



Davide Bucci (DarwinNE)

DISEGNARE CIRCUITI CON FIDOCAD E FIDOCADJ

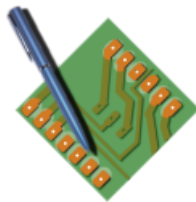
24 April 2010

La filosofia di FidoCad e FidoCadJ

Capita spesso di dover **tracciare un disegno tecnico** al calcolatore, in maniera veloce e senza troppi fronzoli. Quest'esigenza è particolarmente sentita all'interno di **forum** o gruppi di discussione dedicati all'elettronica e non solo.

L'ideale sarebbe ottenere un disegno di qualità, che non occupi troppo spazio e magari anche in un **formato facilmente accessibile**, che consenta anche ad altri di riprendere il risultato ed apportarvi delle modifiche.

Capita spesso infatti che in una discussione su un circuito ci si trovi ad apportare delle varianti. In questo articolo, vedremo la filosofia dei programmi *FidoCad* e *FidoCadJ*, che sono fatti per venire incontro proprio a quest'esigenza.



icona_fidocadj_128x128.png

Il formato FidoCad

Nell'ambiente dei newsgroup italiani (in particolare in **it.hobby.elettronica**, ma non solo), è molto utilizzato un programmino che si chiama *Fidocad per Windows* e che permette di disegnare velocemente schemi elettrici e realizzare sbrogli di piccoli circuiti stampati, un po' come si faceva tanti anni fa con i trasferibili R41. Questo programma, scritto da Lorenzo Lutti, ha avuto il merito di introdurre un formato per la rappresentazione dei disegni che è fatto da codici testuali, è compatto e ben si presta ad essere copiato ed incollato all'interno di messaggi su newsgroup o forum, senza aver a che fare con oscuri formati proprietari, programmi difficili da trovare o molto costosi. Ecco un esempio di un disegno:

```
[FIDOCAD]
LI 55 30 50 30 0
TY 7 9 12 7 0 0 0 * Specchio di corrente
```

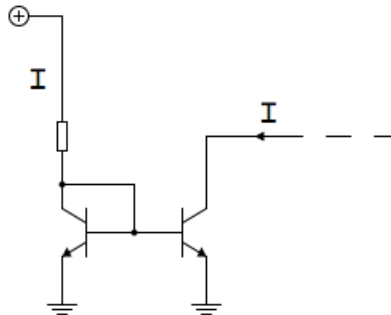
```

LI 55 60 55 65 0
LI 55 65 70 65 0
LI 70 65 70 75 0
SA 55 65 0
SA 70 75 0
LI 85 65 85 55 0
LI 85 55 105 55 0
LI 55 50 55 30 0
TY 48 40 5 4 0 0 0 * I
TY 96 47 5 4 0 0 0 * I
TY 22 100 3 2 0 0 0 * Nei due rami, scorre sempre la stessa corrente
TY 22 105 3 2 0 0 0 * Il circuito funziona molto bene con transistor identici
MC 55 85 0 0 040
MC 85 85 0 0 040
MC 55 50 0 0 115
MC 70 75 0 1 280
MC 70 75 0 0 280
MC 50 30 2 0 010
MC 100 55 2 0 074
LI 110 55 115 55 0
LI 120 55 125 55 0

```

Ed ecco quello che si ottiene con il programma:

Specchio di corrente



Nei due rami, scorre sempre la stessa corrente
 Il circuito funziona molto bene con transistor identici

specchiogz2.png

Grazie al fatto che il programma di Lorenzo Lutti è gratuitamente scaricabile, negli anni è stato molto apprezzato dagli appassionati di elettronica e di fai da te italiani. La decennale esperienza dell'uso di questo formato sui newsgroup Usenet ha dimostrato ampiamente la rapidità e l'efficacia

di questo modo di procedere. *FidoCad per Windows* non è più sviluppato attivamente dal 2001, ma è sempre disponibile dalla pagina del suo autore:

http://www.enetsystems.com/~lorenzo/fidocad_win.asp

Naturalmente, non si tratta di un programma confrontabile con i grandi CAD professionali, ma nonostante le sue limitazioni si rivela insospettabilmente comodo in molte situazioni.

La nascita di FidoCadJ

Io ho abbandonato Windows come sistema operativo principale ormai da molti anni, approdando a Linux e soprattutto a MacOSX. Essendo abbastanza attivo in diversi gruppi di discussione e forum, ho potuto entrare in contatto con *FidoCad*, ma mi sono scontrato con il problema di fare girare il programma sui miei sistemi operativi preferiti. *FidoCad per Windows* gira bene sotto Linux con Wine, ma non c'era una soluzione adatta a MacOSX.

La scelta di un certo sistema operativo è una scelta personale, ma quando manca un certo programma, penso che sia meglio darsi da fare, piuttosto che lagnarsi che sotto un'alternativa a Windows non si trovi una soluzione (come se la colpa fosse del sistema operativo in sé!). Il bravo Lorenzo Lutti aveva scritto da solo *FidoCad per Windows* ed io avrei potuto fare qualcosa di simile.

La prima cosa che ho fatto è stata quindi di studiarli in dettaglio il formato utilizzato dal *FidoCad* e scrivere un'applet Java chiamata *FidoReadJ*, capace di interpretarlo e mostrare il risultato all'interno di una pagina Web. Ho fatto un po' di reverse engineering, ho poi cercato informazioni più o meno dappertutto per capire quali fossero le possibilità e le limitazioni del formato ed infine mi sono scaricato e studiato i sorgenti in C++ (peraltro molto ben fatti e commentati) del *FidoCad* originale. Questo avveniva più o meno verso marzo 2007. Qualche mese più tardi, l'applet era pubblicata sul sito ed era stata ampiamente testata da una parte della comunità gravitante attorno al gruppo **it.hobby.elettronica** e **it.hobby.fai-da-te**. Per chi fosse interessato, è ancora disponibile qui:

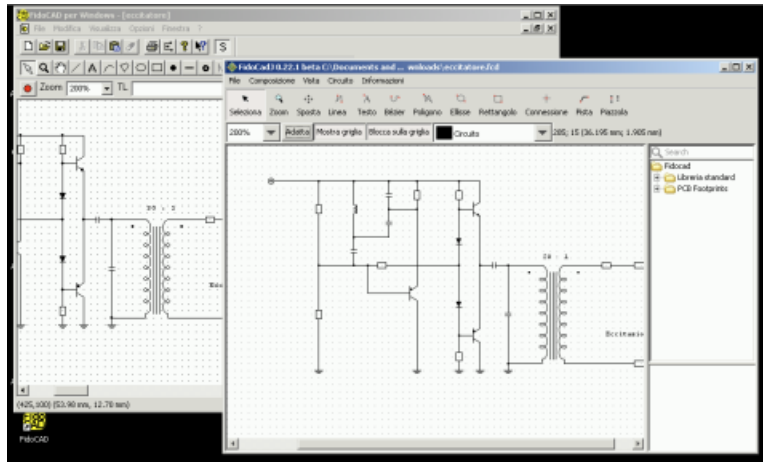
<http://davbucci.chez-alice.fr/index.php?argument=elettronica/fidoreadj/fidoreadj.inc>

Disponendo di un interprete del formato, il lavoro restante per arrivare ad un editor completo è stato quello di mettere a punto la gestione dell'interfaccia utilizzatore e delle varie primitive. Il grosso del lavoro si è protratto in varie fasi da gennaio a luglio 2008, dando origine a *FidoCadJ* (con la "J" alla fine, quindi). *FidoCadJ* non è quindi un porting di *FidoCad* per Windows, ma piuttosto una riscrittura totale del programma. A partire dalla versione 0.21 (pubblicata nel gennaio 2009), *FidoCadJ* propone delle estensioni rispetto al formato *FidoCad* originale. Ho fatto molta attenzione per mantenere comunque una compatibilità all'indietro, anche se a partire dalla versione 0.23 di fine dicembre 2009 ho introdotto delle estensioni per cui questo è impossibile.

FidoCad per Windows e FidoCadJ

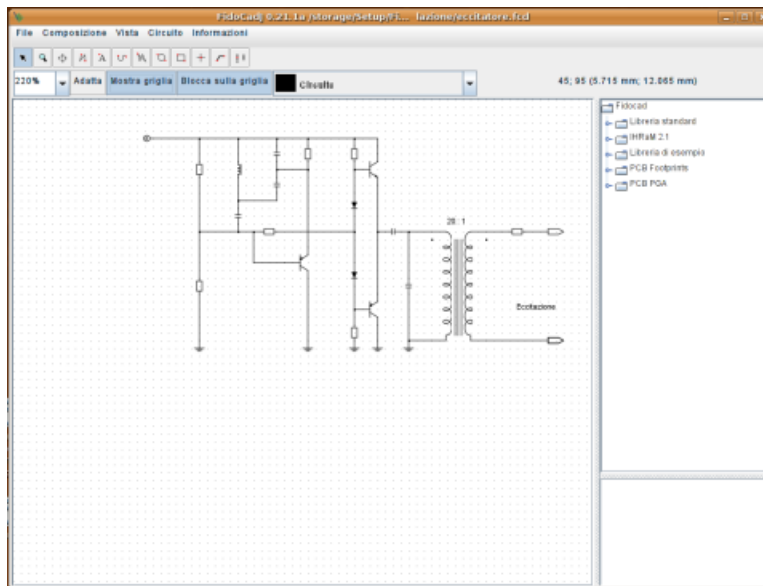
Screenshot

L'immagine seguente mostra l'aspetto che hanno *FidoCadJ* ed il *FidoCad per Windows*. Entrambi mostrano lo stesso circuito:

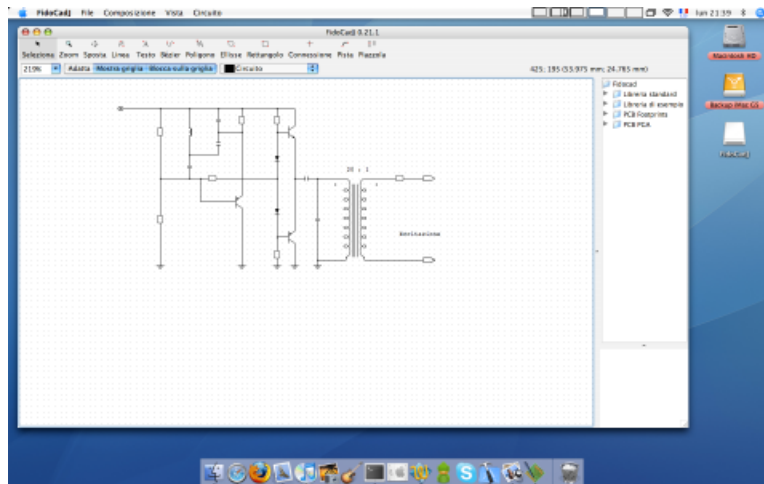


fidocadj_fidocad_t.png

Qui di seguito abbiamo invece una versione di FidoCadJ che gira sotto Linux e MacOSX Tiger:



fidocadj_ubuntu_t.png



fidocadj_t.png

Le differenze tra FidoCad per Windows e FidoCadJ

Cerchiamo adesso di vedere quali sono le differenze tra *FidoCad per Windows* ed il mio *FidoCadJ*. Prima di tutto, c'è una cosa importante da dire relativa a *FidoCadJ*, ovvero che per default integra una libreria in più rispetto al *FidoCad per Windows* e si tratta della libreria messa a punto su **it.hobby.radioamatori.moderato** (IHRaM). E' fatta molto bene ed integra tanti simboli utili. Questa ed altre possono naturalmente essere installate anche su *FidoCad per Windows* e difatti è nata proprio con questo programma. Ho raccolto in questa pagina altre librerie che negli anni sono state messe a disposizione dagli sviluppatori:

<https://darwinne.github.io/FidoCadJ/docs/libraries/>

Attualmente, se cerco di fare un'analisi un po' obiettiva (per quanto posso, essendo ovviamente fortemente legato a *FidoCadJ*), direi che *FidoCadJ* è superiore a *FidoCad per Windows* nei casi seguenti:

- utilizzo delle primitive di disegno estese (linee tratteggiate e frecce alle estremità di segmenti e curve di Bézier)
- possibilità di associare un valore ed un nome ad ogni macro usata negli schemi elettrici
- esportazione in formati grafici vettoriali e raster
- anti aliasing nel disegno degli schemi e resa grafica globalmente migliore
- maggiore velocità di ridisegno (su un JRE decente e installato correttamente)
- ricerca rapida nelle librerie di simboli installate
- sviluppo attivo
- internazionalizzazione dell'interfaccia, delle librerie e dei manuali
- multipiattaforma

Ciò ovviamente non toglie nulla al magnifico lavoro di Lorenzo Lutti e certe mie posizioni sono di sicuro opinabili. *FidoCad* è invece nettamente superiore a *FidoCadJ* per:

- una migliore integrazione in Windows (C++ e MFC)
- un'installazione più semplice in Windows, non richiedendo Java

Ci sono poi alcune differenze di filosofia dovute a delle mie scelte che sono differenti rispetto a quelle fatte da Lorenzo Lutti. Per esempio, se in *FidoCad per Windows* c'è una modalità PCB da attivare che implica che il disegno non viene riscaldato al momento di stampare, questa scelta in *FidoCadJ* viene fatta selezionando o meno l'opzione **Adatta dimensioni alla pagina** che compare quando l'utente vuole stampare. Quello che consiglieri ai nuovi utenti è di provare entrambi i programmi (sempreché si stia usando Windows, altrimenti la scelta è quasi obbligata) e di fare la propria scelta, considerando comunque la compatibilità del formato usato. A molti per esempio non piace Java, quindi possono tranquillamente dimenticarsi di *FidoCadJ* ed installarsi il *FidoCad per Windows*, perdendosi però le novità di questi ultimi anni.

Scaricare FidoCadJ e partecipare allo sviluppo

Al pari di *FidoCAD per Windows*, *FidoCadJ* è liberamente scaricabile ed è disponibile insieme ai suoi sorgenti. La licenza che ho scelto è la GPL v. 3. Ecco la pagina principale relativa a *FidoCadJ*:

<http://darwinne.github.io/FidoCadJ/>

Da qui si può scaricare il necessario (file eseguibili, installazione per Windows, versioni per MacOSX, Linux ed Android):

<http://darwinne.github.io/FidoCadJ/docs/download/>

Ho creato un progetto su GitHub, contenente tutti i sorgenti, per partecipare allo sviluppo del programma:

<https://github.com/DarwinNE/FidoCadJ>

Il programma richiede di avere Java aggiornato alla versione almeno superiore alla 9. Se il programma non funzionasse, verificate la vostra installazione di Java ed eventualmente scaricatevi un JRE aggiornato dal sito di Oracle:

<https://openjdk.java.net/projects/jdk/16/>

E' molto comune che le vecchie versioni di Java rifiutino di aggiornarsi oltre a Java 8 (che è molto vecchia), quindi attenzione! Mi raccomando, fatemi sapere come vi trovate con il programma e mettendomi al corrente dei bug o delle cose che non vanno. Il modo migliore per farlo è aprire una Issue (in inglese) su GitHub, aggiungendo l'etichetta "bug":

<https://github.com/DarwinNE/FidoCadJ/issues>

Tutorial e manuali

In rete ci sono molti tutorial relativi a *FidoCad per Windows* e questo programma ha un help abbastanza ben fatto e fruibile. *FidoCadJ* è dotato di un manuale che contiene un paio di tutorial, uno dedicato alla creazione di uno schema elettrico ed uno invece dedicato ai circuiti stampati.

Ecco il link:

https://github.com/DarwinNE/FidoCadJ/releases/download/vo.24.8/manual_en.pdf

Il manuale presenta fra l'altro informazioni molto dettagliate sul formato *FidoCad* e su quello che c'è da fare per installare *FidoCadJ* su Windows, Linux e MacOSX.

Integrazione di FidoCadJ nel forum di ElectroPortal

Grazie ad una bella idea di [Piercarlo](#) ed al lavoro del webmaster, siamo riusciti ad integrare la possibilità di interpretare e mostrare dei disegni in formato *FidoCad* direttamente all'interno del forum di ElectroPortal. L'idea è che l'utente desideroso di mostrare uno schema elettrico deve solo scaricarsi ed installarsi *FidoCad per Windows*, oppure il mio *FidoCadJ*. Con un semplice copia/incolla basta introdurre il codice *FidoCad* del disegno da mostrare all'interno dei propri interventi, facendolo precedere e seguire dai tag [fcd] e [/fcd]. La tecnica è quindi molto simile a quella utilizzata dall'inclusione del codice LaTeX per le formule. Questo è spesso molto più pratico di uno screenshot da un CAD professionale o una scansione di uno schema cartaceo. Per chi fosse interessato a seguire la discussione che ha reso possibile questa tecnica, può leggere qui:

<http://www.electroportal.net/phpBB2/viewtopic.php?f=4&t=18347>

Per farla breve, *FidoCadJ* gira in maniera batch sul server di ElectroPortal e si occupa silenziosamente di convertire il codice dei disegni in immagini che chiunque può vedere, anche senza utilizzare il programma. Se qualcuno vuole invece modificare i disegni, basterà installare il *FidoCad* di Lorenzo Lutti, oppure il mio *FidoCadJ*, ottenere il codice del disegno e da lì partire con le modifiche.

Quando volete condividere il vostro schema, fate però attenzione. Prima di tutto, se utilizzate librerie non previste dalla distribuzione standard di *FidoCadJ* (che sarebbero la libreria standard, quella per i circuiti stampati e la libreria IHRaM), non dimenticatevi di attivare l'opzione **Suddividere le macro non standard** nelle opzioni del programma. Inoltre, evitate di scegliere font particolari per i disegni (accettabili sono il Courier New, Helvetica, Arial, Times, Times New Roman e Symbol). *FidoCadJ* introduce inoltre alcune estensioni che non sono comprese da *FidoCad*, si tratta della possibilità di associare un nome ed un valore ad ogni macro, nonché di disegnare frecce e linee tratteggiate. Se usate *FidoCadJ* e volete rimanere compatibili con il programma originale (privandovi però delle possibilità avanzate), selezionate la modalità **Modalità di compatibilità con FidoCad** nella configurazione di *FidoCadJ*. Potete trovare più informazioni a questo riguardo sul manuale di *FidoCadJ*.

Conclusione

In questo articolo, ho presentato brevemente la storia del formato *FidoCad* per il disegno di semplici schemi elettrici e circuiti stampati. Ho presentato il programma *FidoCad per Windows*, scritto da Lorenzo Lutti e diffusosi presso la comunità Usenet italiana a partire dalla fine degli anni 90. In seguito, ho brevemente descritto il progetto *FidoCadJ*, scritto in Java, completamente multiplatforma, internazionalizzato e liberamente scaricabile da SourceForge con licenza GPL v. 3.

Grazie al lavoro del webmaster, *FidoCadJ* è utilizzato sul server di ElectroPortal per convertire automaticamente il codice *FidoCad* introdotto all'interno degli interventi del forum, all'interno dei tag [fcd] e [/fcd]. Mi auguro che questa funzionalità contribuisca a rendere il formato *FidoCad* ancora più comodo da utilizzare per le vostre discussioni attorno all'elettronica e non solo. Per quanto riguarda *FidoCadJ* invece, se il programma vi piace non esitate a fargli pubblicità ed a postare magari una recensione su SourceForge o Ohloh. Se avete idee interessanti, volete segnalare un bug o volete partecipare al progetto *FidoCadJ*, non esitate a contattarmi!

Buon lavoro!

Estratto da "<https://www.electroyou.it/mediawiki/index.php?title=UsersPages:Darwinne:fidocadj>"