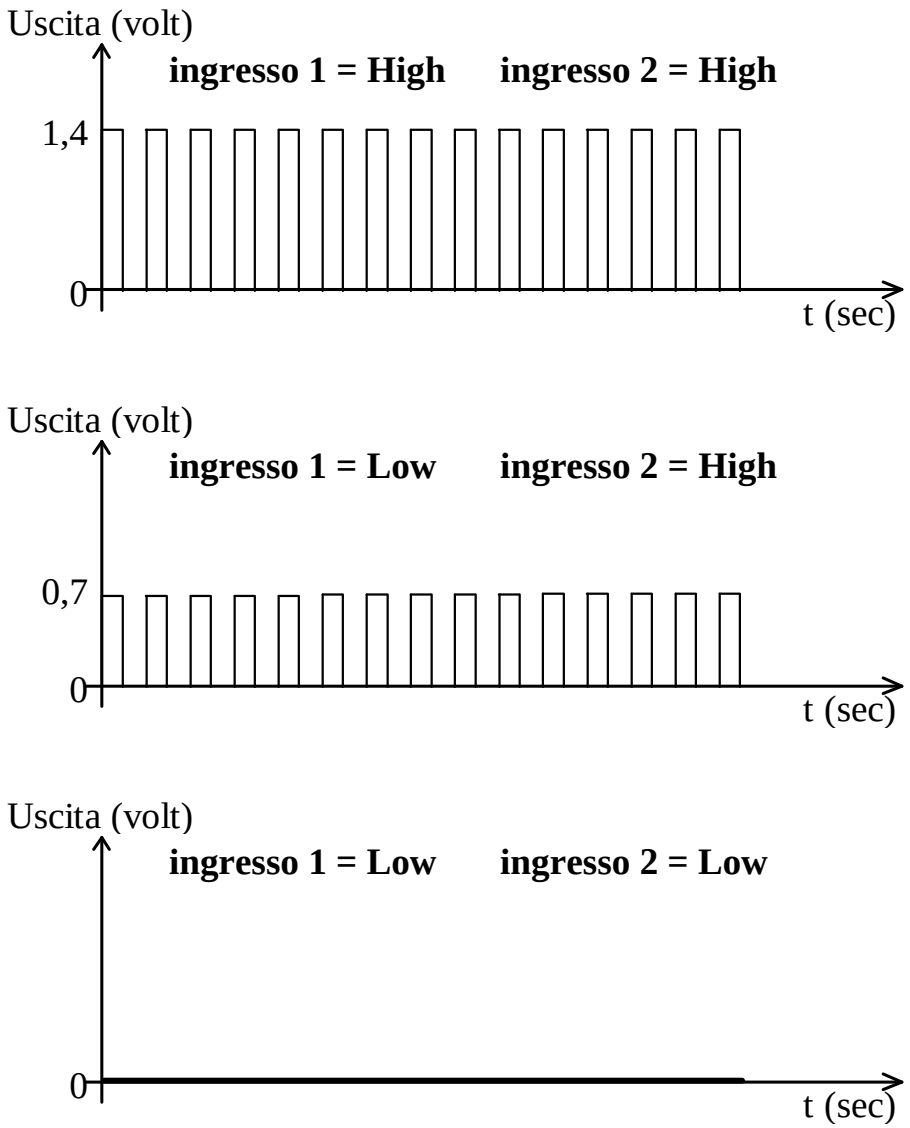


figura 3.2 - Forme d'onda dell'uscita in funzione delle possibili configurazioni dell'ingresso

senza ulteriori codifiche, e successive decodifiche, i sei bit che sono necessari per trasmettere i livelli logici dei tre colori.

La scheda deve elaborare ogni coppia di bit di ingresso e rendere disponibili sulle sue uscite i segnali da inviare agli attuatori; l'intensità di tali segnali varia secondo lo schema in figura 3.2. Nella figura è analizzato il comportamento dell'uscita corrispondente alla coppia di bit che porta l'informazione sulla tonalità del colore rosso: poiché il circuito è replicato tre volte, le forme d'onda illustrate sono valide anche per le uscite corrispondenti al verde e al blu.

I sei segnali d'ingresso devono essere elaborati da uno stadio ad alta impedenza d'ingresso per non caricare eccessivamente il circuito di uscita della porta parallela e quindi per non rischiare il danneggiamento del personal computer; inoltre, attraverso una rete logica e con l'aiuto di un oscillatore, vengono trasformati in onde quadre.

L'informazione digitale dei due bit (per ognuno dei tre colori) viene trasformata in un segnale analogico attraverso un convertitore digitale-analogico, realizzato con due resistenze di valore uguale ed una di valore variabile per ottenere i livelli adeguati necessari per l'elaborazione del segnale, che avviene nello stadio di amplificazione finale.

Per non caricare il convertitore si utilizza un amplificatore operazionale configurato ad inseguitore.

Lo stadio di uscita è costituito da uno stadio amplificatore realizzato con un transistor bipolare configurato a collettore comune. Il segnale di uscita giunge poi ad un attuatore che per il momento è costituito semplicemente da un altoparlante estratto da una cuffia audio.

L'intero circuito funziona con tensione di alimentazione continua tra 0 e 5 volt.

3.2 Oscillatore

L'oscillatore serve per ottenere un'onda quadra di circa 4 volt di ampiezza e 75 Hertz di frequenza: questa, posta in ingresso a tutti gli AND dello stadio iniziale, serve per avere nello stadio di uscita un segnale ad onda quadra invece di un segnale continuo.

Esistono in commercio dei temporizzatori adatti a questo ruolo: il più diffuso è il timer 555, un circuito integrato lineare prodotto dalla Signetics dal 1972 con tecnologia bipolare (figura 3.3).

Il timer 555 può produrre accurate temporizzazioni, da alcuni microsecondi ad alcune decine di minuti, e soprattutto, visto che la sua uscita deve essere messa in ingresso ad un AND TTL, se viene alimentato a 5 volt ha anch'esso un'uscita TTL compatibile. Questo significa che non occorre alcun tipo di interfaccia tra il timer 555 e il gate AND, anch'esso a tecnologia bipolare. Inoltre, caratteristica in questo caso non indispensabile, la presenza di un driver permette di avere all'uscita una corrente fino a 200 mA, assorbita o erogata a seconda del livello rispettivamente basso o alto dell'uscita stessa; questo è comunque garanzia che il carico

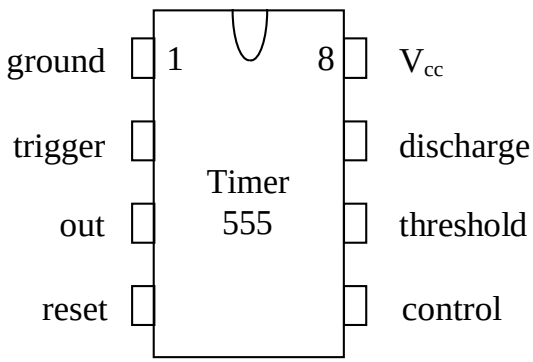


figura 3.3 - Piedinatura del timer 555

applicato in uscita non influirà in nessun caso sulla forma d'onda generata.

Il circuito da realizzare è un circuito astabile che deve oscillare alla frequenza di 75 Hertz e con un duty cycle del 50%. Lo schema del circuito è riportato in figura 3.4, mentre le forme d'onda relative sono in figura 3.5.

Il condensatore C tende a caricarsi alla tensione V_{CC} attraverso le resistenze R_1 ed R_2. Essendo però connesso all'ingresso di threshold del 555, appena V_C raggiunge il valore di soglia ($2V_{CC}/3$), il latch del 555 viene resettato, quindi l'uscita out passa a livello basso e il transistor di scarica entra in saturazione. Il condensatore pertanto si scarica attraverso la resistenza R_2 ed il transistor di scarica che fa capo al pin discharge del 555. La

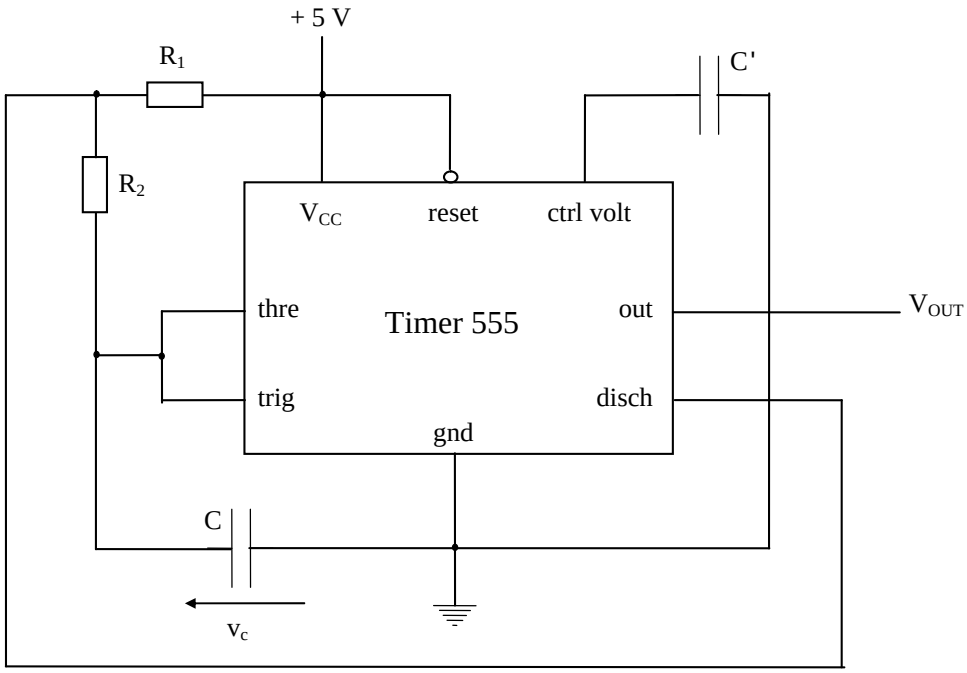


figura 3.4 - Schema del circuito monostabile realizzato con il timer 555

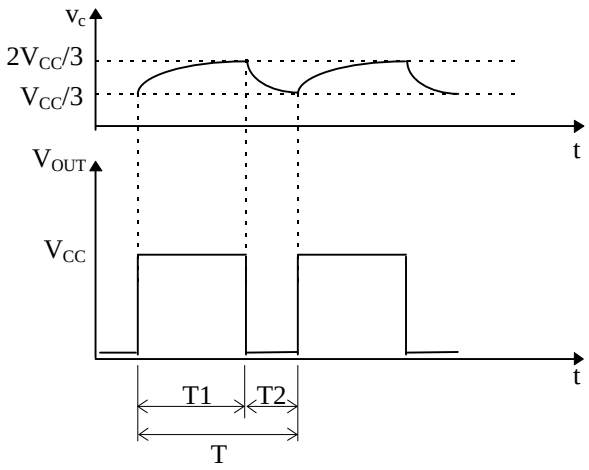


figura 3.5 - Forme d'onda relative al funzionamento del circuito astabile

tensione del condensatore tende a zero, però non riesce a raggiungere tale valore in quanto C è connesso all'ingresso di trigger del 555 e, quando $v_c=V_{CC}/3$, il latch del 555 viene settato, l'uscita out passa a livello alto e il transistor di scarica viene bloccato. Inizia allora una nuova fase di carica del condensatore, ripetendosi il ciclo appena descritto.

Il condensatore C', connesso tra l'ingresso control voltage e massa, serve ad evitare che disturbi sul pin di ingresso non collegato modifichino la durata della forma d'onda di uscita. Dunque il suo valore non è significativo (C'=10nF).

3.2.1 Fase di carica

Il condensatore C è inizialmente carico con una tensione $v_c=V_{CC}/3$ (a regime). Attraverso le resistenze R_1 ed R_2, esso tende

a portarsi alla tensione di alimentazione V_{CC}, con costante di tempo $(R_1+R_2)C$, per cui l'equazione di carica è:

$$v_c = V_{cc} + \left(V_{cc}/3 - V_{cc}\right) * e^{-t'/(R1+R2)*C} . \quad (3.1)$$

La carica è interrotta, però, all'istante t'=T1, in cui $v_c=2V_{CC}/3$, per cui si ha:

$$2V_{cc}/3 = V_{cc} + \left(V_{cc}/3 - V_{cc}\right) * e^{-T1/(R1+R2)*C} , \quad (3.2)$$

da cui si ottiene l'intervallo di tempo di carica T_1:

$$T_1 = \left(R_1 + R_2\right)C \ln 2 = 0{,}693\left(R_1 + R_2\right)C . \quad (3.3)$$

3.2.2 Fase di scarica

Il condensatore C, a regime, è inizialmente carico con una tensione $v_c=2V_{CC}/3$. Attraverso la resistenza R_2 esso tende a scaricarsi con costante di tempo R_2C, per cui l'equazione di scarica è:

$$v_c = \left(2\ V_{cc}/3\right) * e^{-t''/R2*C} ; \quad (3.4)$$

la scarica è però interrotta all'istante t''=T_2, in cui $v_c=V_{CC}/3$, per cui si ha:

$$V_{cc}/3 = \left(2\ V_{cc}/3\right) * e^{-T2/R2*C} , \quad (3.5)$$

da cui si ottiene l'intervallo di tempo di scarica T_2:

$$T_2 = R_2 C \ln 2 = 0{,}693 R_2 C . \tag{3.6}$$

Il periodo T di oscillazione dell'astabile vale pertanto:

$$T = T_1 + T_2 = 0{,}693 \left(R_1 + 2R_2 \right) C . \tag{3.7}$$

La frequenza f di oscillazione è:

$$f = 1/T = \frac{1{,}44}{\left(R_1 + 2R_2 \right) C} . \tag{3.8}$$

Il duty cycle δ, definito come il rapporto fra l'intervallo di tempo in cui l'uscita è alta e l'intero periodo di oscillazione, vale:

$$\delta = T_1/T = \frac{R_1 + R_2}{R_1 + 2R_2} . \tag{3.9}$$

Si può notare come il duty cycle possa essere compreso solo fra circa il 50%, quando R_2 è molto maggiore di R_1, e circa il 100%, quando R_2 è molto minore di R_1.

3.2.3 Scelta dei valori di resistenze e capacità

Per ottenere un duty cycle del 50% circa, è sufficiente che il valore di R_2 sia circa 100 volte quello di R_1: la capacità C resta

dunque funzione della sola frequenza f di oscillazione desiderata.

Posto R_1=4,7 KΩ,

R_2=470 KΩ,

f=75 Hz,

si ha, dalla (3.8):

$$C = \frac{1{,}44}{(4700 + 2*470000)* f} = \frac{1{,}44}{944700*75} \cong 20*10^{-9} = 20\text{nF} \quad (3.10)$$

.

3.3 Stadio di ingresso

Lo stadio di ingresso si propone di acquisire i segnali provenienti dal personal computer. Dovendo fare un AND tra i segnali di ingresso e la forma d'onda generata dal timer 555, è superfluo inserire un ulteriore stadio di separazione per acquisire i segnali dalla porta parallela, in quanto tra le specifiche tecniche della porta logica c'è una resistenza di ingresso molto elevata (tecnologia bipolare) e tra quelle della porta parallela c'è una corrente massima di uscita di 1mA e i livelli della tensione di uscita sono TTL compatibili.

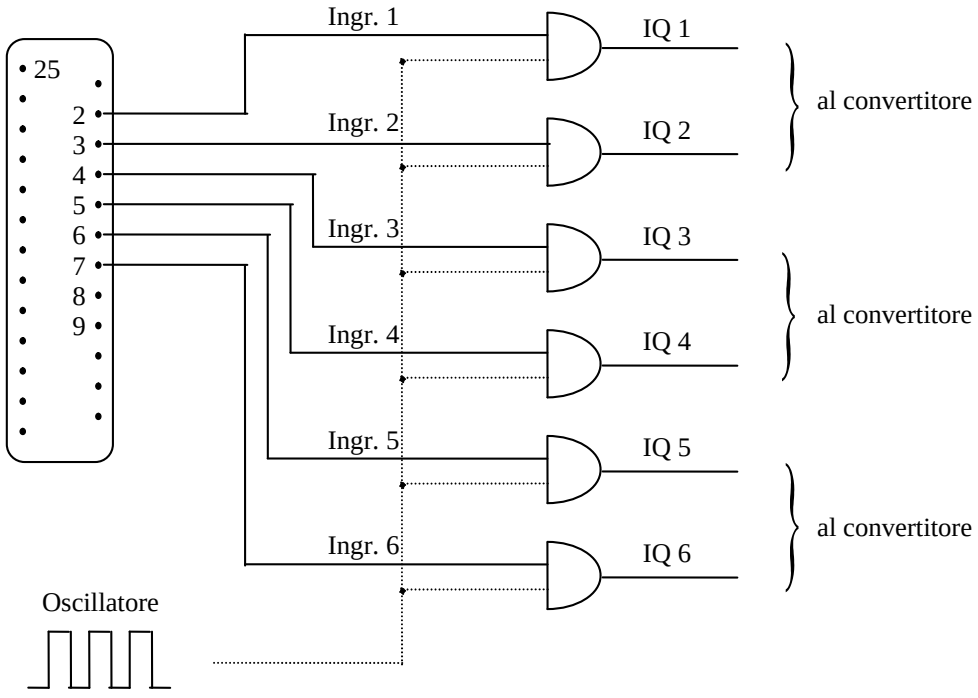


figura 3.6 - Stadio di ingresso

È dunque sufficiente effettuare i collegamenti illustrati in figura 3.6.

I pin 2 e 3, chiamati rispettivamente ingresso 1 ed ingresso 2, portano l'informazione che riguarda il colore rosso, i pin 4 e 5, chiamati ingresso 3 ed ingresso 4, riguardano il colore verde, mentre i pin 6 e 7, chiamati ingresso 5 ed ingresso 6, riguardano il colore blu. I pin 8 e 9 non vengono utilizzati e rimangono pertanto scollegati.

Il significato del livello logico dei sei segnali di ingresso è illustrato nelle seguenti tabelle, dove per ingresso=0 si intende livello logico basso (0 volt) e per ingresso =1 livello logico alto (5 volt):

Ingresso 1	Ingresso 2	Significato
0	0	Non c'è rosso
0	1	Rosso scuro
1	0	*config. non utilizzata*
1	1	Rosso chiaro

Ingresso 3	Ingresso 4	Significato
0	0	Non c'è verde
0	1	Verde scuro
1	0	*config. non utilizzata*
1	1	Verde chiaro

Ingresso 5	Ingresso 6	Significato
0	0	Non c'è blu
0	1	Blu scuro
1	0	*config. non utilizzata*
1	1	Blu chiaro

3.4 Convertitore D/A

Per ognuna delle tre coppie di segnali riguardanti i singoli colori è necessaria una operazione di conversione digitale-analogica. Il modo più semplice di realizzarla, dato il limitato numero di bit, è quello di utilizzare un partitore, secondo lo schema figura 3.7.

Se i due segnali di ingresso IQ_1 e IQ_2, provenienti dallo stadio precedente, sono nello stato logico alto, cioè 5 volt, nelle resistenze R_1 e R_2 non scorre corrente e il segnale di uscita è anch'esso nello stato logico alto. Se invece IQ_1 e IQ_2 sono entrambi nello stato logico basso (0 volt), nelle resistenze non scorre ugualmente corrente, ma il segnale di uscita è nello stato logico basso. Se invece IQ_2 è nello stato logico alto e IQ_1 nello stato basso, nelle resistenze R_1 ed R_2 scorre una corrente pari a $5/(R_1+R_2)$ Ampere e l'uscita si assesta ad un valore pari a $5*R_1/(R_1+R_2)$ volt. Pertanto, affinché sul partitore scorra una corrente di circa 0,25 mA, valore adeguato rispetto alla corrente che deve passare sul partitore per polarizzare lo stadio successivo

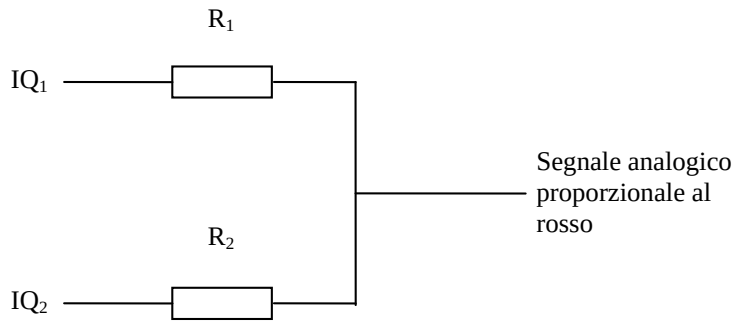


figura 3.7 - Schema di conversione digitale-analogico

ad elevata impedenza d'ingresso, il valore della somma delle resistenze R_1+R_2 deve essere di circa 20 KΩ.

Per avere nel caso della terza configurazione un segnale di uscita di 2,5 volt, è sufficiente scegliere R_1=R_2=10 KΩ. Dovendo però amplificare il segnale nello stadio finale ed essendo quest'ultimo realizzato con tecnologia bipolare, più precisamente con un transistor configurato a collettore comune, non è consigliabile avere un segnale di uscita al convertitore di 2,5 volt, perché bisogna tenere conto della soglia Vbe della giunzione base-emettitore. Se si sceglie 2,5 volt all'uscita del convertitore, non è possibile avere nello stadio finale tre segnali adeguatamente distanziati tra loro. Il valore esatto di tensione all'uscita del convertitore è determinato dopo avere scelto la resistenza di base del transistor.

Per lasciare una possibilità di regolazione di questo livello di tensione, la resistenza R_2 deve essere intesa come una resistenza variabile. Questo è necessario perché, trattandosi della configurazione intermedia tra spento e acceso, è possibile che l'utilizzatore voglia poter variare il livello del segnale che recepisce, alzandolo se lo confonde con quello basso e abbassandolo se non lo distingue da quello alto. Inoltre non è detto che i tre livelli intermedi debbano essere uguali tra di loro, perché i tre attuatori che trasmettono i segnali al corpo umano sono applicati in tre punti diversi del dito.

Per questo motivo si ricorre allo schema precedente con l'aggiunta di una resistenza variabile R_V di 4700 Ω in serie ad R_2, che pertanto assumerà insieme ad R_1 il valore di 10 KΩ.

La corrente che scorre sul partitore varia, a seconda del valore di R_V, da un minimo di 0,2 mA a un massimo di 0,25 mA, mentre la corrente di ingresso allo stadio successivo è stimabile in soli 7 ηA.

3.5 Buffer

Si utilizza, all'uscita di ognuno dei tre convertitori, uno stadio separatore per assorbire la minima corrente possibile dal partitore illustrato nella sezione precedente. Lo stadio separatore è costituito da un amplificatore operazionale con il segnale di ingresso sul terminale positivo e la retroazione dell'uscita sul terminale negativo. L'alimentazione è in tensione continua tra 0 e 5 volt. Lo schema è illustrato in figura 3.8.

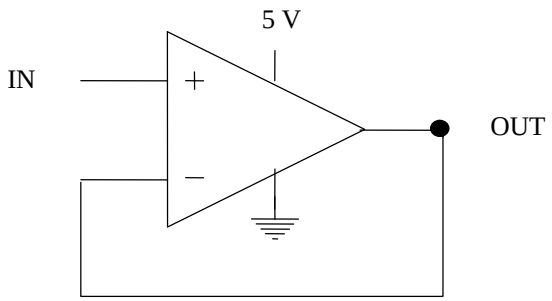


figura 3.8 - Stadio separatore ad elevata impedenza d'ingresso

3.6 Stadio amplificatore di uscita

Lo stadio di uscita è costituito da un transistor bipolare npn configurato a collettore comune (figura 3.9). Ogni stadio di uscita è preceduto dallo stadio separatore appena descritto.

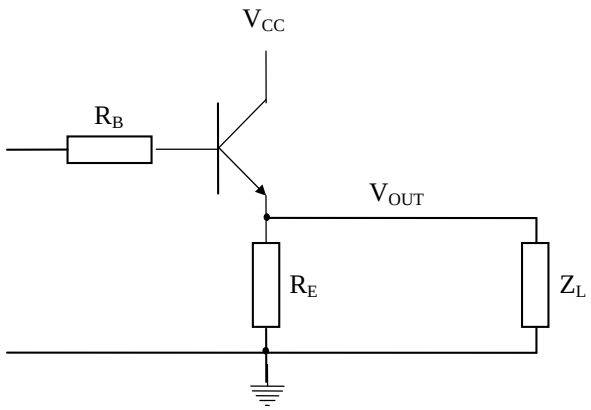


figura 3.9 - Stadio amplificatore di uscita

3.6.1 Calcolo dei valori delle resistenze

Occorre innanzitutto stabilire i livelli massimi desiderati dell'onda quadra di uscita V_{OUT} nei casi di ingresso alto (5 volt), basso (0 volt) ed intermedio. Inoltre bisogna misurare l'assorbimento di corrente da parte del carico nel momento in cui l'uscita è alta.

Si vuole ottenere:

V_{IN}	V_{OUT}	I_{OUT}
bassa	0 volt	0 mA
intermedia	0,7 volt	30 mA
alta	1,4 volt	60 mA

Quando V_{IN} è alta (5 volt) e V_{OUT} è 1,4 volt, il transistor è in conduzione e la giunzione base-emettitore presenta ai suoi capi una tensione di soglia di circa 0,6 volt. Di conseguenza la tensione sul morsetto di base è 0,6+1,4 volt, cioè 2 volt. Poiché sulla R_B c'è una caduta di tensione di circa 3 volt (V_{IN}-V_{BASE}=5V-2V=3V) e su di essa scorre una corrente di base circa 100 volte più piccola di I_{OUT}, si può calcolare il valore di R_B necessario al corretto funzionamento del transistor in queste condizioni:

$$R_B = \frac{V_{IN} - V_{BASE}}{I_{BASE}} = \frac{5 - 2}{\frac{0,06}{100}} = \frac{3}{0,0006} = 5000\Omega = 5K\Omega \ . \qquad (3.11)$$

Quando invece V_{IN} è bassa la tensione ai capi della giunzione base-emettitore non è in ogni caso sufficiente a superare la soglia di accensione, la V_{OUT} è 0 e non c'è corrente in uscita.

Avendo ora calcolato il valore della resistenza di base, è possibile determinare quale valore di V_{IN} è necessario per ottenere V_{OUT}=0,7 volt:

$$V_{IN} = V_{BASE} + R_B * I_{BASE} = 0,7 + 0,6 + 5000*0,0003 = 2,8V \ . \qquad (3.12)$$

V_{IN}=2,8 volt può essere ottenuta dal convertitore digitale-analogico aggiungendo in serie a R2 (figura 3.7) una resistenza

R_3=2727 Ω al posto della resistenza variabile. È per questo motivo che è opportuno scegliere R_V variabile tra 0 e 4700 Ω, cioè per poter regolare a piacimento e personalmente il valore massimo della V_{OUT} intermedia in un intorno del valore ottimale calcolato (2,8 Volt).

Si sceglie inoltre una resistenza di emettitore R_E adeguata: il suo valore deve essere abbastanza alto per non consumare inutilmente corrente; un valore adeguato è 220 KΩ, in modo che la corrente che la attraversa sia limitata a V_{OUT}/220000, che nel caso peggiore (V_{OUT}=1,4 volt) raggiunge l'intensità di 6,3 ηA, valore del tutto trascurabile rispetto all'assorbimento di corrente del carico in questo stato dell'uscita.

Infine, durante la realizzazione del circuito, invece di usare una resistenza di base R_B=5 KΩ, si utilizza una resistenza di valore simile (4,7 KΩ) in quanto molto più facilmente reperibile in commercio.

3.7 Alimentazione

Tutti gli schemi elettrici presentati nelle pagine precedenti funzionano tra 0 e 5 volt, in corrente continua. Bisogna rilevare però che l'assorbimento di corrente, soprattutto da parte dell'oscillatore, avviene a frequenze molto elevate, in quanto il tempo di salita (e di discesa) dell'onda quadra è di circa 100 ns. Questo significa che l'alimentatore deve fornire corrente ad una frequenza di circa $\frac{0{,}35}{100 \cdot 10^{-9}} = 3{,}5\text{MHz}$. Inoltre è elevata anche la corrente assorbita, sempre alla stessa frequenza, dagli attuatori, calcolabile in 60 mA per ognuno. Non considerando il consumo trascurabile di partitori e circuiti logici, l'alimentatore deve fornire poco meno di 200 mA.

Per poter utilizzare un alimentatore a basso costo, senza però avere sbalzi della tensione di alimentazione, si aggiunge, subito prima della sezione di ingresso, un circuito integrato che stabilizzi la propria tensione di uscita a 5 volt, qualunque sia l'ingresso (entro certi limiti). L'integrato presenta il codice 7805, funziona con tensione di ingresso di almeno 6 volt, e va collegato come illustrato in figura 3.10.

L'alimentatore da utilizzare deve avere una tensione di uscita

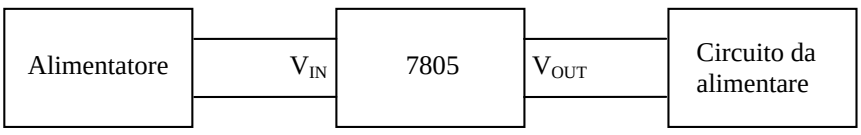


figura 3.10 - Alimentazione

di almeno 6 volt e una corrente massima di erogazione di circa 200 mA.

3.8 Schema completo

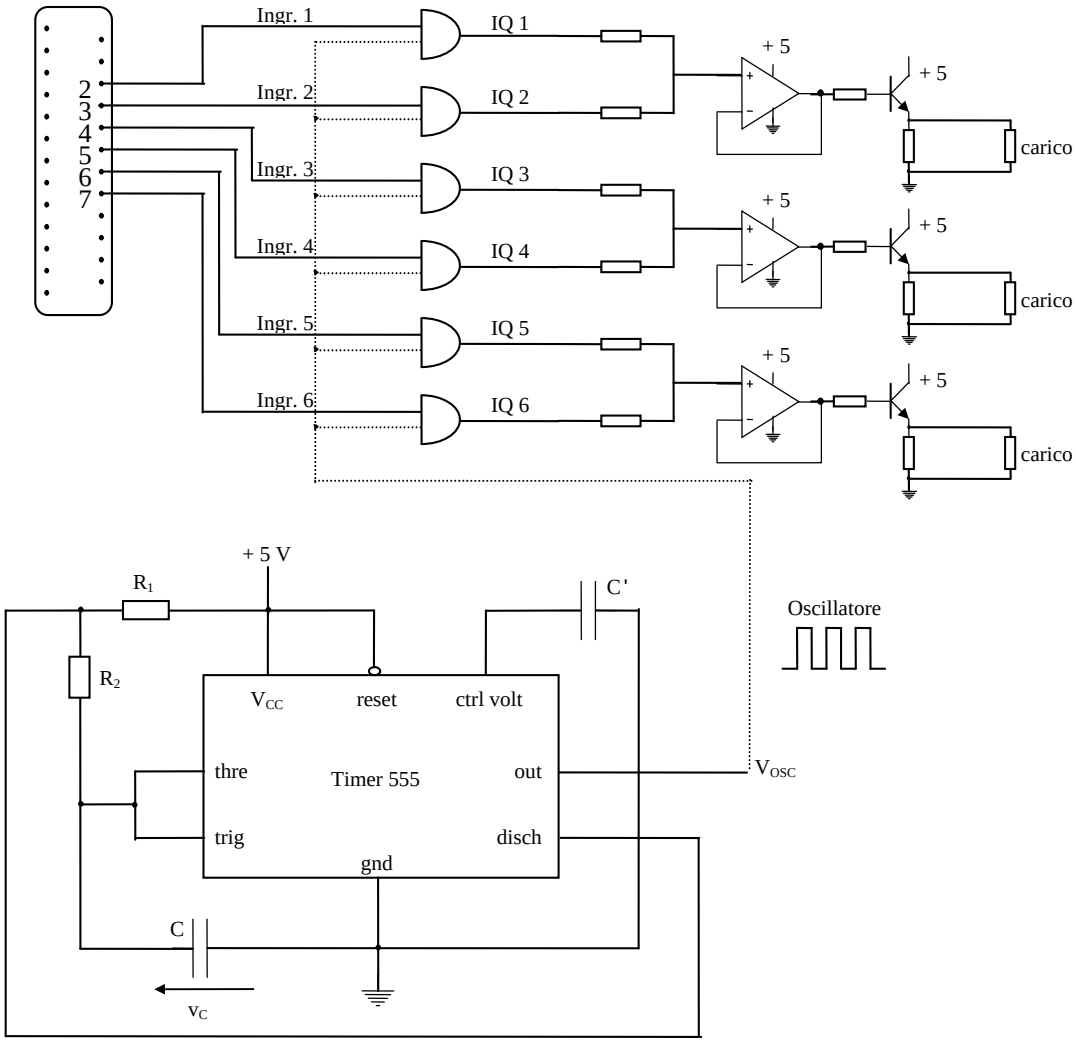


figura 3.11- Schema completo del circuito elettronico

3.9 Simulazione con Spice

Il seguente listato è la descrizione del circuito elettronico da simulare. Seguono la figura 3.12 e la figura 3.13, con le forma d'onda dell'uscita nel caso rispettivamente di ingressi entrambi alti (VINPIU e VINMEN onde quadre con ampiezza uguale a 5 volt) e di un solo ingresso alto.

3.9.1 Descrizione del circuito

```
VCC       1 0       5
VINPIU    3 0 0     pulse(0 5 0 2NS 2NS 0.008 0.016)
VINMEN    4 0 0     pulse(0 5 0 2NS 2NS 0.008 0.016)
REmett    8 0       220K
RLoad     8 0       32
RBase     6 7       5K
RPart1    3 5       10K
RPart2    4 5       10K
EOpAmp    6 0 5 0   1
Q1        1 7 8     2N3904

.OPTIONS ACCT list node
.TRAN 0.0005 0.05 0 0.0005
.MODEL 2N3904 NPN (RB=230       RBM=130    IRB=110UA
                   IS=7.41E-15  BF=280     NF=1.1
                   VAF=143      IKF=30MA   NE=1.33
                   ISE=1.2E-15  BR=2.55    NR=1.05
                   IKR=8.8MA    NC=1.24    ISC=40E-15
                   CJE=7.42PF   VJE=.675   MJE=.348
```

```
                    RE=2.41      RC=0.88     TF=.624Ns
                    XTF=1.09E5   VTF=6.94    ITF=14.4
                    TR=230Ns     CJC=4.56PF  VJC=.503
                    MJC=.354     XCJC=1.0D   XTB=2.0D
                    XTI=3.0D     CJS=0.0D    VJS=.75D
                    MJS=0.0D     EG=1.20D    VAR=22)
.PROBE
.END
```

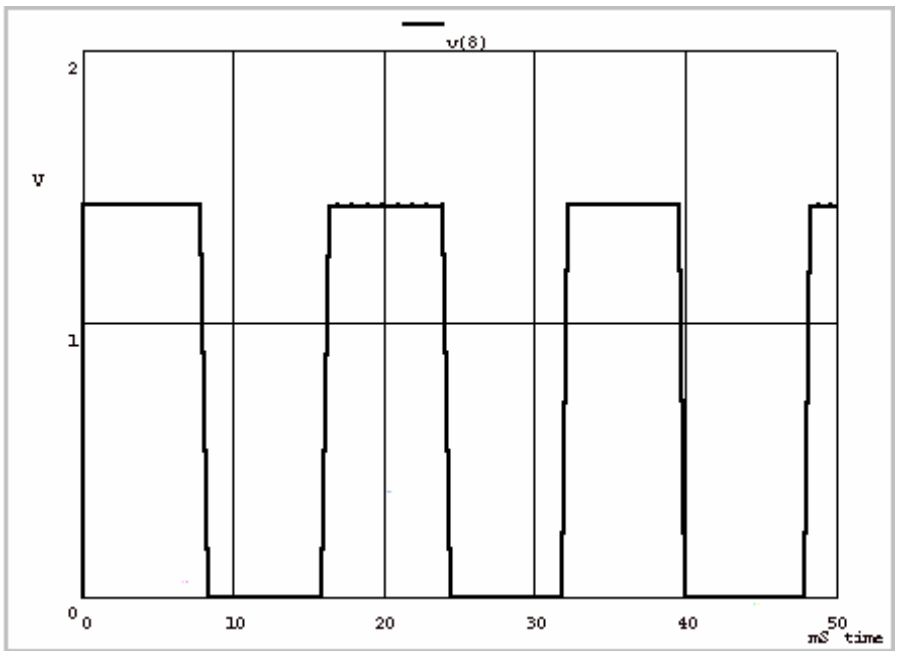


figura 3.12- Segnale di uscita con entrambi gli ingressi alti

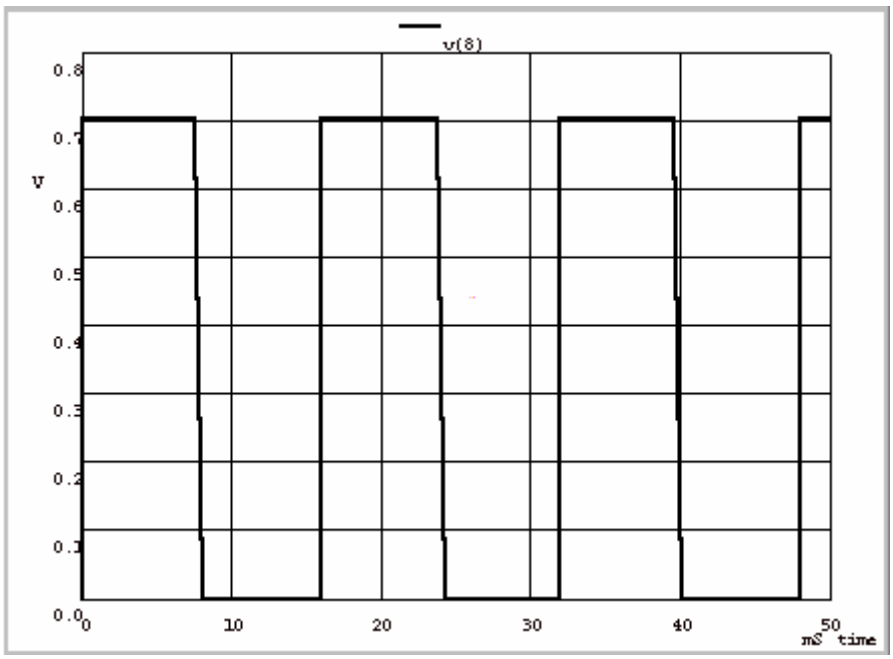
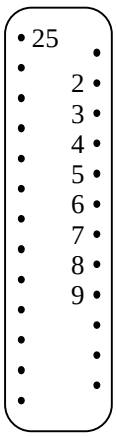


figura 3.13- Segnale di uscita con un solo ingresso alto

SOFTWARE

4.1 Descrizione generale del progetto

Il compito principale del software consiste nel preparare un segnale di uscita dalla porta di output che abbia le caratteristiche logiche necessarie per interfacciarsi al sistema hardware descritto nel capitolo precedente. Il sistema hardware acquisisce tre segnali, ognuno su due bit, che devono essere preparati dal software sui primi sei bit degli otto riservati al *data out* sulla porta parallela. Più precisamente il significato dei *pin* è quello



n° bit	colore	peso
2	rosso	bit più significativo
3	rosso	bit meno significativo
4	verde	bit più significativo
5	verde	bit meno significativo
6	blu	bit più significativo
7	blu	bit meno significativo
8	-	-
9	-	-
25	massa	-

figura 4.1 - Significato della piedinatura della porta parallela

indicato in figura 4.1.

Il software si propone anche di acquisire l'informazione del colore dal VIDET, però al momento non è ancora possibile questo passaggio. Poiché anche il VIDET necessita di un personal computer, almeno nella prima realizzazione, probabilmente l'informazione sul colore potrà essere scambiata via software dai due programmi sullo stesso calcolatore.

Per il momento l'informazione viene generata dal programma stesso, attraverso il puntatore del mouse che "sceglie" un particolare di un'immagine a video. Invece del mouse è possibile utilizzare un *touch screen*, almeno nella fase di prova di riconoscimento del colore puntato da parte di un soggetto videoleso, in modo che quest'ultimo possa, puntando il dito sul monitor, scegliere il particolare dell'immagine di cui vuole riconoscere il colore. Infatti l'utilizzo del mouse non permette di avere un'informazione assoluta sulla posizione del puntatore sullo schermo, ma solamente un'informazione relativa alla direzione dell'ultimo spostamento effettuato; inoltre spesso il mouse non ha un comportamento lineare. Una soluzione intermedia, per quanto riguarda il costo e le prestazioni, può essere l'utilizzo di una tavoletta grafica: questa a differenza del mouse presenta caratteristiche di posizionamento assoluto e di linearità, ma a differenza del *touch screen* non può essere utilizzata strofinando un dito, presupponendo invece l'utilizzo di una penna che deve obbligatoriamente strisciare sulla tavoletta.

Date le caratteristiche del programma da realizzare, è meglio utilizzare un compilatore che preveda un ambiente di programmazione visuale.

Fin dai tempi del Turbo Pascal, Borland ha proposto estensioni al Pascal standard. L'ultimo compilatore, immesso sul mercato nel 1995, si chiama Delphi.

Delphi è un linguaggio di programmazione a basso livello strettamente collegato alle funzioni interne di Windows; esso racchiude queste funzionalità in componenti, oggetti e metodi di

più alto livello. Ciò significa che non è necessario conoscere approfonditamente le funzioni e i comportamenti di Windows per programmare in questo linguaggio, però all'occorrenza non è preclusa alcuna possibilità di interagire a basso livello con il sistema operativo. Delphi è un ambiente orientato agli oggetti e contiene una libreria completa di componenti che rappresentano i più disparati elementi di interfaccia dell'ambiente Windows.

Il file eseguibile creato dal compilatore "gira" su qualunque personal computer 100% IBM compatibile basato su microprocessore 80386 o successivo con almeno 4 MB di RAM e Windows 3.1 o successivi.

Una prima organizzazione della struttura del programma è illustrata in figura 4.2 ed è formata dai seguenti punti:

1) si sceglie un'immagine da visualizzare sul monitor;
2) si proietta l'immagine a schermo intero o quasi intero;
3) si sceglie con il mouse un punto dell'immagine;
4) si "clicca" con il mouse sul punto prescelto (o si muove semplicemente il mouse sull'immagine);
5) si applica l'algoritmo di conversione dell'informazione del colore e si ottengono le tre coppie di bit da mandare in uscita;
6) si prepara un byte con le precedenti informazioni e lo si manda sulla porta parallela;
7) si torna al punto 3 per scegliere un nuovo punto oppure si termina l'esecuzione del programma.

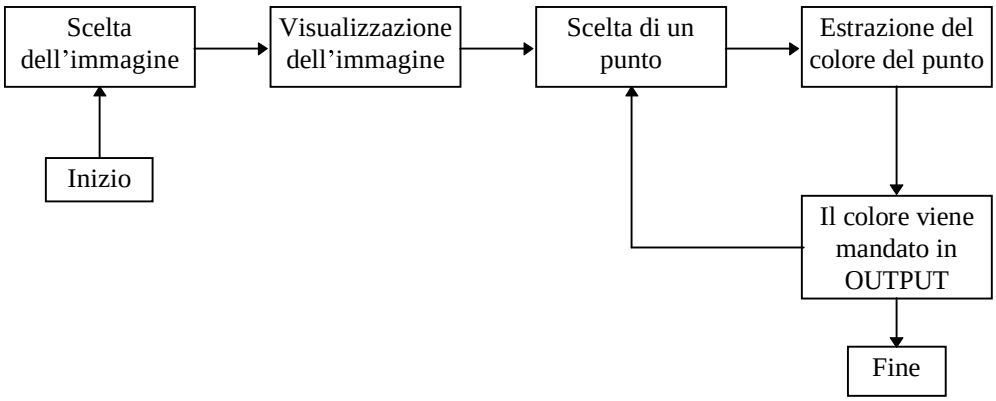


figura 4.2 - Organizzazione del programma

4.2 Approssimazione dell'informazione sul colore

Il punto fondamentale su cui operare è la conversione dell'informazione sul colore, partendo dalla rappresentazione del calcolatore che necessita di otto bit per ognuno dei tre colori fondamentali e volendo giungere ad una rappresentazione che invece deve "occupare" solamente due bit (in totale tre coppie di due bit).

Nella rappresentazione RGB il calcolatore utilizza un byte per il colore rosso, uno per il colore verde e uno per il blu; ogni colore è univocamente associato ad un numero che può andare da 0 a 255. Ci sono naturalmente altri tipi di rappresentazioni possibili, ma questa è la più comoda in quanto si possono trattare i tre colori separatamente, e quindi progettare un circuito elettronico più semplice e replicato tre volte. Inoltre la rappresentazione RGB è la stessa che inconsciamente l'uomo utilizza: gli uomini hanno una visione tricromatica della realtà, in quanto l'occhio umano presenta tre diversi tipi di coni sulla retina, ognuno sensibile ad una frequenza diversa.

Il procedimento utilizzato per la riduzione del numero di colori è quello di suddividere le 256 possibili gradazioni di ciascuno in tre insiemi disgiunti, assegnare ad ogni insieme un valore prestabilito ed approssimare ogni gradazione con il valore predeterminato dell'insieme di cui fa parte. In questo modo dopo l'approssimazione sono distinguibili $3^3=27$ diversi colori; alcuni di questi saranno molto simili tra loro, ma ciò è un vantaggio perché è sempre meglio avere un certo margine di errore, in modo da non dover confondere, ad esempio, un verde con un rosso, ma piuttosto un rosso con un arancione.

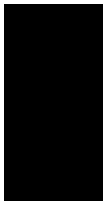
Poiché si dispone di due bit in uscita, è ipotizzabile utilizzare tutte le quattro configurazioni possibili: però una suddivisione in

quattro insiemi, pur fornendo un numero molto più alto di colori, causerebbe una maggiore confusione fra i quattro livelli di segnale da riconoscere. Questa soluzione potrà essere scelta in futuro, qualora si disponesse di attuatori più precisi.

Avendo scelto di quantizzare ogni colore in tre insiemi, questi sono i livelli di quantizzazione risultanti (figura 4.3) e l'elenco dei 27 colori.

min max	approssimazione
0 - 84	0
85 - 170	128
171 - 255	255

figura 4.3 - Quantizzazione del colore



Rosso	0
Verde	0
Blu	0

Rosso	0
Verde	0
Blu	128

Rosso	0
Verde	0
Blu	255

Rosso	0
Verde	128
Blu	0

Rosso	0
Verde	128
Blu	128

Rosso	0
Verde	128
Blu	255

Rosso	0
Verde	255
Blu	0

Rosso	0
Verde	255
Blu	128

Rosso	0
Verde	255
Blu	255

Rosso	128
Verde	0
Blu	0

Rosso	128
Verde	0
Blu	128

Rosso	128
Verde	0
Blu	255

Rosso	128
Verde	128
Blu	0

Rosso	128
Verde	128
Blu	128

Rosso	128
Verde	128
Blu	255

Rosso	128
Verde	255
Blu	0

Rosso	128
Verde	255
Blu	128

Rosso	128
Verde	255
Blu	255

Rosso	255
Verde	0
Blu	0

Rosso	255
Verde	0
Blu	128

Rosso	255
Verde	0
Blu	255

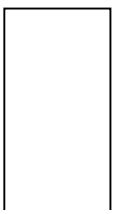
Rosso	255
Verde	128
Blu	0

Rosso	255
Verde	128
Blu	128

Rosso	255
Verde	128
Blu	255

Rosso	255
Verde	255
Blu	0

Rosso	255
Verde	255
Blu	128



Rosso	255
Verde	255
Blu	255

Alma Mater Studiorum
Secularia Nona

Come si può notare dalle 27 possibili configurazioni, ce ne sono alcune che si somigliano notevolmente. Ad esempio, ben quattro colori sono assimilabili al rosso chiaro; guardando in dettaglio quali sono questi quattro colori, si vede che hanno il livello massimo di rosso e il minimo o intermedio di verde e di blu. Questo significa che, se il colore da riconoscere è il rosso vivo (R=255, V=0, B=0), anche se c'è un errore di uno o entrambi gli altri due segnali, il colore viene riconosciuto ugualmente come rosso (o al più arancione). L'importante è che l'errore sia al più di un livello, non importa su quale degli altri due segnali. Se invece l'errore fosse sul segnale del rosso, e ad esempio fosse confuso con il livello intermedio, allora si riconoscerebbe il rosso scuro invece del rosso chiaro: si resta comunque nella gamma del rosso.

Lo stesso discorso vale per le quattro configurazioni del verde e quelle del blu: in tutte il verde (o il blu) è al livello massimo e gli altri due segnali sono intermedi o nulli. Rispetto al caso precedente del rosso, la situazione è migliore perché non esiste un colore vicino al verde (o al blu) così come l'arancione è vicino al rosso.

Un problema potrebbe sorgere quando due dei tre segnali sono al livello massimo e l'altro è a livello intermedio o nullo: la forte vibrazione di due dei tre attuatori potrebbe causare una minore sensibilità sul terzo. Anche in questo caso le configurazioni scelte risultano adeguate, in quanto esistono due tonalità di azzurro chiaro (R=0 V=255 B=255 e R=128 V=255 B=255), due di viola (R=255 V=0 B=255 e R=255 V=128 B=255) e due di giallo (R=255 V=255 B=0 e R=255 V=255 B=128). In ognuno di questi tre casi è "lecito" confondere il livello basso con il livello intermedio del segnale di minore intensità.

Riassumendo, quando c'è un solo livello alto ci si può disinteressare degli altri due e si può presumere che il colore sia proprio quello corrispondente al solo segnale alto, e quando invece si riconoscono due livelli alti si può indicare il colore

corrispondente a quella coppia di segnali senza grosse possibilità di errore: l'importante è aver individuato esattamente quali sono i livelli alti.

Naturalmente il fatto di avere "annullato" alcune possibilità di errore significa disporre di un numero minore di colori rispetto al massimo teorico di 27. Vediamo quali sono i colori distinguibili:

Rosso	Verde	Blu	Colore risultante
0	0	0	Nero
128	0	0	Rosso scuro
255	0	0	Rosso chiaro
255	0	128	Rosso chiaro
255	128	128	Rosso chiaro
255	128	0	Rosso chiaro (arancione)
0	128	0	Verde scuro
0	255	0	Verde chiaro
0	255	128	Verde chiaro
128	255	0	Verde chiaro
128	255	128	Verde chiaro
0	0	255	Blu chiaro
0	128	255	Blu chiaro
128	0	255	Blu chiaro
128	128	255	Blu chiaro
0	0	128	Blu scuro
128	0	128	Viola scuro

0	255	255	Azzurro
128	255	255	Azzurro
255	0	255	Viola
255	128	255	Viola
255	255	0	Giallo
255	255	128	Giallo
255	255	255	Bianco
0	128	0	Verde militare
128	128	0	Verde militare
0	128	128	Grigio
128	128	128	Grigio

I colori sono dunque 16: nero, rosso scuro, rosso chiaro, arancione, verde scuro, verde chiaro, blu chiaro, blu scuro, viola scuro, azzurro, viola, giallo, bianco, verde militare, grigio.

Purtroppo lo svantaggio più pesante di questo tipo di codifica è la mancanza di alcuni colori molto comuni, quali il marrone e il rosa. Il problema è causato dalla scelta dei tre valori di quantizzazione e può essere risolto “sacrificando” uno dei colori sopra elencati (possibilmente il viola scuro o il blu scuro, che pur avendo codifiche diverse sono molto simili) per rappresentare, ad esempio, il marrone. In questo caso, mediante un opportuno addestramento, nella mente del videoleso il marrone prende il posto del viola scuro.

La funzione che si occupa dell’approssimazione del colore è chiamata *Approx*: essa riceve in ingresso la variabile *colore*, di

tipo *longint*[1], e le variabili *min* e *max*, di tipo *byte*[2], che stabiliscono quali debbano essere gli estremi dell'intervallo intermedio, in quanto variabili durante l'esecuzione del programma; in uscita la funzione restituisce un valore di tipo *longint* come quello ricevuto in ingresso. *Approx* utilizza delle variabili temporanee (*c[1]*, *c[2]* e *c[3]*, di tipo *byte*) per memorizzare l'informazione contenuta in *colore* (relativa sia al rosso che al verde che al blu) in tre diverse variabili. Le procedure predefinite *GetRValue*, *GetGValue* e *GetBValue* servono rispettivamente ad estrarre l'informazione sul rosso, sul verde e sul blu dalla variabile *colore*.

Dopo aver estratto e posizionato in *c[1]*, *c[2]* e *c[3]* le tre componenti dei colori fondamentali, il programma si propone di riconoscere se il colore considerato può essere il marrone, perché, come spiegato poco fa, deve essere trattato separatamente. Si considera marrone ogni colore che abbia tonalità di rosso compresa tra 150 e 255, differenza tra tonalità di rosso e tonalità di verde compresa tra 20 e 60, differenza tra tonalità di rosso e tonalità di blu compresa tra 40 e 100.

Lo stesso discorso si può fare con il rosa o con qualunque altro colore si desideri rappresentare al di fuori dei 27 già illustrati. Naturalmente bisogna fare in modo che il colore che viene cancellato sia "dirottato" su uno a lui simile.

Dopo avere accertato di non essere in un colore che va trattato separatamente, si procede con l'approssimazione:

- se la tonalità di rosso è tra 0 e *min*, allora la stessa è approssimata al valore minimo, cioè 0;
- se la tonalità di rosso è tra *min*+1 e *max*, allora la stessa è approssimata al valore intermedio, cioè 128;
- se la tonalità di rosso è tra *max*+1 e 255, allora la stessa è

[1] Il tipo *longint* è un tipo intero predefinito a 32 bit con segno : questo significa che i valori minimo e massimo sono rispettivamente -2147483648 e +2147483647.
[2] Il tipo *byte* è un tipo intero predefinito a 8 bit senza segno : il suo valore minimo è 0 e il massimo è 255.

approssimata al valore massimo, cioè 255.

Questo procedimento va ripetuto per tutti i tre colori. Al termine dell'approssimazione occorre convertire *c[1]*, *c[2]* e *c[3]* nel tipo di dato longint da mandare in uscita: questo compito è svolto dalla funzione predefinita *RGB*.

Un'altra funzione che riveste una certa importanza nell'assicurare il flusso dei dati verso il circuito elettronico di elaborazione dei segnali è la procedura *OutParallela*, che indirizza alla porta parallela un byte ricavato a partire dai valori già approssimati dei tre colori, qui rappresentati ognuno con un byte (R,V,B).

Ricordando che i tre colori occupano i sei bit meno significativi, in particolare che il rosso occupa i due bit meno significativi, e tenendo conto della presenza e della codifica del marrone, si prepara il byte di uscita secondo le regole precedentemente esposte (figura 4.1).

A questo punto si utilizza l'istruzione pascal *PORT*, che serve ad indirizzare una qualunque porta di comunicazione. La sintassi del comando di output è la seguente:

PORT[*indirizzo*]:=Byte[3],

dove *indirizzo* è l'indirizzo della porta che si sta utilizzando[4] e Byte è il byte da mandare in uscita.

[3] Naturalmente la sintassi del comando di input, utile per leggere la parola presente sugli otto bit di dato in uscita, è Byte:=*PORT*[indirizzo].

[4] Per la porta parallela, in particolare, l'indirizzo normalmente è il numero esadecimale $378. È inoltre possibile leggere i dati forzati dall'esterno sui pin 10, 11, 12, 13 e 15 con il comando Byte:=*PORT*[$379] : la porta parallela è infatti bidirezionale.

4.3 Temporizzazione del segnale di uscita

Il modo più rapido per assegnare il byte all'uscita è quello di eseguire l'istruzione *Port* così come spiegato nel paragrafo precedente. Nonostante questa sembri la soluzione migliore, le prime prove effettuate dimostrano che a volte il non vedente non riesce a "sentire" i tre segnali applicati contemporaneamente.

Il problema risiede nell'imprecisione degli attuatori e può manifestarsi quando un segnale "copre" uno degli altri due. Se la vibrazione è troppo forte possono verificarsi due effetti, entrambi dannosi ai fini del riconoscimento del colore: il segnale più forte può coprire gli altri due, oppure può causare una sensazione di vibrazione attribuibile erroneamente a uno degli altri due. Nel primo caso, ad esempio, il bianco (R=255, V=255, B=255) potrebbe essere confuso con il viola (R=255, V=0, B=255) o con l'azzurro (R=0, V=255, B=255), mentre nel secondo il giallo (R=255, V=255, B=0) o il viola potrebbero essere confusi con il bianco.

Una soluzione alternativa all'assegnamento immediato del byte all'uscita è quella di mandare in output una rapida successione dei colori. Questo significa, ad esempio, introdurre un ritardo tra il segnale del rosso e quello del verde, oppure tra quello del verde e quello del blu. In questo modo il soggetto videoleso potrebbe avvertire non solo la presenza dei diversi segnali, ma soprattutto la variazione di ognuno.

Per fare un esempio, i segnali elettrici di uscita rappresentanti il colore bianco sarebbero quelli del grafico in figura 4.4, mentre quelli riguardanti il viola diventerebbero quelli raffigurati in figura 4.5.

Per evitare di confondere le idee, invece di chiarirle, occorre stabilire una convenzione su quale sia l'ordine secondo il quale i segnali vengono ritardati. Poiché è possibile un'applicazione degli attuatori su un unico dito, è consigliabile applicare per primo il segnale sulla punta del dito, poi quello sulla falange mediale e infine quello sulla falange prossimale, in modo da

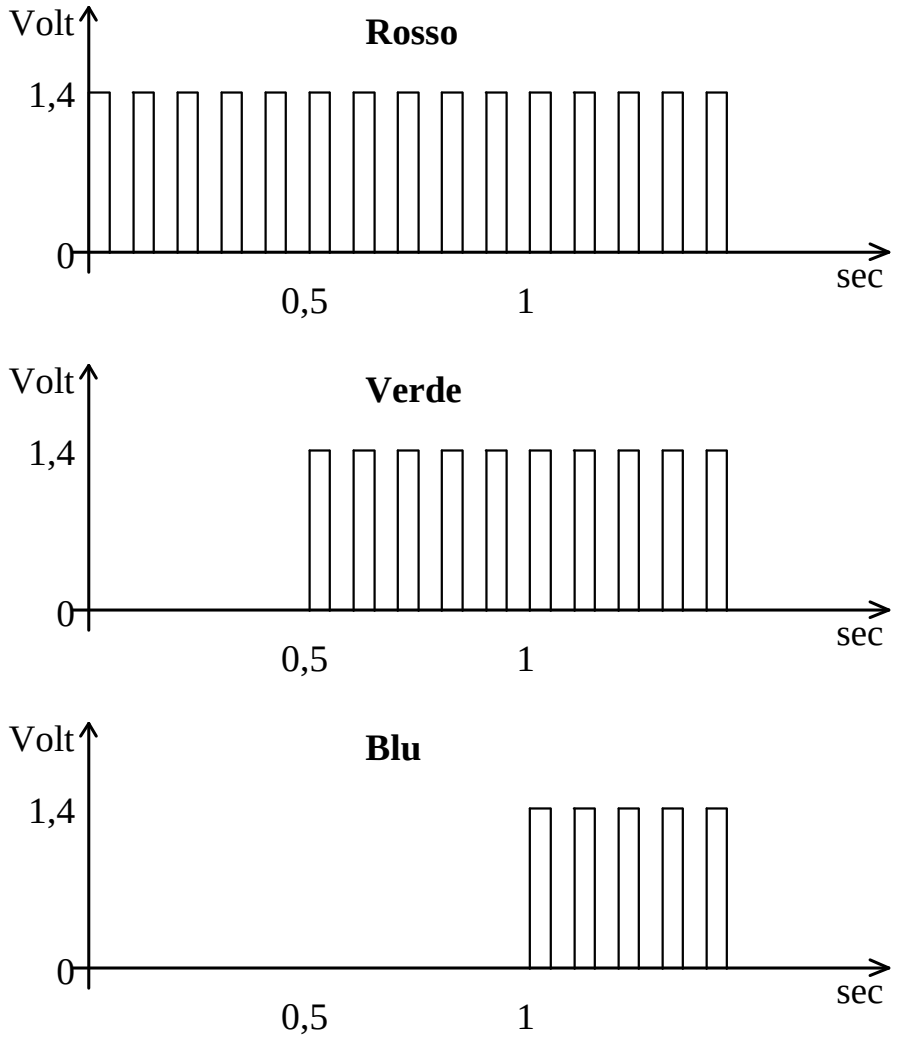


figura 4.4 - Rappresentazione del bianco con introduzione di ritardo tra un colore e l'altro

causare una successione ordinata di sensazioni. Avrebbe effetto analogo l'ordinamento inverso: deve essere evitata, invece, l'applicazione disordinata dei tre segnali.

Supponendo di applicare l'attuatore del rosso alla falange distale, quello del verde alla falange mediale e quello del blu alla falange prossimale, ne scaturisce l'ordine evidenziato nelle

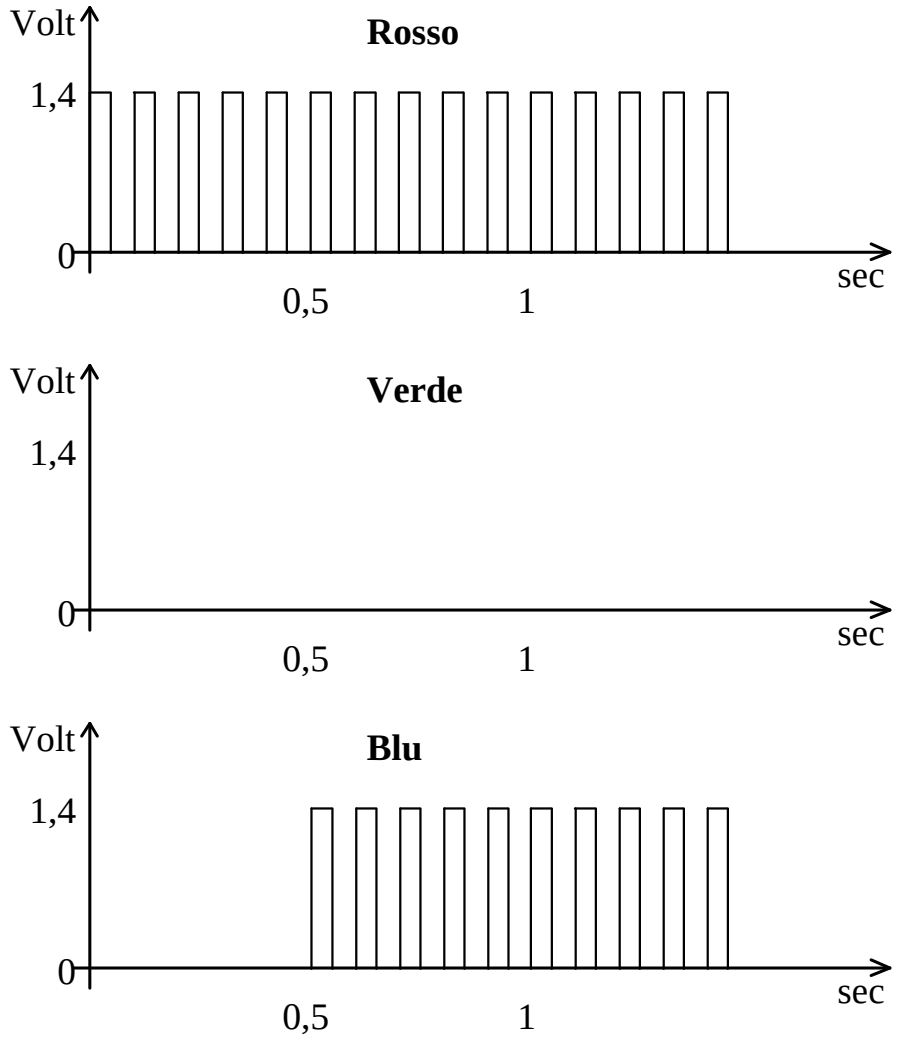


figura 4.5 - Rappresentazione del viola con introduzione di ritardo tra un colore e l'altro

figure: il rosso verrà sempre segnalato immediatamente; il verde, se presente, in ritardo di 500 millisecondi; il blu, se presente insieme agli altri due, in ritardo di un secondo. Naturalmente, in caso di assenza del rosso, sarà il verde ad essere rappresentato senza ritardo, ed il blu, se presente, avrà un ritardo di mezzo secondo.

Dalle prove effettuate si può vedere come questa codifica permetta un riconoscimento molto più sicuro dei colori che con la codifica precedente erano considerati più “difficili” da distinguere. Nonostante il miglioramento a livello del riconoscimento, peraltro fondamentale, l’introduzione del ritardo porta a dover aspettare un periodo di tempo che varia tra i 500 millisecondi e il secondo: anche nel caso che il colore da riconoscere sia un colore non composto, come ad esempio il rosso, occorre aspettare mezzo secondo per essere sicuri che non siano in arrivo altri colori.

Non si può dunque pensare di riconoscere “al volo” i colori puntati dalla freccia del mouse che si muove in tempo reale. Per utilizzare al meglio questa innovazione è necessario poter scegliere tra una rappresentazione dei colori con un ritardo e una rappresentazione in tempo reale con un semplice gesto dell’utilizzatore, come ad esempio un doppio click del mouse o della punta della penna ottica. Il software prevede appunto questa possibilità.

Un ulteriore causa di errore è dato dai livelli intermedi dei segnali: benché sia possibile individuarli, può capitare sia di confonderli con i livelli alti, sia di non sentirli affatto, specialmente se qualcuno degli altri colori è a livello alto. Per questo motivo è necessario fare in modo che siano nettamente distinguibili, non solo per l’intensità intermedia, ma anche per qualche altra caratteristica.

Le caratteristiche dell’ambiente integrato di programmazione permettono in modo rapido e agevole di implementare ritardi: per questo si è pensato si risaltare il livello intermedio mediante un

segnale che ogni 300-400 millisecondi si azzera, per poi tornare al livello desiderato. Per chiarire il concetto, la figura 4.6 rappresenta la forma d'onda di un segnale intermedio: l'andamento è qualitativo, in quanto la frequenza del segnale in uscita è di 75 Hertz, mentre in figura "sembra" di frequenza molto inferiore.

In questo modo, se si è in presenza di un segnale di intensità intermedia, si avverte una pulsazione sull'attuatore del rispettivo colore. La frequenza della pulsazione è regolabile via software, agendo sul valore del tempo che passa (in millisecondi) tra un azzeramento del segnale e il successivo ritorno dello stesso.

Si sarebbero potute realizzare le due modifiche introdotte in questo paragrafo, con le stesse funzionalità, anche dall'hardware. Anche se sarebbe stata più funzionale, nel senso che si sarebbero ottenute temporizzazioni più precise, la realizzazione hardware non avrebbe permesso la facilità di regolazione ottenibile realizzando i ritardi col software. Infatti il vantaggio che si sarebbe ottenuto sarebbe stato quello dell'estrema precisione dei ritardi ottenuti tramite oscillatori, come quello utilizzato sulla scheda hardware per ottenere la frequenza di 75 Hertz. D'altro canto, però, per ottenere un leggero aumento del ritardo standard e per rilevarlo attraverso un display luminoso, sarebbe stato

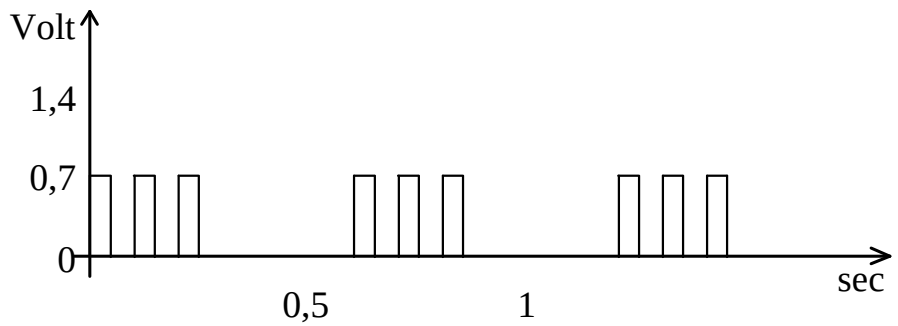


figura 4.6 - Rappresentazione del segnale corrispondente alla tonalità intermedia di uno dei tre colori fondamentali

necessario un set di resistenze o condensatori di precisione, un display e una logica di controllo per lo stesso display. Realizzando il tutto via software si devono fare i conti con i *Timer* di Windows, che segnalano lo scadere del tempo quando il loro valore è azzerato, ma l'azione che devono eseguire viene messa in coda al carico di lavoro che Windows si appresta ad eseguire. In altre parole, può capitare che in un computer particolarmente lento si "perda" qualche scadenza di *Timer*, con la conseguenza di una pulsazione non troppo regolare. Questa, nonostante quello che può sembrare, non è una limitazione troppo stringente, perché la pulsazione è chiaramente avvertibile anche se non è regolare. Il vantaggio dell'implementazione software, cioè la possibilità di regolare la pulsazione (o nel primo caso la possibilità di cambiare i ritardi con cui i colori vengono mandati in uscita) con un semplice *click*, è invece molto apprezzabile, almeno in fase di sviluppo. La realizzazione hardware potrà essere un passo successivo.

4.4 Verifica pratica dell'approssimazione proposta

Resta ora da vedere se l'approssimazione scelta è in grado di riprodurre in maniera sufficientemente fedele immagini del mondo che ci circonda.

È importante innanzitutto vedere se l'immagine resta "comprensibile", nel senso che si capisca di che cosa tratta, e se assomiglia a quella originale, in secondo luogo cercare di individuare se ci sono colori che non solo non sono simili a quelli da approssimare, ma sono molto diversi o addirittura fuorvianti (non solo non aiutano a capire qual è l'oggetto, ma portano a pensarne un altro).

A questo scopo si sono scelte alcune immagini e su ogni pixel di esse, per non perdere in risoluzione, si è operata l'approssimazione da testare con un'apposita procedura chiamata *BitBtnRiduciClick*. Questa procedura è descritta nell'appendice dedicata al codice sorgente pascal di tutti i programmi e le procedure utilizzati; il suo compito è quello di scandire l'immagine orizzontalmente (cioè si occupa di una colonna per volta, procedendo dall'alto verso il basso e da sinistra verso destra), in modo che durante l'approssimazione, che viene mostrata in tempo reale sul video, si possano notare eventuali imprecisioni. Infatti, durante l'esecuzione del programma, si possono modificare gli estremi dell'intervallo di approssimazione intermedio, che normalmente sono 85 e 170, per cercare un'approssimazione migliore. Tutte le prove sono state fatte con i valori di default.

Nelle pagine seguenti viene mostrata una serie di prove con l'immagine approssimata accostata all'immagine originale.

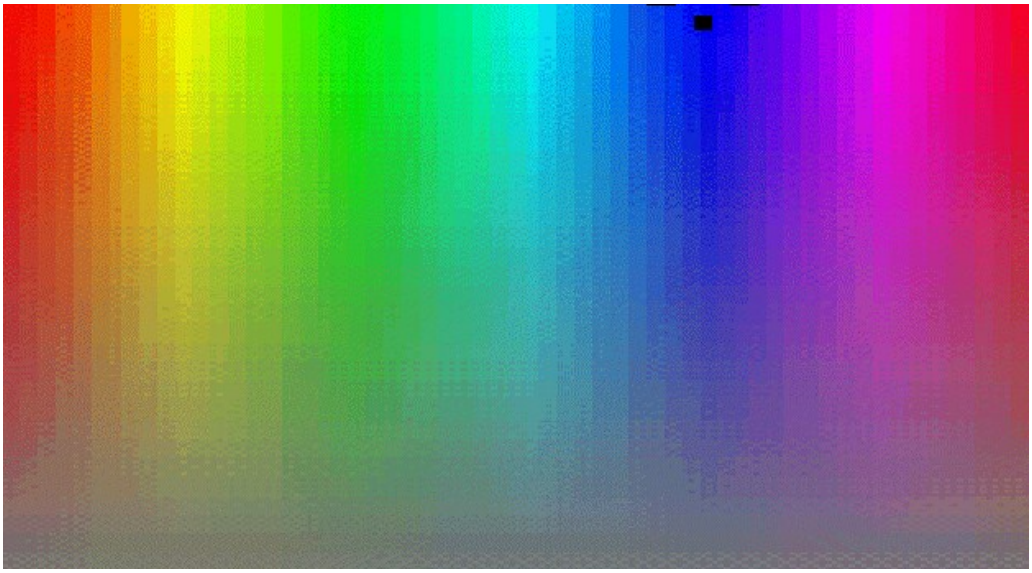


Immagine a 256 colori della mappa dei colori di Windows

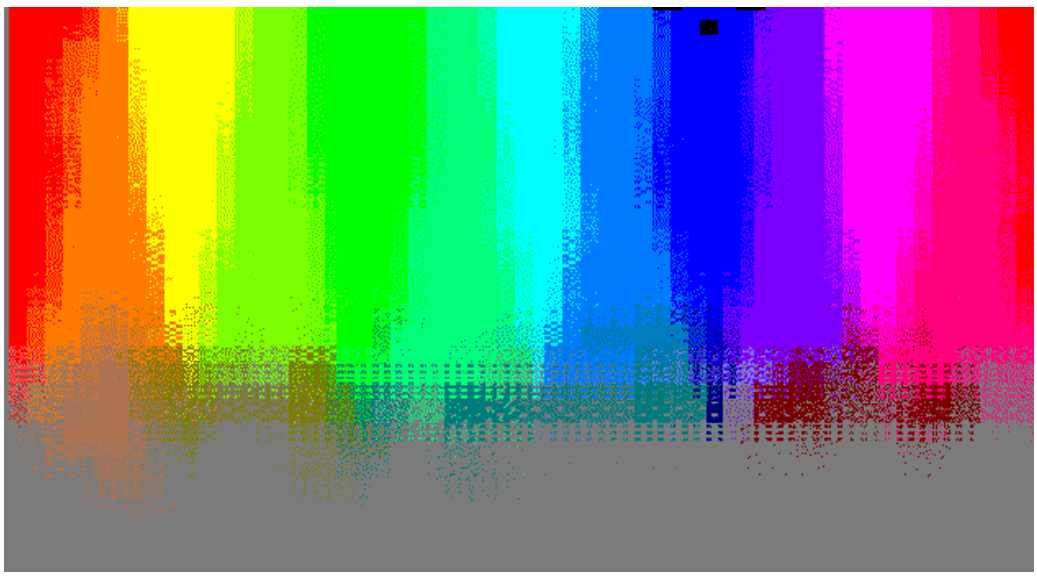


Immagine della mappa dei colori di Windows dopo l'approssimazione

Immagine a 256 colori di un campo di fiori

Immagine del campo di fiori dopo l'approssimazione

Immagine a 256 colori di un fiore

Immagine del fiore dopo l'approssimazione

Immagine a 256 colori del parco di Yosemite

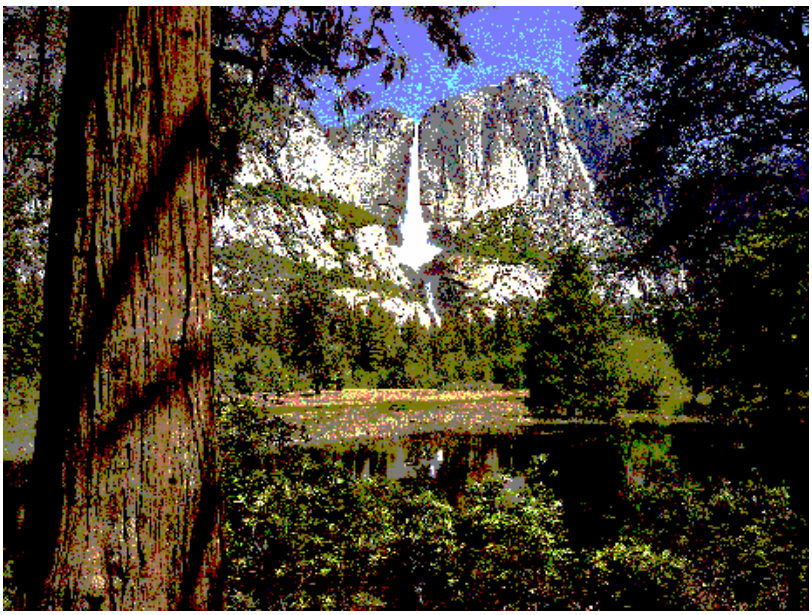


Immagine del parco di Yosemite dopo l'approssimazione

Immagine a 256 colori del Golden Gate (San Francisco)

Immagine del Golden Gate dopo l'approssimazione

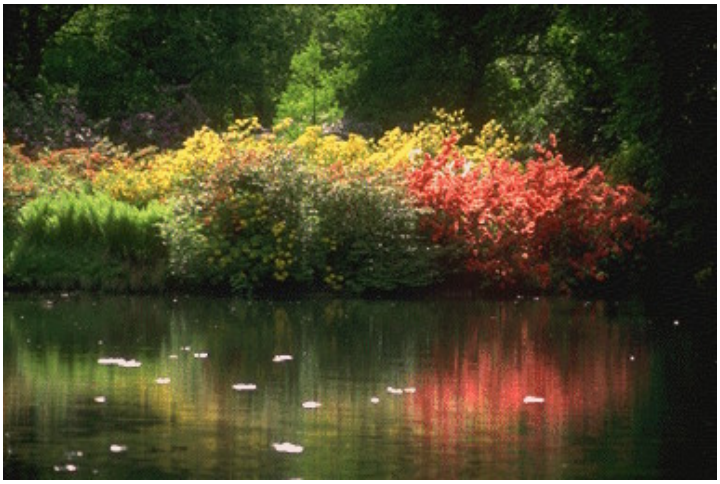


Immagine a 256 colori della riva di un fiume

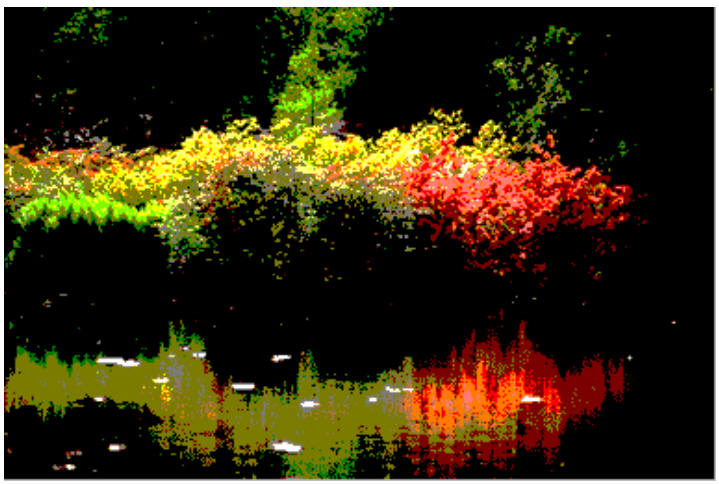


Immagine della riva del fiume dopo l'approssimazione

Immagine a 256 colori della Terra

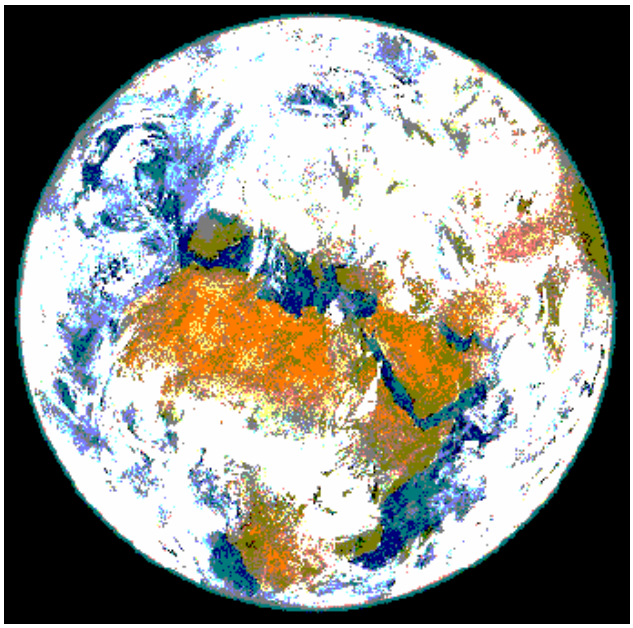


Immagine della Terra dopo l'approssimazione

Immagine a 256 colori di un tramonto

Immagine del tramonto dopo l'approssimazione

Immagine a 256 colori di alcuni tulipani

Immagine dei tulipani dopo l'approssimazione

Immagine a 256 colori di un viale alberato

Immagine del viale alberato dopo l'approssimazione

4.5 Interfaccia grafica

Per agevolare lo svolgimento dell'operazione, è stata sviluppata un'interfaccia grafica dove prende posto l'immagine da approssimare. L'utilizzatore può utilizzare un mouse (o una qualunque periferica che preveda un puntatore sul video) per puntare il colore da approssimare. Il software decodifica la posizione del mouse e avvia la procedura di preparazione dei segnali. Sull'interfaccia grafica prendono posto anche alcune funzioni aggiuntive.

Il significato dei "bottoni" che si vedono nella figura 4.7 e nella figura 4.8 è il seguente:

- *Indirizzo*: qui si sceglie l'indirizzo della porta parallela (il valore di default è 378h);
- *Ritardo*: tempo che passa tra l'applicazione di un colore fondamentale e il successivo;
- *Oscillazione*: tempo durante il quale i livelli di segnale intermedi rimangono alti o bassi;
- *Out*: si può verificare quali sono i livelli di tensione su ciascun pin di output. I bit sono ordinati dal meno significativo (rosso) al più significativo (blu);
- *Riquadro a sinistra*: qui sono indicate le tre gradazioni del colore puntato dal mouse, affiancate dalle tre gradazioni che il colore ha assunto dopo l'approssimazione; sono inoltre presenti le coordinate della posizione del mouse (l'origine è in alto a sinistra);
- *Numero gradazioni*: è il numero degli di intervalli di approssimazione; si è scelto di lavorare con tre intervalli, però è possibile utilizzarne quattro: tuttavia il circuito hardware è stato progettato per tre intervalli;
- *Riduci colori*: opera l'approssimazione di tutta l'immagine

scandendola orizzontalmente;

- *Refresh*: ripristina l'immagine con i colori originali;
- *Stretch*: adatta l'immagine alle dimensioni della finestra;
- *Rileva movimento mouse*: se selezionato, il programma approssima il pixel puntato dal mouse e lo manda in uscita; se non è selezionato, approssima ed aggiorna l'uscita solo quando viene premuto il tasto sinistro;
- *Carica*: avvia la procedura per il caricamento di una nuova immagine, aprendo la finestra di figura 4.8;
- *Auto*: il programma utilizza, per il livello intermedio dell'intervallo di approssimazione, i valori di default; se Auto non è selezionato, i suddetti valori si possono cambiare tramite le due *SpinEdit* sottostanti;
- *Reset*: riporta le due *SpinEdit* ai valori di default.

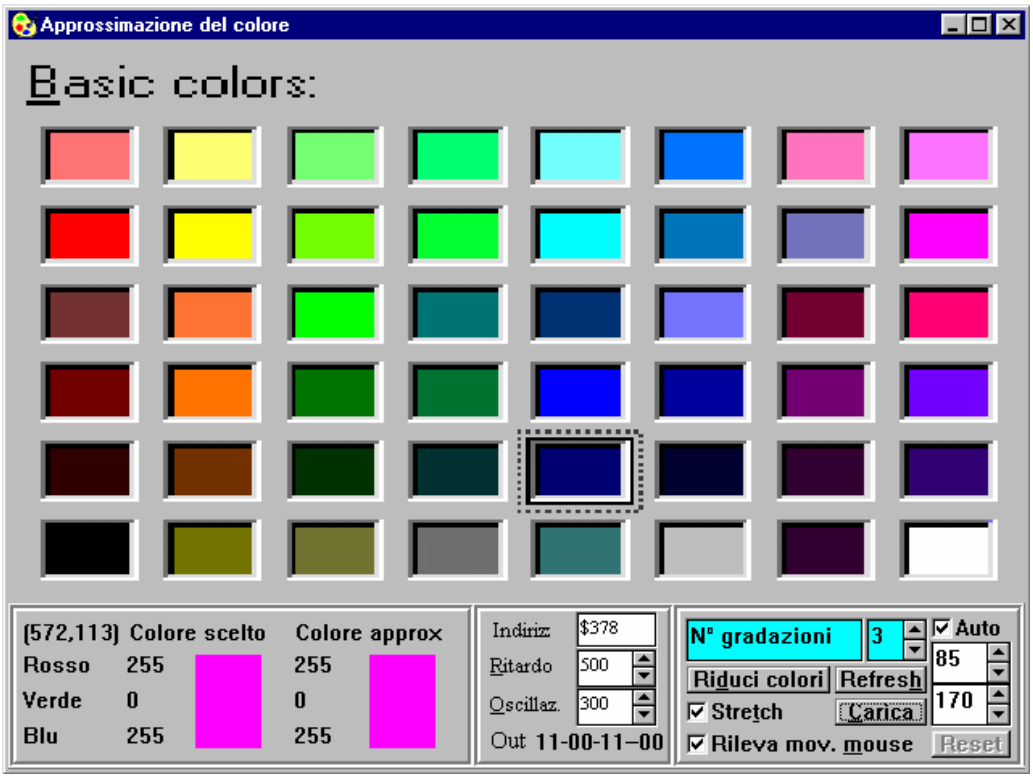


figura 4.7 - Interfaccia grafica di “colori.exe”

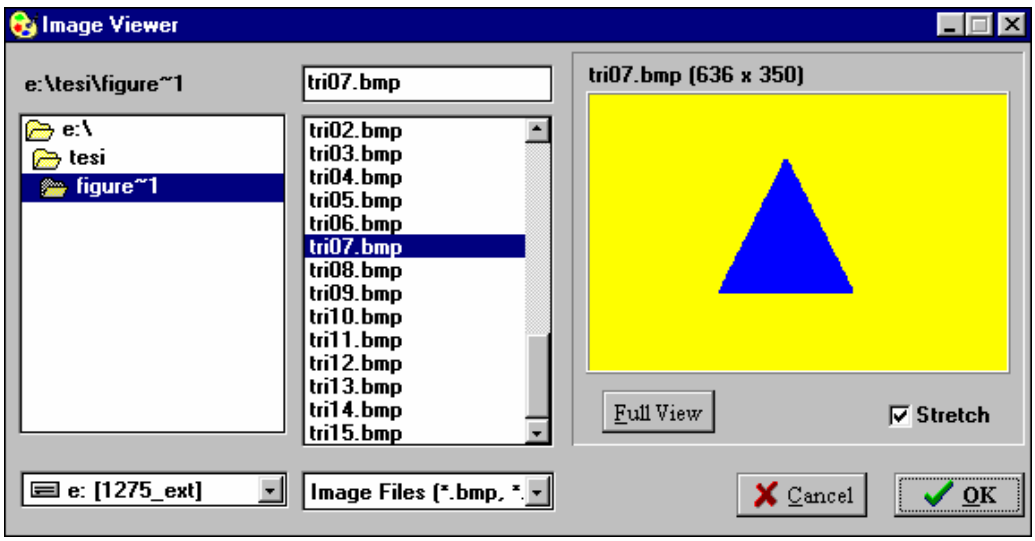


figura 4.8 - Caricamento di un’immagine

Capitolo 5

SPERIMENTAZIONE

La sperimentazione delle codifiche descritte nei capitoli precedenti viene effettuata in due fasi su alcuni soggetti non vedenti ed altri normovedenti. Inoltre, due dei soggetti della prima categoria sono non vedenti dalla nascita, e di conseguenza non hanno la stessa concezione del colore dei soggetti che invece hanno perso la vista in età adulta.

Tutti i soggetti che hanno partecipato ai test di riconoscimento sono stati precedentemente sottoposti ad un addestramento, per insegnargli ad associare i segnali applicati al loro corpo ai colori corrispondenti, così come spiegato nel capitolo 4 dedicato al software.

Per alcuni di loro, come specificato in ogni singola prova, si è utilizzato uno sfasamento temporale dei tre segnali applicati, in modo che si possa distinguere il momento in cui ciascun segnale viene applicato.

In alcuni casi, per ovviare a problemi di inadeguatezza degli attuatori utilizzati, si è preferito applicare ciascun attuatore ad un dito diverso.

Il test di riconoscimento è diviso in due parti:

- la prima riguarda la semplice individuazione di una sequenza di colori;
- la seconda consiste nell'individuare e riconoscere una figura geometrica colorata diversamente dallo sfondo.

I primi test vengono svolti su un numero di colori limitato: i

colori scelti sono quelli la cui codifica approssimata non contiene livelli intermedi. Il riconoscimento si limita, nelle prime prove, ai seguenti colori:

– Bianco (R=255, V=255, B=255);

– Nero (R=0, V=0, B=0);

– Rosso (R=255, V=0, B=0);

– Verde (R=0, V=255, B=0);

– Blu (R=0, V=0, B=255);

– Giallo (R=255, V=255, B=0);

– Viola (R=255, V=0, B=255);

– Azzurro (R=0; V=255, B=255);

Se i soggetti che effettuano il test raggiungono una buona familiarità con i segnali a loro applicati, vengono aggiunti altri colori:

– Rosso scuro (R=128, V=0, B=0);

– Verde scuro (R=0, V=128, B=0);

– Blu scuro (R=0, V=0, B=128);

– Grigio (R=128; V=128, B=128);

– Arancione (R=255; V=128;B=0);

Inoltre uno dei soggetti non vedenti ha accettato di collaborare alla realizzazione di un filmato sulla serie di test a cui si è sottoposto, dall'addestramento al riconoscimento di figure geometriche.

Alma Mater Studiorum
Sæcularia Nona

5.1 Soggetto n°1

Il primo soggetto sottoposto al test è videoleso dalla nascita. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati senza sfasamento tra loro, tutti sul dito indice della mano sinistra.

Il primo test, ripetuto quattro volte, è quello di individuazione di un singolo colore.

Tabella 5.1

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto	Prova n°4 Colore riconosciuto
viola	viola	viola	*bianco*	*bianco*
azzurro	azzurro	azzurro	*verde*	*verde*
blu	*azzurro*	blu	blu	blu
verde	verde	verde	*azzurro*	verde
bianco	*giallo*	*giallo*	*viola*	*viola*
nero	nero	nero	nero	nero
giallo	*rosso*	*rosso*	giallo	giallo
rosso	rosso	rosso	rosso	rosso

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono sottoposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.2

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	bianco	rosso	bianco	rosso
Cerchio	rosso	blu	rosso	blu
Triangolo	azzurro	blu	*verde*	blu
Rettangolo	azzurro	blu	*verde*	blu

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.3

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	giallo	Cerchio	blu	*rosso*
Rettangolo	azzurro	viola	Rettangolo	*verde*	*rosso*

5.2 Soggetto n°2

Il secondo soggetto sottoposto al test è non vedente dall'età adulta. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati senza sfasamento tra loro, tutti sul dito indice della mano sinistra.

Il primo test, ripetuto quattro volte, è quello di individuazione di un singolo colore.

Tabella 5.4

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto	Prova n°4 Colore riconosciuto
azzurro	*giallo*	azzurro	azzurro	*viola*
bianco	*giallo*	*viola*	bianco	bianco
giallo	giallo	giallo	giallo	giallo
nero	nero	nero	nero	nero
rosso	rosso	rosso	rosso	rosso
viola	*blu*	*blu*	*azzurro*	viola
verde	verde	verde	verde	verde
blu	blu	blu	blu	blu

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono sottoposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.5

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	nero	blu	nero
Triangolo	bianco	rosso	*viola*	rosso
Rettangolo	azzurro	blu	azzurro	blu

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.6

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	giallo	Cerchio	blu	*rosso*
Rettangolo	azzurro	viola	Rettangolo	azzurro	viola
Rettangolo	blu	nero	Cerchio	blu	nero
Cerchio	azzurro	giallo	Cerchio	azzurro	giallo

5.3 Soggetto n°3

Il terzo soggetto sottoposto al test è non vedente dalla nascita. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati con uno sfasamento di mezzo secondo tra uno e l'altro, tutti sul dito indice della mano sinistra.

Il primo test, ripetuto quattro volte, è quello di individuazione di un singolo colore. Vengono utilizzati anche i cinque colori aggiuntivi.

Tabella 5.7

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto	Prova n°4 Colore riconosciuto
azzurro	azzurro	azzurro	azzurro	azzurro
viola	viola	viola	viola	viola
blu scuro	blu scuro	blu scuro	blu scuro	blu scuro
bianco	bianco	bianco	bianco	bianco
blu	blu	blu	blu	blu
rosso	rosso	rosso	rosso	rosso
giallo	giallo	giallo	giallo	giallo
rosso scuro	rosso scuro	rosso scuro	rosso scuro	rosso scuro
verde	verde	verde	verde	verde
grigio	grigio	grigio	grigio	grigio
verde scuro	verde scuro	verde scuro	verde scuro	verde scuro
arancione	*giallo*	*giallo*	*rosso*	arancione
nero	nero	nero	nero	nero

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono sottoposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.8

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	nero	blu	nero
Triangolo	bianco	rosso	*giallo*	rosso
Rettangolo	blu	giallo	blu	giallo

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.9

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Cerchio	viola	verde	Cerchio	*azzurro*	verde
Rettangolo	nero	rosso	Rettangolo	nero	rosso
Cerchio	giallo	blu	Rettangolo	giallo	blu

Visti gli ottimi risultati, dovuti allo sfasamento nell'applicazione dei segnali, si effettua una prova analoga senza sfasamento ma con gli attuatori applicati a tre dita diverse (dito indice, medio e anulare della mano sinistra).

Tabella 5.10

Colore da riconoscere	Colore riconosciuto
verde	verde
rosso	rosso
bianco	bianco
grigio	grigio
viola	viola
rosso scuro	rosso scuro
verde	verde
azzurro	azzurro
blu	blu
viola	viola
arancione	arancione
verde	verde
blu	blu
verde	verde
bianco	bianco
grigio	grigio
viola	viola
azzurro	azzurro
verde scuro	verde scuro
blu scuro	blu scuro
azzurro	azzurro
viola	viola

bianco	bianco
rosso scuro	rosso scuro
grigio	grigio
verde scuro	verde scuro
arancione	arancione
azzurro	azzurro
bianco	bianco
arancione	arancione
rosso scuro	rosso scuro

5.4 Soggetto n°4

Il quarto soggetto sottoposto al test è non vedente dall'età di sette anni. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati senza sfasamento tra loro, su tre dita diverse della mano sinistra (indice, medio, anulare).

Il primo test, ripetuto quattro volte, è quello di individuazione di un singolo colore. Vengono utilizzati anche i cinque colori aggiuntivi.

Tabella 5.11

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto	Prova n°4 Colore riconosciuto
azzurro	azzurro	azzurro	azzurro	azzurro
viola	viola	*bianco*	viola	viola
blu scuro	blu scuro	blu scuro	blu scuro	blu scuro
bianco	bianco	*azzurro*	*azzurro*	*viola*
blu	blu	*blu scuro*	blu	blu
rosso	rosso	rosso	rosso	rosso
giallo	giallo	giallo	giallo	giallo
rosso scuro	rosso scuro	rosso scuro	rosso scuro	rosso scuro
verde	verde	verde	verde	verde
grigio	grigio	grigio	grigio	grigio
verde scuro	verde scuro	verde scuro	verde scuro	verde scuro
arancione	*giallo*	*giallo*	*rosso*	arancione
nero	nero	nero	nero	nero

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono sottoposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.12

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	nero	blu	nero
Triangolo	bianco	rosso	*giallo*	rosso
Rettangolo	blu	giallo	blu	giallo

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.13

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Rettangolo	blu	giallo	Rettangolo	*verde*	giallo
Cerchio	viola	azzurro	Cerchio	viola	azzurro
Triangolo	blu	giallo	Triangolo	blu	giallo
Cerchio	rosso	arancione	Cerchio	rosso	arancione

5.5 Soggetto n°5

Il quinto soggetto sottoposto al test è normovedente. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati con sfasamento di mezzo secondo tra uno e l'altro, sul dito indice della mano sinistra.

Il primo test, ripetuto tre volte, è quello di individuazione di un singolo colore. Vengono utilizzati anche i cinque colori aggiuntivi.

Tabella 5.14

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto
azzurro	azzurro	azzurro	azzurro
viola	viola	viola	viola
blu scuro	blu scuro	blu scuro	blu scuro
bianco	bianco	bianco	bianco
blu	blu	blu	blu
rosso	rosso	rosso	rosso
giallo	giallo	giallo	giallo
rosso scuro	rosso scuro	rosso scuro	rosso scuro
verde	verde	verde	verde
grigio	grigio	grigio	grigio
verde scuro	verde scuro	verde scuro	verde scuro
arancione	arancione	arancione	arancione
nero	nero	nero	nero

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono sottoposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.15

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	blu	nero	blu	nero
Triangolo	bianco	rosso	bianco	rosso
Rettangolo	blu	giallo	blu	giallo

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.16

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Cerchio	viola	azzurro	Cerchio	viola	azzurro
Rettangolo	rosso	nero	Rettangolo	rosso	nero
Cerchio	blu	giallo	Cerchio	blu	giallo
Triangolo	rosso	nero	Triangolo	rosso	nero
Triangolo	viola	verde	Triangolo	viola	verde
Rettangolo	bianco	arancione	Rettangolo	bianco	arancione
Cerchio	rosso	arancione	Cerchio	rosso	arancione

5.6 Soggetto n°6

Il sesto soggetto sottoposto al test è normovedente. Questi sono i risultati dei test effettuati dopo un addestramento della durata di qualche minuto.

I segnali sono stati applicati con sfasamento di mezzo secondo tra uno e l'altro, sul dito indice della mano sinistra.

Il primo test, ripetuto tre volte, è quello di individuazione di un singolo colore. Vengono utilizzati anche i cinque colori aggiuntivi.

Tabella 5.17

Colore da riconoscere	Prova n°1 Colore riconosciuto	Prova n°2 Colore riconosciuto	Prova n°3 Colore riconosciuto
azzurro	azzurro	*verde*	azzurro
viola	viola	viola	*bianco*
blu scuro	blu scuro	blu scuro	blu scuro
bianco	*arancione*	bianco	bianco
blu	*azzurro*	blu	blu
rosso	rosso	rosso	rosso
giallo	*viola*	giallo	giallo
rosso scuro	rosso scuro	rosso scuro	rosso scuro
verde	verde	verde	verde
grigio	*arancione*	grigio	*rosso scuro*
verde scuro	verde scuro	verde scuro	verde scuro
arancione	*grigio*	*bianco*	*rosso*
nero	nero	nero	nero

5.7 Soggetto n°7

Il settimo soggetto sottoposto al test è non vedente dall'età adulta. È stata realizzata una videocassetta che contiene tutte le immagini riprese durante questo test.

I segnali sono stati applicati con sfasamento di mezzo secondo tra uno e l'altro, su tre dita diverse della mano destra (indice, medio, anulare).

Il primo test, ripetuto quattro volte, è quello di individuazione di un singolo colore. Vengono utilizzati 12 colori.

Tabella 5.18

Colore da riconoscere	Colore riconosciuto
rosso	rosso
verde	verde
blu	blu
giallo	giallo
azzurro	azzurro
viola	viola
bianco	bianco
verde	verde
rosso	rosso
viola	viola
azzurro	*verde*

azzurro	azzurro
giallo	giallo
azzurro	azzurro
bianco	bianco
blu	blu
viola	viola
rosso	rosso
rosso	rosso
verde	verde
blu	blu
giallo	giallo
azzurro	azzurro
viola	viola
bianco	bianco
rosso scuro	rosso scuro
verde scuro	verde scuro
blu scuro	blu scuro
grigio	*arancione*
rosso	rosso
verde	verde
blu	blu
giallo	giallo
azzurro	azzurro
viola	viola
bianco	bianco

rosso scuro	rosso scuro
verde scuro	verde scuro
blu scuro	blu scuro
grigio	grigio
arancione	arancione
blu	blu
viola	viola
bianco	bianco
rosso	rosso
azzurro	azzurro
grigio	grigio
arancione	arancione
rosso	rosso
blu scuro	blu scuro
azzurro	*verde*
rosso scuro	rosso scuro
grigio	grigio
azzurro	azzurro
viola	viola
verde	verde
blu scuro	blu scuro
arancione	arancione
azzurro	azzurro
bianco	*grigio*
giallo	giallo

blu	*blu scuro*
grigio	*arancione*
grigio	grigio
giallo	giallo
azzurro	azzurro
verde scuro	verde scuro
blu	*blu scuro*
verde	verde
rosso scuro	rosso scuro
viola	viola
blu scuro	blu scuro
bianco	bianco

Il test viene ripetuto applicando i segnali senza sfasamento sulle tre dita.

Tabella 5.19

Colore da riconoscere	Colore riconosciuto
rosso	rosso
verde	verde
blu	blu
giallo	giallo
azzurro	azzurro

viola	viola
bianco	bianco
rosso scuro	rosso scuro
verde scuro	verde scuro
blu	blu
azzurro	azzurro
viola	*rosso*
viola	*arancione*
blu scuro	blu scuro
rosso scuro	rosso scuro
giallo	giallo
verde scuro	verde scuro
azzurro	azzurro
grigio	*arancione*
grigio	-
viola	*rosso*
viola	*giallo*
viola	viola
verde	verde
azzurro	azzurro
viola	viola
viola	*giallo*
viola	viola
bianco	bianco
blu scuro	blu scuro

rosso scuro	rosso scuro
verde	verde
viola	viola
azzurro	azzurro
grigio	*verde scuro*
rosso	rosso
azzurro	azzurro
bianco	bianco
arancione	arancione
rosso	rosso
blu scuro	blu scuro
grigio	*verde scuro*
viola	viola
arancione	*grigio*

Dopo aver lasciato al soggetto qualche minuto di tempo per studiare la forma delle figure geometriche che gli vengono proposte (cerchio, rettangolo, triangolo), gli si chiede di individuare il colore della figura e dello sfondo.

Tabella 5.20

Figura geometrica	Colore figura	Colore sfondo	Colore figura individuato	Colore sfondo individuato
Cerchio	rosso	nero	rosso	nero
Rettangolo	blu	nero	blu	nero
Triangolo	verde	nero	verde	nero

Nella seguente prova il soggetto deve individuare, oltre al colore dello sfondo e della figura, anche la forma geometrica.

Tabella 5.21

Figura geometrica	Colore figura	Colore sfondo	Figura geometrica individuata	Colore figura individuato	Colore sfondo individuato
Rettangolo	blu	giallo	Rettangolo	blu	giallo
Triangolo	rosso	verde	Triangolo	rosso	verde

APPENDICE

Codice sorgente

Il codice sorgente è raccolto in 7 file :

colori.dpr	è il programma principale, quello che genera tutte le finestre;
colore.pas	sorgenti della unit colori (finestra principale);
imagewin.pas	sorgenti della unit imagewin (finestra per il caricamento delle immagini con anteprima);
viewwin.pas	sorgenti della unit view (finestra per la vista in scala 1:1 dell'immagine da caricare);
colore.dfm	dichiarazione di tutti gli oggetti della unit colori;
imagewin.dfw	dichiarazione di tutti gli oggetti della unit imagewin;
viewwin.dfm	dichiarazione di tutti gli oggetti della unit viewwin.

FILE COLORI.DPR

```
PROGRAM COLORI;

uses
  Forms,
  Colore in 'COLORE.PAS' {FormImmagine},
  Imagewin in 'IMAGEWIN.PAS' {ImageForm},
  Viewwin in 'VIEWWIN.PAS' {ViewForm};

{$R *.RES}

begin
  Application.CreateForm(TFormImmagine, FormImmagine);
  Application.CreateForm(TImageForm, ImageForm);
  Application.CreateForm(TViewForm, ViewForm);
  Application.Run;
end.
```

Alma Mater Studiorum
Secularia Nona

FILE COLORE.PAS

```
UNIT COLORE;

{E' LA UNIT PRINCIPALE, QUELLA CHE CONTIENE LA FINESTRA CON}
{L'IMMAGINE DA APPROSSIMARE}

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons, Spin,
  ImageWin;

type
  TFormImmagine = class(TForm)
    ImageCentral: TImage;
    PanelComandi: TPanel;
    PanelParallela: TPanel;
    LabelDataOut: TLabel;
    LabelOut: TLabel;
    LabelIndirizzo: TLabel;
    EditIndirizzo: TEdit;
    LabelPortaParallela: TLabel;
    PanelMostraColori: TPanel;
    LabelRosso: TLabel;
    LabelVerde: TLabel;
    LabelBlu: TLabel;
    LabelValoreVerde: TLabel;
    LabelValoreRosso: TLabel;
    LabelValoreBlu: TLabel;
    LabelPosizione: TLabel;
    LabelColoreScelto: TLabel;
    LabelColoreApprox: TLabel;
    LabelValoreApproxRosso: TLabel;
    LabelValoreApproxVerde: TLabel;
    LabelValoreApproxBlu: TLabel;
    ImageColoreScelto: TImage;
    ImageColoreApprox: TImage;
    PanelGradazioni: TPanel;
    BitBtnRiduci: TBitBtn;
    MemoGradazioni: TMemo;
    SpinEditGradazioni: TSpinEdit;
    BitBtnRefresh: TBitBtn;
    BitBtnCarica: TBitBtn;
```

```
    CheckBoxMouse: TCheckBox;
    CheckBoxAutomatic: TCheckBox;
    SpinEditApproxMin: TSpinEdit;
    SpinEditApproxMax: TSpinEdit;
    ButtonReset: TButton;
    CheckBoxStretch: TCheckBox;
    Timer1: TTimer;
    Timer2: TTimer;
    SpinEditRitardo: TSpinEdit;
    LabelRitardo: TLabel;
    PanelDestro: TPanel;
    Timer3: TTimer;
    LabelOscillazione: TLabel;
    SpinEditOscillazione: TSpinEdit;
    Procedure TrovaDimensioniImmagine(Var L,H:Integer);
    procedure BitBtnRiduciClick(Sender: TObject);
    procedure BitBtnCaricaClick(Sender: TObject);
    procedure SpinEditGradazioniChange(Sender: TObject);
    procedure ImageCentralMouseMove(Sender: TObject; Shift:
              TShiftState; X,Y: Integer);
    procedure ImageCentralMouseDown(Sender: TObject; Button:
              TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure CheckBoxAutomaticClick(Sender: TObject);
    procedure ButtonResetClick(Sender: TObject);
    procedure BitBtnRefreshClick(Sender: TObject);
    procedure CheckBoxStretchClick(Sender: TObject);
    procedure TimerTimer(Sender: TObject);
    procedure ImageCentralDblClick(Sender: TObject);
    procedure Timer3Timer(Sender: TObject);
    procedure SpinEditOscillazioneChange(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormImmagine: TFormImmagine;

implementation

{$R *.DFM}

{PROCEDURA DI AZZERAMENTO DEI VALORI DEL TIMER}
Procedure AzzeraTimer(var Timer:TTimer);
begin
```

```
 Timer.Interval:=0;
 Timer.Tag:=0;
 Timer.Enabled:=False;
end;

{PROCEDURA DI CARICAMENTO DEI VALORI DEL TIMER}
Procedure CaricaTimer(var Timer:TTimer;Msec,Dato:Longint);
begin
 Timer.Interval:=Msec;
 Timer.Tag:=Dato;
 Timer.Enabled:=True;
end;

{PROCEDURA DI RICERCA DELLE DIMENSIONI DELL'IMMAGINE}
Procedure TFormImmagine.TrovaDimensioniImmagine(Var
L,H:Integer);
begin
     With ImageCentral do
     begin
          L:=Width;
          H:=Height;
     end;
end;

{QUESTA FUNZIONI RESTITUISCE IL VALORE DEL COLORE  APPROSSIMATO}
{A PARTIRE DAL COLORE ORIGINALE, SAPENDO QUALI SONO GLI ESTREMI}
{DELL'INTERVALLO INTERMEDIO}
function Approx(Colore:Longint;mode:Byte;min,max:byte):Longint;
var i:Byte; {CONTATORE}
    c:Array[1..3] of Integer; {VETTORE CONTENENTE I TRE COLORI}
    marrone:boolean;
begin
     c[1]:=GetRValue(Colore);   {ROSSO}
     c[2]:=GetGValue(Colore);   {VERDE}
     c[3]:=GetBValue(Colore);   {BLU}
     {SE ROSSO E' MAGGIORE DI 150}
     {SE ROSSO-VERDE E' MAGGIORE DI 20 E MINORE DI 60}
     {SE ROSSO-BLU E' MAGGIORE DI 40 E MINORE DI 100}
     {ALLORA IL COLORE IN QUESTIONE E' MARRONE}
     if (c[1]>=150)
        and ((c[1]-c[2])<=60) and ((c[1]-c[2])>=20)
        and ((c[1]-c[3])<=100) and ((c[1]-c[3])>=40)
            then marrone:=true
            else marrone:=false;
     if not marrone
      then
```

```
begin
     For i:=1 to 3 do
         if FormImmagine.CheckBoxAutomatic.Enabled
         and not FormImmagine.CheckBoxAutomatic.Checked
  {SE NON SI VOGLIONO USARE GLI INTERVALLI DI DEFAULT}
            then
             begin
               if c[i] in [0..min] then c[i]:=0;
               if c[i] in [1+min..max] then c[i]:=128;
               if c[i] in [1+max..255] then c[i]:=255;
             end
        {DA NOTARE CHE IN QUESTO CASO IL PROGRAMMA GIRA}
        {MOLTO PIU' LENTAMENTE :INFATTI L'ISTRUZIONE IF}
        {E' MOLTO PIU' LENTA DELL'ISTRUZIONE  CASE, CHE}
        {PERO' NON SI PUO' USARE PERCHE' NON ACCETTA LE}
        {VARIABILI MIN E MAX TRA I SUOI  PARAMETRI,  MA}
        {VUOLE SOLO COSTANTI}
             else
  {SE INVECE SI USANO GLI INTERVALLI DI DEFAULT}
             begin
        {MODE E' IL NUMERO DEGLI INTERVALLI DI QUANTIZ-}
        {ZIONE: E' IL VALORE  ESPRESSO  NELLO  SPINEDIT}
        {"NUMERO GRADAZIONI"}
                  case mode of
                    2: case c[i] of
                         000..127 : c[i]:=0;
                         128..255 : c[i]:=255;
                       end;
                    3: case c[i] of
                         000..085 : c[i]:=0;
                         086..170 : c[i]:=128;
                         171..255 : c[i]:=255;
                       end;
                    4: case c[i] of
                         000..042 : c[i]:=0;
                         043..127 : c[i]:=85;
                         128..215 : c[i]:=170;
                         216..255 : c[i]:=255;
                       end;
                  end;
             end;
{SE RICONOSCO IL VIOLA SCURO, LO TRASFORMO IN BLU SCURO,}
{PERCHE' LA CONFIGURAZIONE DEL VIOLA ORA E' DEL MARRONE.}
        if (c[1]=128) and (c[2]=0) and (c[3]=128)
           then c[1]:=0;
    end
```

```
      else begin
         {SE RICONOSCO IL MARRONE, USO QUESTA QUANTIZZAZIONE.}
               c[1]:=176;
               c[2]:=120;
               c[3]:=88;
           end;
         {RIUNISCO I TRE BYTE DEI COLORI IN UN UNICA VARIABILE}
         {E LA ASSEGNO ALL'USCITA}
     Approx:=RGB(c[1],c[2],c[3]);
end;

{QUESTA PROCEDURA LEGGE IL BYTE PRESENTE SULLE USCITE}
Function CercaByteOut:Longint;
var B,ByteOut:Longint;
begin
     With FormImmagine do
       begin
            B:=Timer3.Tag;
            ByteOut:=Timer2.Tag;
            if ByteOut=0 then ByteOut:=Timer1.Tag;
            if ByteOut=0
               then ByteOut:=Port[StrToInt(EditIndirizzo.Text)];
            if B>ByteOut then ByteOut:=B;
            CercaByteOut:=ByteOut;
       end;
end;

{LA PROCEDURA ASSEGNAPARALLELA , OGNI VOLTA CHE  VIENE}
{RICHIAMATA, ASSEGNA AI TIMER I VALORI DEGLI EVENTUALI}
{RITARDI E MANDA IN USCITA IL BYTE CHE RAPPRESENTA  IL}
{COLORE RICEVUTO IN INGRESSO}
Procedure AssegnaParallela(p1,p2,p3,p4,p5,p6:Byte);
var Rosso,Verde,Blu:Boolean;
    Ritardo:Longint;
begin
     Ritardo:=FormImmagine.SpinEditRitardo.Value;
     If Ritardo<>0 then With FormImmagine do
     begin
          if p1+p2<>0 then Rosso:=True else Rosso:=False;
          if p3+p4<>0 then Verde:=True else Verde:=False;
          if p5+p6<>0 then Blu:=True else Blu:=False;
          AzzeraTimer(Timer1);
          AzzeraTimer(Timer2);
          AzzeraTimer(Timer3);
          if not (Rosso or Verde or Blu)
             then Port[StrToInt(EditIndirizzo.Text)]:=0;
```

```
          if (Rosso and (not Verde) and (not Blu))
             or ((not Rosso) and Verde and (not Blu))
             or ((not Rosso) and (not Verde) and Blu)
                then Port[StrToInt(EditIndirizzo.Text)]:=
                     p1+p2*2+p3*4+p4*8+p5*16+p6*32;
          if Rosso and Verde and (not Blu)
             then
               begin
                Port[StrToInt(EditIndirizzo.Text)]:=p1+p2*2;
                CaricaTimer(Timer1,Ritardo,p1+p2*2+p3*4+p4*8);
               end;
          if Rosso and (not Verde) and Blu
             then
               begin
                Port[StrToInt(EditIndirizzo.Text)]:=p1+p2*2;
                CaricaTimer(Timer1,Ritardo,p1+p2*2+p5*16+p6*32);
               end;
          if (not Rosso) and Verde and Blu
             then
               begin
                Port[StrToInt(EditIndirizzo.Text)]:=p3*4+p4*8;
                CaricaTimer(Timer1,Ritardo,
                            p3*4+p4*8+p5*16+p6*32);
               end;
          if (Rosso and Verde and Blu)
             then
               begin
                Port[StrToInt(EditIndirizzo.Text)]:=p1+p2*2;
                CaricaTimer(Timer1,Ritardo,p1+p2*2+p3*4+p4*8);
                CaricaTimer(Timer2,2*Ritardo,
                            p1+p2*2+p3*4+p4*8+p5*16+p6*32);
               end;
      end
      else Port[StrToInt(FormImmagine.EditIndirizzo.Text)]:=
                p1+p2*2+p3*4+p4*8+p5*16+p6*32;
      With FormImmagine do
        if (p1+p2=1) or (p3+p4=1) or (p5+p6=1)
          then begin
                 if p1+p2=1 then begin P1:=0; P2:=0 end;
                 if p3+p4=1 then begin P3:=0; P4:=0 end;
                 if p5+p6=1 then begin P5:=0; P6:=0 end;
                 CaricaTimer(Timer3,SpinEditOscillazione.Value,
                             p1+p2*2+p3*4+p4*8+p5*16+p6*32)
               end
          else AzzeraTimer(Timer3);
end;
```

```
{QUESTA FUNZIONE LEGGE L'USCITA DELLA PARALLELA E FORNISCE IN}
{USCITA UNA STRINGA CON IL SUDDETTO VALORE}
Function InParallela:string;
var Input:Byte;
var p1,p2,p3,p4,p5,p6,p7,p8:Byte;
begin
     {LEGGO DALLA PARALLELA IL BYTE PRESENTE SULLE USCITE}
     {PER LEGGERE LE USCITE UTILIZZO LO STESSO  INDIRIZZO}
     {CHE HO USATO PER MANDARE IL BYTE IN USCITA.}
     {ASSEGNO IL BYTE ALLA VARIABILE "INPUT"}
     Input:=Port[StrToInt(FormImmagine.EditIndirizzo.Text)];
     {SE INPUT E' PARI SIGNIFICA CHE IL BIT MENO SIGNIFICATIVO}
     {E' 1}
     if odd(input) then p1:=1 else p1:=0;
     {SE INPUT SHIFTATO A DESTRA DI UNA POSIZIONE E' PARI,}
     {SIGNIFICA CHE IL SECONDO BIT MENO SIGNIFICATIVO E' 1}
     if odd(input shr 1) then p2:=1 else p2:=0;
     {SE INPUT SHIFTATO A DESTRA DI DUE POSIZIONE E' PARI,}
     {SIGNIFICA CHE IL TERZO BIT MENO SIGNIFICATIVO E' 1}
     if odd(input shr 2) then p3:=1 else p3:=0;
     {...E COSI' VIA...}
     if odd(input shr 3) then p4:=1 else p4:=0;
     if odd(input shr 4) then p5:=1 else p5:=0;
     if odd(input shr 5) then p6:=1 else p6:=0;
     if odd(input shr 6) then p7:=1 else p7:=0;
     if odd(input shr 7) then p8:=1 else p8:=0;
     {TRASFORMO I SEI BIT (PIU' DUE CHE NON MI INTERESSANO)}
     {IN UNA STRINGA CHE MANDO IN USCITA}
     Inparallela:=IntToStr(p1)+IntToStr(p2)+'-'+
                  IntToStr(p3)+IntToStr(p4)+'-'+
                  IntToStr(p5)+IntToStr(p6)+'--'+
                  IntToStr(p7)+IntToStr(p8);
end;

{QUESTA PROCEDURA, PARTENDO DAI TRE COLORI IN INGRESSO,}
{PREPARA E MANDA  ALLA  PORTA  PARALLELA  IL  BYTE  CON}
{L'INFORMAZIONE SUL COLORE}
Procedure OutParallela(R,V,B,Mode:Byte);
var p1,p2,p3,p4,p5,p6:Byte;
begin
     {SE ROSSO=255 ALLORA SETTO A 1 ENTRAMBI I SUOI BIT.}
     {FACCIO LA STESSA COSA CON IL VERDE E IL BLU.}
     if R=255 then begin p1:=1; p2:=1 end;
     if V=255 then begin p3:=1; p4:=1 end;
     if B=255 then begin p5:=1; p6:=1 end;
```

```
{SE ROSSO=0 ALLORA SETTO A 0 ENTRAMBI I SUOI BIT.}
{FACCIO LA STESSA COSA CON IL VERDE E IL BLU.}
if R=000 then begin p1:=0; p2:=0 end;
if V=000 then begin p3:=0; p4:=0 end;
if B=000 then begin p5:=0; p6:=0 end;
{PER ASSEGNARE I VALORI DEI BIT QUANDO SIAMO NELL' INTER-}
{VALLO DI QUANTIZZAZIONE INTERMEDIO, DEVO FARE UNA SCELTA}
{A SECONDA DI QUANTI SONO GLI INTERVALLI INTERMEDI}
case mode of
  3: begin
     {SE SONO STATI SCELTI TRE INTERVALLI,ASSEGNO 0 AL BIT}
     {MENO SIGNIFICATIVO E 1 A QUELLO PIU' SIGNIFICATIVO}
         if R=128 then begin p1:=0; p2:=1 end;
         if V=128 then begin p3:=0; p4:=1 end;
         if B=128 then begin p5:=0; p6:=1 end;
     end;
  4: begin
     {SE SONO STATI SCELTI QUATTRO INTERVALLI, ASSEGNO 0}
     {AL BIT  MENO  SIGNIFICATIVO  E  1  A  QUELLO  PIU'}
     {SIGNIFICATIVO SE IL VALORE DEL ROSSO E' 85, MENTRE}
     {FACCIO IL CONTRARIO SE IL VALORE DEL ROSSO E' 170}
         if R=085 then begin p1:=0; p2:=1 end;
         if V=085 then begin p3:=0; p4:=1 end;
         if B=085 then begin p5:=0; p6:=1 end;
         if R=170 then begin p1:=1; p2:=0 end;
         if V=170 then begin p3:=1; p4:=0 end;
         if B=170 then begin p5:=1; p6:=0 end;
     end;
end;
{SE INVECE RICONOSCO IL MARRONE, GLI ASSEGNO LA  CONFI-}
{GURAZIONE DEL VIOLA SCURO, CHE HA ROSSO=128, VERDE=0 E}
{BLU=128}
if (R=176) and (V=120) and (B=88)
   then begin
             p1:=0;
             p2:=1;
             p3:=0;
             p4:=0;
             p5:=0;
             p6:=1;
        end;
{MOLTIPLICO, INFINE, OGNI BIT PER IL SUO PESO}
{E SE E' DIVERSO  DAL  BYTE  CORRENTE  ALLORA}
{ASSEGNO LA SOMMA ALL' ISTRUZIONE  PORT,  CHE}
{INDIRIZZA TUTTO ALLA PORTA PARALLELA.}
If CercaByteOut<>(p1+p2*2+p3*4+p4*8+p5*16+p6*32)
```

```
        then AssegnaParallela(p1,p2,p3,p4,p5,p6);
end;

{QUESTA FUNZIONE RESTITUISCE IL COLORE DI UN DETERMINATO PIXEL}
Function EstraiColore(DC:Hdc;X,Y:Integer):Longint;
begin
     {GETPIXEL FORNISCE IN USCITA IL COLORE (FORMATO RGB) DEL}
     {PIXEL DI COORDINATE X E Y RIFERITE AL DEVICE CONTEXT DC}
     EstraiColore:=GetPixel(DC,X,Y);
end;

{QUESTA PROCEDURA SCANDISCE INTERA L'IMMAGINE E APPROSSIMA AD}
{UNO AD UNO TUTTI I PIXEL CONTENUTI IN ESSA}
procedure TFormImmagine.BitBtnRiduciClick(Sender: TObject);
var X,Y,L,H:integer;
    DC:Hdc;
    Colore,Colore_Approx:Longint;
    Mode,Min,Max:Byte;
begin
     {DC E' IL DEVICE CONTEXT ATTIVO, CIOE' LA FINESTRA ATTIVA}
     DC:=GetDC(GetActiveWindow);
     {MODE E' IL NUMERO DI INTERVALLI DI QUANTIZZAZIONE}
     Mode:=SpinEditgradazioni.Value;
     {MIN E' L'ESTREMO INFERIORE DELL'INTERVALLO INTERMEDIO}
     Min:=SpinEditApproxMin.Value;
     {MAX E' L'ESTREMO SUPERIORE DELL'INTERVALLO INTERMEDIO}
     Max:=SpinEditApproxMax.Value;
     {L ED H SONO RISPETTIVAMENTE LARGHEZZA E ALTEZZA}
     {DELL'IMMAGINE}
     TrovaDimensioniImmagine(L,H);
     {X E Y SCANDISCONO TUTTA L'IMMAGINE, E PER OGNI PIXEL IL}
     {COLORE VIENE LETTO, APPROSSIMATO  E  REINDIRIZZATO  SUL}
     {PIXEL STESSO}
     for X:=0 to L-1 do
      for Y:=0 to H-1 do
       begin
            Colore:=EstraiColore(DC,X,Y);
            Colore_Approx:=Approx(Colore,Mode,Min,Max);
            SetPixel(DC,X,Y,Colore_Approx);
       end;
     {RILASCIO LE LOCAZIONI DI MEMORIA OCCUPATE IN SEDE DI}
     {DEFINIZIONE DELLA FINESTRA ATTIVA}
     ReleaseDC(GetActiveWindow,DC);
end;

{QUESTA PROCEDURA, ATTIVATA QUANDO VIENE PREMUTO IL  TASTO}
```

```
{CARICA, RENDE ATTIVA LA FORM DEL CARICAMENTO DI UNA NUOVA}
{IMMAGINE}
Procedure TFormImmagine.BitBtnCaricaClick(Sender: TObject);
begin
     ImageForm.Show;
end;

{QUESTA PROCEDURA, ATTIVATA QUANDO VARIA  IL  CONTENUTO  DELLO}
{SPINEDIT CHE RIGUARDA LA SCELTA DEL NUMERO DI GRADAZIONI  DEL}
{COLORE (INTERVALLI DI QUANTIZZAZIONE), FA IN MODO CHE, QUANDO}
{IL VALORE SCELTO E' DIVERSO DA TRE,GLI SPINEDIT PER LA SCELTA}
{DEGLI INTERVALLI E IL RESET NON SIANO ATTIVI E L'OPZIONE AUTO}
{NON SIA SELEZIONATA.    INFATTI E' POSSIBILE USARE INTERVALLI}
{PERSONALIZZATI SOLO QUANDO GLI INTERVALLI  DI  QUANTIZZAZIONE}
{SONO TRE.}
Procedure TFormImmagine.SpinEditGradazioniChange(Sender:
TObject);
begin
     if SpinEditGradazioni.Value=3
        then begin
                  CheckBoxAutomatic.Enabled:=True;
                  If CheckBoxAutomatic.Checked=False
                  then begin
                            SpinEditApproxMin.Enabled:=True;
                            SpinEditApproxMax.Enabled:=True;
                            ButtonReset.Enabled:=True;
                       end;
             end
        else begin
                  SpinEditApproxMin.Enabled:=False;
                  SpinEditApproxMax.Enabled:=False;
                  ButtonReset.Enabled:=False;
                  CheckBoxAutomatic.Enabled:=False;
            end;
end;

{QUESTA PROCEDURA, SE E' STATA SCELTA L'OPZIONE  "RILEVA MOVI-}
MENTO MOUSE", OGNI VOLTA CHE "SENTE" L'EVENTO DI MOVIMENTO DEL}
{MOUSE RICHIAMA LA PROCEDURA CHE VERREBBE NORMALMENTE CHIAMATA}
{AL CLICK DEL MOUSE}
Procedure TFormImmagine.ImageCentralMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
var Button:TmouseButton;
begin
     if CheckBoxMouse.State=cbChecked then
```

```
        ImageCentralMouseDown(Self,Button,Shift,x,y);
end;

{QUESTA PROCEDURA , OGNI VOLTA CHE VIENE PREMUTO IL TASTO SINI-}
{STRO DEL MOUSE, LEGGE LA POSIZIONE DEL MOUSE, LEGGE  IL  PIXEL}
{SOTTO IL PUNTATORE, APPROSSIMA  IL  COLORE, RIEMPIE  TUTTE  LE}
{LABEL CONTENENTI LE GRADAZIONI DEI COLORI,MANDA IL COLORE ALLA}
{PARALLELA,RILEGGE LA PARALLELA PER AVERE LA STRINGA IN USCITA,}
{E INFINE PREPARA DUE BITMAP, UNA CON IL COLORE DA APPROSSIMARE}
{E UNA CON IL COLORE DOPO L'APPROSSIMAZIONE,E LE ASSEGNA AI DUE}
{CANVAS DI IMAGECOLORESCELTO E IMAGECOLORAPPROX}
Procedure TFormImmagine.ImageCentralMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
  DC:Hdc;
  Colore,Colore_Approx:Longint;
  Rosso,Verde,Blu:Byte;
  App_Rosso,App_Verde,App_Blu:Byte;
  Mode:byte;
  Rettangolo:TRect;
  L,H:Integer;
begin
     {MODE E' IL NUMERO DI INTERVALLI DI QUANTIZZAZIONE}
     Mode:=SpinEditgradazioni.Value;
     {DC E' IL DEVICE CONTEXT ATTIVO, CIOE' LA FINESTRA ATTIVA}
     DC:=GetDC(GetActiveWindow);
     {SE IL MOUSE E' SULL'IMMAGINE E NON FUORI DAI BORDI}
     TrovaDimensioniImmagine(L,H);
     if (X < L) and (Y < H) then Colore:=EstraiColore(DC,X,Y)
                           else Colore:=0;
       begin
          {ESTRAGGO IL COLORE DEL PIXEL}
          {APPROSSIMO IL COLORE}
          Colore_Approx:=Approx(Colore,Mode,
             SpinEditApproxMin.Value,SpinEditApproxMax.Value);
          Rosso:=GetRValue(Colore);  {ESTRAGGO IL ROSSO}
          Verde:=GetGValue(Colore);  {ESTRAGGO IL VERDE}
          Blu  :=GetBValue(Colore);  {ESTRAGGO IL BLU}
          App_Rosso:=GetRValue(Colore_Approx);
          {APPROSSIMO SOLO IL ROSSO}
          App_Verde:=GetGValue(Colore_Approx);
          {APPROSSIMO SOLO IL VERDE}
          App_Blu  :=GetBValue(Colore_Approx);
          {APPROSSIMO SOLO IL BLU.}
          {SCRIVO LA POSIZIONE DEL MOUSE IN LABELPOSIZIONE}
          LabelPosizione.Caption :=
```

```
          '(' + IntToStr(X) + ',' + IntToStr(Y) + ')';
         {RIEMPIO LE LABEL CON I VALORI DEI COLORI SCELTI}
         LabelValoreRosso.Caption:=IntToStr(Rosso);
         LabelValoreVerde.Caption:=IntToStr(Verde);
         LabelValoreBlu  .Caption:=IntToStr(Blu  );
        {RIEMPIO LE LABEL CON I VALORI APPROSSIMATI DEI COLORI}
         LabelValoreApproxRosso.Caption:=IntToStr(App_Rosso);
         LabelValoreApproxVerde.Caption:=IntToStr(App_Verde);
         LabelValoreApproxBlu  .Caption:=IntToStr(App_Blu  );
         {MANDO I COLORI APPROSSIMATI FUORI DALLA PORTA}
         {PARALLELA}
         OutParallela(App_Rosso,App_Verde,App_Blu,Mode);
         {LEGGO L'USCITA DELLA PARALLELA E LA SCRIVO NELLA SUA}
         {LABEL}
         LabelDataOut.Caption:=InParallela;
         {PREPARO UN RETTANGOLO CON LE GIUSTE DIMENSIONI}
         Rettangolo := Rect(0, 0, ImageColoreScelto.Width,
                       ImageColoreScelto.Height);
         {SCELGO IL COLORE DELLA PENNA UGUALE AL COLORE NON}
         {APPROSSIMATO}
         ImageColoreScelto.Canvas.Brush.Color := Colore;
         {RIEMPIO IL CANVAS CON IL RETTANGOLO E IL COLORE}
         {DELLA PENNA}
         ImageColoreScelto.Canvas.FillRect(Rettangolo);
         {FACCIO LA STESSA COSA CON L'ALTRO RETTANGOLO E IL}
         {COLORE APPROSSIMATO}
         Rettangolo := Rect(0, 0, ImageColoreApprox.Width,
                       ImageColoreApprox.Height);
         ImageColoreApprox.Canvas.Brush.Color := Colore_Approx;
         ImageColoreApprox.Canvas.FillRect(Rettangolo);
       end;
     {RILASCIO LE LOCAZIONI DI MEMORIA OCCUPATE IN SEDE DI}
     {DEFINIZIONE}
     {DELLA FINESTRA ATTIVA}
     ReleaseDC(GetActiveWindow,DC);
end;

{QUESTA PROCEDURA SENTE IL CLICK SULLA OPZIONE "AUTO", E}
{CONTROLLA CHE LE SPINEDIT E IL RESET SIANO O MENO ABILITATE}
Procedure TFormImmagine.CheckBoxAutomaticClick(Sender: TObject);
begin
     if CheckBoxAutomatic.Checked=True
     then begin
               SpinEditApproxMin.Enabled:=False;
               SpinEditApproxMax.Enabled:=False;
               ButtonReset.Enabled:=False;
```

```
          end
     else begin
               SpinEditApproxMin.Enabled:=True;
               SpinEditApproxMax.Enabled:=True;
               ButtonReset.Enabled:=True;
          end;
end;

{QUESTA PROCEDURA RIPORTA NEGLI SPINEDIT MIN E MAX I VALORI}
{DI DEFAULT: VIENE RICHIAMATA DAL CLICK SU RESET}
Procedure TFormImmagine.ButtonResetClick(Sender: TObject);
begin
     SpinEditApproxMin.Value:=85;
     SpinEditApproxMax.Value:=170;
end;

{QUESTA PROCEDURA RISPONDE ALL'EVENTO DI UN CLICK SUL PULSANTE}
{REFRESH, E RIVISUALIZZA L'IMMAGINE PRESENTE NELLA FINESTRA}
Procedure TFormImmagine.BitBtnRefreshClick(Sender: TObject);
begin
     {RICHIAMO IL METODO REFRESH SU IMAGECENTRAL DI}
     {FORMIMMAGINE}
     FormImmagine.ImageCentral.Refresh;
     FormImmagine.Refresh;
end;

{QUESTA PROCEDURA VIENE RICHIAMATA AD OGNI CLICK}
{SULL'OPZIONE STRETCH}
procedure TFormImmagine.CheckBoxStretchClick(Sender: TObject);
begin
     ImageCentral.Stretch:=CheckBoxStretch.Checked;
     Refresh;
end;

{QUESTA PROCEDURA VIENE ESEGUITA QUANDO TIMER1 O TIMER2}
{SEGNALANO LA SCADENZA DEL TEMPO A LORO ASSEGNATO}
procedure TFormImmagine.TimerTimer(Sender: TObject);
var Timer:TTimer;
begin
     Timer:= Sender as TTimer;
     Timer.Enabled:=False;
     With FormImmagine do
     begin
          Port[StrToInt(EditIndirizzo.Text)]:=Timer.Tag;
          LabelDataOut.Caption:=InParallela;
```

```
          Timer3.Interval:=0;
          Timer3.Interval:=SpinEditOscillazione.Value;
     end;
     Timer.Tag:=0;
end;

{QUESTA PROCEDURA VIENE ESEGUITA QUANDO E' NECESSARIO}
{FAR OSCILLARE UN SEGNALE DI USCITA.  IL VALORE DELLA}
{USCITA VIENE SCAMBIATO TRA IL VALORE  ATTUALE  E  IL}
{VALORE DELLA VARIABILE TAG (E' UN PARAMETRO NON USA-}
{TO DA DELPHI MA PRESENTE IN TUTTI I COMPONENTI}
procedure TFormImmagine.Timer3Timer(Sender: TObject);
var Memoria:Longint;
begin
     With FormImmagine do
      If Timer1.Tag+Timer2.Tag=0
        then begin
                 Memoria:=Timer3.Tag;
                 Timer3.Tag:=Port[StrToInt(EditIndirizzo.Text)];
                 Port[StrToInt(EditIndirizzo.Text)]:=Memoria;
                 LabelDataOut.Caption:=InParallela;
             end
        else Timer3.Interval:=SpinEditOscillazione.Value;
             end;
end;

{CON UN DOPPIO CLICK SI PUO'  TOGLIERE  IL  RITARDO}
{OPPURE RIMETTERLO. QUESTA PROCEDURA VIENE LANCIATA}
{IN CASO DI DOPPIO CLICK}
procedure TFormImmagine.ImageCentralDblClick(Sender: TObject);
begin
     If SpinEditRitardo.Text<>'0'
        then begin
                  SpinEditRitardo.Tag:=
                     StrToInt(SpinEditRitardo.Text);
                  Timer1.Enabled:=False;
                  Timer2.Enabled:=False;
                  SpinEditRitardo.Text:='0';
             end
        else begin
                  SpinEditRitardo.Text:=
                      IntToStr(SpinEditRitardo.Tag);
                  SpinEditRitardo.Tag:=0;
             end;
end;
```

```
{OGNI VOLTA CHE  CAMBIA  IL  VALORE  DEL  TEMPO  DI}
{OSCILLAZIONE, VIENE AGGIORNATO IL VALORE SUL TIMER}
procedure TFormImmagine.SpinEditOscillazioneChange(Sender:
TObject);
begin
     With FormImmagine do
       if Timer3.Interval<>0
          then Timer3.Interval:=SpinEditOscillazione.Value;
end;

end.
```

FILE IMAGEWIN.PAS

```
unit ImageWin;

{QUESTA UNIT CONTIENE LA FINESTRA PER IL CARICAMENTO DI}
{UNA NUOVA IMMAGINE}

interface

uses WinTypes, WinProcs, Classes, Graphics, Forms, Controls,
     FileCtrl, StdCtrls, ExtCtrls, Buttons, Spin;

type
  TImageForm = class(TForm)
    DirectoryListBox: TDirectoryListBox;
    DriveComboBox: TDriveComboBox;
    PathLabel: TLabel;
    FileEdit: TEdit;
    PanelPreview: TPanel;
    ImagePreview: TImage;
    FileListBox: TFileListBox;
    LabelNomeImmagine: TLabel;
    ViewBtn: TBitBtn;
    BevelPreview: TBevel;
    FilterComboBox: TFilterComboBox;
    StretchCheck: TCheckBox;
    BitBtnOK: TBitBtn;
    BitBtnCancel: TBitBtn;
    procedure FileListBoxClick(Sender: TObject);
    procedure ViewBtnClick(Sender: TObject);
    procedure StretchCheckClick(Sender: TObject);
    procedure FileEditKeyPress(Sender: TObject; var Key: Char);
    procedure BitBtnOKClick(Sender: TObject);
    procedure BitBtnCancelClick(Sender: TObject);
    procedure FileListBoxKeyPress(Sender: TObject; var Key:
              Char);
  end;

var
  ImageForm: TImageForm;

implementation

uses ViewWin, SysUtils, Colore;
```

```
{$R *.DFM}

{QUESTA PROCEDURA RISPONDE ALL'EVENTO CLICK SULLA LISTA DEI}
{FILE VISUALIZZANDO L'IMMAGINE CORRISPONDENTE IN ANTEPRIMA}
procedure TImageForm.FileListBoxClick(Sender: TObject);
var
  FileExt: string[4];
begin
  {FILEEXT E' UNA STRINGA DI QUATTRO CARATTERI CHE CONTIENE,}
  {IN CARATTERI MAIUSCOLI, L'ESTENZIONE DEL FILE}
  FileExt := UpperCase(ExtractFileExt(FileListBox.Filename));
  {SE IL FILE E' DEL TIPO BITMAP OPPURE ICONA OPPURE WINDOWS}
  {META FILE, ALLORA PUO' ESSERE VISUALIZZATO IN ANTEPRIMA}
  if (FileExt = '.BMP')
    or (FileExt = '.ICO')
    or (FileExt = '.WMF')
     then
      begin
       {IL FILE VIENE CARICATO IN IMAGEPREVIEW}
       ImagePreview.Picture.LoadFromFile(FileListBox.Filename);
       {IL NOME DEL FILE VIENE ASSEGNATO ALLA CAPTION}
       {IMMEDIATAMENTE SOPRA L'IMMAGINE}
       LabelNomeImmagine.Caption:=
         ExtractFilename(FileListBox.Filename);
       {SE IL FILE E' DI TIPO BITMAP, OLTRE A VISUALIZZARLO}
       {VENGONO AGGIUNTE LE DIMENSIONI A FIANCO DEL NOME}
       if (FileExt = '.BMP') then
         begin
           LabelNomeImmagine.Caption:=
              LabelNomeImmagine.Caption +
              Format(' (%d x %d)', [ImagePreview.Picture.Width,
              ImagePreview.Picture.Height]);
           ViewForm.ImageFull.Picture := ImagePreview.Picture;
         end;
       if FileExt = '.ICO'
          then Icon := ImagePreview.Picture.Icon;
       if FileExt = '.WMF'
          then ViewForm.ImageFull.Picture.Metafile:=
               ImagePreview.Picture.Metafile;
      end;
end;

{QUESTA PROCEDURA RISPONDE AL CLICK SUL PULSANTE FULL VIEW}
{CREANDO E ATTIVANDO UNA FINESTRA DELLE DIMENSIONI GIUSTE}
Procedure TImageForm.ViewBtnClick(Sender: TObject);
begin
```

```
  ViewForm.HorzScrollBar.Range := ImagePreview.Picture.Width;
  ViewForm.VertScrollBar.Range := ImagePreview.Picture.Height;
  ViewForm.Caption := LabelNomeImmagine.Caption;
  If ImagePreview.Picture.Width < ViewForm.Width
     then ViewForm.Width:=ImagePreview.Picture.Width+8;
  If ImagePreview.Picture.Height < ViewForm.Height
     then ViewForm.Height:=ImagePreview.Picture.Height+27;
  ViewForm.Show;
end;

{QUESTA PROCEDURA RISPONDE ALL'EVENTO CLICK SULL'OPZIONE}
{STRETCH: IL VALORE DELL'OPZIONE E' PASSATA DIRETTAMENTE}
{ALLA PROPRIETA' STRETCH, CHE SE  E'  IMPOSTATA  A  TRUE}
{PERMETTE ALL'IMMAGINI DI ESPANDERSI ESATTAMENTE FINO AI}
{BORDI DELL'AREA CHE GLI E' STATA ASSEGNATA}
Procedure TImageForm.StretchCheckClick(Sender: TObject);
begin
     ImagePreview.Stretch := StretchCheck.Checked;
end;

{QUESTA PROCEDURA "SENTE" LA PRESSIONE DEL  TASTO  INVIO}
{NELLA FILEEDIT DOVE SI PUO'  SCRIVERE  DIRETTAMENTE  IL}
{NOME DEL FILE IMMAGINE DA VISUALIZZARE}
Procedure TImageForm.FileEditKeyPress(Sender: TObject; var Key:
Char);
begin
  {SE IL TASTO PREMUTO E' INVIO ALLORA IL TESTO PRESENTE}
  {NELLA FILEEDIT E' APPLICATO ALLA FILELISTBOX}
  if Key = #13 then
  begin
    FileListBox.ApplyFilePath(FileEdit.Text);
    Key := #0;
  end;
end;

{QUESTA PROCEDURA "SENTE" LA PRESSIONE DEL TASTO OK E CARICA}
{IL L'IMMAGINE DEL FILE SCELTO  NELLA  FINESTRA  PRINCIPALE,}
{RENDENDO INVISIBILE QUELLA DELLA SCELTA DELL'IMMAGINE}
Procedure TImageForm.BitBtnOKClick(Sender: TObject);
begin
     FormImmagine.ImageCentral.Stretch:=
        FormImmagine.CheckBoxStretch.Checked;
     FormImmagine.ImageCentral.Picture.LoadFromFile
      (FileListBox.Filename);
     ImageForm.Visible:=False;
```

```
end;

{QUESTA PROCEDURA "SENTE" LA PRESSIONE DEL  TASTO  CANCEL E}
{TORNA ALLA FINESTRA PRINCIPALE SENZA CARICARE L'IMMAGINE.}
Procedure TImageForm.BitBtnCancelClick(Sender: TObject);
begin
     ImageForm.Visible:=False;
end;

{QUESTA PROCEDURA "SENTE" LA PRESSIONE DEL TASTO INVIO }
{E TORNA ALLA FINESTRA PRINCIPALE CARICANDO L'IMMAGINE.}
procedure TImageForm.FileListBoxKeyPress(Sender: TObject; var
Key: Char);
begin
     if key=#13 then ImageForm.BitBtnOkClick(Self);
end;

end.
```

FILE VIEWWIN.PAS

```
UNIT VIEWWIN;

{QUESTA UNIT CONTIENE LA FINESTRA CHE SI APRE QUANDO SI CLICCA}
{SUL PULSANTE FULL VIEW DI IMAGE VIEWER. LA FINESTRA CONTIENE }
{L'IMMAGINE NELLE SUE DIMENSIONI ORIGINALI}

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
  Controls, Forms, Dialogs, ExtCtrls;

type
  TViewForm = class(TForm)
    ImageFull: TImage;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ViewForm: TViewForm;

implementation

{$R *.DFM}

end.
```

Alma Mater Studiorum
Secularia Nona

FILE COLORE.DFM

```
object FormImmagine: TFormImmagine
  Left = 146
  Top = 67
  Width = 639
  Height = 479
  Caption = 'Approssimazione del colore'
  Color = -11250604
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = []
  PixelsPerInch = 96
  Position = poScreenCenter
  ShowHint = True
  WindowState = wsMaximized
  TextHeight = 16
  object ImageCentral: TImage
    Left = 0
    Top = 0
    Width = 631
    Height = 349
    Align = alClient
    AutoSize = True
    Center = True
    Picture.Data = {..immagine..}
    Stretch = True
    OnDblClick = ImageCentralDblClick
    OnMouseDown = ImageCentralMouseDown
    OnMouseMove = ImageCentralMouseMove
  end
  object PanelComandi: TPanel
    Left = 0
    Top = 349
    Width = 631
    Height = 103
    Align = alBottom
    TabOrder = 0
    object PanelParallela: TPanel
      Left = 289
      Top = 1
      Width = 124
      Height = 101
      Align = alLeft
```

```
BevelInner = bvRaised
BevelOuter = bvLowered
BevelWidth = 2
TabOrder = 1
object LabelDataOut: TLabel
  Left = 38
  Top = 77
  Width = 76
  Height = 16
  Caption = '00-00-00-00'
end
object LabelOut: TLabel
  Left = 10
  Top = 76
  Width = 22
  Height = 17
  Caption = 'Out'
  Font.Color = clBlack
  Font.Height = -15
  Font.Name = 'Times New Roman'
  Font.Style = []
  ParentFont = False
end
object LabelIndirizzo: TLabel
  Left = 12
  Top = 8
  Width = 35
  Height = 15
  Caption = 'Indirizzo'
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'Times New Roman'
  Font.Style = []
  ParentFont = False
end
object LabelPortaParallela: TLabel
  Left = 16
  Top = 18
  Width = 102
  Height = 19
  Caption = 'Porta  parallela'
  Font.Color = clBlack
  Font.Height = -16
  Font.Name = 'Times New Roman'
  Font.Style = [fsBold, fsUnderline]
  ParentFont = False
```

```
  Visible = False
end
object LabelRitardo: TLabel
  Left = 10
  Top = 30
  Width = 39
  Height = 15
  Caption = '&Ritardo'
  FocusControl = SpinEditRitardo
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'Times New Roman'
  Font.Style = []
  ParentFont = False
end
object LabelOscillazione: TLabel
  Left = 10
  Top = 54
  Width = 44
  Height = 15
  Caption = '&Oscillaz.'
  FocusControl = SpinEditOscillazione
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'Times New Roman'
  Font.Style = []
  ParentFont = False
end
object EditIndirizzo: TEdit
  Left = 64
  Top = 5
  Width = 49
  Height = 21
  Font.Color = clBlack
  Font.Height = -11
  Font.Name = 'Times New Roman'
  Font.Style = []
  ParentFont = False
  TabOrder = 0
  Text = '$378'
end
object SpinEditRitardo: TSpinEdit
  Left = 64
  Top = 28
  Width = 49
  Height = 23
```

```
        Hint = 'Doppio clic sull'#39'immagine per togliere il
ritardo'
        Font.Color = clBlack
        Font.Height = -11
        Font.Name = 'Times New Roman'
        Font.Style = []
        Increment = 100
        MaxValue = 9999
        MinValue = 0
        ParentFont = False
        TabOrder = 1
        Value = 500
      end
      object SpinEditOscillazione: TSpinEdit
        Left = 64
        Top = 52
        Width = 49
        Height = 23
        Hint = 'Impostare a zero per non avere
l'#39'oscillazione'
        Font.Color = clBlack
        Font.Height = -11
        Font.Name = 'Times New Roman'
        Font.Style = []
        Increment = 100
        MaxValue = 9999
        MinValue = 0
        ParentFont = False
        TabOrder = 2
        Value = 300
        OnChange = SpinEditOscillazioneChange
      end
    end
    object PanelMostraColori: TPanel
      Left = 1
      Top = 1
      Width = 288
      Height = 101
      Align = alLeft
      BevelInner = bvRaised
      BevelOuter = bvLowered
      BevelWidth = 2
      TabOrder = 2
      object LabelRosso: TLabel
        Left = 8
        Top = 29
```

```
  Width = 42
  Height = 16
  Caption = 'Rosso'
end
object LabelVerde: TLabel
  Left = 8
  Top = 51
  Width = 37
  Height = 16
  Caption = 'Verde'
end
object LabelBlu: TLabel
  Left = 8
  Top = 73
  Width = 22
  Height = 16
  Caption = 'Blu'
end
object LabelValoreVerde: TLabel
  Left = 72
  Top = 51
  Width = 4
  Height = 16
  Caption = 'LabelValoreVerde'
end
object LabelValoreRosso: TLabel
  Left = 72
  Top = 29
  Width = 4
  Height = 16
  Caption = 'LabelValoreRosso'
end
object LabelValoreBlu: TLabel
  Left = 72
  Top = 73
  Width = 4
  Height = 16
  Caption = 'LabelValoreBlu'
end
object LabelPosizione: TLabel
  Left = 8
  Top = 9
  Width = 57
  Height = 16
  Caption = 'LabelPosizione'
end
```

```
object LabelColoreScelto: TLabel
  Left = 74
  Top = 9
  Width = 85
  Height = 16
  Caption = 'Colore scelto'
end
object LabelColoreApprox: TLabel
  Left = 177
  Top = 9
  Width = 91
  Height = 16
  Caption = 'Colore approx'
end
object LabelValoreApproxRosso: TLabel
  Left = 176
  Top = 29
  Width = 4
  Height = 16
  Caption = 'LabelValoreApproxRosso'
end
object LabelValoreApproxVerde: TLabel
  Left = 176
  Top = 51
  Width = 4
  Height = 16
  Caption = 'LabelValoreApproxVerde'
end
object LabelValoreApproxBlu: TLabel
  Left = 176
  Top = 73
  Width = 4
  Height = 16
  Caption = 'LabelValoreApproxBlu'
end
object ImageColoreScelto: TImage
  Left = 115
  Top = 31
  Width = 41
  Height = 58
end
object ImageColoreApprox: TImage
  Left = 223
  Top = 31
  Width = 41
  Height = 58
```

```
  end
end
object PanelGradazioni: TPanel
  Left = 413
  Top = 1
  Width = 217
  Height = 101
  Align = alLeft
  BevelInner = bvRaised
  BevelOuter = bvLowered
  BevelWidth = 2
  TabOrder = 0
  object BitBtnRiduci: TBitBtn
    Left = 8
    Top = 38
    Width = 89
    Height = 17
    Caption = 'Ri&duci colori'
    Font.Color = clBlack
    Font.Height = -13
    Font.Name = 'System'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 2
    OnClick = BitBtnRiduciClick
  end
  object MemoGradazioni: TMemo
    Left = 8
    Top = 9
    Width = 109
    Height = 26
    Color = clAqua
    Font.Color = clBlack
    Font.Height = -13
    Font.Name = 'System'
    Font.Style = [fsBold]
    Lines.Strings = (
      'N° gradazioni')
    ParentFont = False
    TabOrder = 4
  end
  object SpinEditGradazioni: TSpinEdit
    Left = 120
    Top = 9
    Width = 37
    Height = 26
```

```
  Color = clAqua
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = [fsBold]
  MaxValue = 4
  MinValue = 2
  ParentFont = False
  TabOrder = 5
  Value = 3
  OnChange = SpinEditGradazioniChange
end
object BitBtnRefresh: TBitBtn
  Left = 100
  Top = 38
  Width = 57
  Height = 17
  Caption = 'Refres&h'
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = [fsBold]
  ParentFont = False
  TabOrder = 3
  OnClick = BitBtnRefreshClick
end
object BitBtnCarica: TBitBtn
  Left = 100
  Top = 58
  Width = 57
  Height = 17
  Caption = '&Carica'
  TabOrder = 0
  OnClick = BitBtnCaricaClick
end
object CheckBoxMouse: TCheckBox
  Left = 8
  Top = 78
  Width = 145
  Height = 17
  Caption = 'Rileva mov. &mouse'
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = [fsBold]
  ParentFont = False
```

```
  State = cbChecked
  TabOrder = 6
end
object CheckBoxAutomatic: TCheckBox
  Left = 160
  Top = 6
  Width = 49
  Height = 17
  Caption = 'Auto'
  Color = clBtnFace
  ParentColor = False
  State = cbChecked
  TabOrder = 7
  OnClick = CheckBoxAutomaticClick
end
object SpinEditApproxMin: TSpinEdit
  Left = 160
  Top = 23
  Width = 49
  Height = 26
  Enabled = False
  Font.Color = clBlack
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = [fsBold]
  Increment = 5
  MaxValue = 125
  MinValue = 0
  ParentFont = False
  TabOrder = 8
  Value = 85
end
object SpinEditApproxMax: TSpinEdit
  Left = 160
  Top = 49
  Width = 49
  Height = 26
  Enabled = False
  Increment = 5
  MaxValue = 255
  MinValue = 130
  TabOrder = 9
  Value = 170
end
object ButtonReset: TButton
  Left = 160
```

```
      Top = 78
      Width = 49
      Height = 17
      Caption = 'Reset'
      Enabled = False
      TabOrder = 10
      OnClick = ButtonResetClick
    end
    object CheckBoxStretch: TCheckBox
      Left = 8
      Top = 58
      Width = 69
      Height = 17
      Caption = 'Stre&tch'
      Font.Color = clBlack
      Font.Height = -13
      Font.Name = 'System'
      Font.Style = [fsBold]
      ParentFont = False
      State = cbChecked
      TabOrder = 1
      OnClick = CheckBoxStretchClick
    end
  end
  object PanelDestro: TPanel
    Left = 630
    Top = 1
    Width = 169
    Height = 101
    Align = alClient
    BevelInner = bvRaised
    BevelOuter = bvLowered
    BevelWidth = 2
    TabOrder = 3
  end
end
object Timer1: TTimer
  Interval = 0
  OnTimer = TimerTimer
  Left = 32
  Top = 16
end
object Timer2: TTimer
  Interval = 0
  OnTimer = TimerTimer
  Left = 76
```

```
    Top = 16
  end
  object Timer3: TTimer
    Interval = 0
    OnTimer = Timer3Timer
    Left = 32
    Top = 60
  end
end
```

FILE IMAGEWIN.DFM

```
object ImageForm: TImageForm
  Left = 192
  Top = 162
  ActiveControl = FileListBox
  BorderIcons = [biSystemMenu, biMinimize]
  BorderStyle = bsSingle
  Caption = 'Image Viewer'
  ClientHeight = 275
  ClientWidth = 572
  Font.Color = clBlack
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  KeyPreview = True
  PixelsPerInch = 96
  Position = poScreenCenter
  TextHeight = 13
  object BevelPreview: TBevel
    Left = 316
    Top = 5
    Width = 253
    Height = 220
  end
  object PathLabel: TLabel
    Left = 10
    Top = 16
    Width = 38
    Height = 13
    Caption = 'e:\tesi'
  end
  object LabelNomeImmagine: TLabel
    Left = 325
    Top = 11
    Width = 117
    Height = 13
    Caption = 'LabelNomeImmagine'
  end
  object DirectoryListBox: TDirectoryListBox
    Left = 8
    Top = 40
    Width = 148
    Height = 178
    DirLabel = PathLabel
```

```
  FileList = FileListBox
  IntegralHeight = True
  ItemHeight = 16
  TabOrder = 3
end
object DriveComboBox: TDriveComboBox
  Left = 9
  Top = 240
  Width = 148
  Height = 19
  DirList = DirectoryListBox
  TabOrder = 1
end
object FileEdit: TEdit
  Left = 166
  Top = 13
  Width = 139
  Height = 20
  TabOrder = 2
  Text = '*.bmp;*.ico;*.wmf'
  OnKeyPress = FileEditKeyPress
end
object PanelPreview: TPanel
  Left = 324
  Top = 27
  Width = 237
  Height = 158
  BevelInner = bvLowered
  TabOrder = 9
  object ImagePreview: TImage
    Left = 2
    Top = 2
    Width = 233
    Height = 154
    Align = alClient
    Stretch = True
  end
end
object FileListBox: TFileListBox
  Left = 166
  Top = 41
  Width = 139
  Height = 184
  FileEdit = FileEdit
  ItemHeight = 13
  Mask = '*.bmp;*.ico;*.wmf'
```

```
    TabOrder = 4
    OnClick = FileListBoxClick
    OnKeyPress = FileListBoxKeyPress
  end
  object ViewBtn: TBitBtn
    Left = 333
    Top = 194
    Width = 63
    Height = 24
    Caption = '&Full View'
    Font.Color = clBlack
    Font.Height = -13
    Font.Name = 'Times New Roman'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 7
    OnClick = ViewBtnClick
  end
  object FilterComboBox: TFilterComboBox
    Left = 166
    Top = 240
    Width = 140
    Height = 21
    FileList = FileListBox
    Filter = 'Image Files (*.bmp, *.ico,
*.wmf)|*.bmp;*.ico;*.wmf|Bitmap Files (*.bmp)|*.bmp|Icons
(*.ico)|*.ico|Metafiles (*.wmf)|*.wmf|All files (*.*)|*.*'
    TabOrder = 0
  end
  object StretchCheck: TCheckBox
    Left = 493
    Top = 200
    Width = 64
    Height = 17
    Caption = 'Stretch'
    State = cbChecked
    TabOrder = 8
    OnClick = StretchCheckClick
  end
  object BitBtnOK: TBitBtn
    Left = 496
    Top = 240
    Width = 73
    Height = 25
    Caption = '&OK'
    Font.Color = clBlack
```

```
    Font.Height = -13
    Font.Name = 'Times New Roman'
    Font.Style = [fsBold]
    ModalResult = 1
    ParentFont = False
    TabOrder = 5
    OnClick = BitBtnOKClick
    Glyph.Data = {..immagine  }
    NumGlyphs = 2
  end
  object BitBtnCancel: TBitBtn
    Left = 408
    Top = 240
    Width = 73
    Height = 25
    Caption = '&Cancel'
    Font.Color = clBlack
    Font.Height = -13
    Font.Name = 'Times New Roman'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 6
    OnClick = BitBtnCancelClick
    Kind = bkCancel
  end
end
```

FILE VIEWWIN.DFM

```
object ViewForm: TViewForm
  Left = 135
  Top = 130
  AutoScroll = False
  Caption = 'View Form'
  ClientHeight = 350
  ClientWidth = 640
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'System'
  Font.Style = []
  PixelsPerInch = 96
  Position = poScreenCenter
  TextHeight = 16
  object ImageFull: TImage
    Left = 0
    Top = 0
    Width = 640
    Height = 350
    Align = alClient
    AutoSize = True
  end
end
```

Alma Mater Studiorum
Sæcularia Nona

BIBLIOGRAFIA

EMANUELE BIONDI, PIER LUIGI EMILIANI E PIETRO MORASSO (1995), *Protesi e ausili per la comunicazione*, Pàtron, Modelli del sistema tattile.

GIUSEPPE BIONDO, ENRICO SACCHI, *Manuale di elettronica e telecomunicazioni* (terza edizione), Hoepli, cap.15 - Dispositivi integrati.

P.U. CALZOLARI, S. GRAFFI, *Elementi di Elettronica*, Zanichelli.

NIKLAUS WIRTH, *Algoritmi + Strutture Dati = Programmi*, Tecniche Nuove, 1987.

Guida per la programmazione orientata agli oggetti, EDIA BORLAND.

JON MATCHO, DAVID FAULKNER, *Using Delphi*, QUE, 1995.

MARCO CANTÙ, *Mastering Delphi 2*, SYBEX, 1995

FILIPPO BOSI, *Dev. software sistemi & soluzioni*, EDIZIONI INFOMEDIA; dal numero 24 di Novembre 1994 al numero 36 di Ottobre 1996.

A.K. JAIN, *Fundamentals of digital image processing*, PRENTICE HALL, 1989.

Alma Mater Studiorum
Sæcularia Nona

RINGRAZIAMENTI

Ringrazio il Prof. Massimo Ferri per il lavoro affidatomi e per la fiducia datami durante tutto il lavoro. Grazie alla mia famiglia che mi ha garantito un insostituibile aiuto morale durante tutti i miei studi. Un sentito ringraziamento a Davide Bellini, Alberto Bellini, e Luca Morgantini per il supporto durante la progettazione e la realizzazione dell'hardware e a Filippo Bosi per i preziosi consigli software, oltre a Claudia Piccioni, Antonio Parazza, Bruno Albertazzi, Vittorio Bacchetti, Claudia Graziani e Palmira Lippi per la disponibilità durante la sperimentazione.