

BOARD DISPLAY 7 SEGMENTI

1. COME SI USA

La board mostra un numero a quattro cifre sui display 7 segmenti e viene utilizzata insieme a un microcontrollore che ne comanda l'accensione. Per funzionare la board ha bisogno di essere collegata correttamente alla scheda di sviluppo; i **collegamenti** corretti sono mostrati in Figura 1 per scheda di sviluppo DAM (Digital Analog Microcontroller), e in Figura 2 per una scheda di sviluppo EasyPIC5. Il gruppo di pin a sinistra sono i pin per il controllo dei BJT e vanno collegati a 4 pin qualsiasi del microcontrollore, in Figura 1 viene mostrato il collegamento di questo gruppo con i primi 4 pin della porta A. Il secondo gruppo di pin a destra sono i pin che comandano l'accensione dei segmenti e vanno collegati a 8 pin qualsiasi del microcontrollore, ma per comodità, e per rendere la scrittura del programma molto più semplice, è utile usare un'intera porta a 8 pin del microcontrollore; in Figura 1 è stata usata la porta C.

La board non è stata progettata per funzionare solo con un particolare microcontrollore, ma effettuando correttamente i collegamenti descritti prima si può utilizzare un microcontrollore qualsiasi.

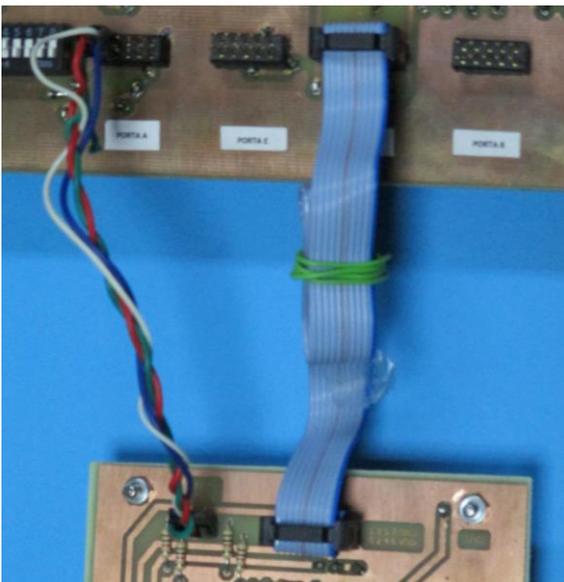
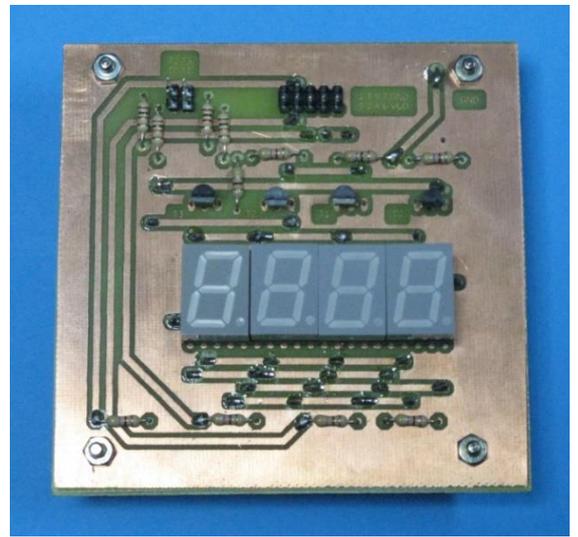


Figura 1. Esempio di collegamento a una scheda di sviluppo DAM

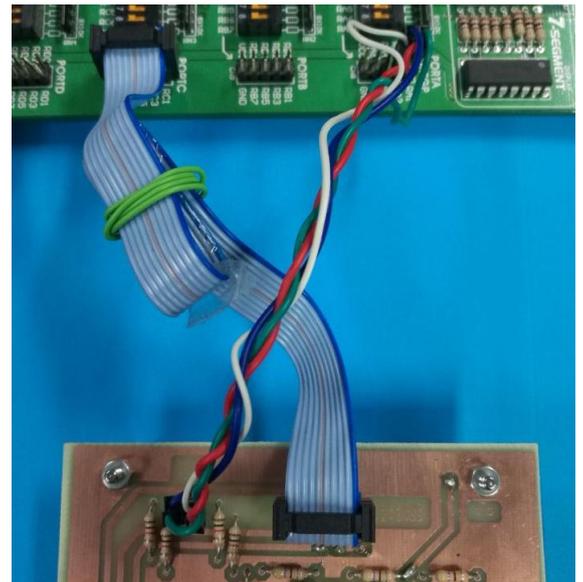


Figura 2. Esempio di collegamento a una scheda di sviluppo EasyPIC5

Dal punto di vista **software** si deve realizzare un programma che faccia accendere alternatamente i display in modo molto veloce, in modo da far sembrare il numero fisso sui display. Un esempio di programma (main + tre funzioni di libreria) per il microcontrollore PIC16F1789 è il seguente:

MAIN:

```

#include <pic16f1789.h>
#include <xc.h>
#include "xc8_header.h"
#include "libreria.h"

void main (void)
{
    while(1){
        Config7seg();
        for (int i = 0; i<= 9999; i++){
            Write7seg(i, 10);
        }
    }
}
    
```

FUNZIONI DI LIBRERIA:

```
void Config7seg () {  
  
    #define uscitaBJT0 PORTAbits.RA0 // chiamo i pin di controllo uscitaBJT#  
    #define uscitaBJT1 PORTAbits.RA1  
    #define uscitaBJT2 PORTAbits.RA2  
    #define uscitaBJT3 PORTAbits.RA3  
  
    TRISAbits.TRISA0 = 0; // PIN DI CONTROLLO DEI BJT  
    TRISAbits.TRISA1 = 0;  
    TRISAbits.TRISA2 = 0;  
    TRISAbits.TRISA3 = 0;  
    ANSELAbits.ANSA0 = 0;  
    ANSELAbits.ANSA1 = 0;  
    ANSELAbits.ANSA2 = 0;  
    ANSELAbits.ANSA3 = 0;  
  
    uscitaBJT0 = 1; // PRIMA LI SPENGO TUTTI  
    uscitaBJT1 = 1;  
    uscitaBJT2 = 1;  
    uscitaBJT3 = 1;  
  
    //-----  
  
    #define uscitaSegmenti PORTC // chiamo i pin dei segmenti uscitaSegmenti  
  
    TRISC = 0 ;  
    ANSELc = 0;  
  
    uscitaSegmenti = 255; // PRIMA LI SPENGO TUTTI  
  
}  
  
void Write7seg (int j, int velocita){  
  
    int migliaia, centinaia, decina, unita, i;  
  
    migliaia = j/1000;  
    centinaia = (j-migliaia*1000)/100;  
    decina = (j-(migliaia*1000+centinaia*100))/10;  
    unita = j-((migliaia*1000)+(centinaia*100)+(decina*10));  
  
    for(i=0; i< velocita; i++) //minore e' "velocita" più sarà veloce  
    {  
  
        uscitaBJT0 = 0; //accendo primo display  
        uscitaSegmenti = conversioneD(unita);  
        __delay_ms(1);  
        uscitaBJT0 = 1; //spengo primo display  
  
        uscitaBJT1 = 0; //accendo secondo display  
        uscitaSegmenti = conversioneD(decina);  
        __delay_ms(1);  
        uscitaBJT1 = 1; //spengo secondo display  
  
        uscitaBJT2 = 0; //accendo terzo display  
        uscitaSegmenti = conversioneD(centinaia);  
        __delay_ms(1);  
        uscitaBJT2 = 1; //spengo terzo display  
  
        uscitaBJT3 = 0; //accendo quarto display  
        uscitaSegmenti = conversioneD(migliaia);  
        __delay_ms(1);  
        uscitaBJT3 = 1; //spengo quarto display  
  
    }  
  
}
```

Per il dettaglio sulle funzioni usate vedi Paragrafo 4.

2.1. DESCRIZIONE: HARDWARE

La board display 7-segmenti è composta dai seguenti **componenti**:

- 4 display 7-segmenti HDSP5551 ad anodo comune;
- 4 transistori PNP BC640-16;
- 4 resistori da 1K Ω ;
- 8 resistori da 470 Ω .

La board presenta 14 **pin di ingresso** di tipo strip-line:

- 8 pin per l'accensione dei segmenti (compreso il punto);
- 4 pin di comando per l'accensione dei BJT;
- 2 pin di alimentazione (VDD, GND).

Lo schematico della basetta e il suo layout (privato dei piani di massa) sono mostrati rispettivamente in Figura 3 e Figura 4. Nello schematico si noti, in particolare, che il pin 10 del connettore strip-line in alto appare scollegato dal resto del circuito: esso in realtà è il pin GND che è scollegato dal circuito perché non è necessario un suo impiego all'interno del circuito stesso, ma è comunque utile portare il valore di riferimento GND all'interno del circuito per creare un adeguato piano di massa.

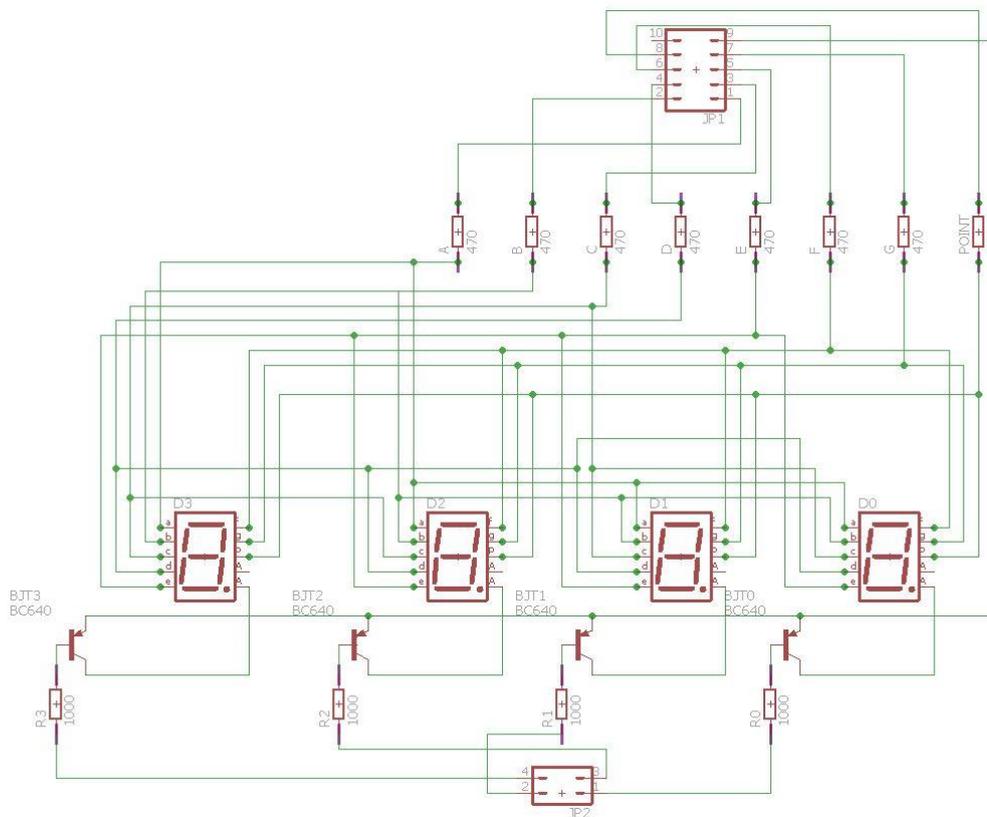


Figura 3. Schematic

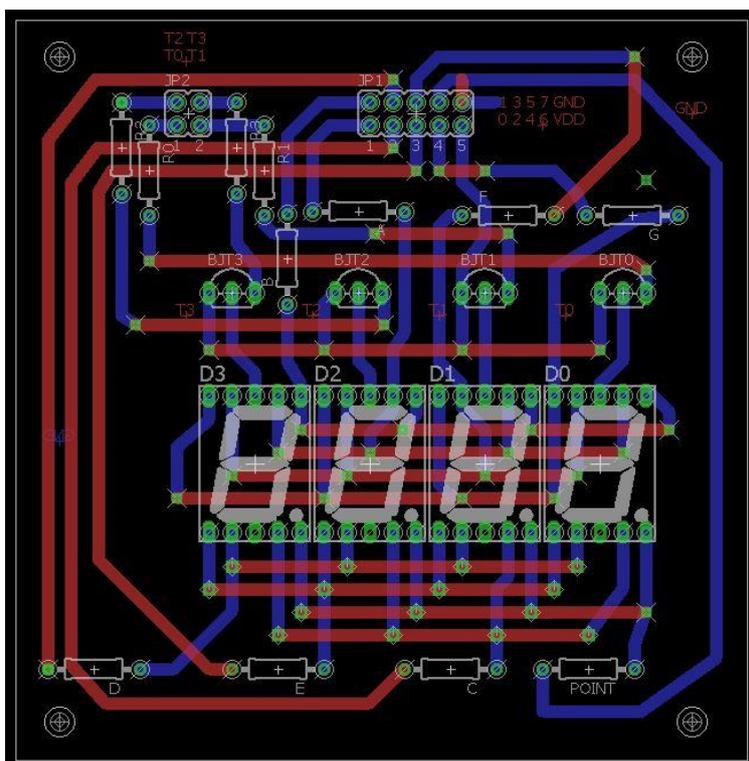


Figura 4. Layout (senza piani di massa): piste e componenti, Top (rosso), Bottom (blu)

2.2. DESCRIZIONE: TOP LAYER E BOTTOM LAYER

La board è dual-layer: i componenti si trovano tutti sul top layer della board, mentre le piste sono state distribuite su entrambe le facciate per ridurre le dimensioni della scheda e cercare di semplificare i collegamenti. Entrambi i layer con i relativi piani di massa sono mostrati e descritti in Figura 5 e in Figura 6.

Su tutta la board il **piano di massa** si trova alla tensione di riferimento GND, ma nel caso di un collegamento a coccodrillo è necessario prestare attenzione ad eventuali piste poste sul bottom layer della basetta; per risolvere questo problema è stato posizionato, vicino all'angolo destro in alto, un pin (indicato in giallo in Figura 5), collegato alla tensione di massa, a cui è possibile collegarsi.

I **connettori strip-line** (evidenziati in blu in Figura 5) sono tutti di tipo maschio e divisi in due gruppi: il gruppo di sinistra (4 pin) contiene i quattro pin di comando per l'accensione dei transistori, mentre il gruppo di destra (10 pin) contiene gli otto pin necessari all'accensione dei segmenti, più due pin per VDD e GND; per entrambi i gruppi è presente sulla board la legenda della piedinatura corretta. Si noti che il pin T0 comanda il transistore T0, il pin T1 comanda il transistore T1 e così via (anche i transistori presentano una legenda indicata sulla board); i pin che comandano i transistori sono collegati alle basi di questi ultimi tramite 4 resistori da 1 K Ω (indicati in marrone in Figura 5). A loro volta i **transistori** (indicati in viola in Figura 5 mostrandone anche la piedinatura) accendono i display corrispondenti (ogni transistore è posto sopra il relativo display). La tensione di comando alla base dei transistori deve essere al valore logico basso per determinare la loro accensione, poiché si tratta di transistori PNP. L'emettitore di tutti i transistori è posto alla tensione VDD, prelevata dal gruppo di pin a destra tramite opportuna pista (sul bottom layer), mentre ogni collettore è collegato all'anodo comune del relativo display.

I **display 7-segimenti** (indicati in arancione in Figura 5) presentano 2 pin di anodo comune (i due centrali indicati in verde in Figura 5) ma viene collegato ai transistori solamente quello posto sul lato superiore di ciascun display. Inoltre i display non sono direttamente saldati su board, ma posti su uno zoccolo in modo da rendere facile la sostituzione di uno di essi qualora non dovesse più funzionare.

Gli otto **resistori** (indicati in azzurro in Figura 5) da 470 Ω servono a creare la giusta differenza di potenziale per la corretta accensione dei segmenti (LED rossi). Ciascun resistore è collegato, quindi, da una parte a un pin di ingresso della board (del gruppo di pin a destra), e dall'altra al segmento corrispondente dei quattro i display (le lettere nere in Figura 5 aiutano a localizzare le corrispondenze fra resistenze e segmenti); per creare quest'ultimo collegamento sono state usate linee orizzontali sul top layer, al fine di creare un "ponte" di contatto fra i quattro segmenti uguali, e linee verticali sul bottom layer (indicate in rosso in Figura 6) per collegare i "ponti" del top (con opportuni via) ai pin corrispondenti dei display; si noti che per questioni di spazio è stato necessario inserire "ponti" orizzontali anche sotto i display sul top layer; si può avere facilmente accesso a queste piste estraendo i display dallo zoccolo. La maggior parte dei piedini dei resistori sono saldati su piste del bottom layer ma alcuni di essi hanno uno dei due terminali saldati su una pista in top layer (resistori C, D, E, F).

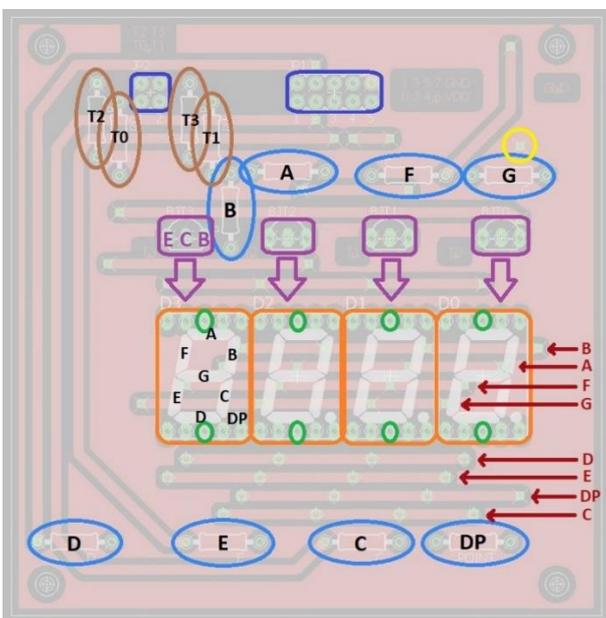


Figura 5. Top Layer

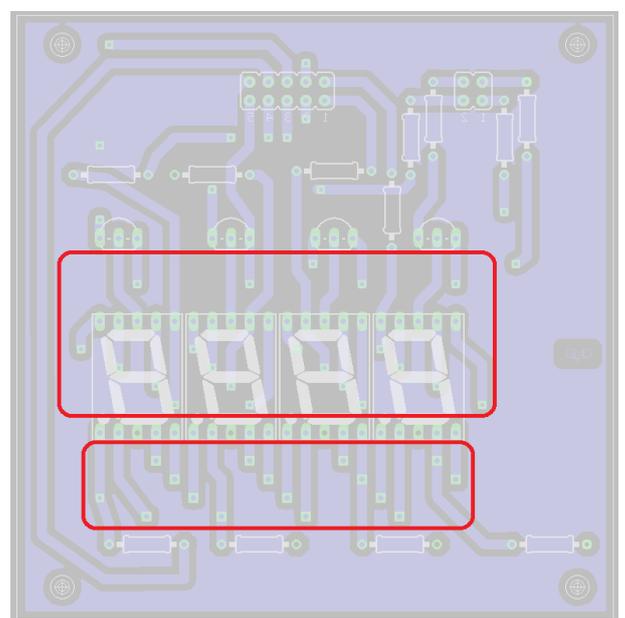


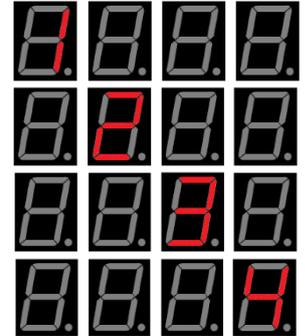
Figura 6. Bottom Layer (Mirrored)

3. COME FUNZIONA

La **logica di funzionamento** della board ha lo scopo di mostrare un'applicazione di MULTIPLEXING con un microcontrollore: essa infatti funziona tramite l'accensione alternata dei 4 display, uno dopo l'altro; i display quindi non saranno mai accesi realmente tutti contemporaneamente, ma il switch fra un display e un altro viene effettuato così velocemente (conviene mantenersi sulle unità di ms) da mostrare il numero come se esso fosse fisso sui display. In questo modo si riesce a risparmiare parecchio in termini del numero di pin utilizzati, visto che adesso non è più necessario dedicare un solo pin ad ogni segmento di ogni display, ma viene utilizzato un solo pin per tutti i segmenti A, uno per tutti i segmenti B, e così via, accendendo tramite i BJT ora uno ora l'altro display.

Se si vuole quindi mostrare il numero 1234 bisogna:

- Spegner tutti i display e accendere per un breve tempo solo il primo display e i segmenti che formano il numero 1;
- spegnere il primo display e accendere per un breve tempo solo il secondo display e i segmenti che formano il numero 2;
- spegnere il secondo display e accendere per un breve tempo solo il terzo display e i segmenti che formano il numero 3;
- spegnere il terzo display e accendere per un breve tempo solo il quarto display e i segmenti che formano il numero 4;
- creare un ciclo che contenga i precedenti punti in modo che il numero scelto resti visibile per il tempo desiderato (il numero di volte in cui viene eseguito il ciclo determina il tempo di permanenza del numero sui display);



4. IL PROGRAMMA PIU' IN DETTAGLIO

Il programma proposto al paragrafo 1 mostra un conteggio da 0 a 9999 sui display. Esso consta di un **main** molto semplice: viene chiamata dapprima, una sola volta, la funzione che ha il compito di configurare i pin di uscita del microcontrollore e poi viene effettuato un ciclo da 0 a 9999 per creare un contatore che scriva passo dopo passo il numero sui quattro display tramite la funzione Write7seg.

Le funzioni necessarie per il corretto funzionamento quindi sono tre: Config7seg, Write7seg, conversioneD.

La funzione **“Config7seg”** ha il compito di configurare i pin di output per la scrittura sui display; nell'esempio sono stati usati i primi quattro pin della PORTA per il controllo dei BJT, e tutta la PORTC per gli 8 segmenti. Sono state definite delle macro per semplificare l'identificazione delle varie uscite e per semplificare un eventuale cambiamento delle porte di uscita.

La funzione principale che effettua la scrittura è **“Write7seg”**, la quale accetta due parametri: il numero passato come primo argomento (“j”) è il numero che deve essere visualizzato sui display, il numero passato come secondo argomento (“velocita”) invece decide per quanto tempo il numero sta fisso sui display, tanto minore è questo numero, quanto minore è il tempo di permanenza di j sui display. Il numero j viene suddiviso nelle sue quattro cifre costitutive (“unita”, “decina”, “centinaia”, “migliaia”) tramite l'algoritmo mostrato nelle prime righe; si effettua poi un ciclo (con tante iterazioni quanto è “velocita”) in cui si accende ora un display, ora l'altro, scrivendo su ogni display la cifra a lui pertinente.

La funzione **“conversioneD”** si occupa di accendere i segmenti corretti di ogni display in relazione alla cifra che deve essere scritta su di esso; per esempio, secondo la convenzione dei display 7 segmenti, per mostrare il numero 6 bisogna accendere i segmenti A, F, E, D, C, G, e quindi nella PORTC vengono posti a 0 i pin corrispondenti ai segmenti da accendere (si ricorda che i display sono ad anodo comune e quindi per accendere i segmenti è necessario portare a 0 i pin corrispondenti).