



Roberto Berardi (Robert8)

ELETRONICA DIGITALE PRATICA IV - LOGIC ANALYZER & CLOCK

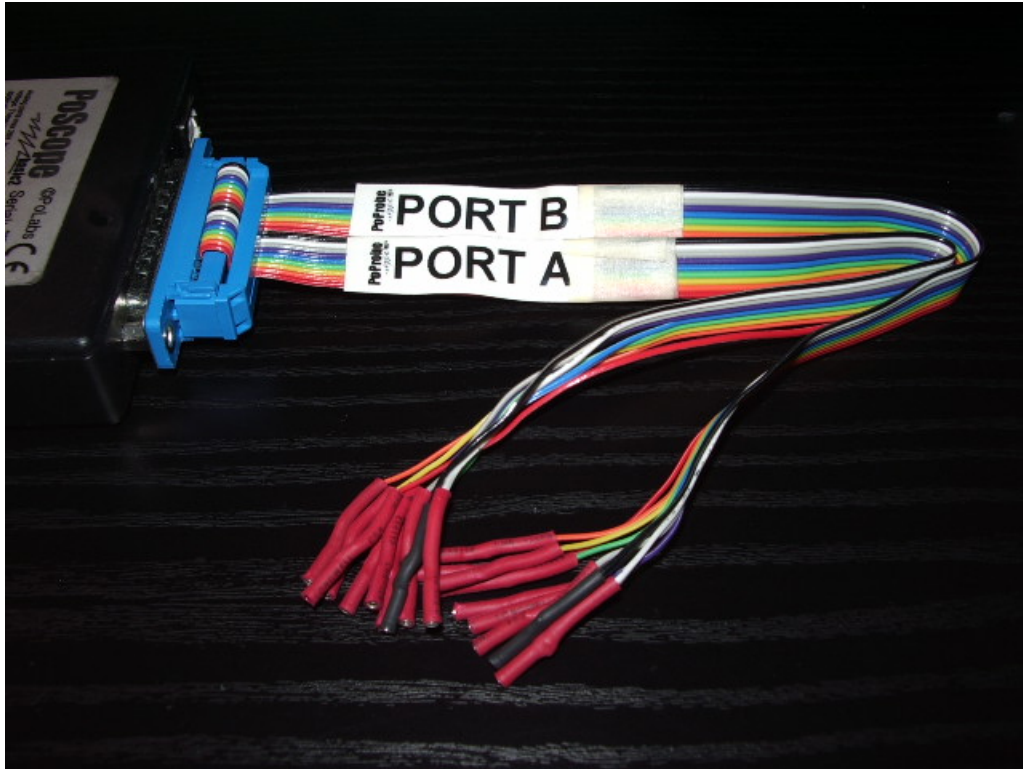
18 May 2012

Introduzione

Nella [scorsa puntata](#) abbiamo affrontato il tema della gestione dell'input e del controllo dell'output negli esperimenti con dispositivi digitali, predisponendo un semplice Logic Switch a controllo **manuale** per gli ingressi ed un Logic Indicator a LED per le uscite. Oggi esaminiamo un'alternativa più sofisticata per la gestione **automatica** sia degli ingressi sia delle uscite. Useremo un analizzatore logico presente come funzione aggiuntiva nell'oscilloscopio USB presentato nel [primo articolo](#): il PoScope Basic2. Grazie a questo economico strumento potremo provare con un solo click, in sequenza, tutte le combinazioni degli ingressi e vedere a video su PC lo stato dell'uscita. Non solo, mentre a mano le prove erano necessariamente statiche, qui potremo testare diverse frequenze partendo da 1 kHz fino ad 1 MHz. Grazie a questo strumento potremo affrontare, nella seconda parte dell'articolo, il tema del **clock** che, d'ora in poi, sarà il **cuore pulsante** dei nostri sistemi digitali. Per la generazione di un segnale di clock il Forum è pieno di articoli e discussioni al riguardo per cui mi limiterò a sperimentare un multivibratore astabile con porte NOT. Cominciamo, quindi, col Logic Analyzer.

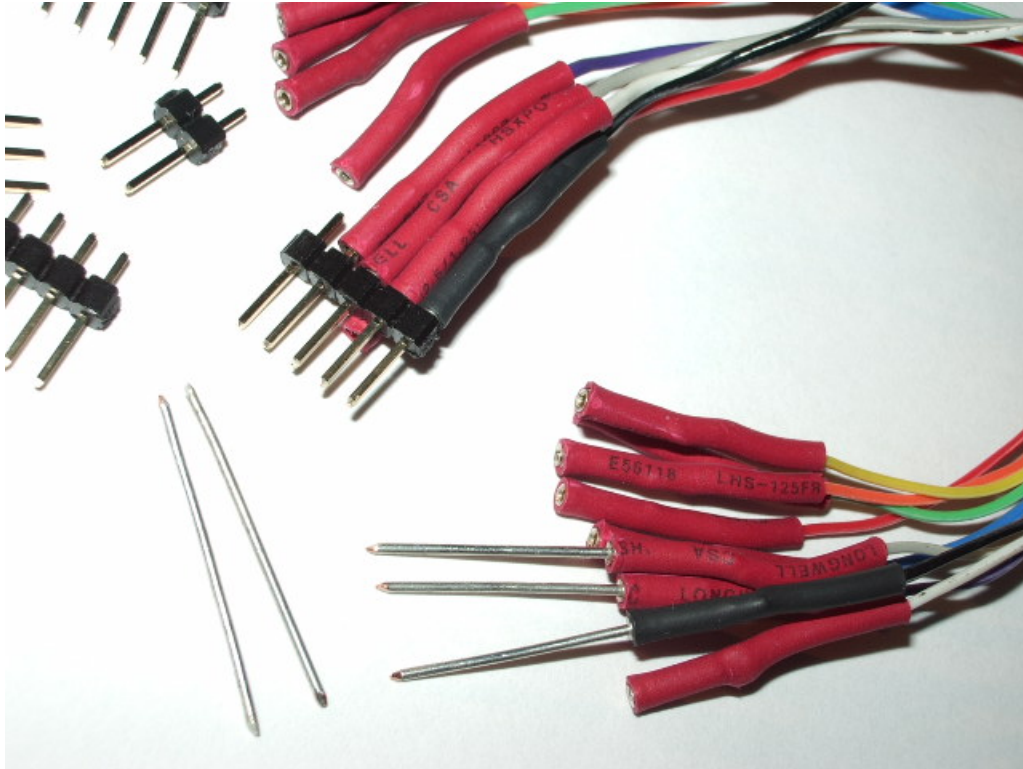
Logic Analyzer

Riporto una foto del PoScope per evidenziarne i connettori utilizzati per l'input/output:



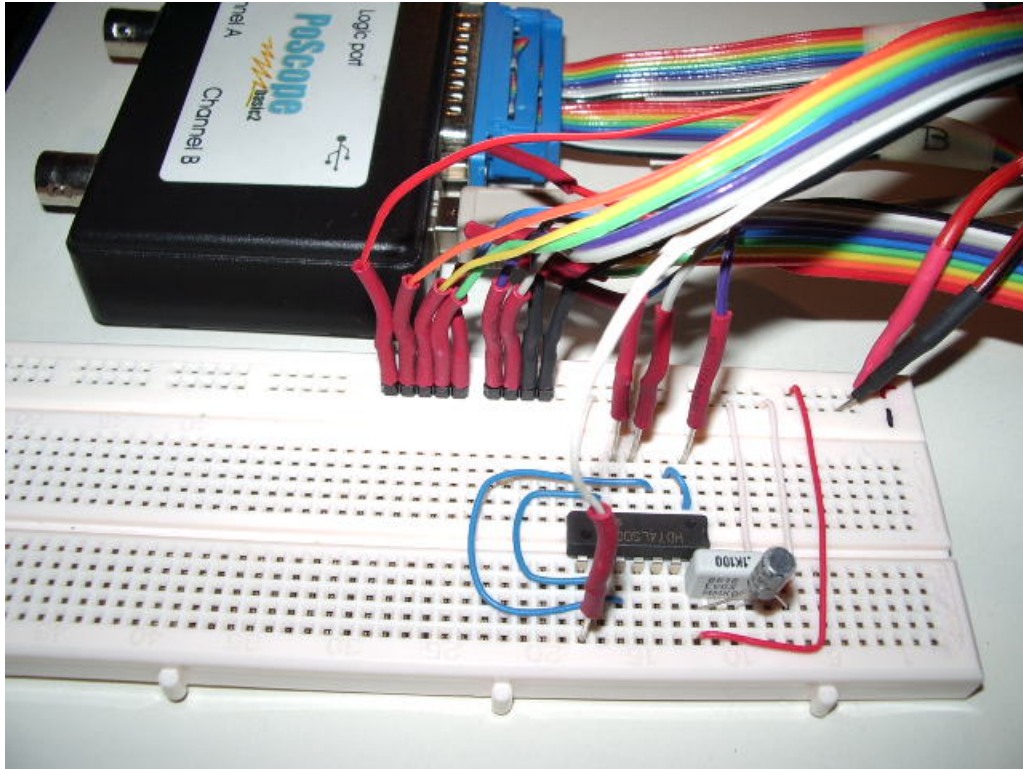
Connettore Logic Analyzer

Notiamo la presenza di 2 canali A e B con 9 cavetti ognuno: 8 di segnale ed 1 di massa. Il canale A viene usato per la rilevazione delle uscite mentre il canale B è usato per generare gli ingressi. Per innestare le boccole sulla BreadBoard occorrono degli spinotti oppure degli spezzoni di filo rigido:



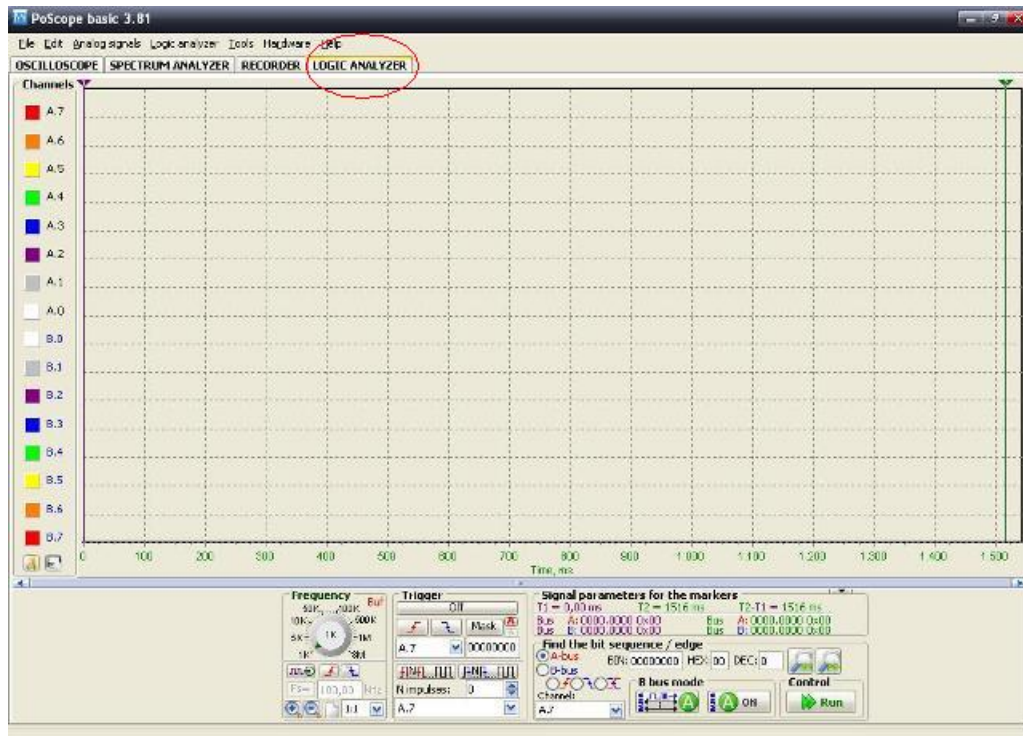
Dettaglio Connettori

Allora sperimentiamo il Logic Analyzer sulla BreadBoard di test descritta nel [secondo articolo](#) con lo scopo di generare tutti gli 8 ingressi possibili e controllare a video lo stato dell'uscita. Appliciamo i primi 3 cavetti del canale B sugli ingressi ed il primo cavetto del canale A sull'uscita come mostrato di seguito:



Test Circuito Ventola

I restanti cavetti del canale B non utilizzati li possiamo lasciare sconnessi, mentre conviene porre a massa quelli che avanzano del canale A per evitare false tracce sul video. Prima di dare corrente alla BreadBoard, prepariamo il generatore di segnale del PoScope: connettiamo il PoScope al PC con l'apposito cavo USB e lanciamo il suo software che apparirà così:



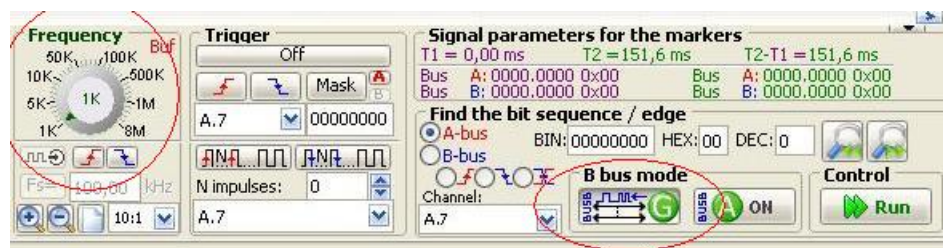
Schermata Logic Analyzer

Dove abbiamo selezionato la quarta funzione disponibile, appunto Logic Analyzer:



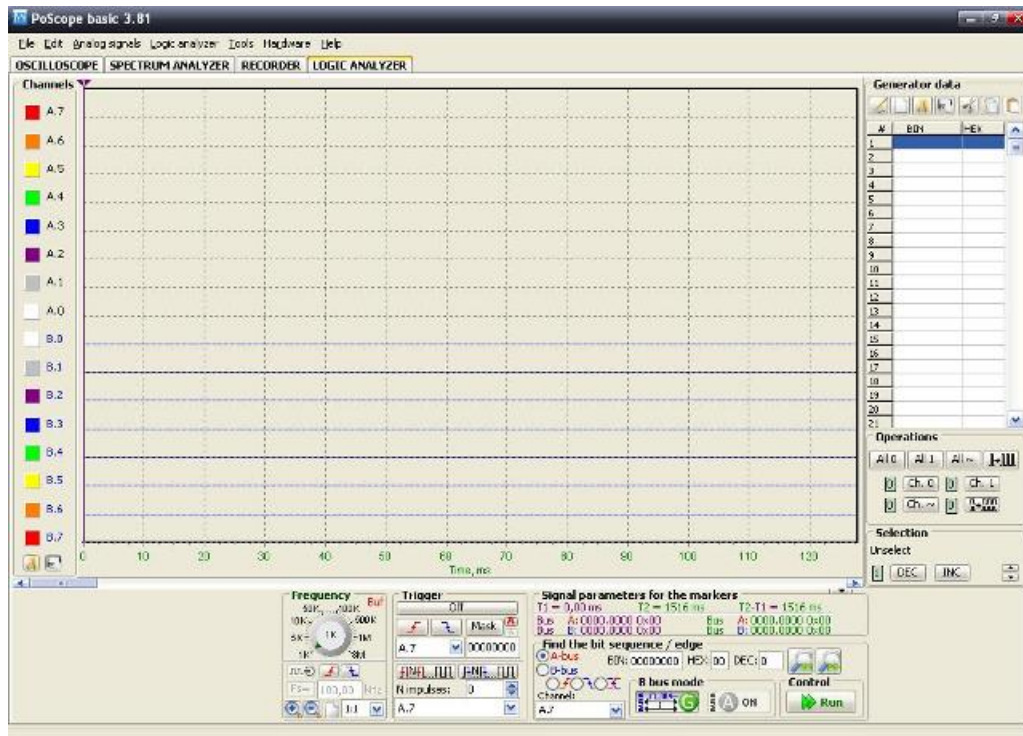
Dettaglio Menù

In basso, al centro, vi è il pulsante di selezione del canale B impostato per default come ingresso, selezioniamolo come generatore ed impostiamo la frequenza su 1 kHz come qui evidenziato:



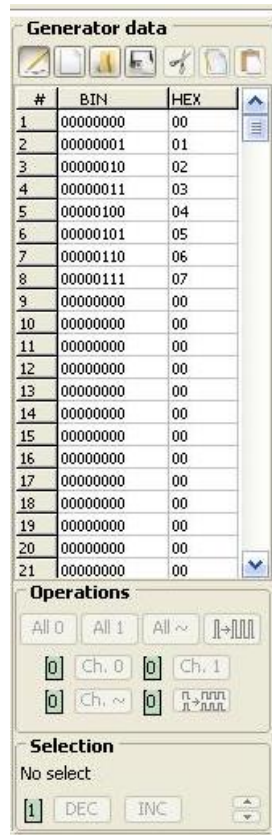
Dettaglio Pannello Controlli

A questo punto apparirà la finestra del generatore a destra:



Schermata Generatore Segnali

Dove possiamo inserire i valori che il generatore emetterà in sequenza, nel nostro caso da 0 a 7:



#	BIN	HEX
1	00000000	00
2	00000001	01
3	00000010	02
4	00000011	03
5	00000100	04
6	00000101	05
7	00000110	06
8	00000111	07
9	00000000	00
10	00000000	00
11	00000000	00
12	00000000	00
13	00000000	00
14	00000000	00
15	00000000	00
16	00000000	00
17	00000000	00
18	00000000	00
19	00000000	00
20	00000000	00
21	00000000	00

Operations

All 0 All 1 All ~ 

 Ch. 0  Ch. 1

 Ch. ~  n, nnn

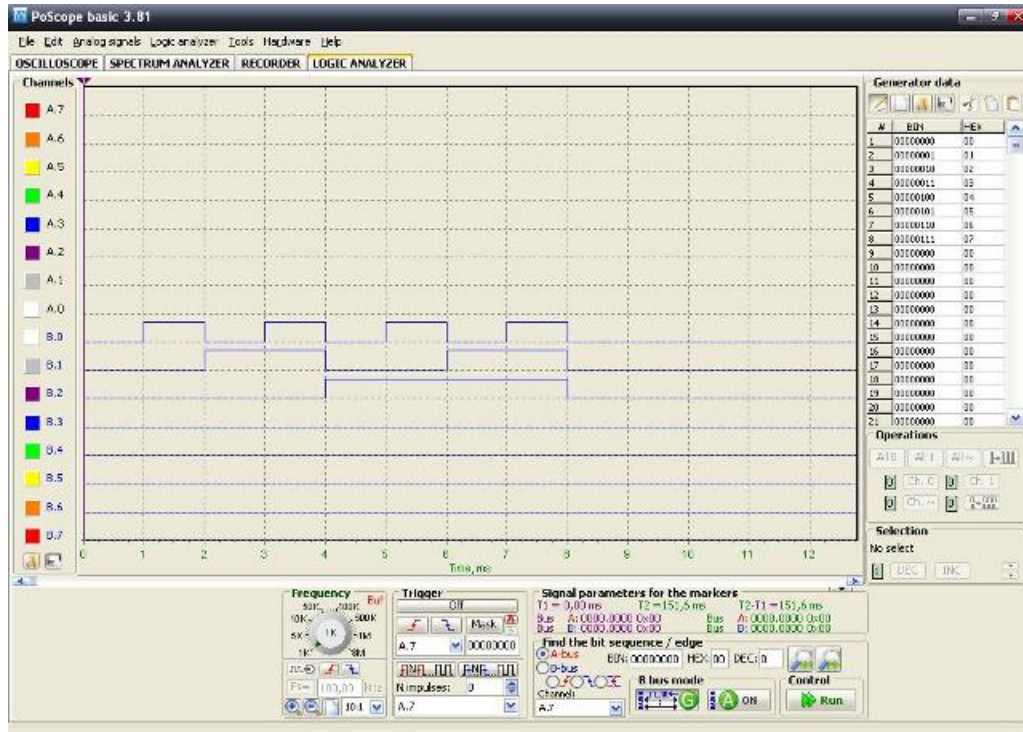
Selection

No select

 DEC INC  

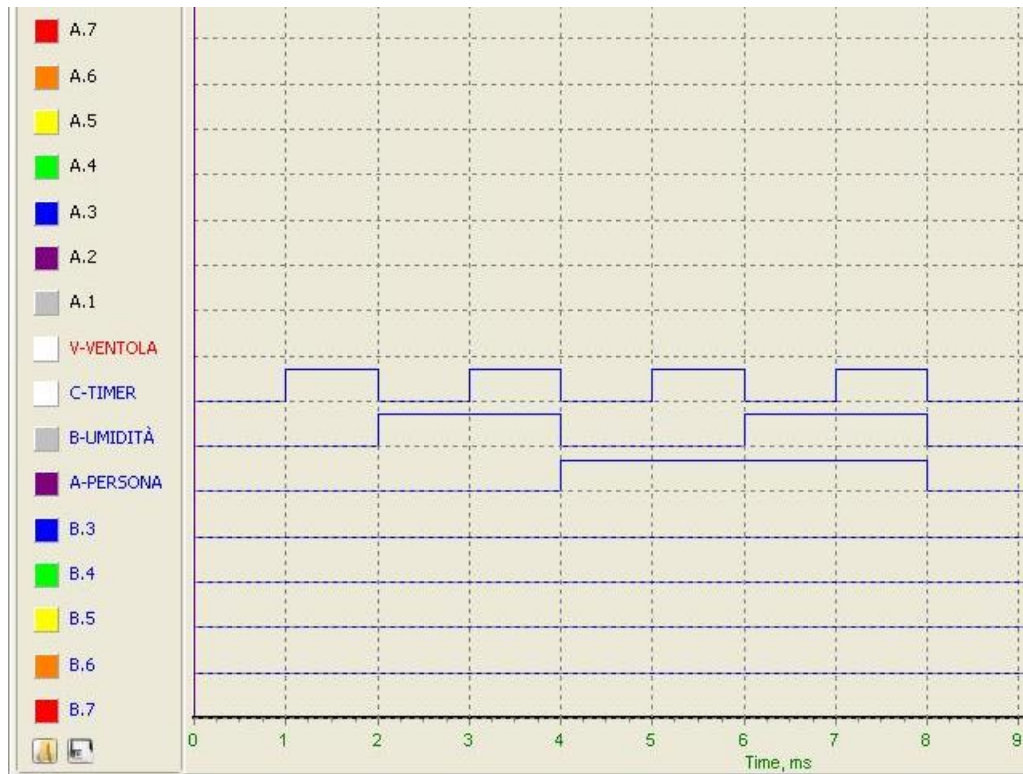
Dettaglio Generator Data

Ecco che sulla schermata appaiono i 3 segnali su B0, B1 e B2 in blu:



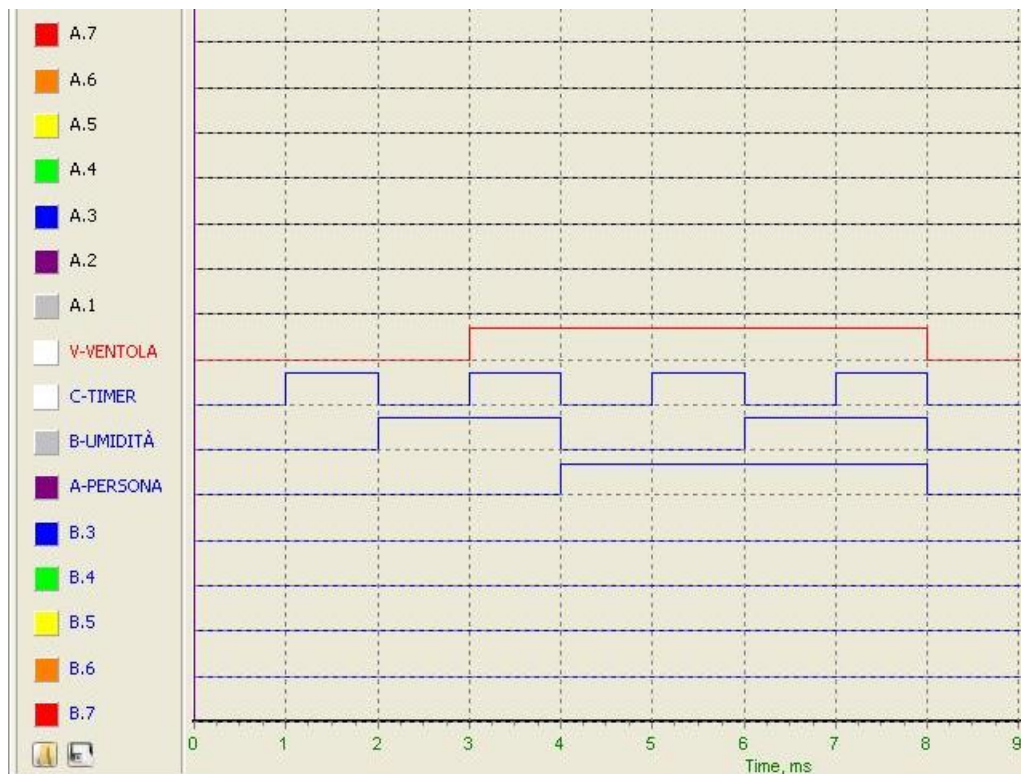
Segnali di ingresso

Per comodità possiamo ridenominare gli anonimi B0,1 e 2 in base alle nostre esigenze dove, ricordiamo, i tre ingressi erano A (presenza di una persona), B (sensore umidità) e C (timer). Invece l'uscita, dove abbiamo posto il cavetto A0, rappresenta la Ventola:



Assegnazione Nomi significativi

Bene! Ci Siamo! Tutto è pronto, diamo corrente alla basetta e pigiamo il pulsante RUN:



RUN

Appare, in rosso, la traccia dell'uscita esattamente come previsto.

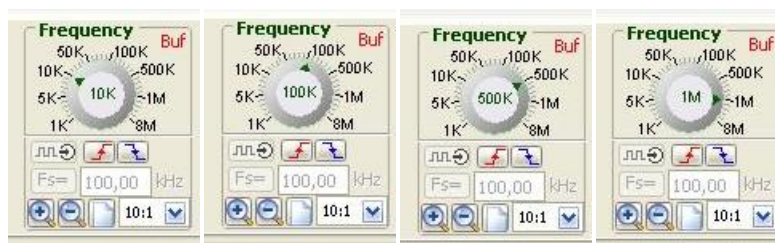
Possiamo provare altre frequenze, semplicemente agendo col mouse sulla manopola, fino a 1 MHz (limite massimo per l'uso contemporaneo di 2 canali):

10 kHz

100 kHz

500 kHz

1 MHz



Con ciò abbiamo concluso la breve panoramica sul Logic Analyzer usato come strumento di test per generare tutti i possibili ingressi al nostro sistema digitale e controllarne l'uscita. Passiamo al clock.

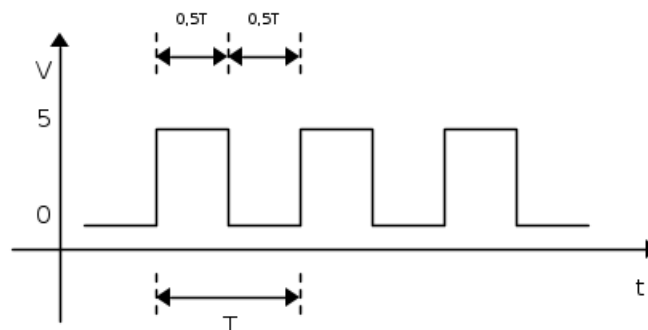
IL CLOCK

Finora abbiamo trattato dispositivi digitali funzionanti in **logica combinatoria** ovvero basati sul concetto che l'uscita dal dispositivo dipende unicamente dai valori degli ingressi. Infatti nel dispositivo [visto](#) sul controllo della ventola questa si accende o spegne in base solo allo stato in cui si trovano gli ingressi provenienti dai tre sensori, o meglio, in base ad un'opportuna combinazione degli ingressi: da qui, appunto, il nome di rete combinatoria.

Quando, invece, l'uscita del dispositivo, oltre a dipendere dagli ingressi attuali, dipende anche dallo stato precedente dell'uscita stessa, allora si parla di **logica sequenziale** perché assume rilevanza, appunto, la sequenza con cui si succedono ingressi e uscite. Un esempio lampante di logica sequenziale è il contatore dove la rete sequenziale con cui è costituito deve valorizzare le uscite successive incrementando il valore delle uscite precedenti che, evidentemente, devono essere state memorizzate per il successivo riutilizzo.

Esistono due tipologie di dispositivi sequenziali: asincroni e sincroni. Si differenziano per come viene gestita la successione temporale degli eventi. Negli **asincroni** la durata, la permanenza, la stabilità degli ingressi, deve rispettare precise tempistiche che spesso vengono formalizzate in protocolli di comunicazione, per esempio nei dispositivi seriali. Nei **sincroni** la successione delle azioni è determinata da un segnale esterno, o comunque indipendente, che scandisce il tempo: il **clock**, il quale ne sincronizza tutte le fasi.

E il clock è appunto l'argomento dei successivi esperimenti che hanno lo scopo di testare diversi modi di ottenere il segnale di clock. Innanzitutto, però, dedichiamo qualche minuto alle caratteristiche del segnale di clock. Il clock è un segnale ad onda quadra o rettangolare la cui rappresentazione ideale è la seguente:



Dove l'ampiezza della tensione deve essere compatibile con i dispositivi che deve sincronizzare: in questo esempio essa varia da 0 a 5 V. L'oscillazione avviene in modo ciclico ogni T secondi e, quindi, T prende il nome di periodo. Per metà del periodo,

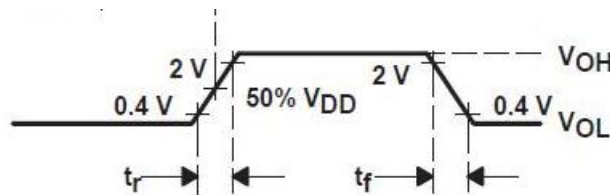
detto semi periodo, la tensione è a livello basso e per l'altro semi periodo è a livello alto.

Con un periodo di 1 secondo si avrà la ripetizione di 1 ciclo ogni secondo, quindi, con frequenza di 1 hertz (Hz). Se il periodo dura 1 ms la frequenza sarà:

$$f = \frac{1}{T} = \frac{1}{0,001} = 1000 \text{ Hz} = 1 \text{ kHz}$$

Naturalmente questo è l'andamento ideale al quale, i dispositivi che generano tale onda quadra, cercano di avvicinarsi divergendone per vari aspetti tra i quali elenchiamo i più significativi:

- **1** - il salto da 0 a 5 V e viceversa da 5 a 0 non potrà essere istantaneo, ma impiegherà almeno qualche frazione di nano secondo anche nei migliori generatori di clock; questa caratteristica viene specificata come **tempo di salita** e **tempo di discesa**, spesso citate nei datasheet con t_r e t_f e riferite a determinate soglie di tensione come nel seguente esempio di un generico generatore di clock a 3 V:



rise time & fall time

Sempre nel datasheet vi troviamo il range di valori tra i quali questi tempi di salita e discesa possono variare; in questo caso tra 0,5 e 2 nano secondi:

switching characteristics over recommended ranges of supply voltage and operating free-air temperature, $C_L = 25 \text{ pF}$, $V_{DD} = 3.3 \text{ V} \pm 0.3 \text{ V}$ (see Note 5)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{pd}	Propagation delay (normalized (see Figure 3))	CLKIN to Yn, f = 66 MHz to 200 MHz	~150	150	ps
$t_{sk(o)}$	Output skew (see Note 6)	Yn to Yn		150	ps
$t_{c(jit_cc)}$	Jitter (cycle to cycle) (see Figure 5)	f = 66 MHz to 200 MHz	70	150	ps
		f = 24 MHz to 50 MHz	200	400	
	Output duty cycle (see Figure 4)	f = 24 MHz to 200 MHz at 50% V_{DD}	45%	55%	
t_r	Rise time	$V_O = 0.4 \text{ V to } 2 \text{ V}$	0.5	2	ns
t_f	Fall time	$V_O = 2 \text{ V to } 0.4 \text{ V}$	0.5	2	ns

Specifiche t_r e t_f

- **2** - i due semi periodi non saranno perfettamente identici e simmetrici, ma potranno variare anche considerevolmente; questa caratteristica viene specificata col termine **Duty Cycle** che rappresenta il rapporto, in percentuale, tra la durata del semi periodo con tensione a livello alto e l'intero periodo. Un Duty Cycle al 50% sarebbe

perfettamente simmetrico. Per fortuna nella maggioranza dei casi una simmetria stretta non è necessaria e l'onda assume una forma rettangolare. Dal datasheet ricaviamo la variazione possibile del Duty Cycle in assoluto:

switching characteristics over recommended ranges of supply voltage and operating free-air temperature, $C_L = 25 \text{ pF}$, $V_{DD} = 3.3 \text{ V} \pm 0.3 \text{ V}$ (see Note 5)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{pd}	Propagation delay (normalized (see Figure 3))	CLIN to Y_n , $f = 66 \text{ MHz to } 200 \text{ MHz}$	-150	150	ps
$t_{sk(o)}$	Output skew (see Note 6)	Y_n to Y_n		150	ps
$t_{c(jit_cc)}$	Jitter (cycle to cycle) (see Figure 5)	$f = 66 \text{ MHz to } 200 \text{ MHz}$	70	150	ps
$t_{c(jit_cc)}$	Jitter (cycle to cycle) (see Figure 5)	$f = 24 \text{ MHz to } 50 \text{ MHz}$	200	400	ps
odc	Output duty cycle (see Figure 4)	$f = 24 \text{ MHz to } 200 \text{ MHz at } 50\% V_{DD}$	45%	55%	
t_r	Rise time	$V_O = 0.4 \text{ V to } 2 \text{ V}$	0.5	2	ns
t_f	Fall time	$V_O = 2 \text{ V to } 0.4 \text{ V}$	0.5	2	ns

Specifiche Duty Cycle

E qui la variazione del Duty Cycle in base alla frequenza di emissione del clock:

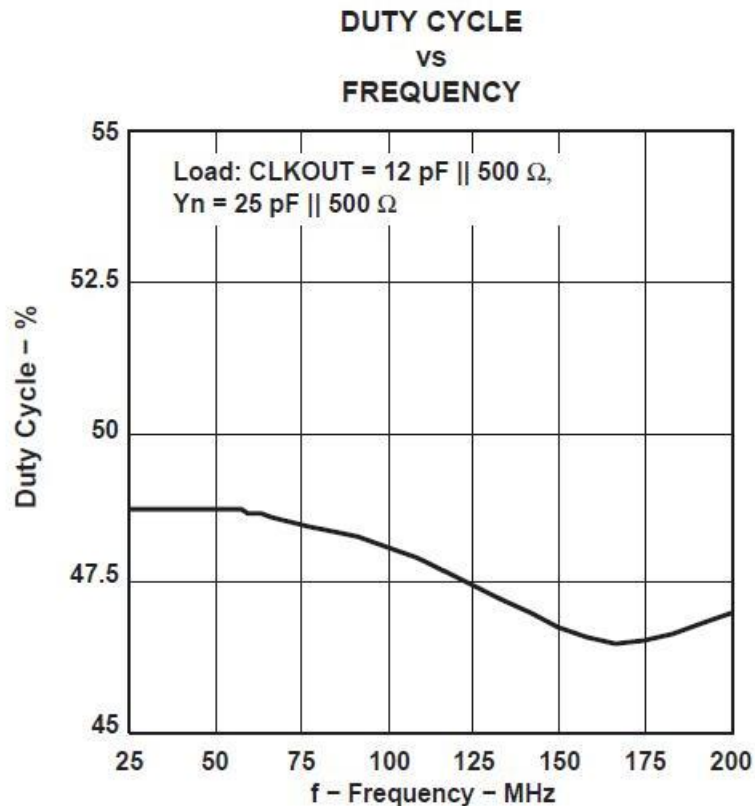


Figure 4

Duty Cycle vs Frequency

- 3 - non tutti i cicli si ripeteranno con lo stesso identico periodo, cioè non avranno tutti l'esatta durata del periodo nominale. Questo fenomeno viene specificato col

termine **Jitter** col quale si misura la variazione dei singoli periodi col periodo medio. Tanto minore sarà la variazione del Jitter tanto più **preciso** sarà il segnale di clock. Il seguente schema mostra il Jitter appunto come differenza tra la durata di due generici periodi:

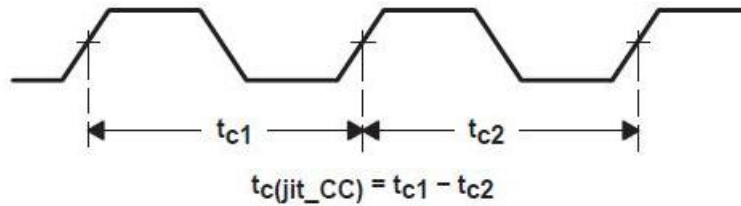


Figure 10. Cycle-to-Cycle Jitter

Jitter

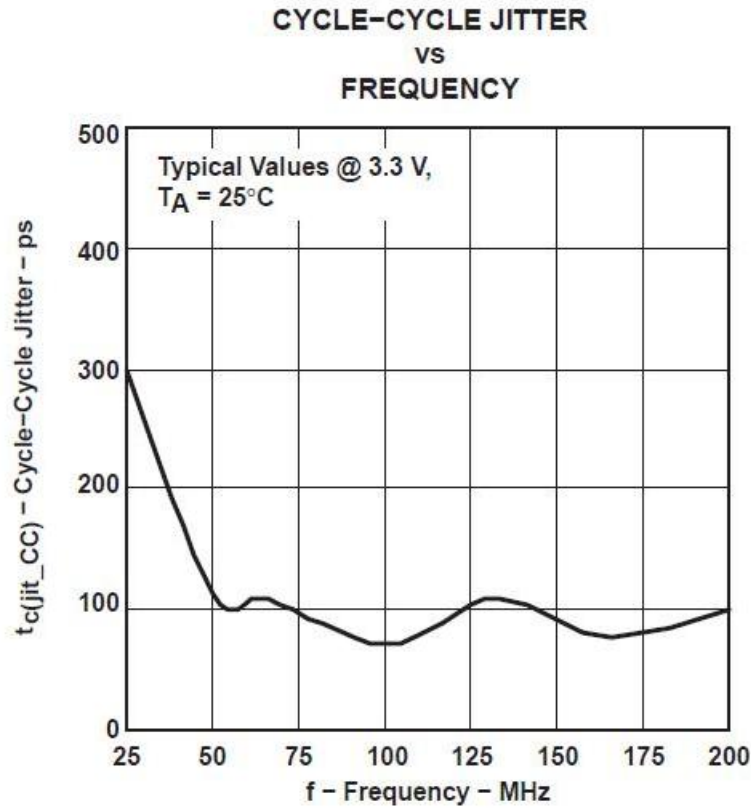
Qui le specifiche del Jitter in valori assoluti:

switching characteristics over recommended ranges of supply voltage and operating free-air temperature, $C_L = 25 \text{ pF}$, $V_{DD} = 3.3 \text{ V} \pm 0.3 \text{ V}$ (see Note 5)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{pd}	Propagation delay (normalized) (see Figure 3)	CLKIN to Yn, f = 66 MHz to 200 MHz	-150	150	ps
$t_{sk(o)}$	Output skew (see Note 6)	$V_O = V_{DD}$	-150	150	ps
$t_{c(jit_cc)}$	Jitter (cycle to cycle) (see Figure 5)	f = 66 MHz to 200 MHz	70	150	ps
		f = 24 MHz to 50 MHz	200	400	
t_{dcl}	Output delay (see Figure 4)	f = 24 MHz to 200 MHz, $V_{DD} = 3.3 \text{ V}$	150	550	ps
t_r	Rise time	$V_O = 0.4 \text{ V to } 2 \text{ V}$	0.5	2	ns
t_f	Fall time	$V_O = 2 \text{ V to } 0.4 \text{ V}$	0.5	2	ns

Specifiche Jitter

E qui la variazione del Jitter in base alla frequenza di emissione del clock:



Jitter vs Frequency

- 4 - dato che non tutti i cicli avranno il medesimo periodo, il segnale di clock tenderà ad assumere un periodo medio che potrà anche essere diverso dal periodo nominale. Questa divergenza viene identificata col termine **Accuratezza** da non confondere con la precisione. Infatti non è detto che un clock preciso sia anche accurato.

Ricapitolando:

- un clock preciso ha una bassa variazione della durata dei singoli periodi;
- un clock accurato ha un periodo medio molto prossimo a quello nominale.

Per visualizzare tali concetti di accuratezza e precisione riporto 3 esempi con bersagli di tiro a segno:

**Tiro preciso ma non
accurato**

**Tiro accurato ma non
preciso**

**Tiro accurato e
preciso**



A



B



C

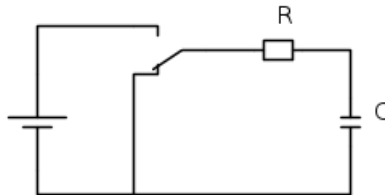
Considerando il centro del bersaglio come rappresentativo del periodo nominale, notiamo:

- caso A, i periodi variano di poco ma hanno un periodo medio distante da quello nominale: il clock è preciso ma non accurato;
- caso B, i periodi variano un po' di più, ma quello medio coincide con quello nominale: il clock è accurato ma non preciso;
- caso C, i periodi variano poco tra di loro e quello medio coincide con quello nominale: clock perfetto.

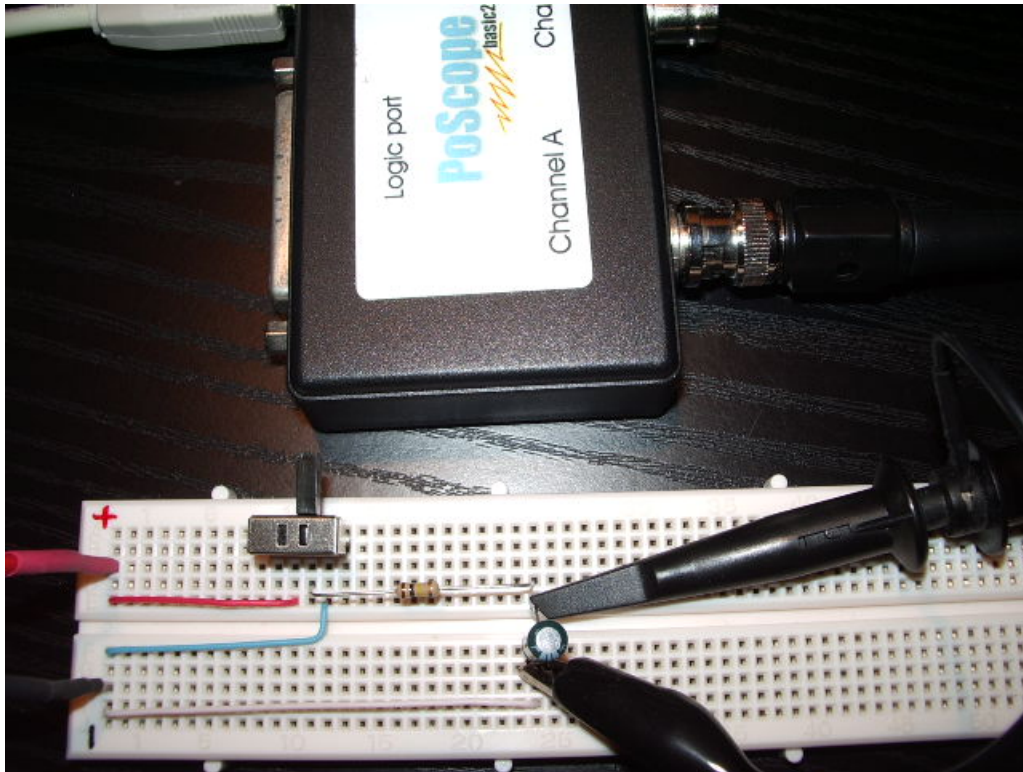
Viste per sommi capi le caratteristiche di un segnale di clock passiamo alla parte pratica su come realizzare un clock con mezzi semplici, di facile realizzazione ed altrettanto facile utilizzo senza ricorrere ad un integrato specializzato e, di conseguenza, senza pretendere prestazioni eccezionali in termini sia di accuratezza che precisione. Per fare ciò abbiamo bisogno di un circuito oscillante, ma andiamo per gradi e analizziamo il semplice circuito RC, ovvero circuito Resistenza - Condensatore.

Circuito RC

Il circuito RC con resistenza e condensatore collegati in serie ed alimentato in continua si presenta così:



Realizziamolo su BreadBoard scegliendo una R da 100 k Ω ed un C da 10 μ F in modo da avere tempi di carica e scarica rilevabili col PoScope:



Test Circuito RC

Collegiamo l'oscilloscopio agli estremi del condensatore e vediamo che succede azionando il deviatore tra + e - per alimentare (5V) e, rispettivamente, scaricare il condensatore. Con la funzione **RECORDER** del PoScope e qualche tentativo per selezionare i migliori parametri di campionamento si riesce a rilevare l'andamento della tensione nelle due fasi:



Carica e Scarica su C

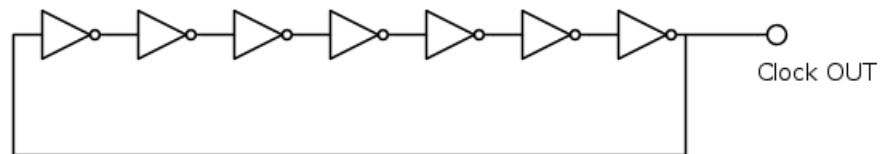
Rileviamo che il condensatore si carica fino a 4,5 V anziché 5 a causa della presenza dell'oscilloscopio che, presumibilmente, ha una resistenza interna di 1 MΩ e quindi realizza un partitore di tensione a circa il 10% con la nostra R da 100 kΩ, ma questo non ci disturba: proseguiamo.

Posizioniamo i marker verticali 1 e 2 al 10% ed al 90% dei 4,5 V sulla curva di salita ed andiamo a leggere, nel riquadro in basso a destra **Signal parameters at the markers**, la differenza T2-T1 di quasi 2 secondi. Stessa procedura sulla curva di discesa e troviamo analogo tempo di scarica. Infatti la teoria ci dice che tale tempo di carica o scarica risulta circa $2,2 * R * C = 2,2 * 100.000 * 10/1000000 = 2,2$ s. La differenza riscontrata è dovuta alle tolleranze sulla resistenza e sul condensatore che non saranno con precisione ne 100 kΩ ne, rispettivamente, 10 μF.

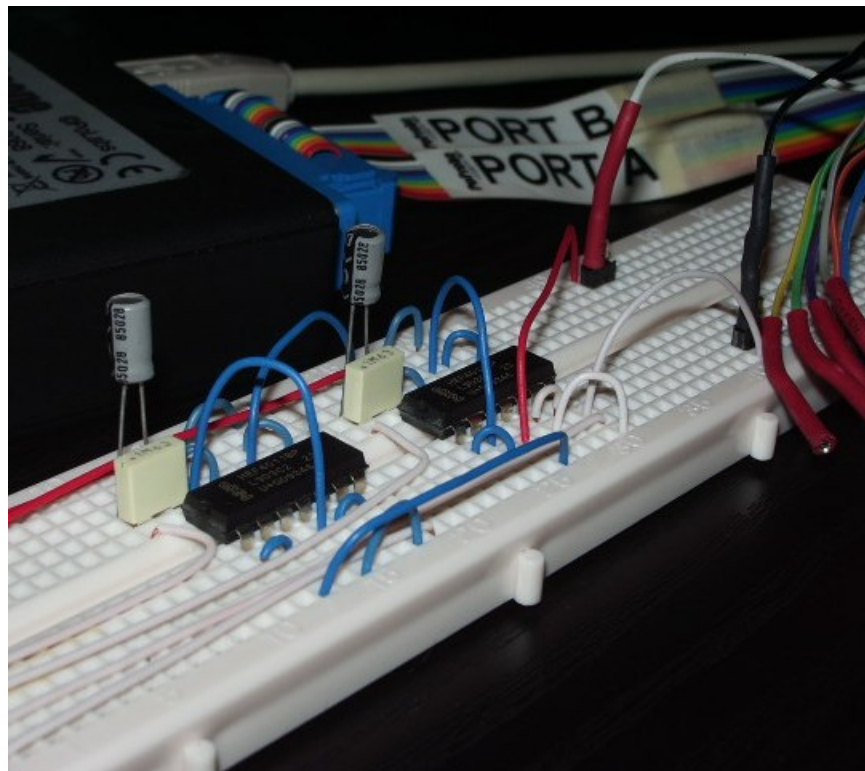
Dunque, abbiamo due fasi di carica e scarica distinte e separate azionate manualmente. Per i nostri scopi, ottenere un'onda quadra, l'ideale sarebbe trovare un meccanismo che faccia succedere una fase all'altra con continuità in modo automatico e che, contemporaneamente, squadrì l'andamento della tensione. Vi sono diverse soluzioni in elettronica analogica con induttanze, transistor e amplificatori operazionali, veramente interessanti che ci farebbero ripercorrere un po' la storia dell'elettronica, ma qui ci occupiamo di elettronica digitale e ci limiteremo ad una soluzione con elementi attivi digitali: le porte logiche NOT.

Multivibratore Astabile con porte NOT

La porta logica NOT ha la caratteristica di invertire il segnale in ingresso, ma lo fa con un certo ritardo che dipende dalle tecniche costruttive. Esaminando il DataSheet dell'integrato 74LS04, nella sezione **Switching Characteristics**, vi troviamo il Propagation Delay Time t_{PLH} o t_{PHL} variabile tra 4 e 15 ns: decisamente rapido, troppo rapido per le nostre capacità cognitive (mi riferisco al PoScope). Proviamo, allora, con una porta NOT in tecnologia CMOS (complementary metal-oxide semiconductor), per esempio la CD4009 o la CD4049 o, ancora, la CD4069 tutte con tempi di propagazione intorno ai 150 ns se alimentate a 5 V. Se ne mettiamo 7 in serie possiamo aumentare il periodo a $2 \times 150 \times 7 = 2100$ ns che, con l'analizzatore logico, dovremmo poter intercettare: proviamo!

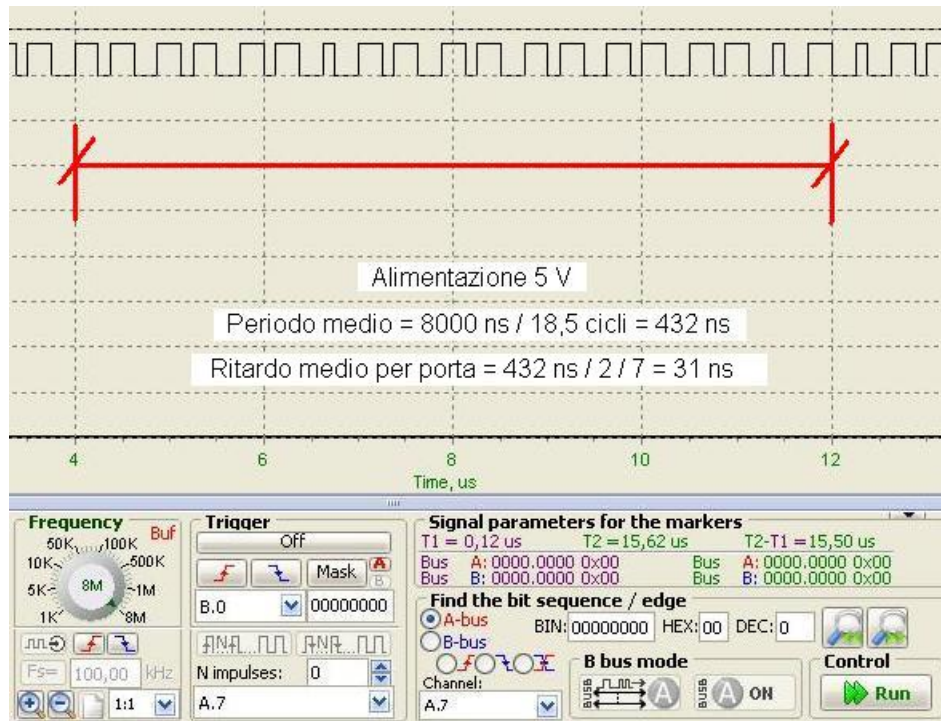


Non disponendo di porte NOT utilizzo le NAND, la sostanza non cambia! Ecco il circuito su BreadBoard con 2 CD4011:



Circuito con 7 NOT

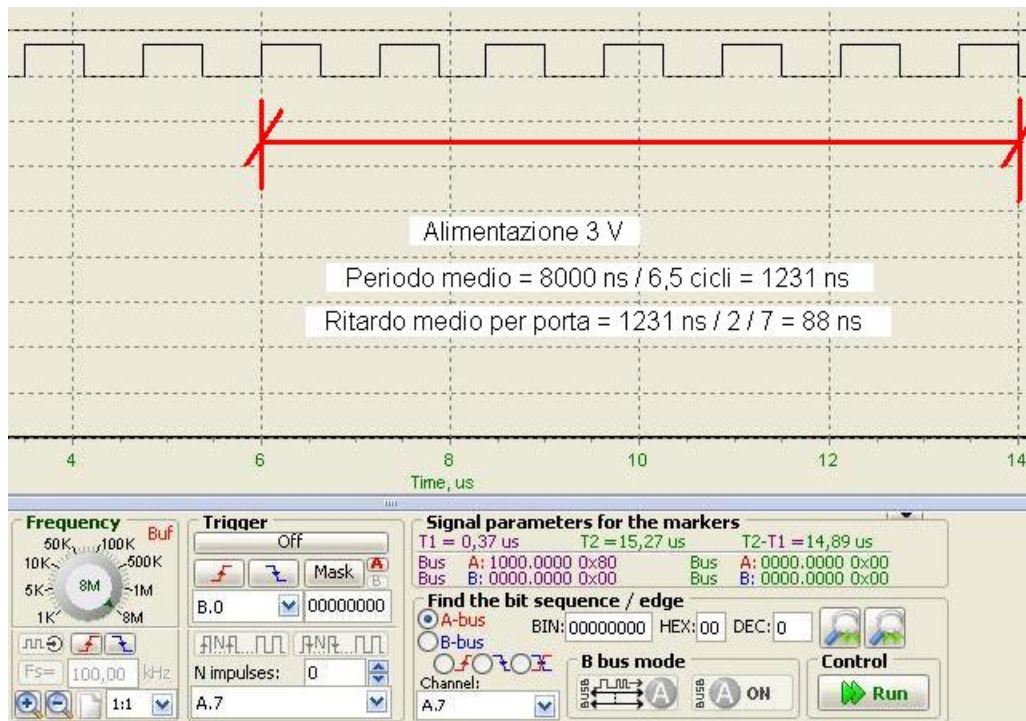
Qui il test con alimentazione a 5 V:



Test a 5 V

Notiamo che il ritardo è inferiore a quanto ci si aspettasse, ma il DataSheet specifica bene che i dati ivi riportati si riferiscono ad un carico di 200 kΩ con capacità 50 pF, mentre il nostro unico carico è rappresentato dall'oscilloscopio stesso e quindi le condizioni son ben diverse. Quindi a 5 V con 7 porte riusciamo ad avere un clock di circa $1 / 430 \text{ ns} = 2,3 \text{ MHz}$.

Vediamo che succede alimentando a 3 V:



Test a 3 V

Qui il ritardo aumenta, come previsto nel DataSheet al diminuire della tensione, ed otteniamo un clock da $1 / 1240 \text{ ns} = 800 \text{ kHz}$. Proviamo pure ad aumentare la tensione fino a 15 V, ma i ritardi diminuiscono troppo ed il nostro PoScope non è più in grado di campionare correttamente.

Bene! Abbiamo verificato che questa configurazione funziona! Però vi sono un sacco di problemi tra cui:

- non è molto pratico essere legati al numero di porte per variare la frequenza del clock;
- il ritardo di propagazione varia molto con la temperatura, con la tensione di alimentazione e col carico esterno.

Insomma il sistema non è per niente stabile.

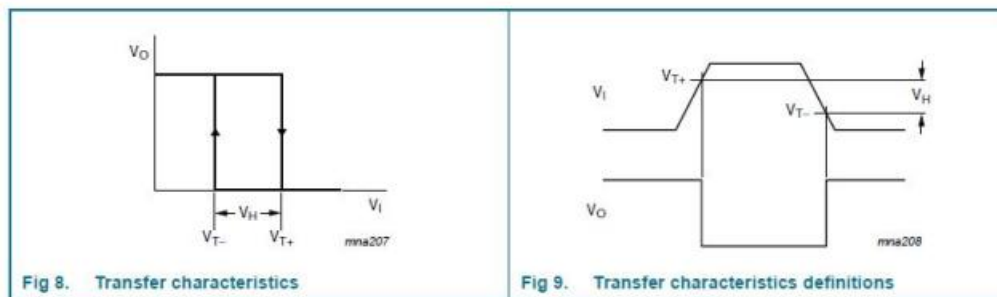
Per ottenere un clock stabile occorre inserire degli elementi passivi che determinino la frequenza e minimizzino le caratteristiche negative dei CMOS. Ed ecco che qui si incontrano le esigenze delle porte NOT, di essere controllate da elementi passivi, con l'esigenza del circuito RC, visto sopra, di un meccanismo che renda cicliche le fasi di carica e scarica del condensatore e ne squadrì l'andamento.

La soluzione consiste, quindi, nell'integrare i due sistemi. Vi sono diversi schemi che realizzano tale connubio, ma uno dei più semplici mi pare quello con un'unica porta NOT che andiamo subito ad esaminare.

Multivibratore Astabile con RC-NOT

Qui si pone il problema di scegliere la porta NOT più adatta. Una normale porta NOT presenta lo svantaggio di mal tollerare ingressi a lenta variazione tipica delle fasi di carica e scarica del nostro circuito RC. Infatti qualunque tensione in ingresso compresa tra la V_{IL} e la V_{IH} determinano instabilità dell'uscita, insomma la porta non "capisce" se l'ingresso ha valore logico 0 o 1 e quindi non "sa" cosa produrre in uscita. Per ovviare all'analogo problema che si riscontrava in elettronica analogica, nel lontano 1934, l'allora studente Otto H. Schmitt inventò il dispositivo che ha preso il suo nome: **Trigger di Schmitt**. Le porte logiche che adottano tale dispositivo sono in grado di trasformare un segnale d'ingresso lento, come lo è appunto quello del nostro condensatore, in un segnale in uscita ben definito e privo di jitter. Questo grazie al fatto che hanno due soglie di tensioni in ingresso sulle quali scattano (trigger = grilletto, scatto, sgancio) invertendo l'uscita solo al superamento di tali soglie. Vediamo uno schema di tali soglie da un generico datasheet di una generica porta NOT Schmitt-Trigger:

14. Transfer characteristics waveforms



Tensioni di soglia NOT Schmitt-Trigger

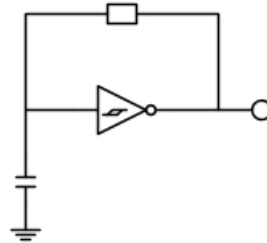
Vediamo nella figura di destra che solo quando la tensione di ingresso V_I supera la soglia V_{T+} o scende sotto la soglia V_{T-} si ha l'inversione dell'uscita. La V_I può essere lenta a piacere e può anche ballare tra le due soglie, ma l'uscita rimane stabile.

Bene! Fa proprio al nostro caso!

Come porta NOT a Trigger di Schmitt potremmo usare una 74LS14, ma è sconsigliato a causa delle resistenze interne non molto elevate e delle scarse e sbilanciate correnti di uscita ($I_{OH} = 0,4 \text{ mA}$ e $I_{OL} = 8 \text{ mA}$) tipiche dei TTL che costringerebbero ad usare una R bassa (inferiore a 1 K) limitando notevolmente le frequenze ottenibili e comunque con risultati scadenti: Duty Cycle molto sproporzionato e scarsa capacità di alimentare carichi. Ci conviene, invece, sfruttare gli integrati CMOS che non presentano tali svantaggi, per esempio il 74HC14 o il 74AC14, ma anche il vecchio

CD40106 va bene per i nostri scopi. Detto ciò vediamo come collegare la porta NOT al circuito RC con uno schema che possiamo trovare nel DataSheet stesso:

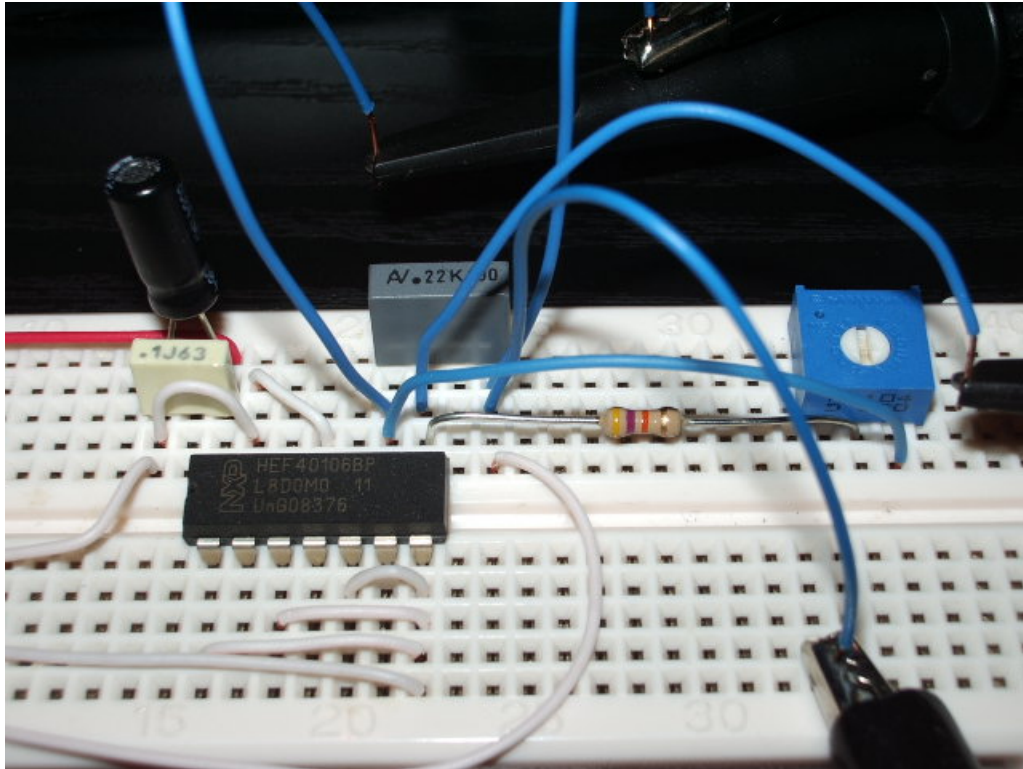
Multivibratore Astabile
con Inverter Schmitt-Trigger



Vediamo come dovrebbe funzionare immaginando una sequenza un po' fantasiosa e poco tecnica, ma che dovrebbe rendere l'idea:

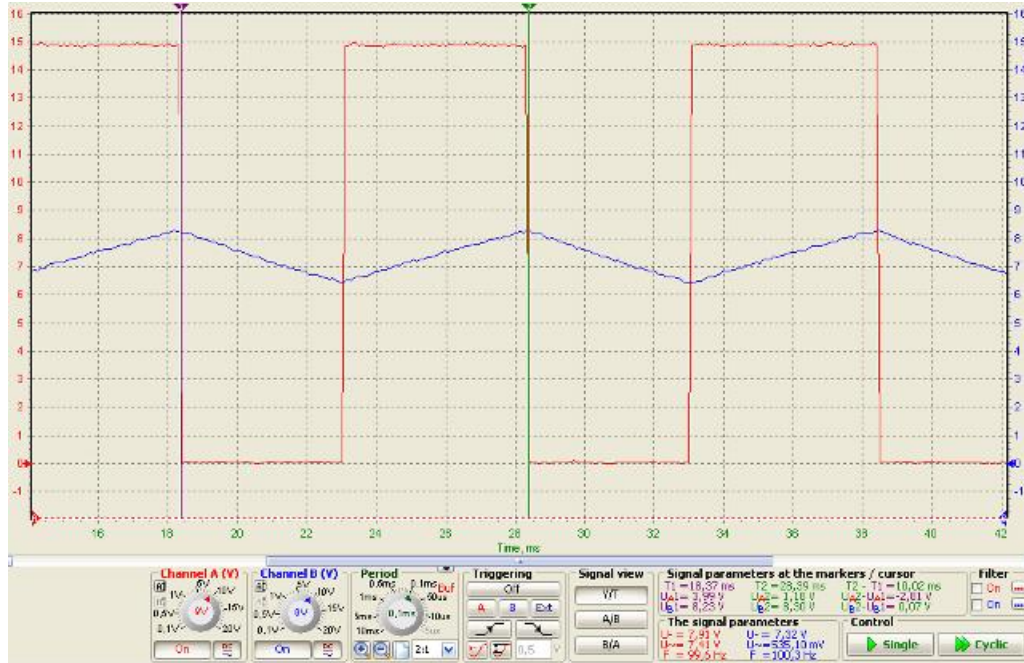
- Supponiamo che nell'istante di accensione il condensatore sia scarico e l'ingresso dell'inverter si trovi a 0 V;
- Passato il tempo di propagazione interno all'inverter, l'uscita si porta a 1 logico (per esempio 5 V);
- Il resistore farà affluire corrente verso il condensatore che inizierà la fase di carica incrementando la sua tensione;
- Solo quando tale tensione supera V_{T+} si ha l'inversione dell'uscita;
- Ma ora l'uscita è a 0 ed assorbe corrente: il condensatore, attraverso il medesimo resistore, inizia a scaricarsi;
- Solo quando la tensione scenderà sotto V_{T-} si avrà il successivo cambio e così via.

Allora non ci resta che sperimentare il circuito su BreadBoard. Dispongo di un CD40106 e questo uso:



Test NOT Schmitt Trigger

Nel datasheet consigliano l'uso di resistori da 50 k Ω a 1 M Ω e condensatori da 100pF a 1 μ F per ottenere periodi da 2 μ s a 0,4 s. Qui uso un C da 220 nF ed una R da 47 k Ω più un trimmer da 100 k Ω per fare un po' di prove. Analizziamo con l'oscilloscopio l'andamento delle due tensioni di ingresso e uscita della porta:



Andamento Vi e Vo a 15 V

Ottimo! Funziona! In blu il segnale in ingresso (carica e scarica del condensatore) ed in rosso l'uscita della porta NOT. Ci si può sbizzarrire provando diverse R, C e tensioni di alimentazione da 3 a 18V. Per il calcolo del periodo T il datasheet riporta la seguente formula che è risultata abbastanza attendibile:

$$T = RC \ln \left[\left(\frac{V_p}{V_n} \right) \left(\frac{V_{dd} - V_n}{V_{dd} - V_p} \right) \right]$$

con:

- T - periodo
- Vp - soglia trigger positiva
- Vn - soglia trigger negativa
- Vdd - alimentazione

Nel nostro caso:

- Vdd = 15 V
- Vp = 8,3 V
- Vn = 6,5 V
- R = 47k la R fissa + 44k del trimmer = 91k
- C = 220 nF

calcoliamo T:

$$T = 91 \cdot 10^3 \cdot 220 \cdot 10^{-9} \cdot \ln \left[\left(\frac{8,3}{6,5} \right) \left(\frac{15 - 6,5}{15 - 8,3} \right) \right] = 9,6 \text{ ms}$$

che corrisponde abbastanza ai circa 10 ms riscontrati.

Conclusione

Abbiamo visto dal vivo l'utilizzo di un economico strumento adatto come oscilloscopio, analizzatore logico e generatore di sequenze digitali. Abbiamo introdotto il clock eseguendo un po' di esperimenti per ottenerlo in modo semplice senza, ovviamente, esigere grande precisione, accuratezza e stabilità, ma sufficiente per i primi esperimenti con piccoli carichi. Quanto piccoli? Basta leggere i datasheet:

- per il CD40106 la corrente in uscita (source o sink) massima è di 6,8 mA
- col 74AC14 si arriva a 24 mA
- col 74HC14 si arriva a 25 mA

Per carichi maggiori sarà necessario realizzare generatori di clock con integrati particolari come l'NE555, ampiamente già descritto e citato in Sitografia.

Sitografia

[Tutorial sul Timer 555](#)

[Accuratezza e generatori di clock](#)

[Multivibratori con porte logiche](#)

[CMOS Oscillators Fairchild AN118](#)

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Robert8:elettronica-digitale-pratica-iv-logic-analyzer-clock>"