



gohan

DADO DIGITALE

18 September 2011

Introduzione

Quello a presentare è stato il **mio primo progetto** elettronico che, come appunto si può leggere dal titolo, consiste in un semplice dado digitale a **logica cablata**, che pur essendo un circuito semplice, come prima esperienza mi dato qualche soddisfazione ed ovviamente mi ha permesso di imparare.

Avendo ricevuto consigli sul forum, avevo promesso di presentarlo, ed anche se sono in ritardo di qualche mese, ho voluto mantenere la promessa.

Infine, ho avuto modo di osservare molti progetti simili al mio in rete.

Su questo portale non ne ho incontrato nessuno fino ad ora, così che ho pensato bene di fare concorrenza e mi è sembrata una buona idea rendere mostrarne il risultato.

Requisiti e schema a blocchi

I requisiti di funzionamento richiedevano che all'accensione del circuito venisse visualizzato un **1** tramite un display a 7 segmenti, ed il **numero casuale** venisse generato tenendo premuto un pulsante (lo switch di comando) durante un tempo indeterminato, perchè un push-button può rimanere premuto teoricamente durante un tempo che non è prestabilito a priori. Al rilascio di questo switch, il numero doveva essere visualizzato sempre sul display.

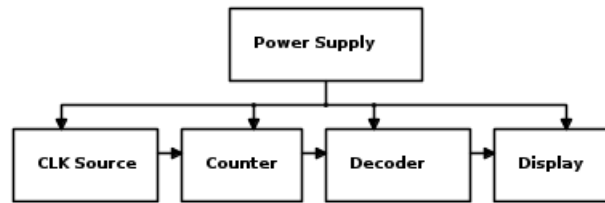
Il range di numeri disponibili era dall'**1** al **6**, come qualsiasi singolo dado da tavolo che si rispetti. Il problema doveva essere risolto usando componenti commerciali.

Ragionando sullo scopo del circuito, la prima funzione che ci viene in mente è quella di un contatore e la soluzione più versatile è quella di scegliere appunto un integrato contatore disponibile in commercio.

Lo stesso contatore deve essere preceduto da una fonte di clock, un circuito oscillatore che controlla la velocità di funzionamento di un dispositivo digitale quale può anche essere anche un microprocessore o microcontrollore.

Una volta scelta la fonte di clock, non rimarrebbe che ultimare il tutto con la logica digitale con cui, a seconda degli output del contatore si determinerebbe la visualizzazione dei numeri tramite il display a 7 segmenti. Si necessita quindi di un decoder a partire dal **BCD** per la visualizzazione sul display.

Avendo determinato la sequenza di funzionamento del circuito, come si fa in qualsiasi progetto, possiamo quindi cominciare a determinare la connessione delle singole funzioni rappresentando uno schema a blocchi

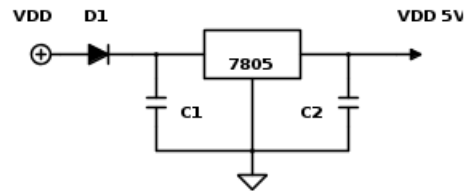


dove:

- **Power Supply:** rappresenta l'alimentazione del circuito, dato che ogni componente ad utilizzare ne avrà bisogno. La sua implementazione fisica è indiscutibile, ma a seconda della famiglia di componenti scelti dovremo alimentare il tutto ad una certa tensione che si è voluto mantenere il più precisa possibile. Secondo questo scopo è stata inclusa nello schema ed il motivo verrà spiegato ulteriormente più avanti.
- **CLK Source:** rappresenta la fonte di clock e indirettamente include il pulsante di controllo tramite il quale viene comandato il contatore tramite l'entrata CLK.
- **Counter:** rappresenta l'integrato contatore che appunto conta ad una determinata frequenza e riporta in uscita il valore binario del numero generato durante il conteggio (in questo caso crescente) in un range dall' 1 a 6.
- **Decoder:** rappresenta l'integrato in grado di decodificare la cifra corrispondente al valore BCD di input per la visualizzazione sul display in output.
- **Display:** rappresenta appunto il componente elettronico che dispone di una determinata piedinatura. Ciascun pin corrisponde ad un proprio segmento, e ciascuno di questi è formato al suo interno da un diodo LED. La scelta di questo componente si relaziona direttamente all'integrato che lo precede nel funzionamento del circuito.

Analisi dei singoli blocchi funzionali e schema completo del circuito

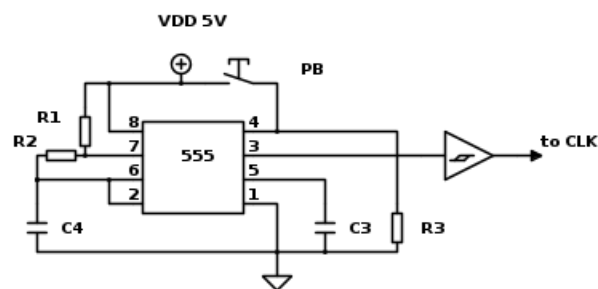
Power Supply



D1 è un **1N4007** e serve come protezione nel caso in cui accidentalmente venga invertita la polarità di alimentazione. Il regolatore di tensione **7805** serve per mantenere la tensione di alimentazione della famiglia **74HC** a $\approx 5\text{ V}$, ed accetta una tensione di input $V_{InMax} \leq \approx 35\text{ V}$, quindi, considerando la V_{fD1} e la V_{Drop} dell'integrato, una tensione di alimentazione minima di $\approx 9\text{ V}$ è sufficiente. Il circuito non consumerà una quantità di corrente notevole tale da dover necessitare di un dissipatore, quindi in questo caso non è necessario. I valori di **C1** & **C2** sono riportati nel datasheet del componente (link sopra); $C1 = 0.33\text{ }\mu\text{F}$ e $C2 = 0.1\text{ }\mu\text{F}$. Io li ho messi tutti e due da $0.1\text{ }\mu\text{F}$, non si sono verificati problemi, ma seguire il datasheet è sempre meglio.

CLK Source

Come fonte di clock è stata scelta una semplice configurazione astabile con l'integrato **555**.



Per il pin **5:Control Voltage**, è stato messo un condensatore con valore di default $0.01\text{ }\mu\text{F}$ per quando il pin non viene utilizzato.

Come al solito, per questa configurazione la frequenza di clock si calcola secondo la formula

$$Freq = \frac{1}{\ln(2)(R1 + 2R2)C4}$$

ed io avevo scelto una frequenza di clock di ≈ 22 kHz.

Per ricavare i valori dei componenti passivi ho prefissato un valore per il condensatore di $0.01 \mu F$, in maniera da risolvere l'equazione calcolando i valori per le resistenze. Detto questo si stabilisce che $R1 + 2R2 = x$, quindi

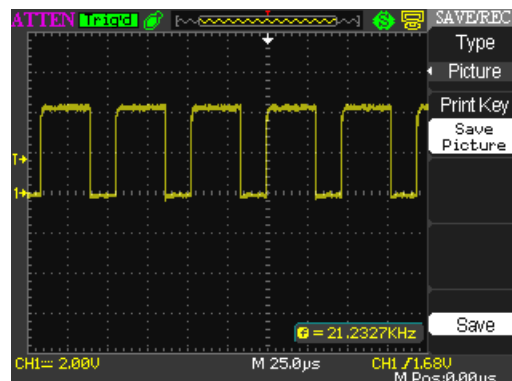
$$22 \text{ kHz} = \frac{1}{\ln(2)C4x}$$

sostituendo si ha che

$$x = \frac{1}{\ln(2)22\text{kHz} \cdot C4} \Rightarrow x = 6557.7 = R1 + 2R2$$

Notiamo subito che tra 6557.7 è un numero di poco più piccolo di 6600, che diviso per tre risulta 2200.

Quindi, se $R1 + 2R2 \approx 6600$ vuol dire che possiamo scegliere due resistenze uguali da 2k2 per avvicinarci il più possibile al valore di frequenza desiderato, infatti risolvendo l'equazione per $R1 = R2 = 2k2$ e $C4 = 0.01 \mu F$ la frequenza teorica (senza tener conto delle tolleranze dei componenti) risulta di $\approx 21.86\text{kHz}$, il quale è un risultato soddisfacente. A seguire possiamo osservare invece l'onda quadrata e la frequenza misurata tramite l'oscilloscopio



Clock

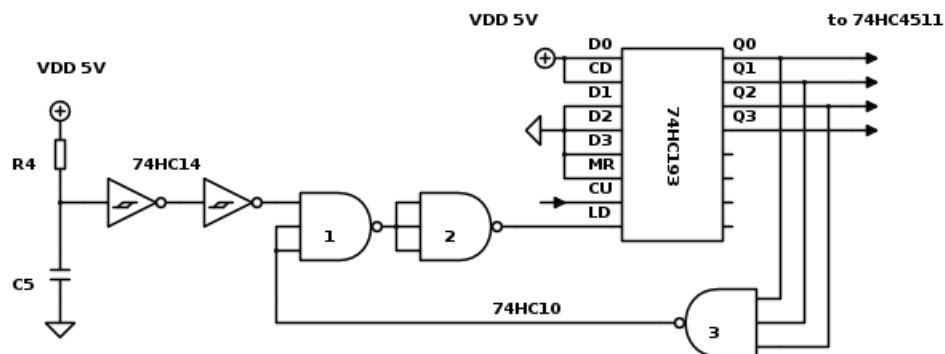
Come avevamo preannunciato, necessitavamo di un pulsante di comando per far sì che venisse generato il numero casuale. Bene, in questo caso la soluzione migliore e più sicura è quella di lasciare l'output dell'oscillatore direttamente collegato all'entrata di clock del contatore, e collegare una resistenza di pull-down (nello schema è $R3 = 1k$) all'entrata di Reset del 555, la quale rimane attiva a stato logico basso. Quindi è facile immaginare che per tutto il tempo che il push-button **PB** non

verrà premuto, il 555 si troverà inattivo e non avremo nessuna onda quadrata in output, lasciando così il contatore in riposo.

Nello schema ho aggiunto il simbolo di un buffer **Schmitt Trigger** (non nel circuito reale) perchè è bene alimentare **qualsiasi** entrata di clock con un segnale ad isteresi, in quanto questo tipo di entrate necessitano dei fronti rapidi per funzionare correttamente. Il 555 ha comunque un output digitale come si può vedere dall'immagine sopra.

Counter

Il contatore è il cuore del circuito. La scelta di un componente adatto caratterizzato dalle sue caratteristiche è ovviamente un dettaglio decisivo per il corretto funzionamento del piccolo circuito. Ricordo che all'accensione, il display avrebbe dovuto visualizzare un **1**, quindi l'output del contatore avrebbe dovuto essere l'equivalente binario **0001**, sempre. Ho così deciso di scegliere un contatore sincrono programmabile ma con LOAD e MR asincrono, precisamente un **74HC193**. In questo modo possiamo programmarne gli input, agire sul LOAD all'accensione, e quando il contatore raggiunga la sequenza di output **0111** riattivare il LOAD riportando le uscite a **0001** immediatamente. Dal datasheet possiamo vedere che questa entrata è attiva a stato logico basso quindi all'accensione bisogna ottenere una transazione positiva (da **0** a **1**) in meno di un secondo. Questo è risolvibile a partire da una rete RC. Gli output del contatore che ci interessano sono i primi tre, e pensando al fatto di rimandare a zero l'output di LOAD (ricordando che è asincrono) quando le prime tre uscite saranno a **1** si è pensato subito ad una NAND a tre input, e l'integrato che ci interessa è il **74HC10**. Il LOAD però necessita al suo ingresso una porta AND con gli ingressi di questa collegati alla rete RC ed alla NAND. Possiamo usare le porte rimanenti del **74HC10**, dato che una AND la possiamo implementare a partire da due NAND in serie. Perfetto, la implementazione del contatore risulta quindi la seguente



Al collegare l'alimentazione, non si sà mai quale possa essere l'output di un contatore, quindi di solito è sempre bene dare un reset iniziale oppure caricare il

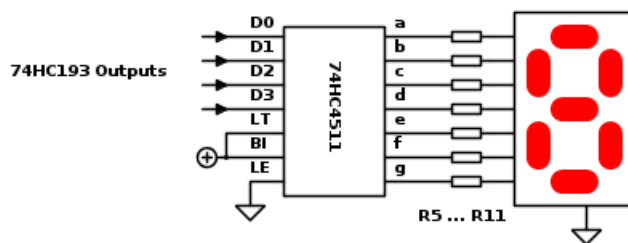
LOAD come nel nostro caso. Nel circuito la NAND 1 & 2 formano una AND (che chiamiamo **AND12**) che riceve in ingresso l'output della NAND 3 ed il segnale di output dal buffer **Schmitt Trigger** (due NOT in serie da un [74HC14](#)).

Grazie alla rete RC ($R4 = 1k$ e $C5 = 1\mu F$) otteniamo la transazione positiva caricando il condensatore al 63.2% in $\tau = R4C5 = 1ms$, anche se il valore di tensione necessario per far sì che cambi lo stato logico dipende dalla soglia dell'invertitore NOT **Schmitt Trigger**. Secondo la logica quindi il LOAD verrà comunque caricato essendo che la AND in qualsiasi caso ha inizialmente in ingresso un segnale logico basso. Tuttavia il segnale in arrivo dalla rete RC non è ad isteresi, per tanto è bene anteporre il buffer **Schmitt Trigger** per rispettare i tempi e la logica del segnale per gli integrati con cui si lavora.

La entrata parallela di preselezione viene settata lasciando solo **D0** collegato a V_{DD} mentre i restanti 3 pin **D1..D3** al potenziale di riferimento. Pure **CD (Count Down Clock Input)** deve essere collegato assieme a **D0** altrimenti il contatore non potrebbe contare (basta osservare la tabella di funzionamento dal datasheet). **CU (Count Up Clock Input)** invece va collegato direttamente all'output del 555. Essendo che il LOAD è asincrono, quando la NAND avrà tutte le entrate a **1**, tramite la AND12 si carica il LOAD.

BCD to 7 Seg & Display

Questi ultimi due blocchi funzionali li descrivo assieme dato che non si tratta solo di collegare le uscite del contatore alle entrate del decodificatore ed i pin del display alle corrispondenti uscite dell'integrato che si usa. Nel circuito ho usato un [74HC4511](#) ed un display a 7 segmenti a catodo comune ([link datasheet](#)) dato che le uscite del decodificatore BCD sono a stato logico alto. Diversamente (nel caso in cui fossero negate), avremmo dovuto scegliere un display ad anodo comune.

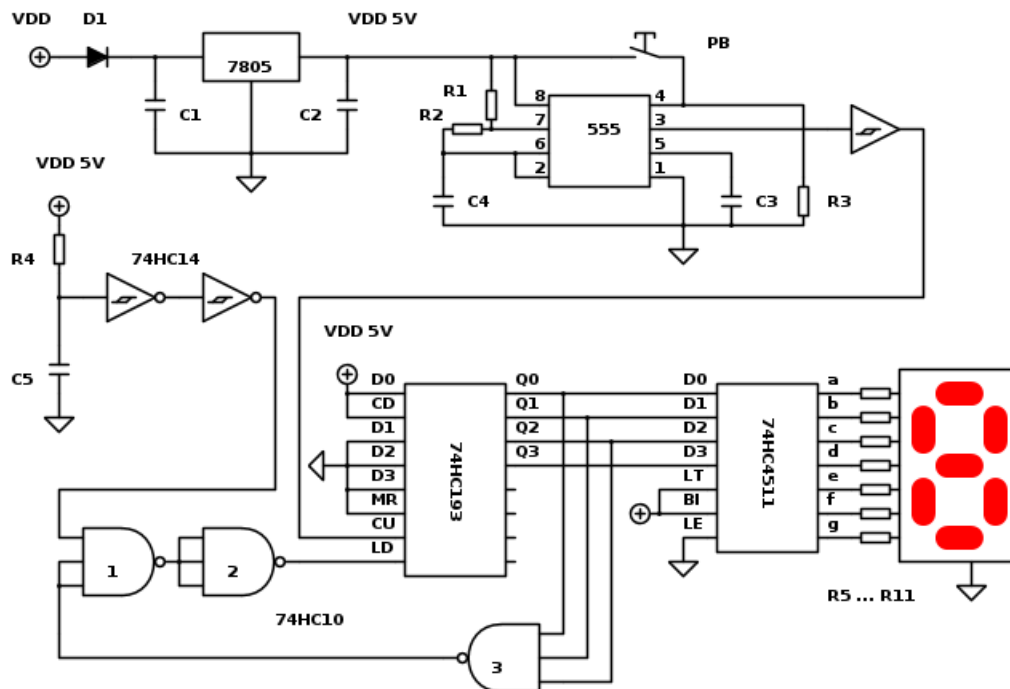


Dato che il display è internamente composto da 7 LED (più il punto decimale che non è stato usato), la resistenza per ogni pin (nel circuito da **R5** a **R11**) si calcola sempre con formula di alimentazione di corrente del LED. Io ho deciso di alimentarli con $I_F \approx 7\text{ mA}$, quindi, sapendo che $V_F \approx 1.8\text{ V}$ e $V_{Out} \approx 5\text{ V}$

$$R_5 \dots R_{11} = \frac{V_{Out} - V_F}{I_F} = 457.14 \Omega \Rightarrow 470 \Omega \text{ valore commerciale.}$$

Schema completo

Ricapitolando, ecco la rappresentazione completa dello schema.

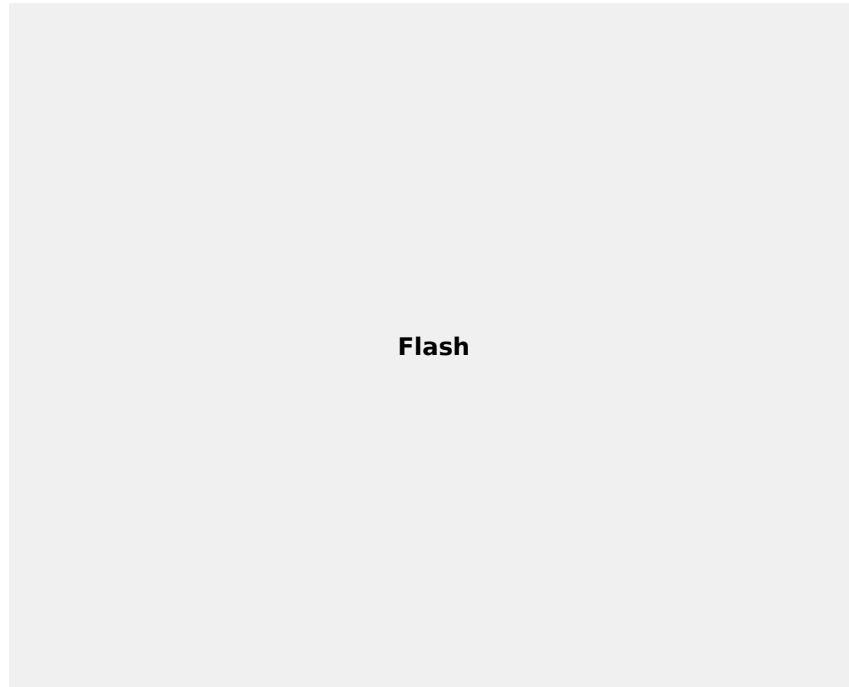


Valori componenti passivi

- $R_1 = R_2 = 2k\Omega$
- $R_3 = R_4 = 1k\Omega$
- $R_5 \dots R_{11} = 470 \Omega$
- $C_1 = C_2 = 0.1 \mu F$
- $C_3 = C_4 = 0.01 \mu F$
- $C_5 = 1 \mu F$

Avrei potuto implementare l'oscillatore che le porte NOT, [qui](#) vi è presente qualche esempio. È un dettaglio che mi è sfuggito al progettare il circuito, pero volevo renderlo noto e segnalare il link dato che può essere utile come riferimento.

Ecco quindi il circuito in azione.



La visualizzazione del numero **8** è dovuta alla frequenza troppo alta perchè l'occhio umano possa percepire il cambio dei numeri.

Il dado con PIC

Dopo aver effettuato il progetto con integrati commerciali e porte logiche, posto il codice di un dado fatto con un PIC16F877A. Era un esercizio svolto in laboratorio, con il push-button collegato al PORTE:0, mentre il display era collegato al PORTC in maniera che **PORTC:0 = a**, **PORTC:1 = b**, **PORTC:2 = c** e così fino a **PORTC:6 = g**.

```
; Name:Dice  
; Pic:PIC16F877A  
; Author:gohan
```

```
LIST p=16f877a  
INCLUDE p16F877A.INC
```

```
__CONFIG __CP_OFF&__DEBUG_OFF&__WRT_OFF&__CPD_OFF&__LVP_OFF  
&__BODEN_ON&__PWRTE_ON&__WDT_OFF&__HS_OSC
```

```
CBLOCK 0x20  
    i
```

ENDC

```

    ORG 0x00

    ;step to bank 1
    bsf STATUS, RP0    ;set RP0
    bcf STATUS, RP1    ;clear RP1

    movlw 0x06
    movwf ADCON1        ;set PORTE as digital

    movlw 0xff
    movwf TRISE        ;set PORTC as input

    movlw 0x00
    movwf TRISC        ;set PORTD as output

    ;return to bank 0
    bcf STATUS, RP0    ;clear RP0
    bcf STATUS, RP1    ;clear RP1

start:
    movlw 0x06        ;load 7Seg value for n° 1
    movwf PORTC        ;display it

    btfss PORTE, 0    ;if push button is not pressed
    goto $-1          ;stay here
    goto continue     ;generate random number

continue:
    movlw 0x5b        ;load 7Seg value for n° 2
    movwf PORTC        ;display it
    btfss PORTE, 0    ;if push button is not pressed
    goto $-1          ;stay here

    movlw 0x4f        ;load 7Seg value for n° 3
    movwf PORTC        ;display it
    btfss PORTE, 0    ;if push button is not pressed
    goto $-1          ;stay here

    movlw 0x66        ;load 7Seg value for n° 4
    movwf PORTC        ;display it
    btfss PORTE, 0    ;if push button is not pressed
    goto $-1          ;stay here

```

```
movlw 0x6d          ;load 7Seg value for n° 5
movwf PORTC         ;display it
btfss PORTE, 0      ;if push button is not pressed
goto  $-1           ;stay here

movlw 0x7d          ;load 7Seg value for n° 6
movwf PORTC         ;display it
btfss PORTE, 0      ;if push button is not pressed
goto  $-1           ;stay here
goto  start         ;restart from 1

end
```

Conclusioni e ringraziamenti

Siccome questo fu il primo progetto in assoluto, ebbi anche io i miei dubbi che mi sono stati risolti grazie a utenti (spero si ricordino) nel forum. Spero di non essere stato troppo stressante in quei giorni, visto che a tutti i costi volevo complicarmi la vita con "porcherie" fatte a transistori. Volevo quindi ringraziare [g.schgor](#) per l'assistenza ed i consigli che mi ha dato a suo tempo. Altrettanto [IsidoroKZ](#), che aveva anche fatto uno schema con la serie 4000 che poi non ho montato (ma gli integrati sono nel cassetto), però si ringrazia tanto per questo come per i suggerimenti importanti riguardo all'utilità delle porte Schmitt Trigger ed alle "porcherie" assolutamente da evitare. Come è giusto che sia, ho imparato anche da questo. Anche [carlo](#) per le sue precisazioni sulla logica CMOS e sul fatto di non lasciare gli ingressi a vuoto. Se ne ho dimenticato qualcuno scusate, ma sappiate che non dimentico :) Grazie a tutti loro, ma anche a tutti gli utenti che leggeranno l'articolo. Spero vi sia piaciuto.

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Gohan:dado-digitale>"