Marco Martines (MarcoMartines)

CIRCUITO DI ALLARME ISA-1

4 March 2010

Introduzione

La realizzazione di un circuito di allarme, seppur elementare, è un argomento che necessita particolare attenzione, specie in fase di progettazione, alle casistiche di funzionamento. Per **casistica di funzionamento** s'intende il modo con cui il nostro circuito risponde agli stimoli di allarme e ai segnali di acquisizione della condizione di allarme, da parte dell'utente utilizzatore. Ogni modalità (**sequenza**) viene identificata con un numero preceduto dalla sigla *ISA*. In commercio è possibile trovare circuiti che settati tramite opportuni switch o jumper consentono di variare la sequenza di funzionamento del sistema di allarme. Nel nostro caso esamineremo un circuito predisposto per lavorare in un'unica sequenza: **ISA-1**

Sequenza ISA-1

La sequenza ISA-1 è la più semplice dal punto di vista di funzionamento e dunque realizzazione. Per realizzare un circuito che rispetti lo standard ISA-1 dobbiamo avere un dispositivo di input, per consentire di segnalare l'avvenuta acquisizione dello stato di allarme da parte dell'utilizzatore (pulsante di AQ) e due dispositivi di output con cui il sistema ci segnala le condizioni di allarme (sirena e led). In condizioni di normalità entrambi i dispositivi di output risultano spenti. Non appena rilevata la condizione di allarme, il circuito la segnalerà alimentando la sirena e con il lampeggiamento del led, sino al riconoscimento AQ. Nel momento in cui avviene l'acquisizione, il circuito deve valutare se la condizione di allarme è ancora attiva (*allarme persistente*) e segnalarlo tramite luce fissa del led, finchè tale condizione permane; altrimenti ripristinerà il sistema alla condizione di normalità.

Pseudocodice

Serviamoci dell'ausilio di uno *pseudocodice* per comprenderne il funzionamento:

```
alarm={true,false};        //condizione dell'allarme
siren={true,false};        //stato della sirena
led={off,on,flashing};        //stato del led
AQ={true,false};        /*stato dell'interruttore
                        di acquisizione della
```

```
                        condizione di allarme*/
if(alarm=true){
        while(AQ=false){
                siren<-true;
                led<-flashing;
                }
        if(alarm=true){
                siren<-false;
                led<-on;
                }
        else(alarm=false){
                siren<-false;
                led<-off;
                }
        }
else{
        siren<-false;
        led<-off
        }
```

Approccio iniziale

Per poter realizzare un circuito del genere, è conveniente servirsi di un software di simulazione (*Micro-Cap 9.0*) per controllarne la correttezza in fase progettuale. Utilizzeremo un approccio *top-down* suddividendo il circuito completo, in sottocircuiti più semplici, in modo da consentirci una realizzazione ed interpretazione più semplice. L'intera circuiteria si dovrà occupare di generare un segnale logico di allarme, rilevare tale segnale e memorizzarlo, riconoscere il segnale di acquisizione, temporizzare l'attivazione della sirena per evitare che si prolunghi nel tempo. Il cuore di tutto il sistema è la parte circuitale che si occupa di rilevare e memorizzare il segnale di allarme da parte di uno o più sensori. La memorizzazione è molto importante, specie nel caso di segnali di allarme di breve durata; è necessario infatti che l'allarme continui a suonare anche in caso di situazioni di allarme brevi nel tempo, sino al riconoscimento AQ; è opportuno inoltre memorizzare anche quest'ultima. Dovremmo disporre anche di un circuito in grado di generare un impulso di tipo *onda quadra* per consentire a led di lampeggiare (che utilizzeremo anche nella temporizzazione della sirena) e di un ulteriore circuito affinché il segnale generato dal sensore sia un segnale logico. Utilizzeremo dunque due *flip-flop* per memorizzare i due segnali principali (allarme, AQ); un *comparatore* per asserire a un livello logico alto il segnale di allarme, quando la tensione prodotta dal sensore utilizzato per rilevare il movimento supera una determinata soglia.

Memorizzazione impulsi esterni

Come descritto in **Approccio Iniziale** la parte principale del sistema è definita dai due flip-flop che ci consentiranno di: rilevare un segnale di allarme, consentire l'acquisizione da parte dell'utente ed infine segnalare all'utente se nel momento dell'acquisizione la condizione di allarme è ancora verificata. Vediamo l'implementazione circuitale, ricordando che ai fini della realizzazione pratica di tale sistema è necessario utilizzare un integrato che contenga le porte logiche che andremo ad impiegare (*es. CM4011B*).

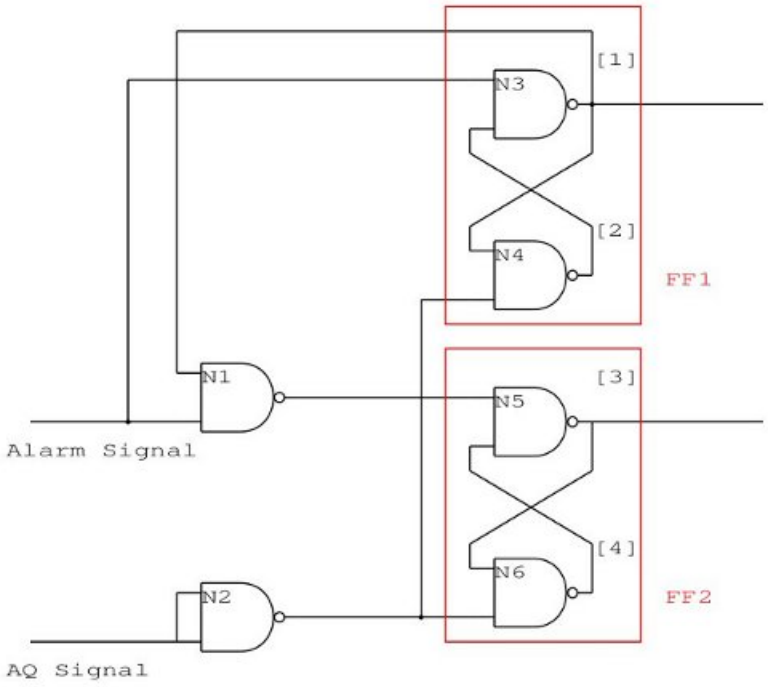

Figura 1: I due flip-flop

Nel circuito viene usata una NAND (N2) al posto di un *inverter*; questo poiché in fase di realizzazione sarà più economico, sia come costi che come spazio, andare ad utilizzare tale porta, già presente negli integrati impiegati per i flip-flop piuttosto che inserire un integrato apposito che contenga l'inverter. Vediamo la tabella delle verità dei due flip-flop, basati sulle NAND, visti in **Figura 1**, ricordando che i flip-flop sono elementi sequenziali e non combinatori.

Alarm Signal	AQ Signal	nodo [1]	nodo [3]
0	1	1	0
0	0	1	0
1	0	1	1

0	0	1	1
0	1	1	0
0	0	1	0
1	0	1	1
1	1	0	0
1	0	0	0
0	0	1	0

Il circuito va inizializzato con *AQ Signal* asserito, e *Alarm Signal* deasserito. La coppia di uscite dei flip-flop può assumere solamente 3 valori *10*, *11*, *00*, che rispettivamente corrispondo alla condizioni di normalità, allarme ed allarme persistente.

Condizione di allarme

Il comparatore, in **Figura 2** ci consentirà di asserire alto il segnale Alarm Signal, se il valore di tensione prodotto dal sensore sarà maggiore del valore di tensione di soglia imposto. Sarà possibile regolare il valore della soglia agendo sul resistore variabile del partitore.

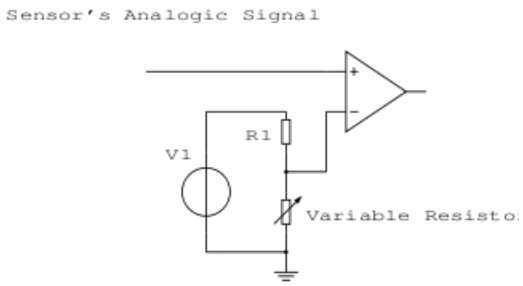

Figura 2: Comparatore di tensione

Segnale intermittente

Questa parte di circuito, schematizzato in **Figura 3** serve a generare un segnale pulsante, spesso utilizzato come segnale di *clock* in molti altri circuiti. Non analizzeremo in dettaglio il funzionamento poiché si discosterebbe in modo significativo dall'argomento trattato, ma ci limiteremo a studiare la configurazione, in modalità astabile, dell'integrato NE555 che utilizzeremo a tale scopo. Il segnale intermittente verrà asserito sulla linea *Output*.

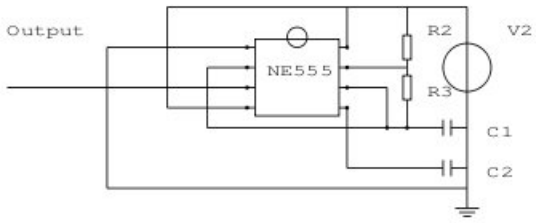

Figura 3: Circuito generatore segnale di clock

Componentel	Valore
C1	470 nF
C2	470 nF
R1	$3\,k\Omega$
R2	$3\,k\Omega$
R3	$100\,k\Omega$
C3	1 mF
C4	470 nF
R4	$3\,k\Omega$
R5	$100\,k\Omega$

Sono riportati in tabella i valori dei componenti impiegati per i segnali di clock del led e della sirena, nei due circuiti contenenti l'NE555.

Circuito completo

Collegando tra loro le varie componenti, sopra analizzate, otteniamo il circuito finale in **Figura 4**. Effettuiamo dunque una simulazione del circuito. Essendo la versione freeware di Micro-Cap 9.0 non possiamo simulare un circuito con un numero elevato di nodi; ci limiteremo dunque ad utilizzare un unico NE555 trascurando la temporizzazione della sirena. Inoltre per poter apprezzare il lampeggiamento del led, simulando tempi molto brevi, è stato utilizzato un condensatore da 47nF al posto di quello da 470nF. I grafici della simulazione in **Figura 6** mostrano, in ordine, il segnale digitale di allarme, il segnale di AQ ed i valori delle uscite del led e della sirena. Concludendo osserviamo che qualora la complessità richiesta dal sistema risulti maggiore è conveniente pensare all'utilizzo di un microcontrollore al posto dei circuiti integrati utilizzati.

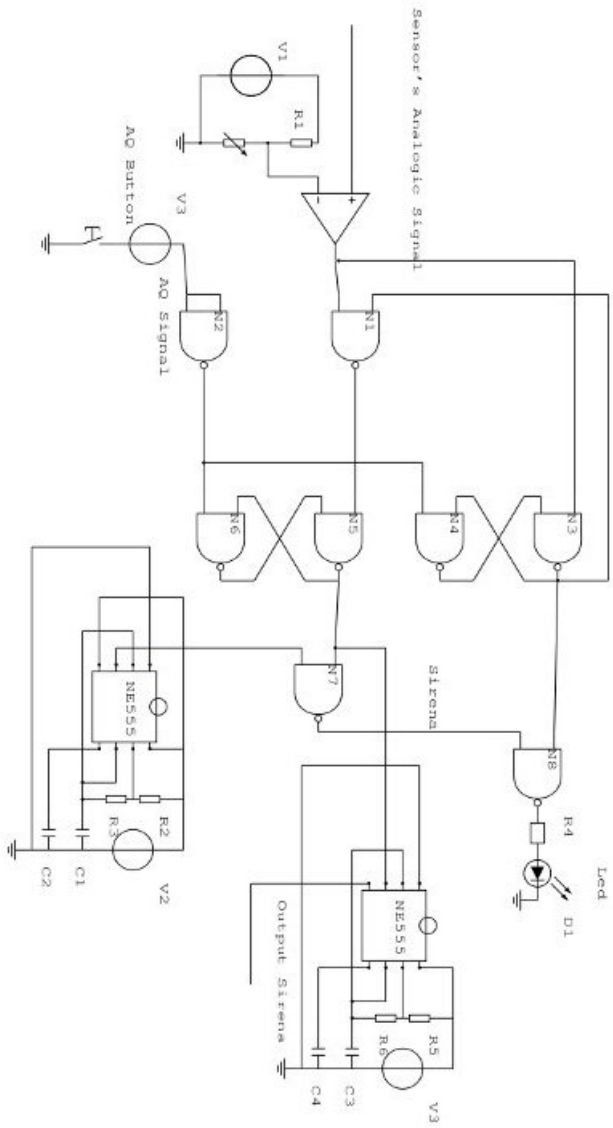

Figura 4: Circuito completo

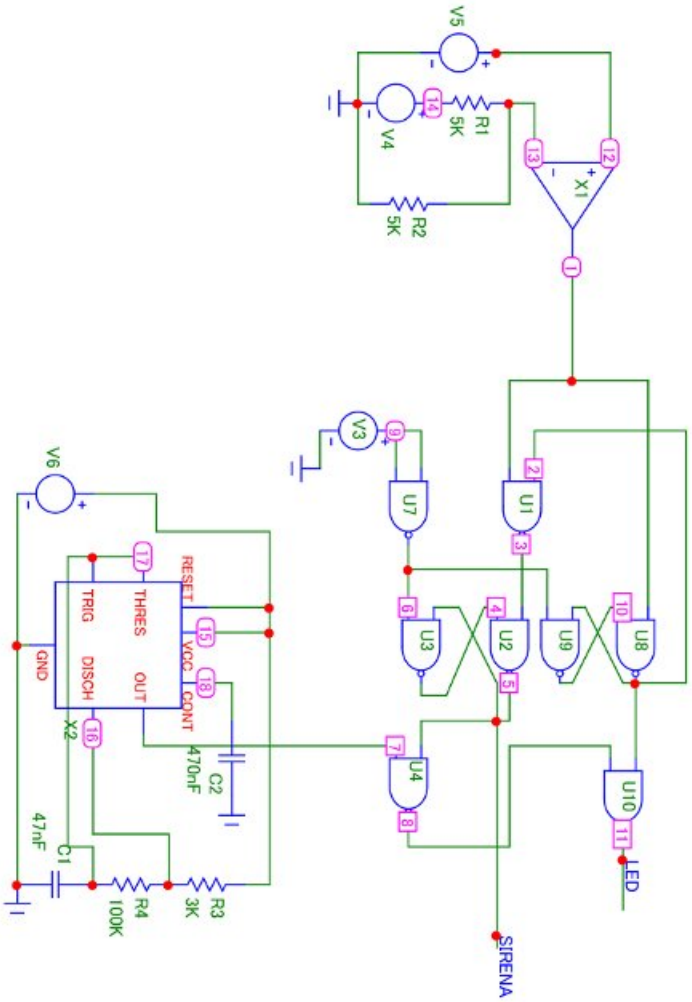

Figura 5: Circuito simulato in Micro-Cap 9.0

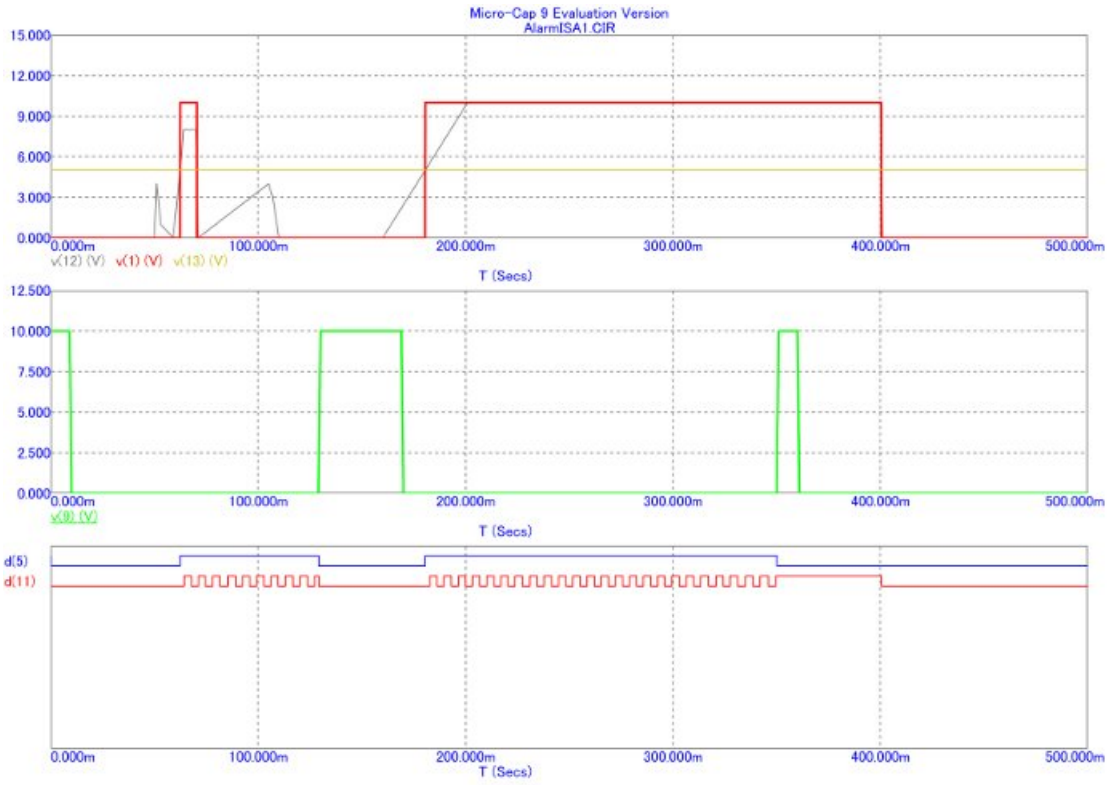

Figura 6: Simulazione effettuata con Micro-Cap 9.0

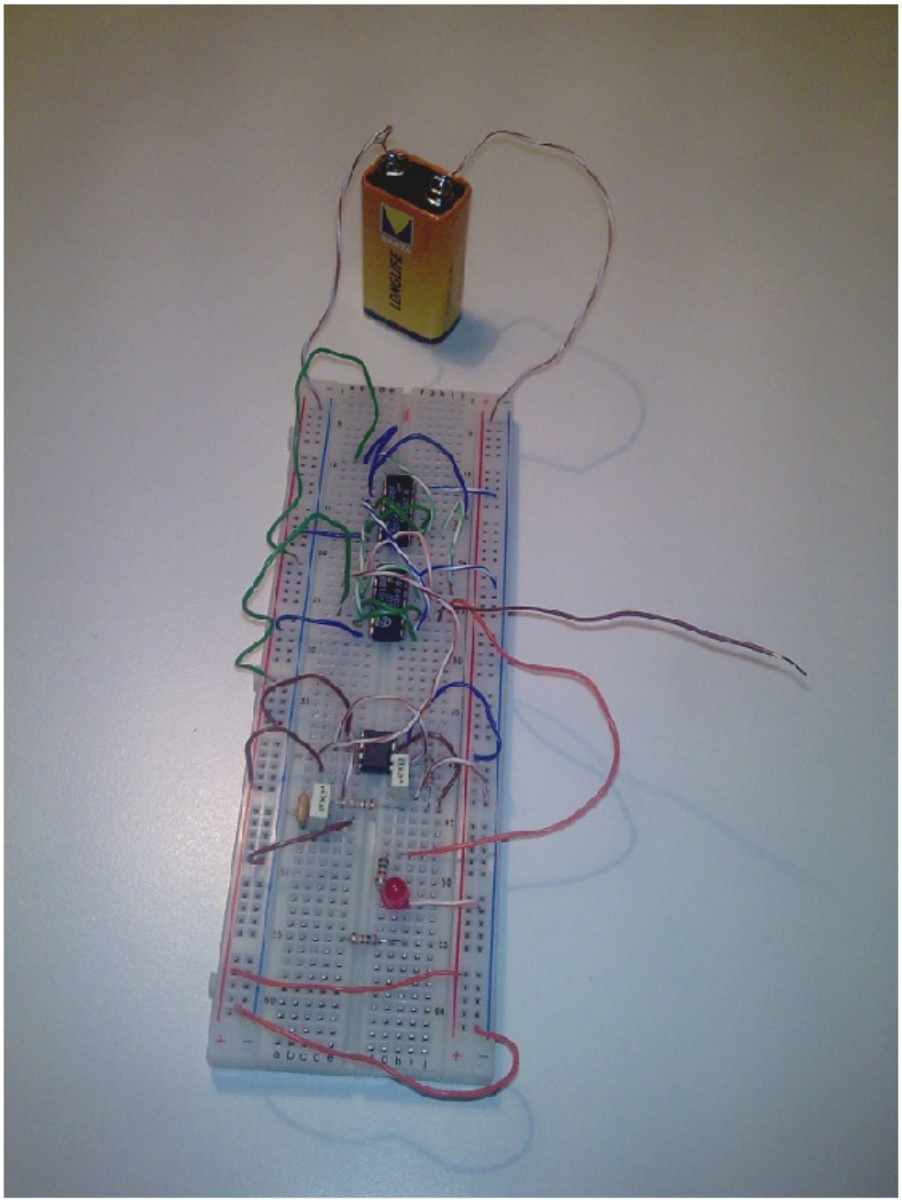

Figura 7: Parte del circuito realizzato su breadboard

Marco Martines

Estratto da "http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Marcomartines:allarme"