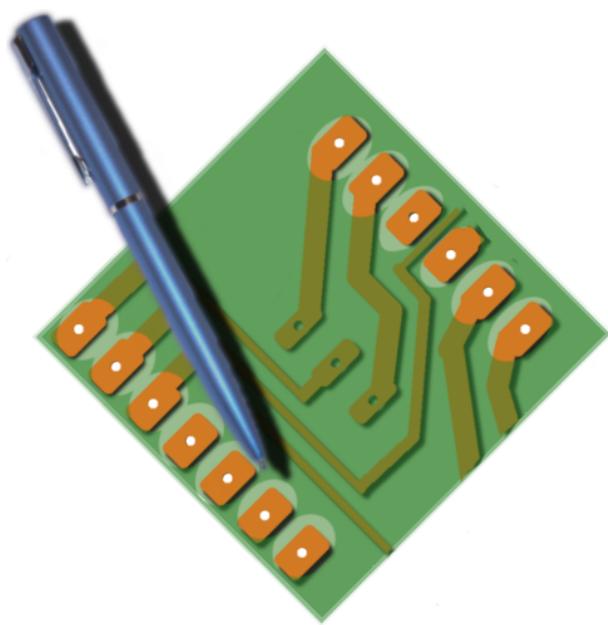


FidoCadJ 0.24.8 gebruikers handleiding

Davide Bucci



31 mei 2023

Deze handleiding valt onder de Creative Commons Public License versie 2.5 of recenter. De volledige tekst van deze licentie is beschikbaar op het adres <http://creativecommons.org/licenses/by-nc-nd/3.0/>.

U bent vrij om dit werk te reproduceren, te verspreiden, te communiceren of in het openbaar te vertonen, te vertegenwoordigen, uit te voeren en af te spelen onder de volgende voorwaarden:

Naamsvermelding U moet het werk toeschrijven op de manier die is gespecificeerd door de auteur of licentiegever (maar op geen enkele manier die suggereert dat zij u of uw gebruik van het werk steunen).

Niet-commercieel U mag dit werk niet voor commerciële doeleinden gebruiken.

Geen afgeleide werken U mag dit werk niet veranderen, transformeren of erop voortbouwen.

Van elk van de bovenstaande voorwaarden kan worden afgeweken als u toestemming krijgt van de auteursrechthebbende (Davide Bucci).

Alle commerciële namen, logo's en handelsmerken die in dit werk worden genoemd, zijn geregistreerd door hun eigenaren.

Abstract

Dit document is de FidoCadJ gebruikershandleiding. Na een korte inleiding over de geschiedenis en het ontstaan van deze software wordt het basisgebruik van FidoCadJ uitgelegd. Ons doel is om te laten zien hoe je zeer eenvoudige elektronische schema's en printplaten kunt tekenen. De handleiding eindigt met een gedetailleerde beschrijving van het FidoCadJ-formaat dat eerder ook door FidoCAD werd gebruikt. Tot slot geven we enkele tips over het downloaden en installeren van FidoCadJ met behulp van de meest voorkomende besturingssystemen: Linux, macOS en Windows.

Dit document presenteert alleen het programma zelf en de bestandsinleiding. Het gaat niet over de integratie van FidoCadJ in websites, noch over de organisatie van de code en de aspecten van ontwikkeling. Raadpleeg de GitHub-repository als u wilt deelnemen aan de ontwikkeling!

Dankbetuigingen

Een aantal mensen hebben deze software sinds de eerste versies gebruikt en mij met hun adviezen geholpen. Ik wil dus de [it.hobby.elettronica](#) newsgroup deelnemers bedanken voor de zeer vruchtbare discussies die we samen hadden.

Deze software is zeer zorgvuldig getest op Linux dankzij het geduld van Stefano Martini. Hij was een zeer attente alfa en beta tester! Ik wil Olaf Marzocchi en Emanuele Baggetta bedanken voor hun tests op macOS.

Ik wil graag “F. Bertolazzi”, die zeer nuttige adviezen gaf over de bruikbaarheid van deze software, bedanken. Hij heeft ook de CadSoft Eagle compatibiliteits bibliotheek samengesteld, handig om FidoCadJ-tekeningen naar Eagle te exporteren. Veel dank aan “Celsius”, die de softwarefunctionaliteiten voor PCB’s en de bijbehorende bibliotheken heeft getest.

Dank ook aan Andrea D’Amore voor zijn adviezen over de FidoCadJ gebruikersinterface op de Apple Macintosh. Ik wil Roby IZ1CYN bedanken voor de discussies over bibliotheken en voor het schrijven van sectie [A.2](#) van deze handleiding en over de installatie ervan op Linux. Met dank aan Max2433BO voor advies over Linux en voor zijn geduldige werk aan GitHub-problemen.

Macintosh-gebruikers kunnen FidoCadJ gebruiken dat goed geïntegreerd is in het uiterlijk van hun besturingssysteem dankzij de Quaqua uitlerijk en beleving, geleverd door Werner Randelshofer en meer recentelijk dankzij de VAqua7 uiterlijk en beleving. Werner heeft ook nuttige adviezen gegeven over de gebruikersinterface van FidoCadJ: bedankt daarvoor!

Een deel van deze handleiding is vertaald naar het Engels door “Pasu”. Ik wil hem bedanken voor zijn harde werk. Ik wil Miles “qhg007” ook bedanken voor de zorgvuldige controle en zeer nuttige opmerkingen. De Engelstalige handleiding is vertaald naar het Nederlands door Joop Nijenhuis als dank je wel voor FidoCadJ. De Nederlandstalige handleiding is met behulp van o.a. Google vertaald naar het Duits.

In april 2010 is FidoCadJ geïntegreerd in de bekende Italiaanse website www.electroyou.it. Het draait nu geruisloos op de server en converteert automatisch tekeningen die op het forum zijn geplaatst. Ik wil mijn dank uitspreken aan de beheerder Zeno Martini en webmaster Nicolò Martini en aan alle gebruikers van deze site. Ze gaven me veel zeer nuttige ideeën over batchcontrole FidoCadJ: er is een veelbelovend pad uitgestippeld dat het verdient om volledig te worden onderzocht. FidoCadJ wordt ook gebruikt op <http://www.matematicamente.it>. Het is gebruikt door de populaire website www.grix.it dankzij de inspanningen van Arniek, Sstrix en Stan. Veel complimenten voor hen!

Nuttige hints over de website van FidoCadJ kwamen van Federica Garin en Emanuele Baggetta. Sergio Juarez heeft er ook veel aan bijgedragen door de technische aspecten en de organisatie te verbeteren.

FidoCadJ Licentie

Copyright © 2007-2023 het FidoCadJ ontwikkelteam.

Deze software is gratis: u kunt deze opnieuw distribueren en/of wijzigen onder de voorwaarden van de GNU General Public License zoals gepubliceerd door de Free Software Foundation, versie 3 van de licentie.

Dit programma wordt verspreid in de hoop dat het nuttig zal zijn, maar ZONDER ENIGE GARANTIE; zonder zelfs de impliciete garantie van VERKOOPBAARHEID of GESCHIKTHEID VOOR EEN BEPAALD DOEL. Zie de GNU General Public License voor meer details.

U zou samen met dit programma een kopie van de GNU General Public License moeten hebben ontvangen. Zo niet, zie <http://www.gnu.org/licenses/>.

Inhoudsopgave

1 Inleiding	1
1.1 De filosofie van FidoCadJ	1
1.2 Geschiedenis van FidoCadJ	2
1.3 FidoCadJ en de toekomst	4
2 Tekenen met FidoCadJ	6
2.1 Teken hulpmiddelen	6
2.2 Een eenvoudig schema	10
2.3 De lagen	15
2.4 Het raster	15
2.5 Een eenvoudige printplaat	15
2.6 De liniaal gebruiken	22
2.7 Pijl- en lijnstijlen	22
2.8 Exporteren	24
2.9 Opdrachtregel opties	26
2.10 Beheer van de bibliotheek	29
2.10.1 Bibliotheekbestanden gebruiken	29
2.10.2 Nieuwe symbolen definiëren	30
2.10.3 Wijzigen van bestaande symbolen	33
3 Tekenformaat, macro's & bibliotheek	34
3.1 Koptekst beschrijving	34
3.2 Coördinaten systeem	34
3.3 Elementen tekenen	35
3.4 FidoCadJ extensies	41
3.4.1 Laag instelling	42
3.4.2 Instelling elektrische aansluiting	43
3.4.3 Lijnbreedte	43
3.5 Tolerantie voor syntaxfouten	43
3.6 Bibliotheek formaat	44
3.7 Standaard bibliotheken	44
4 Conclusie	46
A Platformspecifieke informatie	47
A.1 macOS	47
A.1.1 Hoe FidoCadJ op macOS te downloaden en uit te voeren	47
A.2 Linux	48

A.2.1	Via elk platform, vanaf de terminal	48
A.2.2	Op een grafisch systeem	49
A.2.3	“Portable” installatie	50
A.2.4	Raspberry Pi 400	51
A.3	Windows	52
	A.3.1 Hoe FidoCadJ te downloaden en uit te voeren	52
A.4	Compileren van sources	52
A.5	Android	53
B	FidoCadJ voorbeelden	55
C	FidoCadJ art	61

Lijst van figuren

1.1	Deel van een van mijn berichten op www.electroyou.it . Met slechts één klik zoomt u in op het schema. Met een tweede kunt u onmiddellijk de broncode verkrijgen die u op FidoCadJ kunt plakken om deze te wijzigen.	4
1.2	Een voorbeeld van een FidoCadJ-tekening geïntegreerd in een www.grix.it forumbericht. U kunt de broncode van de tekening met slechts één klik opvragen.	5
2.1	De drie menu balken van FidoCadJ 0.24.8.	6
2.2	Een FidoCadJ 0.24.8 sessie dat draait op een macOS 10.13 High Sierra. Appendix A beschrijft de eigenaardigheden van de versie voor Macintosh-computers.	7
2.3	Een FidoCadJ 0.24.8 sessie dat draait op een Raspberry Pi met aanzien en beleving Metal.	7
2.4	De snelzoekfunctie voor de geïnstalleerde bibliotheken.	8
2.5	Dialoogvenster voor de parameters van een symbool in een FidoCadJ-tekening. Het besturingssysteem is Raspberry Pi OS, in het Nederlands. Druk op de knop linksonder om symbolen in te voeren of hints te krijgen.	10
2.6	Het referentieschema: stroomspiegel gemaakt met NPN-transistors.	10
2.7	We beginnen met het tekenen van een paar transistors.	11
2.8	Selecteer en spiegel de transistor aan de linkerkant door te drukken op S	11
2.9	We zijn te dicht bij de bovenrand van het blad: selecteer de hele tekening en verplaats deze naar het midden.	12
2.10	Het circuit is bijna voltooid.	13
2.11	Het complete circuit.	13
2.12	Een zeer eenvoudige versterkertrap die een NPN-transistor gebruikt die is aangesloten in een gemeenschappelijke emitterconfiguratie.	16
2.13	De belangrijkste onderdelen worden op het bord geplaatst.	17
2.14	Voedingsaansluitingen toegevoegd met polygonale lijnen.	17
2.15	De resterende verbindingen toegevoegd PCB-sporen.	18
2.16	De printplaat is bijna voltooid.	18
2.17	De klus geklaard met de zeefdruk.	19
2.18	De printplaat, zoals deze wordt afgedrukt (gespiegeld) op een ISO-UNI A4-vel.	20

2.19 Klik met de rechtermuisknop en sleep om de FidoCadJ-liniaal te activeren.	22
2.20 Een elektrische tekening (een GIC van Antoniou) waarin enkele FidoCadJ-extensies zijn gebruikt.	23
2.21 Het parametervenster van de getoonde Bézier-curve gebruikt in de schema's van figuur 2.20.	23
2.22 Het exporteer naar afbeelding instel menu.	24
2.23 Het uiterlijk van FidoCadJ, met behulp van de Motif look & feel.	29
2.24 Het pop-upmenu dat verschijnt als u met de rechtermuisknop klikt maakt het mogelijk om tekenelementen om te zetten in een symbool	31
2.25 Het nieuwe dialoogvenster voor symbooldefinities. Hier kunt u alle belangrijke kenmerken van het symbool instellen. Let op de oorsprong gedefinieerd door de twee rode assen.	31
2.26 Het nieuw aangemaakte symbool, weergegeven in de symbolenlijst en in de tekening. Aan de linkerkant staat nog steeds de tekening die is gebruikt voor de symbooldefinitie. Merk op dat er slechts één controlepunt aanwezig is voor het nieuwe bibliotheeksymbool in de tekening.	32
2.27 Het pop-up menu dat wordt gebruikt voor het wijzigen van symbool eigenschappen in een gebruikersbibliotheek.	33
3.1 Figuur 2.20 zoals het zou verschijnen op FidoCAD voor Windows.	42
3.2 Voorbeeld van hoe bibliotheken in de bibliothekenmap verschijnen.	45
A.1 Het instellen van bestandsrechten, op Ubuntu 8.04.	50
A.2 FidoCadJ voor Android, draaiend op een Samsung S5-smartphone.	53
B.1 Voorbeeld van gebruiker DanteCpp (Oostenrijk)	56
B.2 Een voorbeeld van een FidoCadJ tekening met andere inzet van de beschikbare lagen (lees kleuren).	57
B.3 Modulair samengesteld effecttrack. Door op de pagina in te zoomen (>200%) kunnen knoppen en teksten beter leesbaar worden.	57
B.4 Tuner samengesteld uit mengversterkereenheden van Philips. Door op de pagina in te zoomen (>200%) kunnen knoppen en teksten beter leesbaar worden.	58
B.5 Voorbeeld van een geluidsstudio indeling (de werkelijke opzet is veel gecompliceerder en past niet op een A4, het idee zal wel duidelijk zijn). Door op de pagina in te zoomen (>200%) kunnen teksten beter leesbaar worden.	59
B.6 Mijn poging voor een nieuwe indeling van de woonkamer.	60

Lijst van tabellen

2.1	Samenvatting van de tekenopdrachten die beschikbaar zijn in FidoCadJ. De toets die in de meest linkse kolom wordt weergegeven, maakt een snelle selectie met het toetsenbord mogelijk. Klik met de rechtermuisknop in een van de primitieve plaatsingsmodi om toegang te krijgen tot de eigenschappen ervan.	9
2.2	Deze code beschrijft het circuit uit ons voorbeeld.	14
2.3	Deze code beschrijft de printplaat uit ons voorbeeld.	21
2.4	Lijst met alle exportbestandsindelingen die beschikbaar zijn in FidoCadJ.	25
3.1	Betekenis van de parameter a voor de aanwezigheid van een pijlpunt aan de zijkanten van een lijn of Bézier-primitief.	36
3.2	Betekenis van de parameter b voor de pijlpuntstijl.	36
3.3	Functie van de bits in de term tekststijl.	37
A.1	Lijst met beschikbare make doelen om FidoCadJ uit bronnen te compileren.	53

Hoofdstuk 1

Inleiding

In dit hoofdstuk zullen we FidoCadJ kort introduceren. We zullen in het bijzonder een beschrijving geven van de filosofie achter deze software, evenals een korte geschiedenis van de ontwikkeling ervan. Als u FidoCadJ moet installeren en uitvoeren, ga dan naar appendix A, waarin wordt beschreven hoe u FidoCadJ installeert op de meest voorkomende besturingssystemen.

1.1 De filosofie van FidoCadJ

FidoCAD (zonder de J aan het einde) was een vectortekensetware die bijzonder geschikt was voor elektrische schema's en voor eenvoudige printplaten. Het werd eind jaren negentig gebruikt door de Italiaanse Usenet-gemeenschap.

Het kan nog steeds gratis worden gedownload (een Windows-versie genationaliseerd in de Italiaanse taal) van Lorenzo Lutti's pagina:

<http://www.enetsystems.com/~lorenzo/fidocad.asp>

De uitvoerbestanden die door FidoCAD werden gegenereerd waren een zeer compacte tekstcode. Deze functie maakte het heel gemakkelijk om tekeningen in tekstberichten op te nemen, zoals die worden gebruikt in niet-binaire Usenet-groepen.

Helaas bestond FidoCAD alleen in een Windows-versie. Linux-gebruikers konden het gebruiken met WInE, maar degenen die afhankelijk waren van andere platforms zoals ik (ik gebruik macOS) moesten een andere oplossing vinden. Ik besloot daarom een kleine bijdrage te leveren aan de gemeenschap door FidoCadJ te schrijven (met de laatste J, deze keer). Deze editor is geschreven in pure Java en is volledig multi-platform. Met FidoCadJ kunt u een tekening weergeven en wijzigen met behulp van de bestandsindeling FidoCAD. Tegenwoordig werk ik niet meer alleen aan het project en is FidoCadJ een volwaardig en ik durf te schrijven een succesvol Open Source project met bijdragen van over de hele wereld.

Wie in het verleden FidoCAD heeft gebruikt, zou heel snel bekend moeten zijn met FidoCadJ, aangezien veel commando's en procedures vrijwel gelijkaardig zijn aan de originele applicatie. FidoCadJ is bijna volledig compatibel met de originele FidoCAD, op enkele details na. Het doel van volledige compatibiliteit is nagestreefd voor zover mogelijk, maar vragen van nieuwe gebruikers hebben geleid tot enkele uitbreidingen van het oorspronkelijke formaat.

Enkele functies die door FidoCadJ worden aangeboden en niet aanwezig zijn in de originele FidoCAD, zijn de beschikbare exportmogelijkheden. Aangezien ik een L^AT_EX-gebruiker ben, heb ik besloten een exportfunctie op te nemen voor een aantal vectorformaten, waaronder Encapsulated PostScript (EPS) en scripts voor de PGF pakket. Natuurlijk kan FidoCadJ exporteren naar het zeer bekende PDF formaat, met enkele beperkingen.

1.2 Geschiedenis van FidoCadJ

Ik ben al lang geïnteresseerd in elektronische schakelingen. Toen ik verschillende speciale Italiaanse Usenet-nieuwsgroepen begon te volgen, merkte ik dat er veel schema's werden verstrekt met behulp van het FidoCAD voor Windows-formaat. Dit was veel beter dan onhandige ASCII-tekeningen. Aangezien ik geen Windows gebruik, was het voor mij bijna onmogelijk om ernaar te kijken en ik wilde proberen iets te doen om dit probleem op te lossen.

Het eerste dat ik eind 2006 deed, was het bestuderen van het bestandsformaat dat wordt gebruikt door FidoCAD en het schrijven van een Java-applet genaamd FidoReadJ. Het was in staat om het circuit te ontleden om het in een webpagina te tonen. Het internet was toen anders! Om dat te doen, heb ik min of meer overal gezocht (oude berichten, webpagina's), en heb ik veel reverse-engineering gedaan vanuit bestaande FidoCAD-bestanden. Ik heb ook de FidoCAD-bronnen gedownload en bestudeerd, geschreven in een vrij duidelijke C++ door Lorenzo Lutti.

Ik schreef de kern van de applet in maart 2007. Een paar maanden later was de applet online en werd hij getest door een deel van de gemeenschap die zich aangetrokken voelde tot it.hobby.elettronica en it.hobby.fai-da-te.¹

Aangezien ik een interpreter van het FidoCAD-formaat had, was het interessant om een complete editor te maken. Veel van het initiële werk werd in verschillende stappen gedaan, tussen januari en juli 2008: FidoCadJ is geen port van FidoCAD voor Windows, maar het is een nieuw programma, helemaal opnieuw geschreven.

De keuze om Java te gebruiken is vanwege het feit dat ik momenteel veel verschillende besturingssystemen gebruik. Tijd en energie steken in iets dat niet helemaal overdraagbaar is spreekt mij niet meer aan. De inspanning om het Cocoa-framework te leren zou waarschijnlijk een beter resultaat hebben opgeleverd op macOS, maar het zou FidoCadJ volledig niet overdraagbaar hebben gemaakt. Ik ben geen computerman. De tijd die ik besteed aan het programmeren is tijd die wordt weggenomen van mijn elektronische interesses. In feite, vooral in het begin, toonde de FidoCadJ-codebron aan dat ik niet erg Java en objectgeoriënteerd programmeren purist ben en verschillende oplossingen waren meer praktisch dan elegant. Ik heb echter geprobeerd om de algehele codekwaliteit van FidoCadJ continu te verbeteren. Het is behoorlijk succesvol vergeleken met andere Open Source-projecten, tenminste bij één gelegenheid². Op deze

Trouwens, ik denk dat het beter is om actief aan een oplossing te werken in plaats van zeuren, omdat een ander besturingssysteem dan Windows bepaalde software mist.

Om eerlijk te zijn, deed ik rond 1993 een eerste poging om een 2D-vectortekensysteem te schrijven

¹FidoReadJ is nu nog steeds beschikbaar op het volgende adres, maar het is onwaarschijnlijk dat beschikbare webbrowsers Java applets nog steeds ondersteunen:
<http://davbucci.chez-alice.fr/index.php?argument=elettronica/fidoreadj/fidoreadj.inc>.

²Zie bijvoorbeeld Rahman, M. M., Roy, C. K., and Keivanloo, I. "Subjective Evaluation of Software Quality Using Crowdsource Knowledge: An Exploratory Study". Univ. of Saskatchewan, Department of Computer Science technical report 2013-01

manier doorgaand, is statische en dynamische analyse van de broncode gebruikt om aan te geven waar de code-organisatie kan worden aangepast en verbeterd.

Waar het om gaat zijn de ervaringen van degenen die het programma gebruiken, veel meer dan de keuze van een bepaalde taal. Om deze reden luister ik altijd naar uw suggesties, om te begrijpen hoe dit project verder kan worden ontwikkeld. Java is misschien niet de perfecte keuze of de oplossing voor elk probleem. Het is echter flexibel genoeg gebleken voor een tool als FidoCadJ.

Zonder te streven naar perfectie en de grenzen van mijn programmeervaardigheden te kennen, is het mijn bedoeling ervoor te zorgen dat FidoCadJ GEEN applicatie van slechte kwaliteit zal zijn. Om deze reden is elk bugrapport of commentaar over de bruikbaarheid van het programma meer dan welkom.

In november 2009 opende ik een SourceForge-project gewijd aan FidoCadJ. Vanaf deze plek kon je alle uitvoerbare bestanden, handleidingen en de broncode downloaden. Helaas heeft SourceForge sinds 2013 geleidelijk enkele duistere praktijken overgenomen: advertenties met valse downloadknoppen verschenen overal en crap-ware toegevoegd aan installatieprogramma's voor sommige projecten. In 2015 was de site ongeveer 15 dagen offline: het werd duidelijk dat het tijd was om te veranderen en we migreerden het FidoCadJ-project op GitHub. Vanaf augustus 2015 wordt alle ontwikkeling op FidoCadJ gedaan met Git: voel je vrij om FidoCadJ te forken en de bronnen te bekijken. De SourceForge-pagina blijft echter bestaan, omdat iemand nog steeds graag pakketten van SourceForge laadt (maar de broncode is niet up-to-date).

Een grondige herziening van de bestaande code is tussen 2013 en 2014 door Kohta Ozaki en mijzelf gedaan, en werd vervolgens geleidelijk voortgezet in 2015. Dit maakte het mogelijk om het programma over te zetten naar Android-platforms, wat mogelijk werd gemaakt door de bijdrage van andere mensen, in het bijzonder Dante Loi. Vele andere personen hebben op dat moment een handje geholpen. Het werk aan Android loopt helaas sinds 2016 vast, ondanks dat de port succesvol was en een pre-release bèta klaar en gedistribueerd was.

Als u de gemeenschap wilt helpen met FidoCadJ, maakt u geen zorgen: u hoeft geen ervaren Java-programmeur te zijn. U kunt bijvoorbeeld de gebruikersinterface of de handleidingen in een nieuwe taal vertalen, de documentatie zorgvuldig controleren op inconsistenties, typefouten en fouten. FidoCadJ is momenteel beschikbaar in het Italiaans, Frans, Engels, Spaans, Duits, Chinees, Japans, Grieks, Tsjechisch en Nederlands. Sommige vertalingen zijn mogelijk niet meer up-to-date en vereisen enig werk van de gemeenschap. Zoek tussen de problemen op GitHub (met de tag "translation") om te zien wat er nodig is. Het is vrij eenvoudig om menu's en commando's te vertalen, dus als u FidoCadJ in uw moedertaal wilt zien, maak dan een nieuw "issue" aan op GitHub!

Er is ook werk aan de winkel om de bibliotheken (tenminste de standaard) en natuurlijk deze handleiding te controleren. U kunt ook aan de standaardbibliotheek... werken. Ik ben van mening dat slechts de helft van de tijd die ik aan FidoCadJ besteed in werkelijkheid besteed is aan de codering zelf. De rest wordt besteed aan het beantwoorden van vragen van gebruikers, het schrijven en verbeteren van de documentatie en dergelijke. Op GitHub kunt u deelnemen aan de discussies, verbeteringen voorstellen of een bugrapport geven. Aarzel niet om deel te nemen:

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

1.3 FidoCadJ en de toekomst

In april 2010 stuitte ik bijna per ongeluk op een thread op een bekende Italiaanse website www.electroyou.it. Piercarlo Boletti stelde daar voor om een schematisch opname hulpmiddel rechtstreeks in het forum te integreren, aangezien iets soortgelijks al is gedaan met L^AT_EX-vergelijkingen. Omdat FidoCadJ werd geciteerd in zijn tussenkomst, heb ik een account aangemaakt om in het forum te kunnen schrijven en heb ik aangeboden om samen te werken. Ik heb een paar dagen contact gehad met de zeer aardige webmaster en het systeem was klaar: een simpel kopiëren en plakken van de FidoCadJ-code en een paar tags. Het forum draait automatisch FidoCadJ op zijn servers en maakt direct een afbeelding uit de code. Deze afbeelding wordt dan getoond in de forumpost, maar de broncode blijft altijd beschikbaar. Op deze manier hoeft degene die de discussie leest niets te hebben geïnstalleerd, maar als men iets wil veranderen, kan de code met een paar klikken worden verkregen.

Figuur 1.1 toont een deel van een van mijn berichten op www.electroyou.it. Dit systeem is zo krachtig en flexibel dat ik de eerste was die verbaasd was over het enthousiasme van de gebruikers! ³

The screenshot shows a forum post titled "[12] Re: Orologi e orologiai" by DarwinNE. The post contains a message from TardoFreak and a detailed schematic diagram of an electronic circuit. The schematic includes various components like transistors (Q1-Q8), resistors (R1-R12), capacitors (C1-C5), and diodes (D1-D4). Power supplies of +17V and -17V are shown. A note at the bottom of the post reads: "Penso proprio che i ragazzi abbiano intenzione di lasciare fuori i microcontrollori, per fare un ottimo esercizio di stile. Perché non utilizzare un contatore analogico? Anni fa ne avevo realizzato uno per un tracciacurve, è un circuito un bel po' critico, ma abbastanza affascinante. Penso che parte della criticità venga dal fatto che ho simulato degli unijunzione programmabili con una coppia di transistor NPN+PNP (Q2 e Q3, Q7 e Q8). Inoltre, i generatori di corrente che ho adottato sono probabilmente migliorabili." Below the schematic is a note: "Sorgente FidoCadJ | Ingrandisci | Cos'è Fidocad".

Figuur 1.1: Deel van een van mijn berichten op www.electroyou.it. Met slechts één klik zoomt u in op het schema. Met een tweede kunt u onmiddellijk de broncode verkrijgen die u op FidoCadJ kunt plakken om deze te wijzigen.

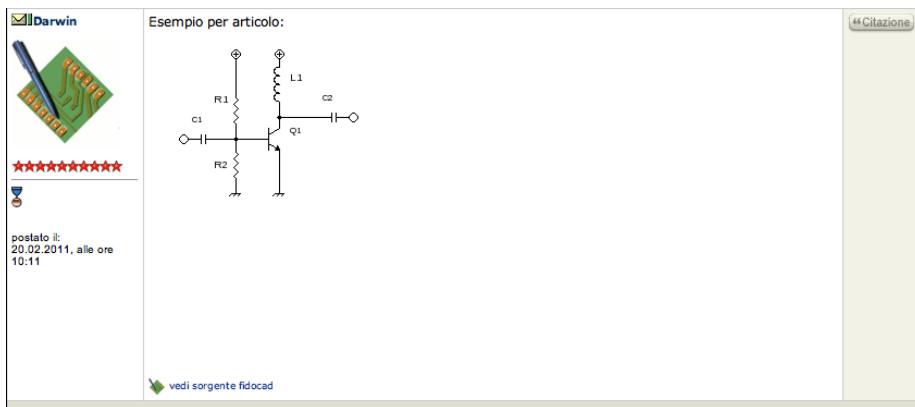
³ Als je Italiaans kunt lezen, hier is een artikel dat ik heb geschreven: <http://www.electroyou.it/darwinne/wiki/fidocadj>

FidoCadJ is ook nuttig gebleken voor tekeningen die niet direct met elektronica te maken hebben (zie bijlagen B en C om te zien wat ik bedoel).

Het succes dat FidoCadJ had op www.electroyou.it stimuleerde enkele verzoeken om een soortgelijk systeem op andere platforms te implementeren, en in het bijzonder op www.grix.it, een andere bekende Italiaanse elektronicawebsite. FidoCadJ wordt ook gebruikt door de **Matematicamente** gemeenschap. In het geval van www.grix.it, kon de server geen Java-programma uitvoeren en daarom is de FidoReadPHP-klasse geschreven. Het kan worden uitgevoerd op een PHP-interpreter om de afbeeldingen van de FidoCadJ-broncode te verkrijgen. Het FidoReadPHP-project is open source en is beschikbaar op SourceForge:

<https://sourceforge.net/projects/fidoreadphp/>

maar het wordt momenteel helaas niet actief onderhouden. Het resultaat is enigszins vergelijkbaar met wat het is verkregen op www.electroyou.it, zelfs als de grafische mogelijkheden van PHP vrij beperkt zijn met betrekking tot Java⁴. Figuren 1.2 toont een voorbeeld van een tekening verkregen met FidoReadPHP.



Figuur 1.2: Een voorbeeld van een FidoCadJ-tekening geïntegreerd in een www.grix.it forumbericht. U kunt de broncode van de tekening met slechts één klik opvragen.

Ik denk dat de toekomst van FidoCadJ niet complexer zal worden, als een complete CAD voor elektronica. In plaats daarvan verdient de integratie met discussiegroepen en forums te worden ontwikkeld. De ervaring leert dat dit mogelijk is en het enthousiasme van de gebruikers prikkelt.

⁴Nogmaals, als je Italiaans kunt lezen, hier is een artikel:
<http://www.grix.it/viewer.php?page=9335>

Hoofdstuk 2

Tekenen met FidoCadJ

Expert Mac-gebruikers zullen merken dat alle menu's op hun eigen plaats staan!

Het gebruik van FidoCadJ zou vrij intuïtief moeten zijn voor wie al een vector tekening programma gebruikt. Een schermafbeelding van het programma dat draait op macOS wordt getoond in Fig. 2.2; Een paar details kunnen anders zijn bij gebruik op andere besturingssystemen (bijvoorbeeld Fig. 2.3 toont het resultaat met aanzien en beleven Metal op een Raspberry Pi 400), maar de filosofie blijft hetzelfde. We zullen de kenmerken van het programma beschrijven, evenals de basiselementen (de primitieven), waaruit een FidoCadJ-tekening bestaat. Onderstaand een afbeelding met daarin de drie werkbalken en hoe deze worden aangegeven in FidoCadJ.



Figuur 2.1: De drie menu balken van FidoCadJ 0.24.8.

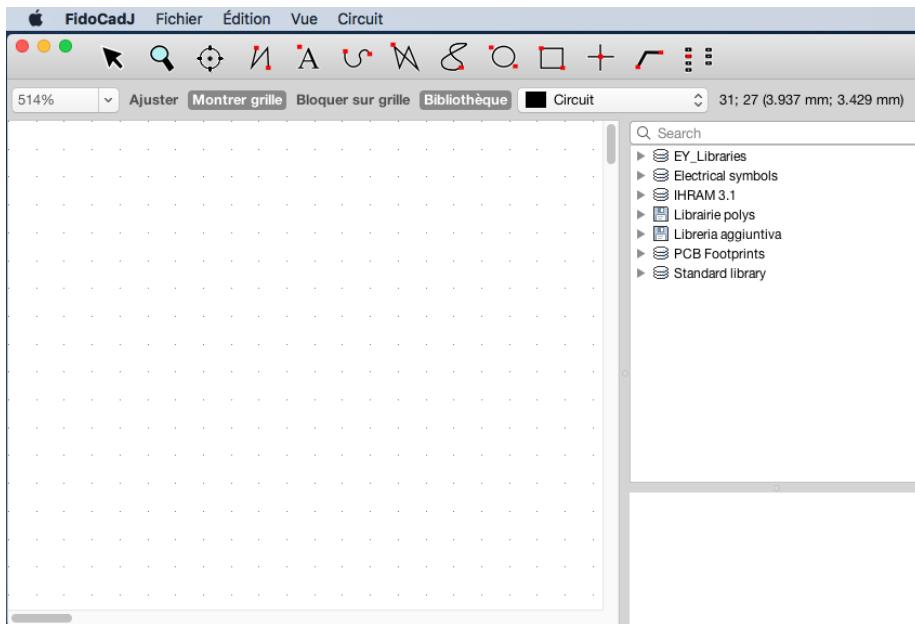
2.1 Teken hulpmiddelen

In de gereedschapbalk bovenaan het venster kunnen we de meest gebruikte functies vinden voor het maken en bewerken van een tekening. Table 2.1 toont een korte samenvatting van de functionaliteiten, de opdrachten en de mogelijke acties. U zult merken dat als een knop in de gereedschapsbalk eenmaal is ingedrukt, deze op die positie blijft totdat een andere functie van de gereedschapsbalk wordt geselecteerd. In de gereedschapbalk kunnen we selecteren welk tekening element zal worden gebruikt¹. In de Werkvlakbalk wordt rechts de huidige werklaag getoond.(zie 2.3 voor meer informatie).

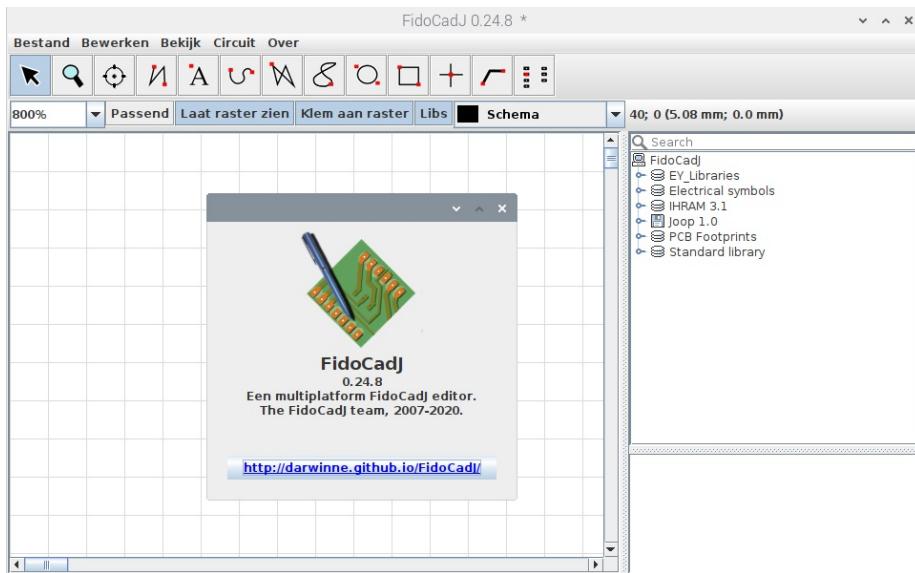
Dit gedrag in gebruikersinterfaces wordt meestal "drukknoppen" genoemd en is geïnspireerd op de oude buizenradio's en de schakelaars die tot 1970 in de mode waren.

De gereedschapbalk kan gedeeltelijk worden aangepast. We kunnen kiezen tussen alleen pictogrammen of pictogrammen met een omschrijving. De pictogrammen zijn in twee formaten beschikbaar. Om deze instellingen te wijzigen

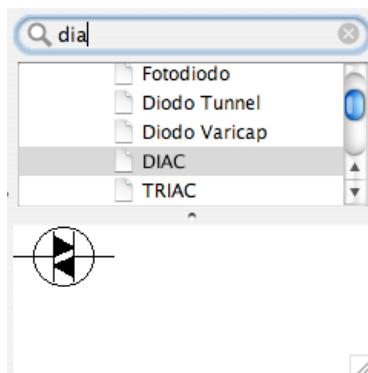
¹Voor meer informatie over de tekenelementen, zie 3.3.



Figuur 2.2: Een FidoCadJ 0.24.8 sessie dat draait op een macOS 10.13 High Sierra. Appendix A beschrijft de eigenaardigheden van de versie voor Macintosh-computers.



Figuur 2.3: Een FidoCadJ 0.24.8 sessie dat draait op een Raspberry Pi met aanzien en beleving Metal.



Figuur 2.4: De snelzoekfunctie voor de geïnstalleerde bibliotheken.

kunt u in de menubalk het menu “Bestand/Opties” selecteren ². Dergelijke wijzigingen in de instellingen worden pas van kracht na een herstart van FidoCadJ omdat het niet denkbaar is dat we deze instellingen elke dag willen veranderen. Figuren 2.2 en 2.3 tonen de gereedschapbalk geconfigureerd om geen tekst en pictogrammen in hun normale omvang weer te geven. In de Werkvlakbalk kunnen we de zoominstellingen zien en de knoppen “Passend”, “Toon raster”, “Snap to grid” en “Libs”. De eerste maakt het mogelijk om automatisch de meest geschikte zoominstellingen te selecteren om de hele tekening op het scherm te tonen. De tweede schakelt tussen een zichtbaar en onzichtbaar raster. De derde knop regelt het snappen, d.w.z. de automatische uitlijning van elementen op het raster. Met de laatste kan de zichtbaarheid van het bibliotheekpaneel aan de rechterkant van het venster worden gewijzigd. Als u de elementen zorgvuldig moet uitlijnen, kan het handig zijn om **Alt** ingedrukt te houden terwijl u de cursoroefsen gebruikt: de geselecteerde objecten worden als één geheel in de gewenste richting verplaatst.

Aan de rechterkant van het hoofdvenster van FidoCadJ worden de symbolen (ook wel macro’s genoemd) weergegeven in de geladen bibliotheken, weergegeven als een boom. Om een element uit de bibliotheek in de tekening in te voegen, selecteert u het in de lijst en klikt u op de tekening. FidoCAD-bibliotheken bevatten alle standaardsymbolen die worden gebruikt in elektrische schema’s en een brede selectie van footprints voor het tekenen van PCB’s.

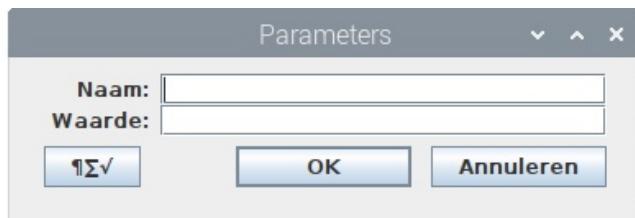
U kunt ook snel zoeken in de bibliotheken. Typ gewoon iets in het tekstveld dat net boven de bibliotheekstructuur verschijnt (kijk naar figuur 2.4). U kunt door de resultaten navigeren door return te typen.

Figure 2.5 toont een voorbeeld van wat kan worden verkregen door te dubbelklikken, in selectiemodus, op een tekenelement (in dit geval een transistor symbool). Binnen dit venster is het mogelijk om alle parameters (coördinaten, rotatie...) van elk tekenelement te wijzigen. Het aspect van dit venster is afhankelijk van het geselecteerde element.

²Behalve op macOS waar dit item in het menu van FidoCadJ te vinden is onder de aanduiding “Voorkeuren”.

Tabel 2.1: Samenvatting van de tekenopdrachten die beschikbaar zijn in FidoCadJ. De toets die in de meest linkse kolom wordt weergegeven, maakt een snelle selectie met het toetsenbord mogelijk. Klik met de rechtermuisknop in een van de primitieve plaatsingsmodi om toegang te krijgen tot de eigenschappen ervan.

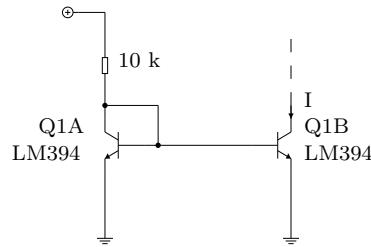
Toets	Opdracht	Gebruik
[A], [ESC] of spatie	▼ SELECT	Selecteer een of meer grafische elementen. Gebruik [Control] ([Command] in macOS) voor meerdere selecties of om slechts één element te deselecteren. Klik en sleep om meerdere elementen in een gebied te selecteren. Gebruik [R] om de geselecteerde elementen te roteren. Gebruik [S] om de geselecteerde elementen te spiegelen. Dubbelklik op een element om de eigenschappen ervan te wijzigen.
	🔍 ZOOM	Klik met de linkermuisknop om het zoomniveau te vergroten. Klik met de rechtermuisknop om het te verkleinen.
	❖ VERSCHUIVEN	Klik op de tekening en beweeg de muis om de tekening te verplaatsen.
[L]	✗ LIJN	Voeg een lijn of een reeks lijnen in. Druk op [Esc] of dubbelklik om het invoegen te beëindigen.
[T] [B] [P]	• TEKST ‣ BÉZIER ‣ VEELHOEK	Voeg een tekenreeks in. Teken een Bézier-curve Teken een veelhoek curve gevuld of leeg. Dubbelklik of druk op [Esc], om het invoegen van nieuwe hoekpunten te beëindigen.
[K]	曲线 KROMME	Open of gesloten natuurlijke kromme curve. Dubbelklik of druk op [Esc] om het invoegen van nieuwe hoekpunten te beëindigen.
[E]	○ ELLIPS	Teken een ellips gevuld of leeg (hou [Control] of [Command] in macOS ingedrukt voor een cirkel).
[G]	□ RECHTHOEK.	Teken een rechthoek gevuld of leeg (hou [Control] of [Command] in macOS ingedrukt voor een vierkant).
[C] [I]	⊕ CONNECTIE ‣ PCB SPOOR	Plaats een elektrische verbinding. Teken een PCB-spoor. De standaardbreedte kan worden gewijzigd via het dialoogvenster dat toegankelijk is via het menu “Bestand/Opties”.
[Z]	[:] PCB OOG	Tekent een PCB-oog. De standaardgrootte kan worden gewijzigd via het menu “Bestand/Opties”.



Figuur 2.5: Dialoogvenster voor de parameters van een symbool in een FidoCadJ-tekening. Het besturingssysteem is Raspberry Pi OS, in het Nederlands. Druk op de knop linksonder om symbolen in te voeren of hints te krijgen.

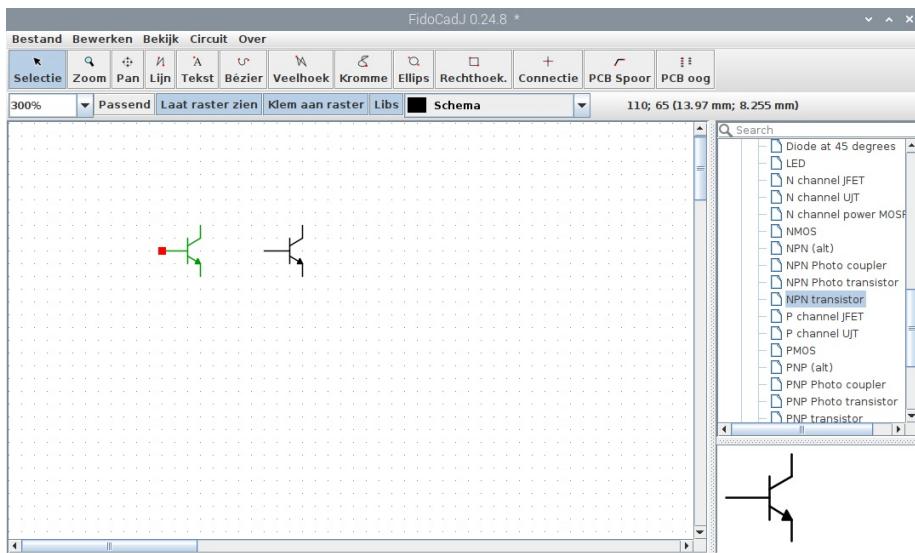
2.2 Een eenvoudig schema

Als voorbeeld van het gebruik van deze applicatie laten we zien hoe u het eenvoudige elektrische schema van de figuur kunt tekenen [2.6](#).



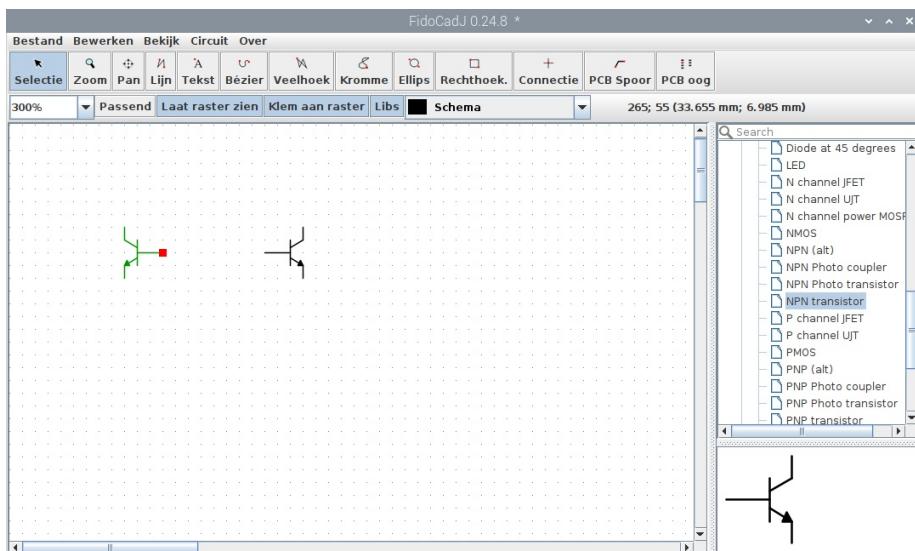
Figuur 2.6: Het referentieschema: stroomspiegel gemaakt met NPN-transistors.

Zodra FidoCadJ draait, maakt u een nieuwe tekening via de menubalk en menu “Bestand/Nieuw”. We beginnen dan met het plaatsen van de twee transistors in het Werkvlak waaromheen ons schema is opgebouwd. Om dit te doen hebben we de symbolen (ook wel macro’s genoemd) nodig die in de standaardbibliotheek staan en standaard wordt geladen. De bibliotheek bevindt zich aan de rechterkant van het venster. Het symbool dat we gebruiken wordt “NPN-transistor” genoemd en is opgenomen in de categorie “Diodes en transistors” van de “Standaardbibliotheek”. Door op de naam van het gewenste symbool te klikken wordt het symbool geselecteerd en kan vervolgens overal in de tekening geplaatst worden (FidoCadJ toont een voorbeeld) door een tweede keer op de gewenste locatie te klikken. We zouden nu in een stadium moeten zijn dat vergelijkbaar is met dat in de figuur [2.7](#).



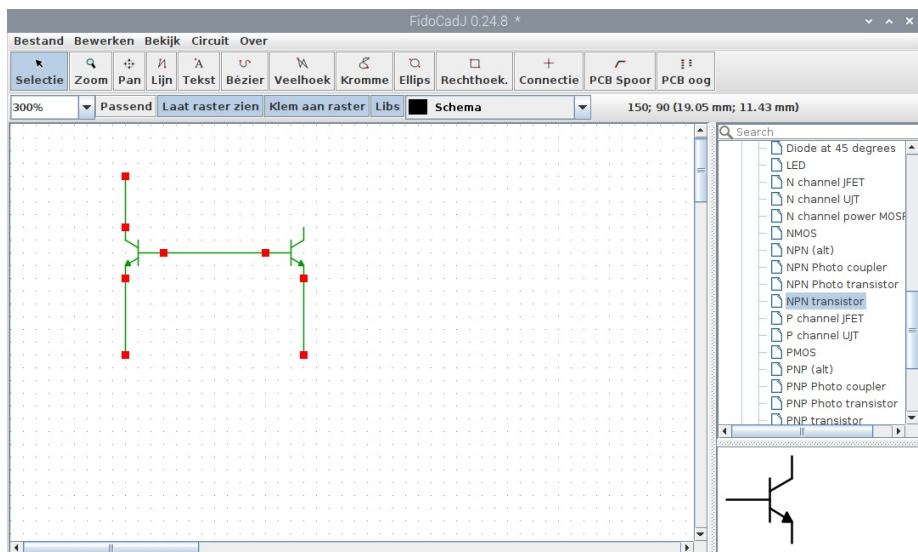
Figuur 2.7: We beginnen met het tekenen van een paar transistors.

We kunnen opmerken dat de bipolaire transistor aan de linkerkant niet correct is georiënteerd. Om het probleem op te lossen volstaat het om in de werk balk op "Selectie" te klikken, vervolgens op de transistor te klikken (die groen wordt gemarkerd, met een controlepunt geïdentificeerd door een klein rood vierkantje) en druk vervolgens op **S** om een gespiegelde versie te verkrijgen. U kunt ook op **S** drukken als u het symbool in de bibliotheken heeft geselecteerd en u het in de tekening gaat plaatsen. We krijgen dus een resultaat dat vergelijkbaar is met het resultaat dat in de figuur 2.8 wordt weergegeven.



Figuur 2.8: Selecteer en spiegel de transistor aan de linkerkant door te drukken op **S**.

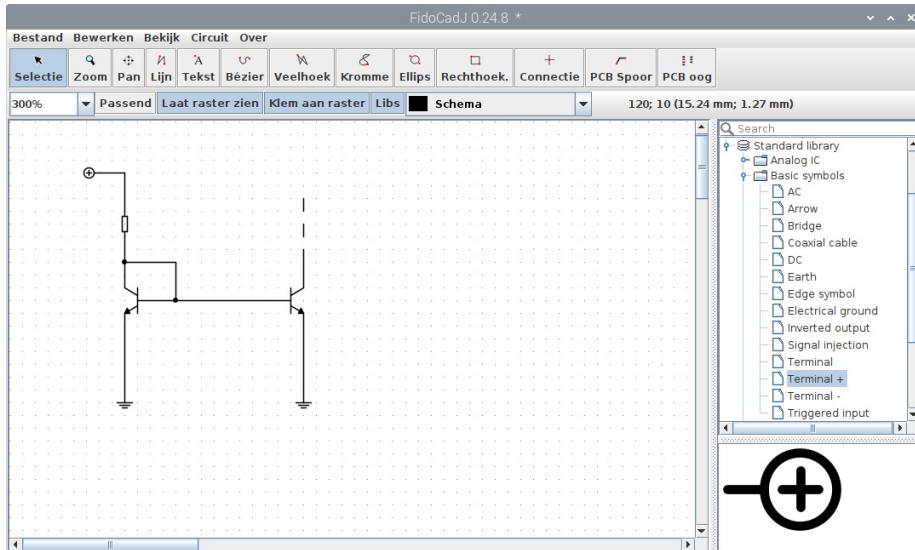
Door het gereedschap “Lijn” van de gereedschapsbalk te gebruiken, kunnen we een paar elektrische verbindingen tekenen, totdat we ons realiseren dat we met tekenen te dicht bij de randen van het Werkvlak begonnen zijn. Dit kan eenvoudig worden opgelost door de hele tekening te selecteren: in de modus “Selecteren” kunnen we op de linkerbovenhoek van de tekening klikken en, terwijl we de linkermuisknop ingedrukt houden, de cursor naar de rechteronderhoek slepen. Er verschijnt een rechthoek met een groene contour om aan te geven dat we alle elementen erin proberen te selecteren. Omdat we alles willen verplaatsen wat we tot nu toe hebben getekend, moeten we ze eerst allemaal selecteren (zie figuur 2.9). Nu, nog steeds in de selectiemodus, kunnen we op elk geselecteerd element klikken om de gehele selectie naar de gewenste positie te slepen.



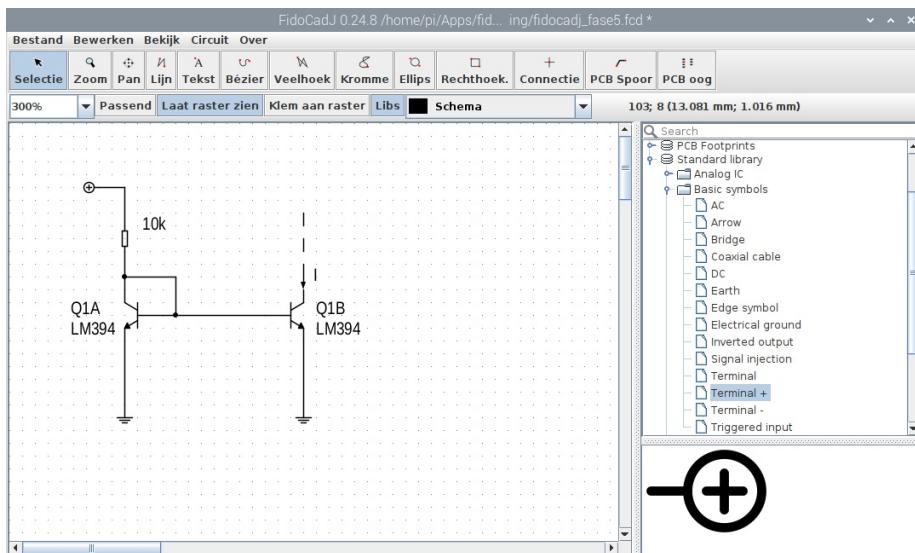
Figuur 2.9: We zijn te dicht bij de bovenrand van het blad: selecteer de hele tekening en verplaats deze naar het midden.

We kunnen dan doorgaan met het plaatsen van de andere delen van de schakeling, een weerstand (Standard Library/Discrete devices/Resistor) en het label voor de positieve voeding (Standard Library/Basic symbols/ Terminal +). We moeten de laatste draaien om in de gewenste positie te plaatsen. We kunnen het selecteren en op toets **R** drukken totdat we het gewenste resultaat hebben verkregen. We zouden nu een tekening moeten hebben die lijkt op die in de figuur 2.10.

Om het schema compleet te maken hoeven we alleen de tekstreeksen en de pijl toe te voegen om de richting van de stroom aan te geven. Voor dat laatste is er een symbool genaamd “Arrow”, opgenomen in “Standard library/Basic symbols”. Om de tekst te plaatsen, kunnen we dubbelklikken op het symbool (zie figuur 2.5). Het onderdeelnummer en de gebruikte naam van de transistoren worden gespecificeerd in de velden “Naam” en “Waarde”. De functie waarmee een naam en een waarde aan een element kunnen worden toegevoegd, is een extensie die is geïntroduceerd met FidoCadJ . Zie sectie 3.4 voor meer informatie over compatibiliteit. De tekstafmeting is 4 units verticaal en 3 units horizontaal. Het volledige circuit wordt getoond in figuur 2.11.



Figuur 2.10: Het circuit is bijna voltooid.



Figuur 2.11: Het complete circuit.

Tabel 2.2: Deze code beschrijft het circuit uit ons voorbeeld.

```
[FIDOCAD]
MC 95 65 0 0 280
FCJ
TY 115 60 4 3 0 0 0 * Q1B
TY 115 65 4 3 0 0 0 * LM394
MC 55 65 0 1 280
FCJ
TY 20 60 4 3 0 0 0 * Q1A
TY 20 65 4 3 0 0 0 * LM394
LI 55 65 95 65 0
LI 40 75 40 95 0
LI 110 75 110 95 0
LI 40 40 40 55 0
MC 40 30 0 0 115
LI 40 15 40 30 0
LI 30 15 40 15 0
MC 30 15 2 0 010
LI 40 50 60 50 0
LI 60 50 60 65 0
SA 60 65 0
SA 40 50 0
LI 110 45 110 55 0
LI 110 35 110 40 0
LI 110 25 110 30 0
MC 40 95 0 0 040
MC 110 95 0 0 040
TY 45 30 4 3 0 0 0 * 10 k
TY 115 50 4 3 0 0 0 * I
MC 110 50 1 0 074
```

Als curiositeit, de code die het circuit van ons voorbeeld beschrijft staat hierboven. Om toegang te krijgen tot deze code volstaat het om “Broncode” te selecteren in het menu “Circuit”. We zijn nu klaar om ons circuit te kopiëren en te plakken in een e-mailbericht, in een nieuwsgroep of in een forum bericht.

Als u geïnteresseerd bent in het exportformaat dat door FidoCadJ wordt gebruikt, dan vindt u een gedetailleerde beschrijving in hoofdstuk 3. Het is echter niet nodig om uit het dialoogvenster “Broncode” te gebruiken; je kunt gewoon de hele tekening selecteren, kopiëren (door “Bewerken/Kopiëren” te selecteren of door op **Ctrl+C** te drukken) en plakken in het bericht dat we aan het schrijven zijn en de code wordt automatisch toegevoegd. FidoCadJ past meestal een diagonale shift toe op de gekopieerde elementen; de *x* en *y* breedte en hoogte zijn gelijk aan de rasterafstand. Het standaardgedrag is om zo te werk te gaan, om de geplakte elementen te onderscheiden van de originele. Aangezien dit in sommige situaties problematisch kan zijn, kan deze instelling worden gewijzigd in de programmaopties.

2.3 De lagen

Een manier om lagen voor te stellen, is door te denken aan een tekening gemaakt van bovenop elkaar liggende acetaatfolie vellen. Elke laag wordt gekenmerkt door een andere kleur en kan zichtbaar of verborgen zijn. Deze aanpak is gebruikelijk in veel CAD-pakketten, omdat het een eenvoudige weergave en beheer mogelijk maakt van verschillende delen van de tekening die over elkaar zijn geplaatst, bijvoorbeeld in een printplaatontwerp. FidoCadJ staat maximaal 16 lagen toe, genummerd van 0 tot 15. Conventioneel hebben sommige lagen een specifiek doel. Laag nul wordt gewoonlijk gebruikt voor elektrische schema's, laag 1 voor de kopersoldeerzijde, laag 2 voor de kopercomponentenzijde en laag 3 voor de zeefdruk . De overige lagen hebben geen vooraf bepaald doel en kunnen vrij worden gebruikt. Naam en kleur van elke laag kan worden opgegeven met behulp van de menubalk en het menu "Bekijk/Lagen". Vanuit hetzelfde menu kunnen we ook de lagen selecteren die we op het scherm willen zien of die we willen afdrukken. De volgorde van de lagen is belangrijk, omdat lagen met een lager nummer eerder worden getekend: elementen op opeenvolgende lagen bedekken de reeds getekende.

2.4 Het raster

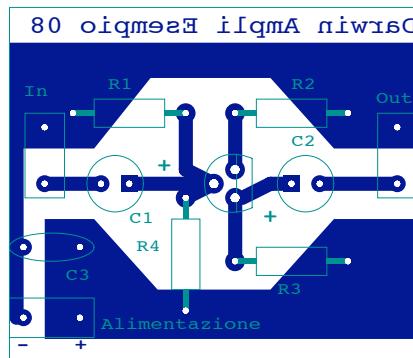
De logische eenheid in FidoCadJ is 5 mils (127 micron) en "halve eenheden" zijn niet toegestaan, wat betekent dat de coördinaten van elk grafisch element gehele getallen moeten zijn. Dit maakt het mogelijk een resolutie te verkrijgen die voldoende fijn is om een elektrisch schema en de meeste printplaten te tekenen. Om het tekenen te vergemakkelijken, maakt de toepassing het echter mogelijk om een grover raster in te stellen en de muis te dwingen uit te lijnen met het dichtstbijzijnde punt van het raster. Om deze functionaliteit in te schakelen zijn er twee knoppen in de Werkvlakbalk, "Laat raster zien" en "Klem aan raster", die het mogelijk maken om te schakelen tussen zichtbaar/verborgen raster en de muiscursor op het raster te forceren of vrij te laten bewegen. De rasterstap kan worden gekozen via een dialoogvenster dat toegankelijk is via de menubalk en het menu "Bestand/Opties tabblad Tekening".

2.5 Een eenvoudige printplaat

Om te oefenen wat we tot nu toe hebben geleerd, zullen we zien hoe we een eenvoudige printplaat kunnen ontwerpen. In tegenstelling tot andere software voor elektronische CAD die erg krachtig maar soms behoorlijk moeilijk te beheren is, biedt FidoCadJ een elektronische versie van de goede oude R41-transfers, of de tape op mylar-vellen. Het werken op een computer maakt het natuurlijk mogelijk om te profiteren van alle flexibiliteit die de machine biedt.

Opgemerkt moet worden dat het ontwerp van een printplaat, vooral als dit een complexe is, geen gemakkelijke taak is. De functies autoplacer en autorouter beloven wonderen in de publiciteitsfolders van grote CAD-bedrijven, maar het lijdt geen twijfel dat dit nog steeds taken zijn waarbij de ervaring van de gebruiker nog steeds een belangrijke rol speelt. FidoCadJ vormt een zeer directe en snelle manier om kleine printplaten op doe-het-zelf-schaal te tekenen. Deze

Een paar krabbels gemaakt met papier en potlood (en veel gummen) resulteren in tijdwinst door een duidelijk idee te verkrijgen dat met behulp van de computer moet worden ontwikkeld.



Figuur 2.12: Een zeer eenvoudige versterkertrap die een NPN-transistor gebruikt die is aangesloten in een gemeenschappelijke emitterconfiguratie.

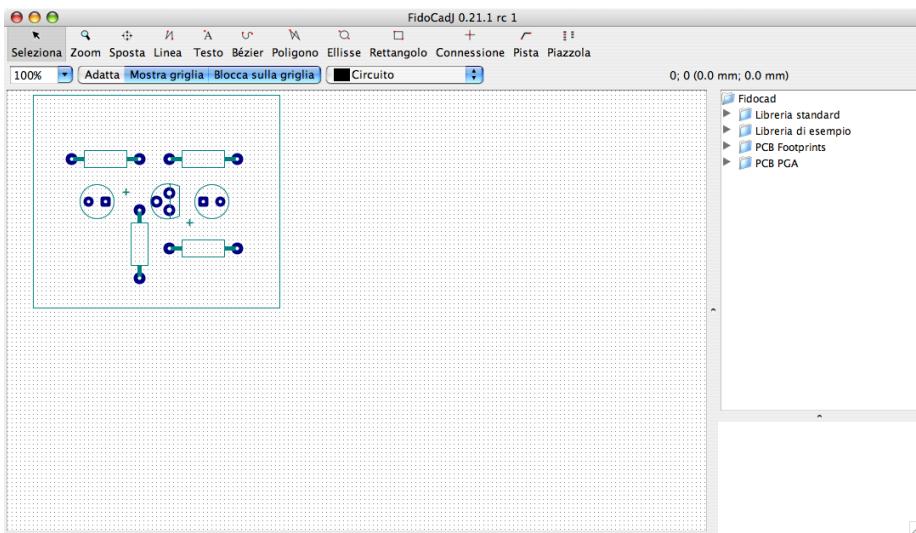
benadering is heel eenvoudig, maar als u een zeer complexe printplaat moet ontwerpen, dan is FidoCadJ NIET de tool voor u: gebruik dan bijvoorbeeld de uitstekende KiCad: de extra complexiteit is dan gerechtvaardigd. We zullen hier zien hoe je een heel eenvoudige, maar complete tekening kunt maken.

Ik zou willen voorstellen om te beginnen met een duidelijk idee van waar elk onderdeel moet worden geplaatst en hoe alle sporen moeten worden getekend zodat ze elkaar zo min mogelijk kruisen.

Hier spelen we een beetje vals en gaan we uit van het resultaat dat we willen bereiken, zoals weergegeven in figuur 2.12. Het is een eenvoudige common-emitter versterker opgebouwd rond een NPN-transistor, type BC547 of vergelijkbaar. Het is handig om aan het bord te denken alsof het transparant was, door vanaf de kant van de componenten te kijken. Hiervoor is de zeefdruk met de onderdelentekening erg handig, al willen we die waarschijnlijk niet op het eigenlijke bord afdrukken voor een doe-het-zelf-project.

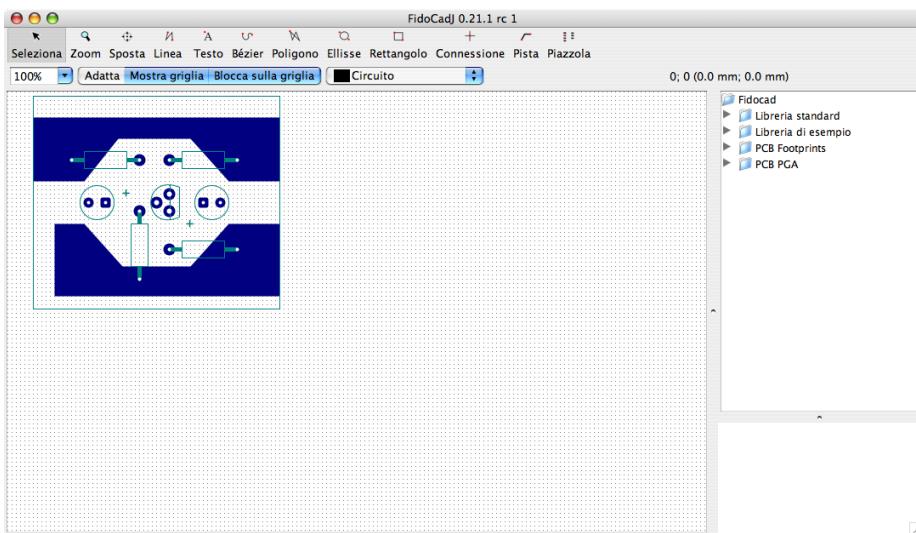
Het eerste dat ik zou willen voorstellen, is om alle componenten zo goed mogelijk te plaatsen. In ons voorbeeld zijn dit de transistor (uit de bibliotheek “PCB footprints/3 terminals semiconductors/TO92”), de weerstanden (“PCB footprints/Resistors/Resistor 1/4 W 0,4i resistor”), en de elektrolytische condensatoren (“PCB-footprints/Electrolytic capacitors/Vert. diam. 5 mm 2.5 mm pitch”). Om het bord af te bakenen, kan het handig zijn om een lege rechthoek op de zeefdruklaag te plaatsen (laag nr. 3). Om dit te doen kunnen we de *rechthoek*, primitieve gebruiken om er zeker van te zijn dat we eerst de juiste laag hebben geselecteerd. We zouden een resultaat moeten krijgen dat lijkt op het resultaat in de afbeelding 2.13.

We kunnen dan de koperen gebieden introduceren die de positieve en negatieve voeding leveren. Deze worden gespecificeerd door een polygoon te tekenen (met behulp van de *veelhoek* primitieve) en te dubbelklikken nabij de grens in selectiemodus om FidoCadJ te vertellen (via dialoogvenster) dat we een gevulde veelhoek willen hebben. Voordat u de polygoon plaatst, moet u ervoor zorgen dat de huidige laag degene is waar we het kopergebied in willen plaatsen (laag nr. 1 of koperkant). Het gebruik van een opgevuld gebied voor de voedingslijnen



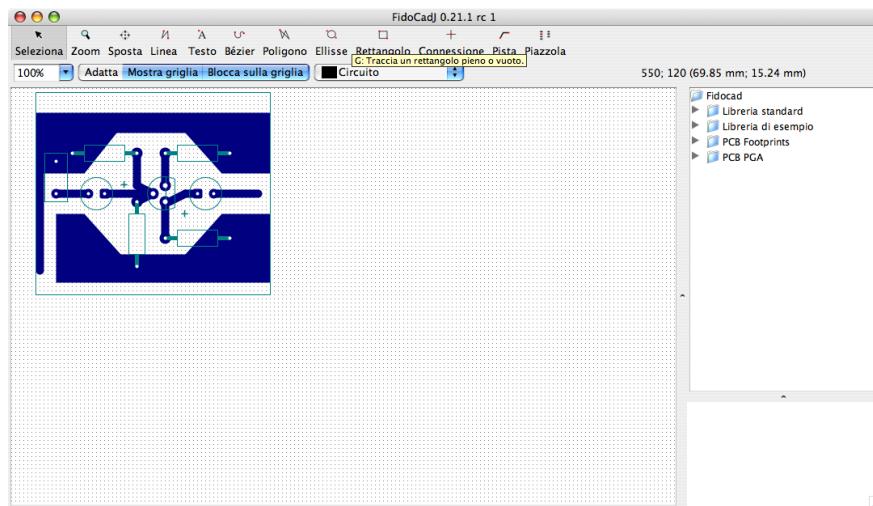
Figuur 2.13: De belangrijkste onderdelen worden op het bord geplaatst.

kan nuttig zijn om ervoor te zorgen dat deze verbindingen lage strooi-inducties hebben. We zouden het resultaat zoals in de figuur 2.14 moeten kunnen reproduceren. Om de elektrische verbindingen te voltooien kunnen we de *PCB Spoor*



Figuur 2.14: Voedingsaansluitingen toegevoegd met polygonale lijnen.

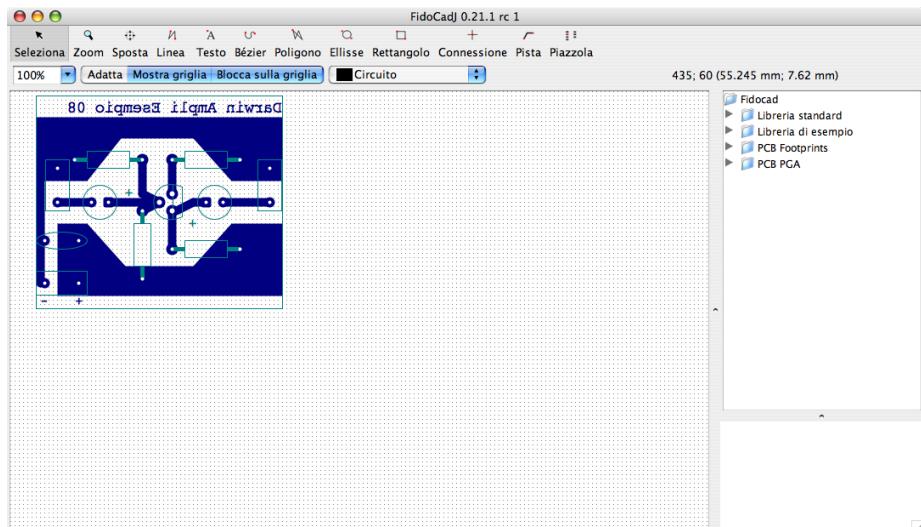
primitieve gebruiken. Ik koos voor een spoordikte van 10 eenheden (1,27 mm), wat het soldeerproces ten goede komt, zie figuur 2.15. We realiseren ons dat we een paar connectoren nodig hebben: één voor de input, één voor de output en één voor de voeding. We kunnen de voetafdruk gebruiken die is ontworpen voor een polyester condensator. Het heeft waarschijnlijk de juiste afmetingen. Laten we niet vergeten dat FidoCadJ wordt beschouwd als een vervanging voor de transfers...



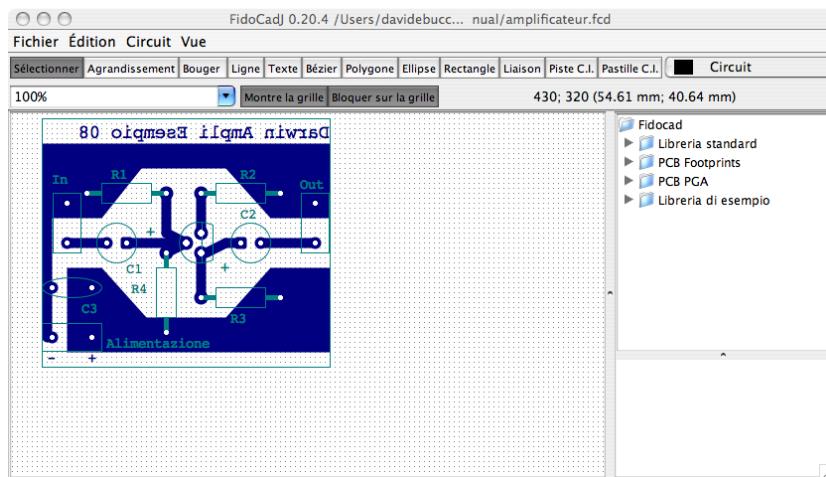
Figuur 2.15: De resterende verbindingen toegevoegd PCB-sporen.

Wees voorzichtig met de spoorbreedtes: iets dat op het scherm op een snelweg lijkt, zal waarschijnlijk een spoor zijn dat zo dun is dat het tijdens het solderen loskomt van het bord.

We kunnen ook een “+” en een “-“ op de koperlaag plaatsen en parallel aan de voeding een keramische condensator plaatsen. Laten we ook tekst op de bovenkant zetten. Om op de koperlaag te schrijven is na het selecteren van de juiste laag het ook nodig om alle teksten te spiegelen. Dit is eenvoudig te doen in het gebruikelijke eigenschappendialoogvenster dat toegankelijk is door te dubbelklikken op de tekenreeks die we willen wijzigen wanneer we in de selectiemodus zijn. Mogelijk moet u een beetje experimenteren voor de juiste grootte van de tekens. Om een idee te geven is het handig om een verhouding aan te houden van 3/4 tussen de horizontale en de verticale afmetingen van de tekens. Figuur 2.16 toont het verkregen resultaat met 11 eenheden voor de horizontale en 18 eenheden voor de verticale dimensie van de tekst.



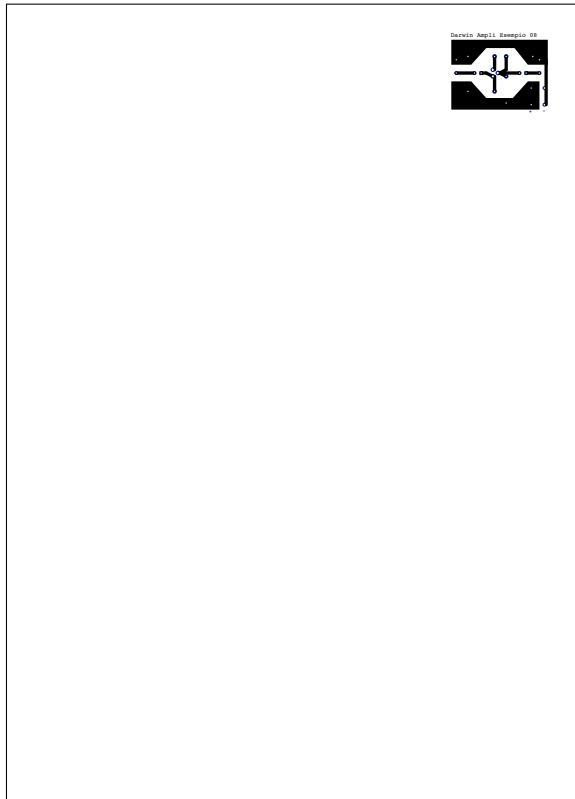
Figuur 2.16: De printplaat is bijna voltooid.



Figuur 2.17: De klus geklaard met de zeefdruk.

In dit stadium ontbreekt alleen de tekst met de naam van elk onderdeel, die op laag 3 (Componentenkant) kan worden geplaatst. Het programma met voltooide printplaat wordt getoond in figuur 2.17.

Zodra de tekening is voltooid, moeten we deze waarschijnlijk afdrukken op een transparant om het te gebruiken met een ontwikkelmachine of om het te gebruiken met andere methoden zoals de “Press&Peel”’ transfersystemen. Om dit te doen moeten we alle lagen die we niet willen printen onzichtbaar maken. Dit wordt gedaan vanuit het dialoogvenster dat toegankelijk is via de menubalk en het menu “Bekijk/Lagen”. In ons geval is het voldoende om laag 3 met de componentenkant te verbergen. Het programma toont dan alleen de koperlaag.



Figuur 2.18: De printplaat, zoals deze wordt afgedrukt (gespiegeld) op een ISO-UNI A4-vel.

Vervolgens drukken we alles af wat op het scherm wordt weergegeven (uiteraard NIET aangepast aan de grootte van de pagina, aangezien we de afmetingen in de tekening willen hebben), waarbij we ervoor zorgen dat de zwart-wit afdruk wordt geselecteerd om het maximale contrast te garanderen. Het kan handig zijn om de hele tekening te spiegelen, afhankelijk van de gekozen techniek voor de productie van de printplaat. Aangezien onze printplaat vrij klein is, neemt hij in zijn werkelijke afmetingen slechts een klein hoekje van het blad in beslag (uitgaande van een standaard ISO-UNI A4), zoals we kunnen zien in figuur 2.18.

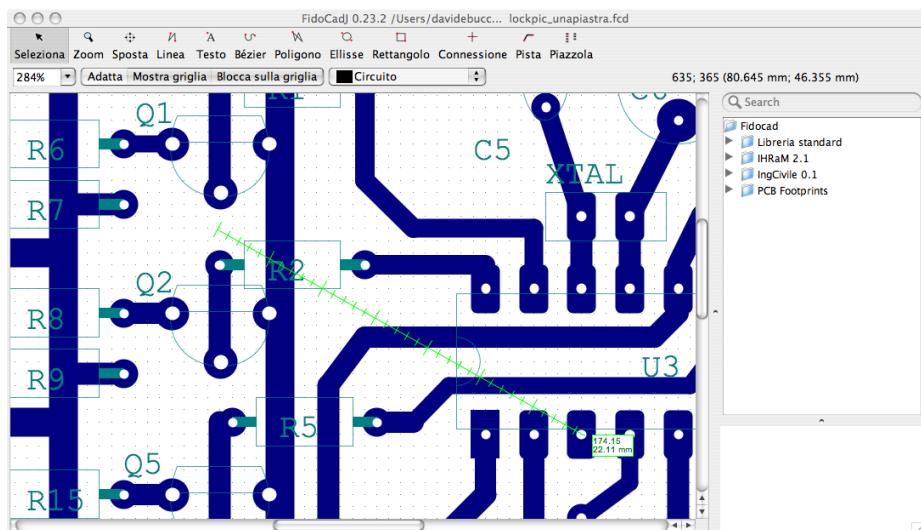
Tabel 2.3: Deze code beschrijft de printplaat uit ons voorbeeld.

Ter informatie is hieronder de code gegenereerd voor de printplaat van het bovenstaande voorbeeld (pas op met lange regels!):

```
[FIDOCAD]
TY 320 10 18 11 0 4 1 * Darwin Ampli Esempio 08
TY 85 240 12 8 0 5 1 * +
TY 44 239 12 8 0 5 1 * -
PL 35 90 35 225 10 1
PL 55 130 95 130 10 1
PL 250 130 305 130 10 1
PL 215 130 230 130 10 1
PL 195 140 215 130 10 1
PL 115 130 175 130 10 1
MC 155 220 3 0 PCB.R01
MC 75 80 0 0 PCB.R01
MC 270 185 2 0 PCB.R01
MC 270 80 2 0 PCB.R01
MC 230 130 3 0 PCB.CE00
MC 115 130 1 0 PCB.CE00
MC 40 175 0 0 PCB.CC50
PL 190 80 190 120 10 1
PL 190 140 190 185 10 1
PL 155 80 155 120 10 1
PL 155 120 175 130 10 1
PL 155 140 175 130 10 1
PP 30 30 30 105 90 105 130 55 215 55 260 105 320 105 320 30
    1
PP 320 240 320 155 260 155 215 205 135 205 90 155 55 155 55
    240 1
MC 190 120 0 0 PCB.T092
MC 305 90 1 0 PCB.CPBX352
MC 55 90 1 0 PCB.CPBX352
MC 80 225 2 0 PCB.CPBX352
TY 290 65 12 8 0 0 3 * Out
TY 40 60 12 8 0 0 3 * In
TY 95 225 12 8 0 0 3 * Alimentazione
TY 70 190 12 8 0 0 3 * C3
TY 230 95 12 8 0 0 3 * C2
TY 115 150 12 8 0 0 3 * C1
TY 120 170 12 8 0 0 3 * R4
TY 220 200 12 8 0 0 3 * R3
TY 230 55 12 8 0 0 3 * R2
TY 100 55 12 8 0 0 3 * R1
RV 30 5 320 255 3
```

2.6 De liniaal gebruiken

Bij het tekenen van een printplaat is het vaak handig om afstanden in het werkgebied te meten. U kunt bijvoorbeeld een spoorbreedte, de ruimte tussen twee sporen of de totale grootte van een print controleren. FidoCadJ biedt (sinds versie 0.23.2) een linialfunctie waarmee u deze taken gemakkelijk kunt uitvoeren. Gewoon met de rechtermuisknop klikken en slepen. Je zou een groene liniaal moeten krijgen, zoals die getoond in figuur 2.19. Als rechtsklikken en slepen niet gemakkelijk werkt in uw systeem, kunt u ook linksklikken en slepen terwijl u op de **Shift** toets drukt. De totale gemeten lengte wordt weergegeven

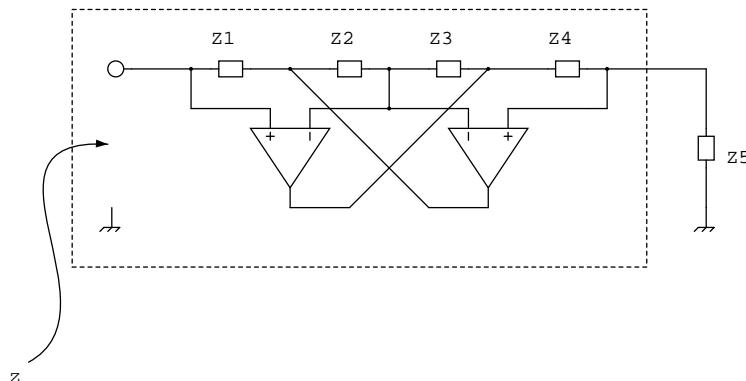


Figuur 2.19: Klik met de rechtermuisknop en sleep om de FidoCadJ-liniaal te activeren.

in FidoCadJ logische eenheden en ook in millimeters. Dit is handig wanneer de tekening wordt afdrukt in de 1:1-modus (zoals voor printplaten).

2.7 Pijl- en lijnstijlen

Met FidoCadJ kunnen pijlpunten worden getekend aan het begin en aan het einde van lijnen, Bézier-curven en natuurlijke kromme curves. Het is mogelijk om te kiezen of de pijl aan het begin of aan het einde van een element (of beide) moet staan en geeft een keuze tussen verschillende pijlstijlen. Pijllengte kan negatief zijn: in dit geval steekt de pijl buiten de lijn of curve uit. Experimenteer gerust! Er zijn ook enkele lijnstijlen beschikbaar, voor technische of mechanische tekeningen. Figuur 2.20 toont een voorbeeld waarin een elektrisch circuit (een GIC) is ingesloten in een gestreepte rechthoek. er wordt ook een pijl aan het einde van een Bézier-curve gebruikt. Door op dit element te dubbelklikken, toont FidoCadJ een dialoogvenster zoals getoond in figuur 2.21. U ziet dat het keuzevak “Pijl aan het begin” is aangevinkt. FidoCadJ zorgt ervoor dat de pijlpunt correct wordt georiënteerd. Er zijn verschillende tekenstijlen voor de pijl en ook voor streepjes . Je kunt er een beetje mee proberen te spelen om



Figuur 2.20: Een elektrische tekening (een GIC van Antoniou) waarin enkele FidoCadJ-extensies zijn gebruikt.



Figuur 2.21: Het parametervenster van de getoonde Bézier-curve gebruikt in de schema's van figuur 2.20.

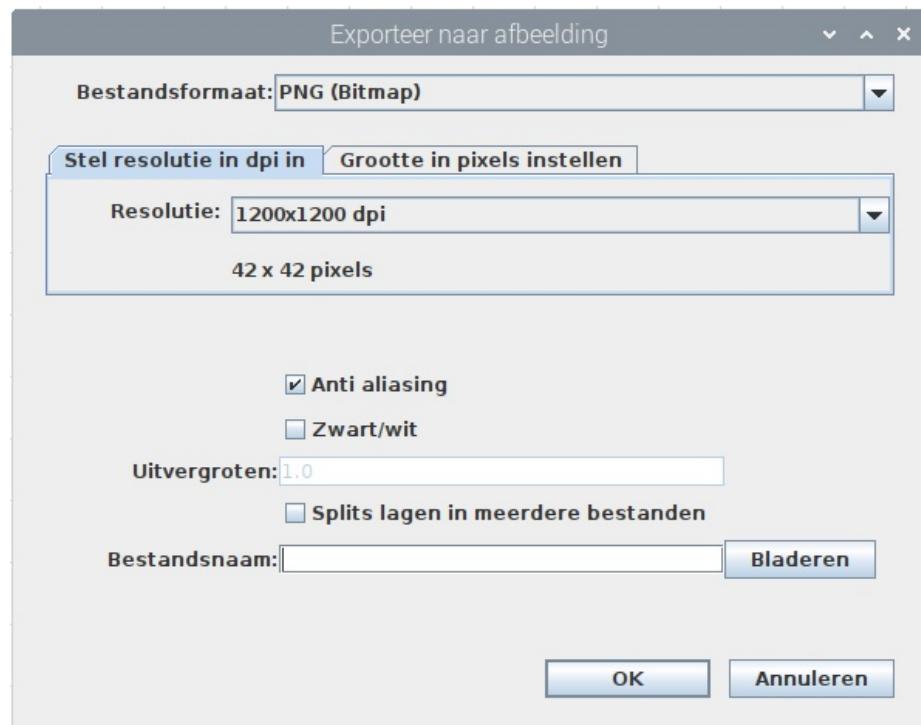
hun verschillen te ontdekken. Als de pijllengte negatief is, strekt de pijl zich uit buiten de lengte van het element en wijst ernaar.

De mogelijkheid om een streepjes stijl te kiezen en pijlpunten te plaatsen was niet aanwezig in het originele FidoCAD-formaat. Dit betekent helaas dat FidoCadJ-tekeningen die deze functionaliteiten gebruiken niet volledig achterwaarts compatibel zijn met FidoCAD voor Windows. Als u volledige FidoCAD-compatibiliteit nodig heeft, kunt u de optie "Strikte FidoCAD compatibiliteit" activeren op het tabblad "FidoCadJ extensies" van het venster "FidoCadJ voorkeuren". Op deze manier staat het programma u niet toe om grafische elementen in te voeren die compatibiliteitsproblemen zouden geven met FidoCAD. Als je meer informatie nodig hebt over de compatibiliteit van de nieuwe grafische elementen met FidoCAD, kijk dan in de sectie 3.4.

2.8 Exporteren

Een vectorformaat slaat de elementen op waaruit de tekening bestaat. Een bitmapformaat werkt op een matrix van pixels.

Een van de belangrijkste dingen aan FidoCadJ voor mij is de mogelijkheid om eenvoudige schema's te maken voor typografisch gebruik. Om deze reden heb ik een functie geïntroduceerd die de export van tekeningen via een aantal verschillende bestandsindelingen mogelijk maakt. Om de huidige tekening te exporteren, selecteert u in de menubalk uit het menu "Bestand" de opdracht "Exporteren". De tabel 2.4 toont een lijst met momenteel beschikbare grafische bestandsindelingen. Voor elk bestandsformaat geeft de tabel (maar ook het exportdialoogvenster in FidoCadJ) aan of het vector- of bitmap³. Voor de bitmap bestandsformaten kan het handig zijn om de optie "Anti-aliasing" in te schakelen, om het vervelende effect van de kwantisatie, vooral zichtbaar op diagonale lijnen, te verminderen. De resolutie en de "Anti-aliasing"-opties worden niet gebruikt bij het exporteren naar een vectorbestandsindeling. In dit geval kunt u in plaats daarvan een schaalfactor opgeven. Met de optie "Zwart/Wit" kan elke zichtbare laag in effen zwart worden afgedrukt. Dit is belangrijk voor de voorbereiding van films voor typografische doeleinden of een ontwikkelmanchine.



Figuur 2.22: Het exporteer naar afbeelding instel menu.

³De codestructuur van FidoCadJ staat de toevoeging van een ander bestandsformaat vrij eenvoudig toe. Neem contact met me op als je wilt deelnemen aan het project.

Tabel 2.4: Lijst met alle exportbestandsindelingen die beschikbaar zijn in FidoCadJ.

Formaat	Opmerking
JPG	Alomtegenwoordig bitmapformaat. De gebruikte compressie is lossy, daarom is het niet perfect voor het exporteren van FidoCadJ-schema's, omdat het enkele zichtbare artefacten kan opleveren.
PNG	Gecomprimeerd bitmapformaat, geschikt voor het exporteren van schema's. Dit is waarschijnlijk de beste manier om een FidoCadJ-tekening te exporteren wanneer een vectorindeling niet kan worden gebruikt.
SVG	W3C standaard vectorformaat. Internetbrowsers kunnen het binnen een webpagina weergeven. Zeer goed formaat voor afbeeldingen en schema's, het kan worden gebruikt met toepassingen zoals Inkscape, om de tekeningen gemaakt met FidoCadJ te wijzigen. Er zijn enkele beperkingen bij het exporteren van geroteerde en gespiegeld teksten.
EPS	Ingekapseld Postscript-vectorformaat. Gebruikt op professionele grafische toepassingen, handig voor het integreren van een FidoCadJ-tekening in een L ^A T _E X-document. Dit is gebruikt om figuur 2.12, pagina 16 in deze handleiding te verkrijgen (via een PDF-conversie, aangezien ik PDFL ^A T _E X gebruik).
PDF	De zeer bekende Portable Document File. Geëxporteerde bestanden bevatten geen lettertypen, dus de weergave van tekst kan op sommige platforms enigszins worden gewijzigd.
PGF	Vectorformaat om direct te gebruiken in een L ^A T _E X-document bij gebruik van het <i>pgf</i> -pakket, beschikbaar in het CTAN-archief. Dit is de gedachte exportoptie om schema's te exporteren naar bewerkbaar script. Tekstattributen worden niet vertaald. Dit maakt de introductie van L ^A T _E X-code direct in de tekening mogelijk en dit is de techniek die in deze handleiding wordt gebruikt om Figuur 2.6, pagina 10 te verkrijgen.
SCR	Met FidoCadJ kan een tekening worden geëxporteerd naar een script dat kan worden geïmporteerd in CadSoft Eagle. Om deze functie te gebruiken is het noodzakelijk om de bibliotheek FidoCadJLIB.1br in de map 1br van de huidige installatie van Eagle te installeren. De bibliotheek kan worden gedownload van de website van FidoCadJ. Op het moment van schrijven werkt deze optie alleen met schema's die alleen de meest voorkomende symbolen bevatten. Sommige tekenelementen zoals pads en tracks zijn niet beschikbaar en worden niet geëxporteerd.

2.9 Opdrachtregel opties

De applicatie wordt gedistribueerd als een bestand **.jar**, wat een Java-archief is.⁴ In veel besturingssystemen zou het voldoende moeten zijn om op het bestand te dubbelklikken om de applicatie uit te voeren, vooropgesteld dat een recente versie van Java op de machine is geïnstalleerd. In Oracle's terminologie, de zogenaamde JRE, of de Java Runtime Environment, is alles wat nodig is om een in Java geschreven programma uit te voeren (maar niet om het te schrijven, in dat geval zou de SDK nodig zijn...). De minimale Java-versie die nodig is om FidoCadJ uit te voeren is versie 9 die al een paar jaar bestaat.

In sommige gevallen kan het handig zijn om FidoCadJ uit te voeren vanaf een opdrachtregel (de terminal in Unix-systemen, of de MS-DOS Prompt in Windows). Hiervoor volstaat het om het commando **java** uit te voeren, met de optie **-jar**:

```
java -jar fidocadj-0.24.8.jar
```

Als een bestand is opgegeven in de opdrachtregel, probeert FidoCadJ het te openen. Bijvoorbeeld (op een Unix-machine):

```
java -jar fidocadj-0.24.8.jar ~/FidoCadJ/test.fcd
```

FidoCadJ wordt uitgevoerd en probeert het bestand **~/FidoCadJ/test.fcd** (mits deze bestaat). Er zijn nog enkele interessante dingen die FidoCadJ kan doen. Optie **-h** toont een Engelstalige lijst van de FidoCadJ-opties:

```
[davidebucci@davide-bucci-portable]$ java -jar fidocadj-0.24.8.jar -h
Dit is FidoCadJ, versie 0.24.8.
Door het FidoCadJ-team, 2007-2023.

Gebruik: java -jar fidocadj-0.24.8.jar [-opties] [bestand]
waarbij de opties zijn (vertaald):

-n      Start de grafische gebruikersinterface niet (headless-modus)

-d      Stel de externe bibliotheekdirectory in
Gebruik: -d dir
waarbij 'dir' het pad is van de map die u wilt gebruiken.

-c      Converteer het opgegeven bestand naar een grafisch formaat.
Gebruik: -c sx sy eps|pdf|svg|png|jpg|fcd|sch uitgaand_bestand
Als u deze opdrachtregeloptie gebruikt, *moet* u een FidoCadJ-bestand
specificeren om te converteren.
Een alternatief is om de resolutie in pixels per logische eenheid op
te geven door deze vooraf te laten gaan door de letter 'r' (zonder
spaties), in plaats van sx en sy op te geven.
OPMERKING: de juistheid van de bestandsextensie wordt gecontroleerd,
tenzij de optie -f is opgegeven.

-m      Als een bestand wordt geëxporteerd naar een vectorafbeeldings-
bestandsformaat, splitst u de lagen en schrijft u één bestand
voor elke laag. De bestandsnaam wordt verkregen door _ gevolgd
door het laagnummer toe te voegen aan de opgegeven bestandsnaam.
Met de volgende opdracht worden bijvoorbeeld de bestanden
test_0.svg, test_1.svg ... gemaakt van de tekening in test.fcd:

java -jar fidocadj.jar -n -m -c r2 svg test.svg test.fcd
```

⁴Met uitzondering van de Macintosh-versie, die een stand-alone applicatie is.

```

vervolg lijst van de FidoCadJ-opties

-s      Druk de grootte van het opgegeven bestand af in logische eenheden.
-h      Druk deze hulp af en sluit af.
-t      Druk de tijd af die FidoCadJ heeft gebruikt voor de opgegeven bewerking.
-p      Activeer sommige platformafhankelijke optimalisaties niet. U kunt deze optie proberen als FidoCadJ vastloopt of tergend traag is.
-l      Forceer FidoCadJ om een bepaalde landinstelling te gebruiken (de code kan onmiddellijk volgen of worden gescheiden door een optionele spatie).
-k      Toon de huidige landinstelling.
-f      Forceer FidoCadJ om enkele gezondheidstests op de invoergegevens over te slaan.

[bestand] Het optionele (behalve als u de -d of -s opties gebruikt) FidoCadJ-bestand om te laden tijdens het opstarten.

Voorbeeld: laad en converteer een FidoCadJ-tekening naar een png-bestand van 800x600 pixels zonder de GUI te gebruiken.
    java -jar fidocadj.jar -n -c 800 600 png out1.png test1.fcd

Voorbeeld: laad en converteer een FidoCadJ-tekening naar een png-bestand zonder gebruik te maken van de grafische gebruikersinterface (de zogenaamde headless-modus). Elke logische eenheid van FidoCadJ wordt geconverteerd in 2 pixels op de afbeelding.
    java -jar fidocadj.jar -n -c r2 png out2.png test2.fcd

Voorbeeld: laad FidoCadJ en forceer de locale naar vereenvoudigd Chinees (zh)
    .
    java -jar fidocadj.jar -l zh

[davidebucci@davide-bucci-portable]$
```

De eenvoudigste optie is **-n**, waarmee de software... niets doet, d.w.z. het activeert de GUI niet en sluit gewoon af. In dit geval is de Java-omgevingsvariabele `java.awt.headless` ingesteld op "waar". Vanzelfsprekend is alleen deze optie niet zo nuttig, maar wel waardevol bij gebruik in combinatie met andere functionaleiten die we gaan beschrijven. Optie **-d** maakt het mogelijk om de map te specificeren waarin FidoCadJ zoekt naar bibliotheken die bij het opstarten moeten worden geladen. Optie **-c** maakt het mogelijk om FidoCadJ een FidoCAD-bestand (dat moet worden opgegeven) te laten converteren naar een afbeelding in vector- of rasterformaat. Dit kan erg handig zijn, omdat FidoCadJ op een niet-interactieve manier als een converter kan worden gebruikt (samen met de optie **-n**).

We kunnen dus kijken naar het eerste voorbeeld in de help:

```
java -jar fidocadj.jar -n -c 800 600 png out1.png test1.fcd
```

FidoCadJ draait zonder de GUI te activeren en exporteert in het png formaat het bestand `test1.fcd`. Het uitvoerbestand krijgt de naam `out1.png` en heeft een pixelgrootte van 800x600. Er is een alternatieve versie van de **-c** optie, waarmee kan worden opgegeven hoeveel pixels moeten worden gebruikt om één logische eenheid te converteren (we noemen deze factor r_p). FidoCadJ werkt niet met halve logische eenheden (het zijn altijd gehele getallen), door bijvoorbeeld r_p

te kiezen, dat gelijk is aan twee pixels per logische eenheid, zorgt ervoor dat schema's altijd begrijpelijk zijn (ook al is het waarschijnlijk een beetje klein). De factor r_p kan een niet geheel getal zijn, zoals in het volgende voorbeeld:

```
java -jar fidocadj.jar -n -c r1.25 png out2.png test2.fcd
```

Om de totale grootte (in logische eenheden) van een schema te kennen, kun je de **-s** keuze gebruiken. Onthoud hoe dan ook dat FidoCadJ tijdens het exporteren altijd een marge van $t_{margin} = 3$ logische eenheden toevoegt voor elke zijde van de tekening. Dus als t_w de breedte van de tekening is in logische eenheden gegeven door **-s**, zou je kunnen verwachten dat p_w de breedte van je tekening in pixels wordt berekend als volgt:

$$p_w = r_p(t_w + 2t_{margin}) \quad (2.1)$$

Een andere interessante optie, hoewel een kenmerk van Java meer dan van FidoCadJ, is de mogelijkheid om het uiterlijk van de applicatie te wijzigen (in Java-jargon genaamd look & feel). U kunt de look&feel kiezen die u wilt zonder een enkele regel code te wijzigen. Hier is iets dat Linux-gebruikers waarderen, de GTK+-look:

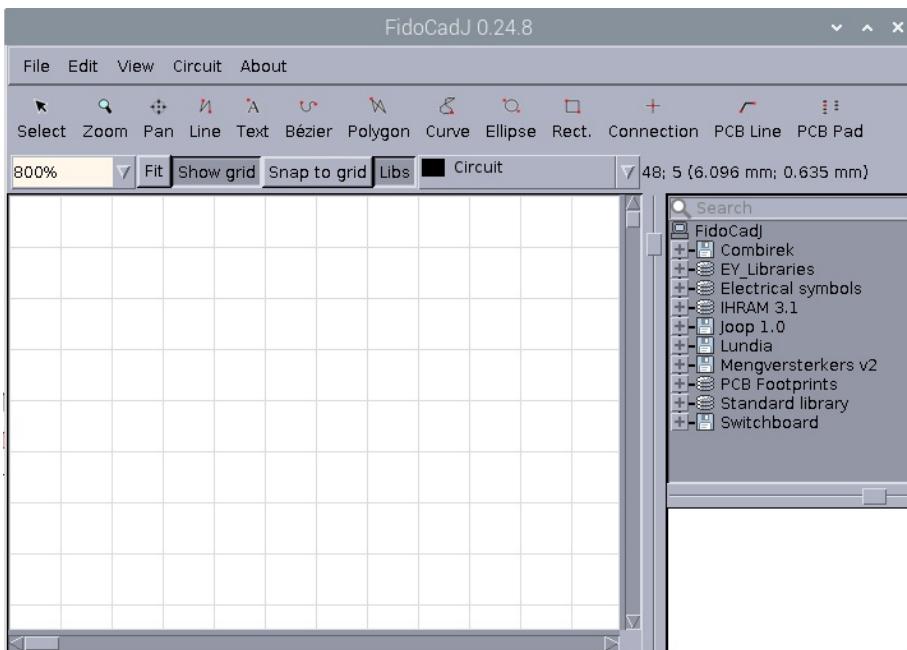
```
java -Dswing.defaultlaf=com.sun.java.swing.plaf.gtk.
      GTKLookAndFeel -jar fidocadj.jar
```

Er is ook de klassieke Motif look & feel, getoond in Figure 2.23⁵:

```
java -Dswing.defaultlaf=com.sun.java.swing.plaf.motif.
      MotifLookAndFeel -jar fidocadj.jar
```

Vanzelfsprekend zijn de hierboven genoemde commando's bedoeld om vanaf een terminal te worden verzonden, waarbij ervoor moet worden gezorgd dat de huidige map het bestand **fidocad.jar** bevat en alles op dezelfde regel geschreven wordt.

⁵Deze stijl kan op de een of andere manier mensen choqueren die goed bekend zijn met zeer verfijnde grafische interfaces zoals Aqua, met macOS. Ik zag echter vele jaren geleden een synchrotronbesturingssysteem met een grafische interface gebaseerd op Motif en dat dwong respect af.



Figuur 2.23: Het uiterlijk van FidoCadJ, met behulp van de Motif look & feel.

2.10 Beheer van de bibliotheek

2.10.1 Bibliotheekbestanden gebruiken

Een bibliotheek is een verzameling symbolen die de gebruiker in zijn tekeningen kan opnemen. Er is een verzameling bibliotheken opgenomen in het FidoCadJ-pakket, evenals de mogelijkheid om nieuwe symbolen en bibliotheken te definiëren. FidoCadJ maakt het zelfs mogelijk om een directory op te geven waarin alle gebruikersbibliotheek-bestanden (met de bestandsextensie .fcl) worden geplaatst. Dit kan via de menubalk en het menu “Bestand/Opties”.

In enkele zeer speciale gevallen (of om te testen) kunnen de bibliotheken in het FidoCadJ-pakket worden vervangen door externe. Als een bestand met de naam **FCDstdlib.fcl** aanwezig is, vervangt de inhoud de standaard bibliotheek die direct beschikbaar is in de toepassing. Analoog, als een bestand met de naam **PCB.fcl** aanwezig is, vervangt de inhoud de PCB-bibliotheek⁶.

In versie 0.23 kon ik dankzij Roby IZ1CYN de IHRaM 3.1-bibliotheek rechtstreeks in de FidoCadJ-distributie opnemen. Ik heb dit gedaan omdat van alle bibliotheken die ik heb gezien, deze een van de meest complete en rationeel opgebouwde is gebleken. Precies zoals het gebeurt voor de andere bibliotheken die in FidoCadJ zijn ingesloten, als er een bestand is met de naam **IHRAM.FCL** in het huidige zoekpad van de bibliotheek wordt deze geladen op de plaats van de versie die in het programma is ingesloten. Je hebt ook een bibliotheek met

⁶Let op het gebruik van hoofdletters, als uw besturingssysteem onderscheid maakt tussen hoofdletters en kleine letters in het bestandsbeheer.

elektrische symbolen waarvan het bestand `elettrotecnica.fcl` heet.

Andere bestanden met `.fcl` extensie in het zoekpad van de bibliotheek worden beschouwd als bibliotheken en FidoCadJ probeert ze te laden wanneer het wordt gestart. Dit wordt gedaan als de applicatie start, wanneer de gebruiker het zoekpad van de bibliotheek wijzigt, of wanneer in de menubalk en het menu “Circuit” de optie “Bibliotheken bijwerken” wordt gekozen.

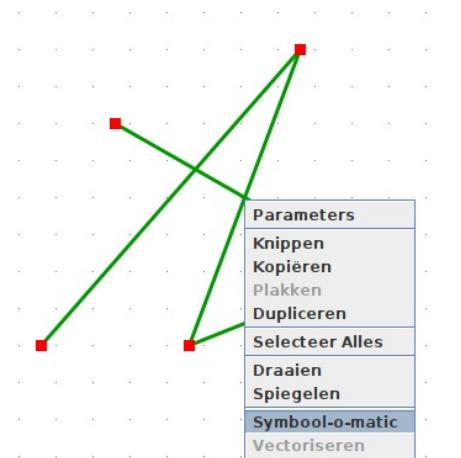
Met FidoCadJ kunnen niet-standaard symbolen worden gesplitst (zoals de originele FidoCAD doet). Dit kan erg handig zijn bij het plaatsen van een tekening in een nieuwsgroep, omdat op deze manier elk symbool dat niet tot de standaard FidoCAD-bibliotheken behoort, wordt omgezet naar zijn grafische primitieven. Wie uw bericht leest, hoeft dus niet dezelfde bibliotheken te hebben die u in uw systeem heeft geïnstalleerd. Voor kopiëren/plakken is een opdracht en een snelkoppeling beschikbaar in de menubalk in het menu “Bewerken”, genaamd “Kopiëren, split de niet standaard macros” (of `Control+I7`), om ervoor te zorgen dat de gekopieerde tekening alle niet-standaard macro’s splitst. U kunt een bestand ook opslaan met de niet-standaard macro’s gesplitst, door in de menubalk in het menu “Bestand” en “Opslaan als..., split de niet standaard macros” te kiezen. Er verschijnt een dialoogvenster waarin u een nieuwe bestandsnaam kunt kiezen, aangezien u het bestand waarmee u werkt meestal niet wilt overschrijven.

2.10.2 Nieuwe symbolen definiëren

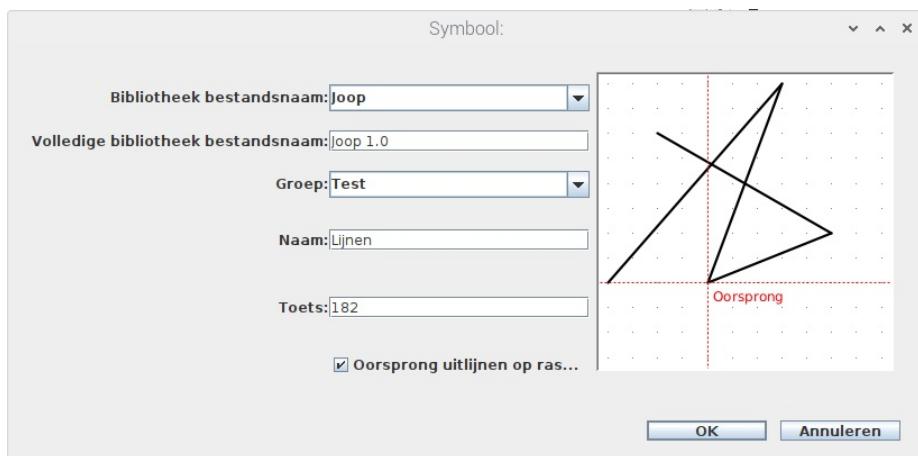
Met FidoCadJ kunnen nieuwe bibliotheken en nieuwe symbolen worden gemaakt op een manier die redelijk intuïtief zou moeten zijn. Dit zijn de te volgen stappen:

- FidoCadJ heeft een plaats nodig om de geproduceerde bestanden op te slaan. Zorg ervoor dat er een map met de gebruikersbibliotheken is gedefinieerd. Zo niet, specificeer het dan in de FidoCadJ-gebruikersinstellingen.
- Teken je nieuwe symbool.
- Selecteer wat je zojuist hebt getekend en klik er met de rechtermuisknop op. Er verschijnt een pop-up menu, zoals in figuur 2.24.
- Kies “Symbol-o-matic.” Er verschijnt een dialoog voor de definitie van de details van het nieuw gecreëerde symbool (zie figuur 2.25).
- In het eerste veld “Bibliotheek bestandsnaam:” vindt u de bestandsnaam van de bibliotheek waarin het nieuwe symbool moet worden opgenomen. Als u daar een nieuwe naam typt, maakt FidoCadJ een nieuw bestand aan.
- Het tweede veld “Volledige bibliotheek bestandsnaam:” is de volledige naam van de bibliotheek, degene die getoond wordt in de bibliotheekboom. Je zou daar kunnen schrijven wat je wilt, maar het is beter om een korte en zinvolle naam te kiezen.

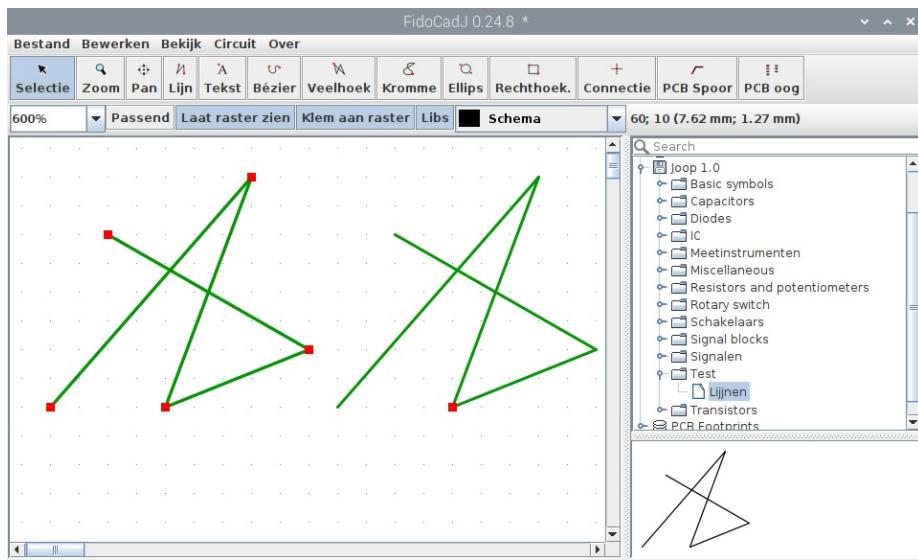
⁷Op macOS systemen, is de snelkoppeling `Command+I`



Figuur 2.24: Het pop-upmenu dat verschijnt als u met de rechtermuisknop klikt maakt het mogelijk om tekenelementen om te zetten in een symbool



Figuur 2.25: Het nieuwe dialoogvenster voor symbooldefinities. Hier kunt u alle belangrijke kenmerken van het symbool instellen. Let op de oorsprong gedefinieerd door de twee rode assen.



Figuur 2.26: Het nieuw aangemaakte symbool, weergegeven in de symbolenlijst en in de tekening. Aan de linkerkant staat nog steeds de tekening die is gebruikt voor de symbooldefinitie. Merk op dat er slechts één controlepunt aanwezig is voor het nieuwe bibliotheeksymbool in de tekening.

- Het derde veld “Groep:” stelt je in staat om te kiezen in welke groep je het nieuwe symbool wilt plaatsen binnen de bibliotheek. Nogmaals, als u een nieuwe naam typt, wordt er een nieuwe groep gemaakt.
- Het vierde veld “Naam:” is de naam van het symbool, zoals het getoond wordt in de bibliotheekboom. Kies opnieuw een kort maar betekenisvolle naam.
- Het vijfde veld “Toets:” is een zeer korte tag die het symbool in de code identificeert. Het moet binnen elke bibliotheek uniek zijn en mag geen spaties bevatten, evenals tekens zoals haakjes, punten, enzovoort. FidoCadJ stelt een korte numerieke code voor, maar u kunt ook geheugensteuntjes gebruiken.
- Door in het veld aan de rechterkant te klikken, moet u de Oorsprong van het symbool kiezen, het punt dat als referentie moet worden gebruikt om het symbool in de tekening te plaatsen. U kunt ervoor kiezen om de gewenste positie op het raster vast te klikken door te klikken op “Oorsprong uitlijnen op raster”.
- Zodra u op “OK” klikt, zou u uw nieuw gecreëerde symbool moeten vinden in de gebruikersbibliotheek in de bibliotheekboom in het hoofdvenster van FidoCadJ.

Een voorbeeld van een gebruikerssymbool dat in een tekening wordt gebruikt, wordt getoond in figuur 2.26. Let op het verschil tussen het geselecteerde deel van de tekening aan de linkerkant (eigenlijk wat is gebruikt om een nieuw



Figuur 2.27: Het pop-up menu dat wordt gebruikt voor het wijzigen van symbool eigenschappen in een gebruikersbibliotheek.

symbool te definiëren) en het nieuwe symbool dat aan de rechterkant is geplaatst. In feite is er slechts één controlepunt beschikbaar voor de plaatsing van het symbool en dit controlepunt komt overeen met de oorsprong tijdens de symbooldefinitie. Het is dus handig om daarvoor een punt te kiezen dat enige relevantie heeft in je symbool. Je kunt een symbool weer splitsen in zijn primitieven door het te selecteren en op “Vectoriseren” te klikken.

2.10.3 Wijzigen van bestaande symbolen

Er zijn enkele menu-opties beschikbaar voor symbolen in de bibliotheekstructuur. Zodra u een symbool in een gebruikersbibliotheek selecteert en er met de rechtermuisknop op klikt, verschijnt er een pop-up menu, zoals weergegeven in figuur 2.27. Hier heb je verschillende opties:

- “Kopiëren” kopieert een hele bibliotheek of een bibliotheeksymbool.
- “Plakken” plakt op de gewenste plaats de hele bibliotheek of een bibliotheeksymbool.
- “Hernoemen” verandert de weergegeven naam van het symbool.
- “Verwijderen” gooit het symbool weg. Wees hier voorzichtig mee: als een tekening het symbool bevat, wordt het ongeldig en verdwijnt het.
- “Wijzigen sneltoets” is handig om de sneltoets die bij het symbool hoort te wijzigen. Nogmaals, dit is iets dat moet worden gedaan als het symbool niet in de tekening is gebruikt.

Door slepen en neerzetten te gebruiken, kunt u de bibliotheek of de categorie waarop een symbool is gecategoriseerd wijzigen. Houd er rekening mee dat een symbool compleet identificatielabel is samengesteld uit de naam van de bibliotheek, gevolgd door de sleutel van het symbool. Het verplaatsen van een symbool van de ene bibliotheek naar een andere maakt dus tekeningen die het bevatten ongeldig.

Alle bibliotheekbewerkingen kunnen ongedaan worden gemaakt, precies zoals tekening bewerkingen. Houd er rekening mee dat lege bibliotheken of groepen niet worden weergegeven.

Hoofdstuk 3

Tekenformaat, macro's & bibliotheek

Dit hoofdstuk bevat een gedetailleerde beschrijving van het formaat dat door FidoCAD en, als gevolg daarvan, door FidoCadJ wordt gebruikt om een tekening op te slaan. Het is een eenvoudig tekstformaat dat als voordeel heeft dat het zeer compact en efficiënt is. Aangezien het formaat nooit in detail in een document is beschreven, probeer ik alles wat ik erover heb geleerd samen te vatten. Aangezien ik een paar extensies heb toegevoegd die alleen beschikbaar zijn in FidoCadJ, beschrijf ik deze ook hier. Onthoud dat FidoCadJ sinds versie 0.23.4 alleen de UTF-8-codering op alle platforms gebruikt.

3.1 Koptekst beschrijving

Alle bestanden met een tekening in het FidoCAD -formaat moeten beginnen met de tag *[FIDOCAD]*. Een programma kan daarom de aanwezigheid van FidoCAD-commando's herkennen door deze tag te lezen. In dit opzicht is FidoCadJ toleranter dan de originele FidoCAD en herkent en interpreteert een bestand dat niet de standaard header bevat. Zelfs commando's met tekst kunnen daarom correct worden geïnterpreteerd, zolang het aantal foute regels een intern in het programma ingestelde waarde (ongeveer 100) niet overschrijdt. Dit zorgt ervoor dat FidoCadJ geen minuten tijd verspilt aan het proberen van een zeer groot binair bestand te openen.

3.2 Coördinaten systeem

FidoCadJ werkt op een zeer eenvoudig coördinatensysteem. In de praktijk heeft het een zeer groot gebied tot zijn beschikking dat alleen wordt geïdentificeerd door hele en positieve coördinaten. De lengte van elke eenheid in x en in y is vastgesteld op $127\mu\text{m}$, een waarde die het mogelijk maakt om zelfs voor het kleinste SMD-pakket een goede resolutie te verkrijgen zonder te fijn te zijn voor dagelijks gebruik. In typografische termen is de FidoCadJ-resolutie 200 dots per inch.

De originele FidoCAD had twee verschillende verwerkingsmodi: PCB en elektrisch schema. In FidoCadJ is dit verschil vervaagd en verschijnt pas op het moment dat de tekening wordt afdrukt. Het is daarom aan te raden om het programma zo in te stellen dat het formaat van een elektrisch schema wordt aangepast om beter gebruik te maken van het paginaformaat, anders heeft het afdrukte resultaat het formaat van een postzegel.

3.3 Elementen tekenen

FidoCadJ kan als volgt 12 tekenelementen beheren

- Lijn
- Gevulde of lege rechthoek
- Eenvoudige tekst (verouderd)
- Geavanceerde tekst
- Gevulde of lege veelhoek
- Gevulde of lege ellips
- Bézier curve
- Natuurlijke kubieke kromme
- Elektrisch knooppunt
- PCB oog
- PCB spoor
- Macro

We zullen ze allemaal beschrijven. Over het algemeen wordt elk element geïdentificeerd door een commando en een aantal parameters (meestal gehele getallen of tekenreeksen) op dezelfde regel geplaatst en gescheiden door een spatie.

Lijn

De lijn primitief wordt geïdentificeerd door de opdracht **LI** en de definitie ervan vereist alleen begin- en eindcoördinaten en de laag:

```
LI x1 y1 x2 y2 l
```

Wiskundigen zouden waarschijnlijk de term "segment" geschikter vinden.

de punten (x_1, y_1) en (x_2, y_2) vertegenwoordigen respectievelijk de begin- en eindcoördinaten, terwijl l de laag is, gekenmerkt door een geheel getal tussen 0 en 15.

FidoCadJ extension: vanaf FidoCadJ versie 0.23 kan **LI** in de volgende regel worden gevolgd door een extensie:

```
FCJ a b c d e nv
```

Tabel 3.1: Betekenis van de parameter a voor de aanwezigheid van een pijlpunt aan de zijkanten van een lijn of Bézier-primitief.

a	Pijlpunt
0	geen
1	aan de startzijde
2	aan de eindzijde
3	beide kanten

Tabel 3.2: Betekenis van de parameter b voor de pijlpuntstijl.

b	Pijlpunt stijl
0	gevulde standaard pijlpunt
1	gevulde standaard pijlpunt met maatlijn
2	lege standaard pijlpunt
3	lege standaard pijlpunt met maatlijn

waarbij a een geheel getal is dat de aanwezigheid of niet-aanwezigheid van de pijlpunten aan de uiteinden van het segment weergeeft (kijk eens naar tabel 3.1), b vertegenwoordigt de stijl van de pijlpunt (als tabel 3.2). Parameters c en d geven respectievelijk de totale lengte en de halve breedte van de pijlpunt, terwijl e een geheel getal is dat de streepstijl geeft. Als nv gelijk is aan 1, moet de opdracht **FCJ** gevuld worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Gevulde of lege rechthoek

Een rechthoek gevuld of leeg wordt geïdentificeerd door respectievelijk de opdrachten **RP** en **RV**, gevuld door de coördinaten van de twee hoekpunten op een van de twee diagonalen en de laag.

```
RP x1 y1 x2 y2 1
RV x1 y1 x2 y2 1
```

de punten (x_1, y_1) en (x_2, y_2) vertegenwoordigen de twee hoekpunten en l is de laag, gekenmerkt door een geheel getal tussen 0 en 15 .

FidoCadJ extension: vanaf FidoCadJ versie 0.23 kunnen **RP** en **RV** in de volgende regel gevuld worden door een extensie:

```
FCJ e nv
```

waarbij e een geheel getal is dat de streepstijl aangeeft. Als nv gelijk is aan 1, moet de **FCJ** opdracht worden gevuld door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Eenvoudige tekst (verouderd)

De eenvoudige tekst was de eerste primitieve die werd geleverd met de vroege versies van FidoCAD. FidoCadJ herkent het en schrijft eenvoudig tekst met grootte 12, ongeacht de huidige zoom.

Aangezien deze primitieve als achterhaald werd beschouwd door Lorenzo Lutti, de maker van FidoCAD, doet FidoCadJ precies hetzelfde en hoewel dit

Tabel 3.3: Functie van de bits in de term tekststijl.

Bit	Waarde	Gedrag
0	1	tekst vetgedrukt
2	4	gespiegelde tekst

correct wordt geïnterpreteerd, is het niet beschikbaar op de gereedschapbalk. FidoCadJ slaat dit element precies op als of het geavanceerde tekst was en gebruikt de opdracht **TY** bij het opslaan van het bestand.

De opdracht is **TE** en het formaat is als volgt:

```
TE x1 y1 te schrijven tekst
```

het punt (x_1, y_1) is waar de string “te schrijven tekst” moet worden geplaatst. Merk op dat de laaginformatie ontbreekt. FidoCadJ behandelt dit object als of het op laag nul is geplaatst (circuit).

Geavanceerde tekst

De primitieve geavanceerde tekst biedt veel meer flexibiliteit met betrekking tot eenvoudige tekst die hierboven is geïntroduceerd.

Het wordt geïdentificeerd door de opdracht **TY**, gevolgd door een aantal parameters om de tekstoriëntatie te bepalen (geroteerd of gespiegeld), en voor afmetingen x en y en het gebruikte lettertype. Vanwege de hoeveelheid informatie die moet worden verstrekt, is de resulterende opdrachtreeks vrij complex:

```
TY x1 y1 sy sx a s l f te schrijven tekst
```

Het punt (x_1, y_1) is waar de string “te schrijven tekst” moet worden geplaatst. De waarde van s_y en s_x geeft de horizontale en verticale afmetingen van de tekst aan in logische eenheden. FidoCadJ respecteert de verticale dimensie, uitgaande van de horizontale, en die beeldverhouding wordt alleen gewijzigd als dit strikt noodzakelijk is. De tekstroteratie wordt beschreven door de term a , uitgedrukt in sexagesimale graden, terwijl de waarde van s de tekststijl bepaalt, volgens tabel 3.3. De laag wordt gegeven door de gebruikelijke term l , en f geeft het te gebruiken lettertype aan, of het kan een sterretje zijn om het gebruik van standaard Courier New font aan te geven. Als de naam van het lettertype spaties bevat moeten deze worden vervangen door “++”.

Tekstopdrachten ondersteunen subscripts: om ze te gebruiken voegt u tekst in tussen het onderstrepingssteken $_$ en het dakje $^$; bijvoorbeeld $_12^$ schrijft 12 als subscript. Ze ondersteunen ook superscript: voeg tekst in tussen het dakje $^$ en het onderstrepingssteken $_$; bijvoorbeeld 56_ schrijft 56 als superscript. In beide gevallen zijn de laatste $^$ of $_$ optioneel. Gebruik een backspace om het teken $_$ of $^$ af te drukken, d.w.z. $\backslash_$ voor $_$ en $\backslash^$ voor $^$. Gebruik twee backspaces voor één \backslash . Voorbeeld van tekst met superscript: $3^2_+4^2_=5^2$ of C_12

De maximale lengte van de tekst is ongeveer 80 woorden. Het tellen gebeurt in woorden en niet in tekens omdat in de interne structuur van het programma de woorden (en opdracht termen) gescheiden zijn wanneer een regel geïnterpreteerd wordt.

Gevulde of Lege veelhoek

Een gevulde of lege veelhoek wordt aangegeven met respectievelijk de opdrachten **PP** en **PV**, gevolgd door de coördinaten van de hoekpunten die de veelhoek definiëren en de laag. De opdrachten zijn

```
PP x1 y1 x2 y2 ... l
PV x1 y1 x2 y2 ... l
```

waarbij de punten (x_1, y_1) , (x_2, y_2) ... de hoekpunten zijn die de veelhoek definiëren en l de laag is, gekenmerkt door een geheel getal tussen 0 en 15. De lengte van de opdrachtregel kan dus variëren afhankelijk van het aantal gebruikte hoekpunten. Het maximum aantal beschikbare hoekpunten is willekeurig vastgesteld op iets minder dan 5000.

FidoCadJ-extensie: vanaf FidoCadJ versie 0.23, **PP** en **PV** kan in de volgende regel gevolgd worden door een extensie:

```
FCJ e nv
```

waarbij e een geheel getal is dat de streepstijl aangeeft. Als nv gelijk is aan 1, moet de **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Gevulde of lege ellips

Een ellips gevuld of leeg wordt geïdentificeerd door respectievelijk de opdrachten **EP** en **EV**, gevolgd door de coördinaten van de twee hoekpunten op de diagonaal en het laagnummer.

```
EP x1 y1 x2 y2 1
EV x1 y1 x2 y2 1
```

het punt (x_1, y_1) vertegenwoordigt het eerste hoekpunt op de diagonaal, (x_2, y_2) is het tweede hoekpunt en l is de geïdentificeerde laag met een geheel getal tussen 0 en 15.

FidoCadJ-extensie: vanaf FidoCadJ versie 0.23, **EP** en **EV** kan in de volgende regel gevolgd worden door een extensie:

```
FCJ e nv
```

waarbij e een geheel getal is dat de streepstijl aangeeft. Als nv gelijk is aan 1, moet de **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Bézier-curve

Een Bézier-curve segment, in zijn kubische variant, wordt geïdentificeerd door vier hoekpunten, die dus vereist zijn in de bijbehorende opdracht **BE**:

```
BE x1 y1 x2 y2 x3 y3 x4 y4 1
```

waarbij $P_1 \equiv (x_1, y_1)$, $P_2 \equiv (x_2, y_2)$, $P_3 \equiv (x_3, y_3)$ en $P_4 \equiv (x_4, y_4)$ de vier controlepunten van het Bézier-curve segment zijn, terwijl l de laag is, geïdentificeerd

door een geheel getal tussen 0 en 15. Gegeven de vier hierboven gedefinieerde punten, wordt het curvesegment berekend met de uitdrukking

$$B(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4, \quad (3.1)$$

waarbij $t \in [0, 1]$ een parameter is.

FidoCadJ-extensie: vanaf FidoCadJ versie 0.23, **BE** kan in de volgende regel gevuld worden door een extensie:

```
FCJ a b c d e nv
```

waar a een geheel getal is dat de aanwezigheid of niet-aanwezigheid van pijlpunten aan de uiteinden van de curve weergeeft (kijk eens naar tabel 3.1), b vertegenwoordigt de stijl van de pijlpunt (als tabel 3.2). Parameters c en d geven respectievelijk de totale lengte en de halve breedte van de pijlpunt, terwijl e een geheel getal is dat de streepstijl geeft. Als nv gelijk is aan 1, moet de **FCJ** opdracht gevuld worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Natuurlijke kubieke kromme (complexe kromme)

Een natuurlijke kubieke kromme wordt gedefinieerd door een bepaald aantal hoekpunten. De curve doorkruist elk hoekpunt en wordt op zo'n manier berekend dat het erg vloeiend is. In het FidoCadJ-formaat wordt een kromme geïdentificeerd door de opdrachten **CV** en **CP**:

```
CV aa x1 y1 x2 y2 ... 1
CP aa x1 y1 x2 y2 ... 1
```

de parameter aa is gelijk aan 1 als de curve gesloten is, anders is hij 0. Hoekpunten $(x_1, y_1), (x_2, y_2)\dots$ definiëren de kromme en l is de laag, een geheel getal van 0 tot 15. De lengte van de opdrachtregel kan dus variëren afhankelijk van hoeveel hoekpunten er worden opgenomen. Wat polygonen betreft, is het maximale aantal hoekpunten dat beschikbaar is intern vastgesteld op iets minder dan 5000. Deze primitief is geïntroduceerd in versie 0.24 en is niet aanwezig in de originele FidoCAD.

FidoCadJ-extensie: **CV** en **CP** kan in de volgende regel gevuld worden door een extensie:

```
FCJ a b c d e nv
```

waar a een geheel getal is dat de aanwezigheid of niet-aanwezigheid van de pijlpunten aan de uiteinden van de curve weergeeft (kijk eens naar tabel 3.1), b vertegenwoordigt de stijl van de pijlpunt (als tabel 3.2). Parameters c en d geven respectievelijk de totale lengte en de halve breedte van de pijlpunt, terwijl e een geheel getal is dat de streepstijl geeft. Als nv gelijk is aan 1, moet de **FCJ** opdracht worden gevuld door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Elektrisch knooppunt

De primitieve elektrisch knooppunt is gewoon een gevulde cirkel met constante afmetingen en wordt gebruikt om een verbinding in een elektrisch schema weer te geven. Het wordt geïdentificeerd door de opdracht **SA** en het vereist alleen de coördinaten en de laag:

```
SA x1 y1 1
```

Met FidoCadJ wordt de diameter van de cirkel intern in het programma vastgelegd in twee logische eenheden.

Als **SA** wordt gevolgd door **FCJ** moet **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

PCB oog

Een PCB oog wordt geïdentificeerd door de opdracht **PA** en wordt gekenmerkt door zijn stijl (rond, rechthoekig, rechthoekig met afgevlakte hoeken) en de diameter van het interne gat:

```
PA x1 y1 dx dy si st 1
```

waarbij het punt (x_1, y_1) de positie van het oog vertegenwoordigt, d_x de breedte van het oog is (langs de x -as), d_y de hoogte (langs de y -as). De waarde s_i is de diameter van het gat in het oog, terwijl s_t de stijl van het oog is:

- 0 ovaal oog
- 1 rechthoekig oog
- 2 rechthoekig oog met afgeronde hoeken

De waarde van l moet een geheel getal zijn om de laag aan te geven waarin het oog moet worden geplaatst. Als **PA** wordt gevolgd door **FCJ**, moet de **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

PCB spoor

De PCB-spoor is in wezen een segment waarvan de breedte kan worden gespecificeerd. De hoeken aan de uiteinden van het segment zijn altijd afgerond om de verbinding met andere PCB-sporen of -ogen te vergemakkelijken. De te gebruiken opdracht is **PL**, met het volgende formaat:

```
PL x1 y1 x2 y2 di 1
```

Het spoor wordt getekend tussen de punten (x_1, y_1) en (x_2, y_2) , met een totale breedte van d_i , en de gebruikte laag is l . De breedte d_i kan een gebroken getal zijn. Als **PL** gevolgd wordt door **FCJ**, moet de **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

Macro-oproep

Een macro is een tekening of een symbool in een bibliotheek. Over het algemeen is dit de manier waarop veel gebruikte elektrische symbolen worden weergegeven. De opdracht die gebruikt wordt om een macro aan te roepen is **MC**, en het aanroepen gaat als volgt:

```
MC x1 y1 o m n
```

Het symbool wordt getekend met positie (x_1, y_1) als referentie, en de oriëntatie wordt bepaald door de waarde van o (vermenigvuldigd met 90° rechtsom) en als m gelijk is aan 1 wordt de macro gespiegeld. de laatste parameter, n , is de naam van de macro in de bibliotheek, gespecificeerd als *library.code*.

Als **MC** gevolgd wordt door **FCJ**, moet de **FCJ** opdracht gevolgd worden door twee **TY** opdrachten die de naam en de waarde geven die aan dit element zijn gekoppeld.

3.4 FidoCadJ extensies

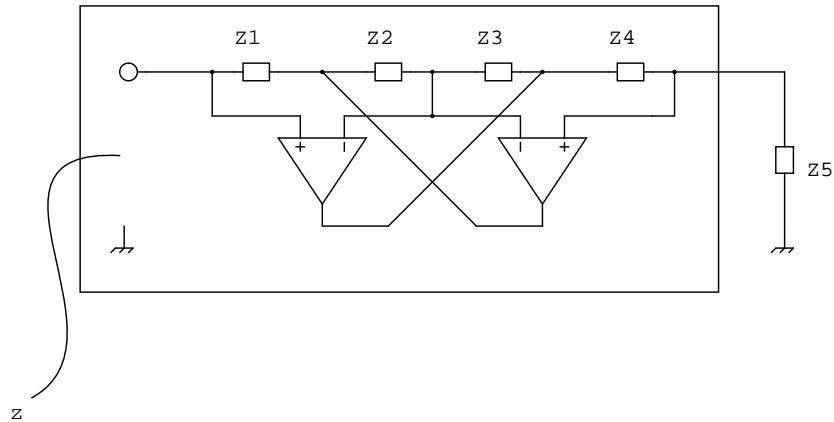
Sinds versie 0.21 is FidoCadJ begonnen met het introduceren van enkele verfijningten ten opzichte van het oorspronkelijke FidoCAD-formaat. De FidoCadJ-extensies worden in de code weergegeven door de opdracht **FCJ**. Deze opdracht wordt niet alleen gebruikt, maar betekent dat FidoCadJ aanvullende informatie moet specificeren over wat er in de vorige regel is gespecificeerd. Hier is een voorbeeld:

```
[FIDOCAD]
MC 40 30 0 0 080
FCJ
TY 50 35 4 3 0 0 0 * R1
TY 50 40 4 3 0 0 0 * 47k
```

De aanwezigheid van de **FCJ** opdracht geeft aan dat de naam en de waarde van de macro gespecificeerd in de eerste regel worden gegeven door de twee **TY** opdrachten die volgen. Alleen de coördinaten en de lettertypen worden in aanmerking genomen voor de tekstweergave.

Met FidoCadJ kan een “strikte FidoCAD-compatibiliteitsmodus” worden geactiveerd, waarin alle extensies zijn uitgeschakeld. Op deze manier is FidoCadJ perfect compatibel met de originele FidoCAD, behalve dat FidoCadJ de UTF-8-codering blijft gebruiken in plaats van de oude CP-1252 gebruikt door FidoCAD. FidoCAD voor Windows kan de aanvullende informatie die door de opdracht **FCJ** wordt overgedragen, niet begrijpen. Bij het lezen met de originele FidoCAD van een FidoCadJ-tekening die een extensie bevat, geeft het programma enkele fouten en geeft een resultaat dat afwijkt van de originele FidoCadJ-tekening. De resultaten zijn afhankelijk van de gebruikte extensie: een stippellijn wordt weergegeven als ononderbroken en de pijlpunten worden niet getekend. Aan de andere kant wordt de tekst die is gekoppeld aan de naam en de waarde van een macro perfect weergegeven.

Figuur 3.1 laat zien wat kan worden verkregen met FidoCAD om het FidoCadJ-bestand te lezen dat figuur 2.20 beschrijft. Na het negeren van de FidoCAD-fouten, ontbreken er enkele details (de streepjes en de pijl), maar de algehele tekening is nog steeds begrijpelijk.



Figuur 3.1: Figuur 2.20 zoals het zou verschijnen op FidoCAD voor Windows.

Een belangrijk verschil met de oorspronkelijke FidoCAD is dat FidoCadJ het mogelijk maakt om een bepaald aantal configuratiertips op te slaan in de uitvoerbestanden. De opdracht die daarvoor wordt gebruikt is **FJC** en moet normaal helemaal aan het begin van het bestand worden geplaatst. In de volgende paragrafen worden de casussen beschreven die worden behandeld.

3.4.1 Laag instelling

De opbouw van de lagen wordt gespecificeerd met de opdracht **FJC L**. FidoCadJ slaat alleen gegevens op als de corresponderende laag is gewijzigd ten opzichte van de standaard laag. De syntaxis is als volgt:

```
FJC L n xxxx yy
```

waar *n* staat voor het nummer van de laag (variërend van 0 tot 15), *xxxx* is een 32-bits geheel getal met instellingen over de te gebruiken RGB-kleur. De rode component zit in de bits 16-23, de groene in de bits 8-15 en de blauwe in de bits 0-7. De waarde *yy* is een floating point constante met een enkele precisie die de laagtransparantie tussen 0,0 (volledig transparant) en 1,0 (volledig ondoorzichtig) geeft.

Een tweede belangrijke informatie is de naam van de laag, gespecificeerd (als de gebruiker deze heeft gewijzigd) als volgt:

```
FJC N n aaaaa
```

waarbij *n* het nummer van de laag is (geheel getal van 0 tot 15), terwijl *aaaa* de naam is van de laag die in aanmerking moet worden genomen. Als deze opdracht niet aanwezig is, is de naam van de laag de naam die standaard door FidoCadJ wordt gebruikt, afhankelijk van de taal en de lokale configuratie van het besturingssysteem.

3.4.2 Instelling elektrische aansluiting

De grootte van de zwarte cirkel die wordt gebruikt om een elektrische verbinding aan te geven kan door de gebruiker worden gewijzigd. Wanneer de FidoCadJ-extensies actief zijn, wordt de geselecteerde waarde als volgt in het bestand opgeslagen met de opdracht **FJC C**:

```
FJC C aaaa
```

waarbij *aaaa* een floating point waarde met dubbele precisie is (positief) die de diameter van de elektrische verbinding geeft, in FidoCadJ logische eenheden.

3.4.3 Lijnbreedte

De lijnbreedte voor de “pen” die wordt gebruikt tijdens het tekenen van elektrische schema’s kan worden gewijzigd. FidoCadJ gebruikt het commando **FJC A** om de lijnbreedte van de lijnen, ovalen, Bézier, complexe krommen, rechthoeken en polygonaal te specificeren. De breedte kan een dubbele precisieconstante zijn.

```
FJC A aaaa
```

Waar *aaaa* staat voor de lijnbreedte (in logische eenheden). De standaardbreedte is intern gedefinieerd op 0,5 logische eenheden. In oudere versies van FidoCadJ werd een commando **FJC B bbbb** gebruikt om de lijnbreedte van gebogen lijnen (ovalen, Bézier) te specificeren. Sinds versie 0.23.6 is dit onderscheid geëlimineerd en wordt dit commando niet meer overgenomen.

3.5 Tolerantie voor syntaxisfouten

FidoCadJ is ontworpen om fouten en/of bepaalde syntaxisfouten in de opdrachten die aan het programma worden doorgegeven te tolereren. Het is duidelijk dat, tenzij je een kristallen bol hebt aangesloten als een USB-apparaat, het programma geen fouten kan corrigeren, maar gewoon alle betrokken regels overslaat (en verwijdert).

Een uitzondering op dit gedrag is dat een aantal elementen kan worden opgegeven zonder de laag, die wordt beschouwd als onderdeel van laag 0 (schema’s). Dit is voor achterwaartse compatibiliteit met FidoCAD.

3.6 Bibliotheek formaat

De bestandsstructuur van een bibliotheekbestand is vrij eenvoudig:

```
[FIDOLIB Librairie de base]
{Sybols de base}
[000 Terminal]
LI 100 100 102 100
EV 102 98 106 102
[010 Terminal +]
LI 100 100 102 100
EV 102 98 106 102
LI 103 100 105 100
LI 104 99 104 101
[020 Terminal -]
LI 100 100 102 100
EV 102 98 106 102
LI 103 100 105 100
...
...
```

De eerste regel bevat, tussen vierkante haakjes, de naam van de bibliotheek (voorafgegaan door “FIDOLIB”). De tweede regel moet tussen accolades de bibliotheekcategorie bevatten waaronder de (later) in het bestand gespecificeerde macro's moeten worden opgeslagen.

Elke macro bestaat uit een header (tussen vierkante haken) en een reeks opdrachten. De header bestaat uit een *onderdeelnaam* (die uniek moet zijn binnen de bibliotheek) en de beschrijving ervan. De onderdeelnaam wordt gebruikt in een FidoCadJ-script, terwijl de beschrijving de gebruiker helpt bij het bladeren door alle macro's in het bestand. De opdrachten zijn niets anders dan FidoCadJ-tekeningen, waarbij het coördinaatpunt (100.100) als oorsprong wordt gebruikt. Dit is het punt dat moet worden gebruikt als referentie wanneer de macro wordt aangeroepen. In een FidoCadJ-script wordt een macro geïdentificeerd door “library.macro” gebruikt met de MC-opdracht.

Niets verhindert het aanroepen van een macro binnen een andere macro. Recursie (d.w.z. een macro die zichzelf aanroeft) moet echter worden vermeden.

Een bibliotheek *mag niet* onder de macrodefinities enige informatie over de FidoCadJ-configuratie *bevatten*. Met andere woorden, opdrachten **FJC** zouden *nooit* moeten verschijnen in een bibliotheekbestand.

3.7 Standaard bibliotheken

FidoCadJ bevat twee bibliotheken die traditioneel worden geleverd met FidoCAD. Dit zijn de standaardbibliotheek en de bibliotheek die de PCB-symbolen bevat. Een andere bibliotheek die is ontwikkeld voor FidoCAD en zo nuttig is dat deze is opgenomen als onderdeel van FidoCadJ, is de IHRAM-bibliotheek. IHRAM staat voor **it.hobby.radioamatori.moderato** en het is een Italiaanse Usenet-discussiegroep.

Het is echter mogelijk om de inhoud van deze interne bibliotheken te overschrijven door (menubalk en dan menu “Bestand/Opties/Bibliothekenmap”) een map op te geven die de te laden bibliotheken bevat. Als een bestand met

de naam `FCDstdlib.fcl` in deze map aanwezig is, wordt de inhoud ervan gebruikt in plaats van de standaardbibliotheek. Als een bestand met de naam `PCB.fcl` aanwezig is, wordt de inhoud ervan gebruikt in plaats van de interne PCB-bibliotheek. Andere bibliotheken met andere bestandsnamen (maar nog steeds met de extensie `fcl`) worden samen met de standaardbibliotheeken geladen. Als er een bestand met de naam `IHRAM.FCL` in het huidige zoekpad van de bibliotheek wordt deze geladen op de plaats van de versie die in het programma is ingesloten. Je hebt ook een bibliotheek met elektrische symbolen waarvan het bestand `elettrotecnica.fcl` heet.

De standaard bibliotheken zijn in FidoCadJ buiten de naam te herkennen aan het symbool met drie schijfjes boven elkaar (van “harddisk”). Eigen en andere bibliotheken die in de gedefinieerde map “Bestand/Opties/Bibliothekenmap” staan zijn te herkennen aan het diskette symbool.



Figuur 3.2: Voorbeeld van hoe bibliotheken in de bibliothekenmap verschijnen.

FidoCadJ beschouwt de bovenstaande bibliotheken als onderdeel van de standaardset. Met andere woorden, het splitst de symbolen in tekeningen niet, als ze tot die bibliotheken behoren, behalve wanneer de optie “Strikte compatibiliteit met FidoCAD” actief is. In dit geval wordt alleen de originele FidoCAD-bibliotheek als standaard beschouwd, precies zoals vroeger de originele software.

Hoofdstuk 4

Conclusie

We hebben in deze handleiding gezien hoe FidoCadJ gebruikt kan worden om een elektrisch schema of een eenvoudige printplaat te tekenen. In dit stadium moet de lezer over alle elementen beschikken die nodig zijn om FidoCadJ effectief te gebruiken voor zijn behoeften.

FidoCadJ mag niet als uniek worden beschouwd voor elektronica-ontwerpen. Het kan voor elk type 2D-tekening en in veel situaties worden gebruikt, op voorwaarde dat er specifieke bibliotheken beschikbaar zijn.

De voordelen van een gratis programma zijn dat het volledig openstaat naar de gebruikersgemeenschap. Om deze reden is uw feedback erg belangrijk (in ieder geval om te begrijpen of het project de moeite waard is om verder ontwikkeld te worden, en in welke richting). Als u contact met mij wilt opnemen, opent u een “issue” op het GitHub FidoCadJ-project¹.

¹<https://github.com/DarwinNE/FidoCadJ/issues>

Bijlage A

Platformspecifieke informatie

A.1 macOS

Een van de meest voorkomende kritieken op de vroege versies van FidoCadJ van Macintosh-gebruikers (zoals ik trouwens) was de slechte integratie van het programma op macOS. Vanaf versie 0.21.1 levert FidoCadJ specifieke inspanningen om beter te voldoen aan het uiterlijk en de filosofie van Mac's native applicaties. Om deze reden zijn sommige details in het programma enigszins anders wanneer FidoCadJ op een Apple-platform wordt uitgevoerd:

- standaard gebruikt FidoCadJ de Quaqua¹ look and feel wanneer de volledige applicatie (FidoCadJ.app in plaats van fidocadj.jar) wordt uitgevoerd. Aangezien Quaqua de prestaties op niet zo recente machines kan vertragen, is het mogelijk om het uit te schakelen via de instellingen van het menu Voorkeuren.
- De menubalk wordt weergegeven op zijn plaats, dat is de bovenkant van het scherm.
- De menu-items "Voorkeuren" en "Over FidoCadJ" staan op hun plaats, namelijk in het FidoCadJ-menu.
- Het programma vertelt het besturingssysteem dat het .fcd bestanden kan openen; Deze zijn gekoppeld aan een specifiek icoon, dat voldoende suggestief moet zijn.

A.1.1 Hoe FidoCadJ op macOS te downloaden en uit te voeren

FidoCadJ kan werken met een macOS-versie die recenter is dan 10.6 (Snow Leopard). Onder de motorkap heeft FidoCadJ minimaal versie 9 van Java nodig. Om marketing redenen installeert Apple Java niet met de laatste versies van macOS. Trouwens, Apple staat niet toe om zijn App Store-software te verspreiden

¹<http://www.randelshofer.ch/quaqua/>

op basis van Java of gedistribueerd onder GPL. Dit is een van de belangrijkste redenen waarom het vrij onwaarschijnlijk is dat FidoCadJ zal worden overgezet naar iPad en iPhone.

Zelfs als u het Java-archief `fidocadj.jar` gebruikt zoals u zou doen op andere besturingssystemen, kunt u op macOS de specifiek op maat gemaakte applicatie gebruiken. Alles werkt net als een native applicatie: u kunt de schijfkopie downloaden via de volgende link:

https://github.com/DarwinNE/FidoCadJ/releases/download/v0.24.8/FidoCadJ_MacOSX.dmg

U kunt dan de kopie openen en `FidoCadJ.app` verplaatsen naar de **Toepassingen** map, waar u het op precies dezelfde manier kunt gebruiken als elke andere Macintosh-toepassing. Om FidoCadJ te verwijderen, sleept u gewoon `FidoCadJ.app` naar de prullenbak.

A.2 Linux

door Roby IZ1CYN

Voorwaarde: JRE 9 van Oracle en/of OpenJDK 9 JRE (of opeenvolgende versies die compatibel zijn met de specificaties van het programma) moeten zijn geïnstalleerd. In paragraaf A.2.1 wordt de installatie van het programma met alleen commando's van een terminal beschreven. In paragraaf A.2.2 wordt in plaats daarvan de interactie met een grafische omgeving gebruikt. Een slechte configuratie van uw Java-runtime-omgeving geeft slechte prestaties van FidoCadJ².

A.2.1 Via elk platform, vanaf de terminal

Download het programma met behulp van de `wget` commando:

```
$ wget https://github.com/DarwinNE/FidoCadJ/releases/
  download/v0.24.8/fidocadj-0.24.8.jar
--00:48:18-- https://github.com/DarwinNE/FidoCadJ/releases/
  download/v0.24.8/fidocadj-0.24.8.jar
=> 'fidocadj.jar'
Resolution of github.com is being done... xxx.xxx.xxx.xxx
Connection to github.com is being done... xxx.xxx.xxx.xxx
          connected.
HTTP request sent, waiting for answer... 200 OK
Length: 343,207 (335K) [application/java-archive]

100% [=====] 343,207
        422.48K/s

00:48:30 (420.55 KB/s) - ''fidocadj-0.24.8.jar'' saved [xxx/
yyy]
$
```

²Opmerking redacteur: Ik zweer het, het is waar! *Alsjeblieft*, beledig me niet als je favoriete Linux-distributie wordt geleverd met een zeer onbetrouwbare versie van Java.

Als alternatief, of als u problemen ondervindt, kunt u het bestand vanuit elke browser downloaden via de volgende URL:

<https://github.com/DarwinNE/FidoCadJ/releases/download/v0.24.8/fidocadj-0.24.8.jar>

en sla het bestand bijvoorbeeld op in uw `/home/[user_name]/Download` directory (veel moderne browsers doen dat standaard).

Maak een nieuwe map aan (maar in eerste instantie worden we een superuser door `su` of `sudo -s`):

```
$ su
-> enter password
# mkdir /usr/local/bin/fidocadj
```

... en we kunnen het gedownloade bestand daar naartoe verplaatsen (vervang `<user>` door de gebruikersnaam van het account waar het bestand is gedownload):

```
# mv /home/[user_name]/Download/fidocadj-0.24.8.jar /usr/
    local/bin/fidocadj
```

Maak van het bestand een uitvoerbaar bestand

```
# chmod +x /usr/local/bin/fidocadj/fidocadj-0.24.8.jar
```

En we mogen niet vergeten om weer normale gebruikers te worden:

```
# exit
```

En nu kunnen we het programma uitvoeren:

```
$ /usr/local/bin/fidocadj/fidocadj-0.24.8.jar
```

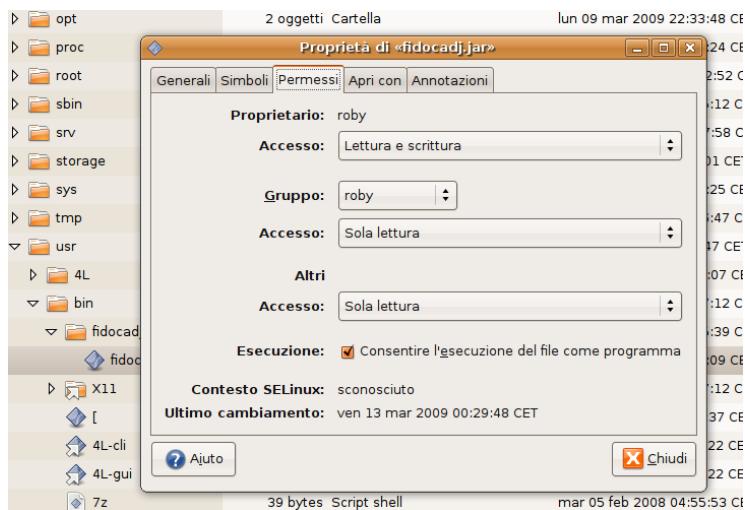
A.2.2 Op een grafisch systeem

In het voorbeeld gebruiken we Ubuntu 8.04, maar voor oudere of nieuwere versies verandert er niet veel:

- Download het bestand vanuit de browser of met Gwget of vergelijkbare tools.
- Start onze Bestandsbeheerder (Nautilus, Konqueror...) als root (als we geen specifiek commando in het menu hebben, hoeven we het alleen maar vanaf de console te starten met behulp van `sudo nautilus`). We kunnen de volgende directory maken:

```
/usr/local/bin/fidocadj
```

en verplaats daar het bestand dat we zojuist hebben gedownload, dat in veel gevallen in een map staat zoals `/home/[gebruikersnaam]/Download/` na het downloaden.



Figuur A.1: Het instellen van bestandsrechten, op Ubuntu 8.04.

- Klik met de rechtermuisknop op het bestand, vanuit het venster selecteren we het tabblad “rechten”, we voegen het vinkje toe voor “Sta de uitvoering van het bestand toe als een programma”, zoals weergegeven in de figuur A.1 (alleen dan op zijn Italiaans...)
- Selecteer het tabblad “Openen als” en selecteer “OpenJDK Java 9 Runtime” of “Oracle Java 9 Runtime” (of een volgende versie).
- Klik op “Sluiten” en we zijn nu klaar om FidoCadJ uit te voeren: een dubbelklik op het uitvoerbare bestand, of we kunnen het toevoegen aan het hoofdmenu. De opdracht om toe te voegen is gewoon `/usr/local/bin/fidocadj/fidocadj-0.24.8.jar`.

A.2.3 “Portable” installatie

door Max2433BO

Als u niets op het systeem kunt installeren, kan FidoCadJ nog steeds worden gebruikt als een “draagbare” toepassing die overal kan worden uitgevoerd zonder dat hiervoor supergebruikersrechten nodig zijn.

Maak hiervoor een map `/home/[user_name]/fidocadj/` waar de volgende bestanden worden geplaatst:

- `fidocadj-0.24.8.jar`
- `jdk-9.0.4` (javaSE 9, of groter)
- a script `fidocadj-0.24.8.sh` om FidoCadJ te starten, met de juiste Java-versie.

```
#!/bin/bash
export PATH=.:./fidocadj/jdk-9.0.4/bin
export CLASSPATH=.:./fidocadj/jdk-9.0.4/lib:./fidocadj
./fidocadj/jdk-9.0.4/bin/java -jar ./fidocadj/fidocadj_0
.24.8.jar
```

Het script moet uitvoerbaar worden gemaakt met `chmod +x fidocadj-0.24.8.sh` in de directory waar het script aanwezig is. Natuurlijk moet u de Java-versie en de bijbehorende directory aanpassen aan uw behoeften.

A.2.4 Raspberry Pi 400

door *Joop Nijenhuis*

De Raspberry Pi 400 is een zeer goedkope computer. Goedkoop zijn betekent hier niet een armetierig uitgevoerd systeem. Dit is wat de toekomst zou kunnen zijn. Binnenin een 64-bit quad-core ARM-processor met 4Gb ram, Gigabit Ethernet, Wifi, Bluetooth. Het heeft 2 HDMI-videoopoorten, dus het kan op bijna alles worden aangesloten en het gebruikt een microSD-kaart als hoofdstation. Er zijn 1 USB 2.0-poort en 2 USB 3.0-poorten. Voeding via een USB type-C-poort. Er is een uitbreidingspoort voor experimenten. Hij verbruikt maar 15 Watt aan energie, waarschijnlijk veel minder omdat je niet alles gebruikt. Dit type computer wordt onder jongeren gepromoot om ze te laten programmeren en ze meer inzicht te geven in wat een computer is en kan. Raspberry Pi wordt geleverd met zijn eigen op Debian gebaseerde OS-distributie genaamd NOOBS. Op de kaart staat ook OpenJDK Java versie 11.x.x. Alles is al geïnstalleerd! Je zou de microSD-kaart achter in de kaartreader kunnen steken en het systeem laten draaien, of dat verstandig is, is een andere zaak. Als je een andere achtergrond hebt, dan vind je Linux misschien niet gemakkelijk. Wees je ervan bewust dat het ook anders kan! Je kunt de hierboven beschreven weg gaan of je kunt het op mijn manier doen. Om de een of andere reden houdt de Raspberry Pi-gemeenschap niet van Java, weet niet waarom, dus geen antwoorden daar. Ik vond “native” oplossingen niet beter en soms slechter dan wat ik al in Java had draaien. FidoCadJ is er een van en werkt prima met/op de Pi. Hoe nu verder? Maak in /home/pi een nieuwe map aan en geef deze de naam “Apps” of iets dergelijks. Open deze map en maak opnieuw een nieuwe map aan met de naam “fidocadj”. Kopieer de download “fidocadj-0.24.8.jar” naar deze map.

Start de teksteditor en kopieer de volgende regels erin:

```
#!/bin/sh

#go to script's directory
progdir=${0%/*}
cd $progdir

#launch fidocadj
java -Duser.home=$progdir -jar fidocadj-0.24.8.jar 2>
fidocadj-bugs.txt
```

Sla het op als “fidocadj.sh” in dezelfde map als FidoCadJ. U kunt dit bestand toevoegen aan het programma “Main Menu Editor” in de NOOBS-distributie. Met enige aanpassing kan deze opstelling voor elk ander Java-programma worden gebruikt. De “-Duser.home=\$progdir” zorgt ervoor dat altijd “\$progdir” wordt gebruikt. De “2>fidocadj-bugs.txt” stuurt elk probleem naar het bestand “fidocadj-bugs.txt”. Maak je geen zorgen, het bestand blijft leeg op mijn systeem, maar voor het geval er iets misgaat, heb ik waarschijnlijk enkele hints waar ik het probleem kan zoeken.

Het mooie van dit systeem is dat het altijd werkt (tot nu toe) en dat je ALLE dingen over het programma in ÉÉN map kunt vinden, in geval van problemen of als je gewoon alles wilt wissen. Omdat OpenJDK Java, versie 11, deel uitmaakt van de NOOBS-distributie, krijgt het updates elke keer dat u het Pi-systeem bijwerkt en als er een update voor Java is. Nog iets: ik werk met een Wacom Tablet en een pen als muis. Werkt heel goed met FidoCadJ. Als je meer wilt lezen; <https://forums.raspberrypi.com/viewtopic.php?t=320892>

A.3 Windows

Wanneer FidoCadJ een Windows-platform herkent, zal het proberen de native Look and Feel te gebruiken.

A.3.1 Hoe FidoCadJ te downloaden en uit te voeren

Heel vaak, als je Java hebt geïnstalleerd, hoef je alleen maar `fidocadj-0.24.8.jar` bestand te downloaden en met een dubbelklik te laten starten. Als uw besturingssysteem het na het downloaden als een `zip` archive ziet, is Java waarschijnlijk niet beschikbaar op uw computer. Met Oracle kunt u de huidige versie van de Java-runtime gratis downloaden en installeren (vervang “it” eventueel voor “nl” of de aanduiding voor uw land): <http://www.java.com/it/download/> OpenJDK is o.a. te downloaden van: <https://bell-sw.com/pages/downloads/>

A.4 Compileren van sources

Als u wilt compileren uit FidoCadJ-bronnen, is dat gelukkig een eenvoudige taak, omdat u alleen een werkende Java SDK en GNU-make-up nodig hebt. Meer details worden gegeven in het bestand README.md. Het FidoCadJ-project heeft een build-automatiseringssysteem, gebaseerd op make. Je moet eerst de volledige bron repository downloaden van GitHub. De make-regels in tabel A.1 zijn geïmplementeerd en kunnen vanaf de opdrachtregel worden gebruikt. Als u de broncode van FidoCadJ wilt bestuderen, voert u `make createdoc` in om de Javadoc-beschrijving te verkrijgen die geplaatst wordt in de `doc` submap.

Als u Windows gebruikt, kan het handig zijn om de `winbuild.bat` script te gebruiken in plaats van `make`. Het script wordt in de `dev_tools` directory geplaatst en moet worden gebruikt met een actie gekozen uit `{run|clean|compile|force|rebuild}`.

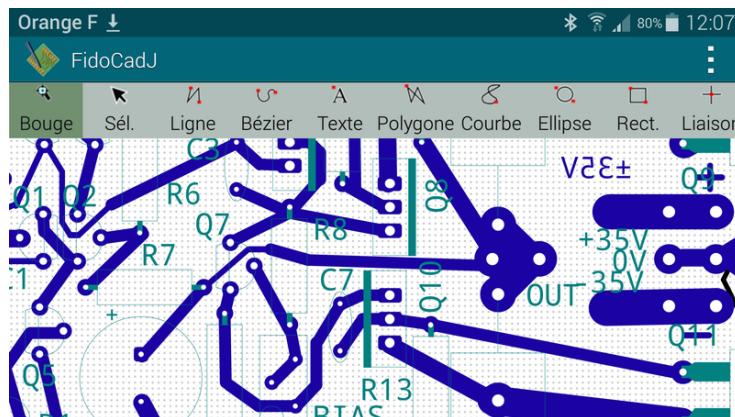
Tabel A.1: Lijst met beschikbare make doelen om FidoCadJ uit bronnen te compileren.

Regel	Beschrijving
<code>make</code>	Compile FidoCadJ (implicit)
<code>make clean</code>	Erase all the compiled classes
<code>make cleanall</code>	Do a clean, erase fidocadj.jar, Javadocs
<code>make compile</code>	Compile FidoCadJ (explicit)
<code>make createdoc</code>	Run Javadoc on all source files
<code>make createjar</code>	Prepare jar/fidocadj.jar
<code>make rebuild</code>	Do a clean and then run FidoCadJ
<code>make run</code>	Run FidoCadJ

A.5 Android

In 2015 is een specifieke versie van FidoCadJ voor Android ontwikkeld. Het deelt veel van de broncode met de pc-versie, maar alle grafische gebruikersinterfacecode moest worden herschreven. Het zou moeten werken op verschillende apparaten met minimaal Android 4.0: zowel smartphones als tablets. Figuur A.2 toont bijvoorbeeld het uiterlijk van FidoCadJ voor Android met een Samsung S5-smartphone.

Er zijn enkele kleine verschillen tussen de Android-applicatie en de pc-applicatie die in de handleiding wordt beschreven. Ze zijn natuurlijk perfect compatibel met het bestandsformaat en kunnen tekeningen uitwisselen.



Figuur A.2: FidoCadJ voor Android, draaiend op een Samsung S5-smartphone.

- U kunt geen gebruikersbibliotheeken maken en bewerken, maar u kunt wel de op een pc gebouwde bibliotheken gebruiken.
- U slaat de gebruikersbibliotheeken op in het externe bestandssysteem (de “SD-kaart”), in de `FidoCadJ/Libs` map.
- De symbolen worden niet de hele tijd getoond om ruimte te besparen: je moet met je vinger over de rechterkant van het scherm vegen om ze te tonen.

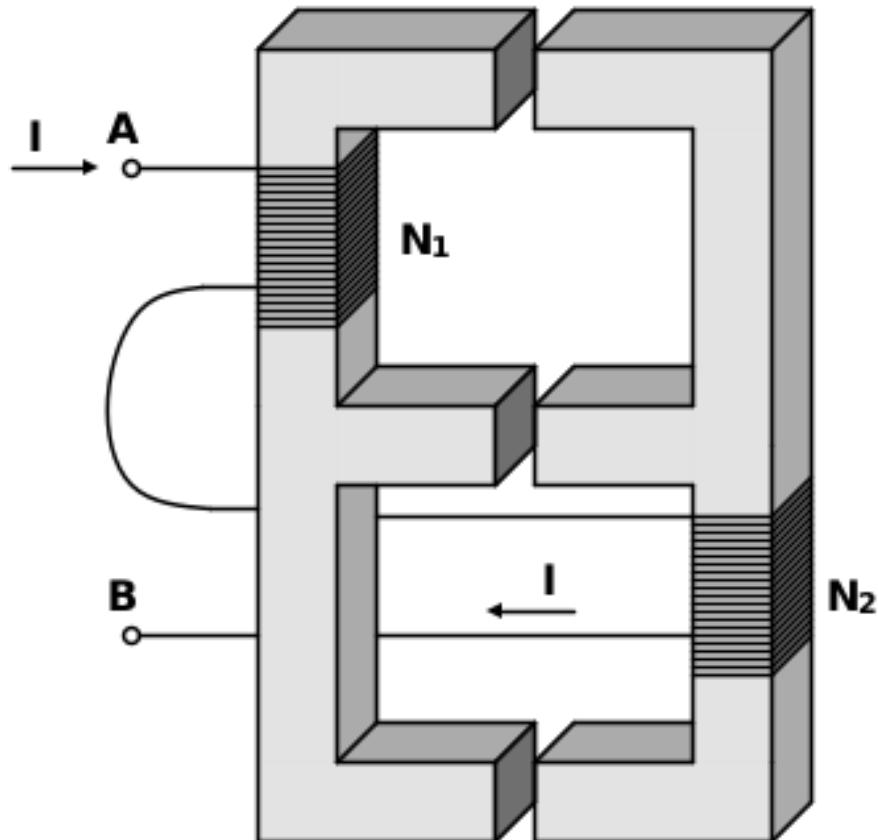
- Je hebt een extra knop in de werkbalk, waarmee je het getoonde gedeelte van de tekening kunt verplaatsen en zoomen. Terwijl u zich in deze “Verplaatsen”-status bevindt, raakt u het scherm aan en beweegt u uw vinger om de tekening te verplaatsen(pannen). Gebruik twee vingers en “knijp” om de zoom te vergroten of te verkleinen.

In 2020 ziet de toekomst van de Android-applicatie er echter nogal somber uit. Allereerst lijkt het erop dat Google Play de enige effectieve manier is om de applicatie te verspreiden. Een ontwikkelaarsaccount om een applicatie te publiceren kost geld. Alternatieven bestaan, maar zijn niet wijdverbreid. Bovendien is de applicatie in 2014 ontwikkeld met technieken die nu blijkbaar door Google worden uitgefaseerd. Wat nog belangrijker is, het onderhouden van een moderne app is gewoon niet iets dat ik alleen wil doen, gezien de tijd die ik heb gekozen om aan FidoCadJ te besteden. Als iemand ervoor wil zorgen, stap dan naar voren.

Bijlage B

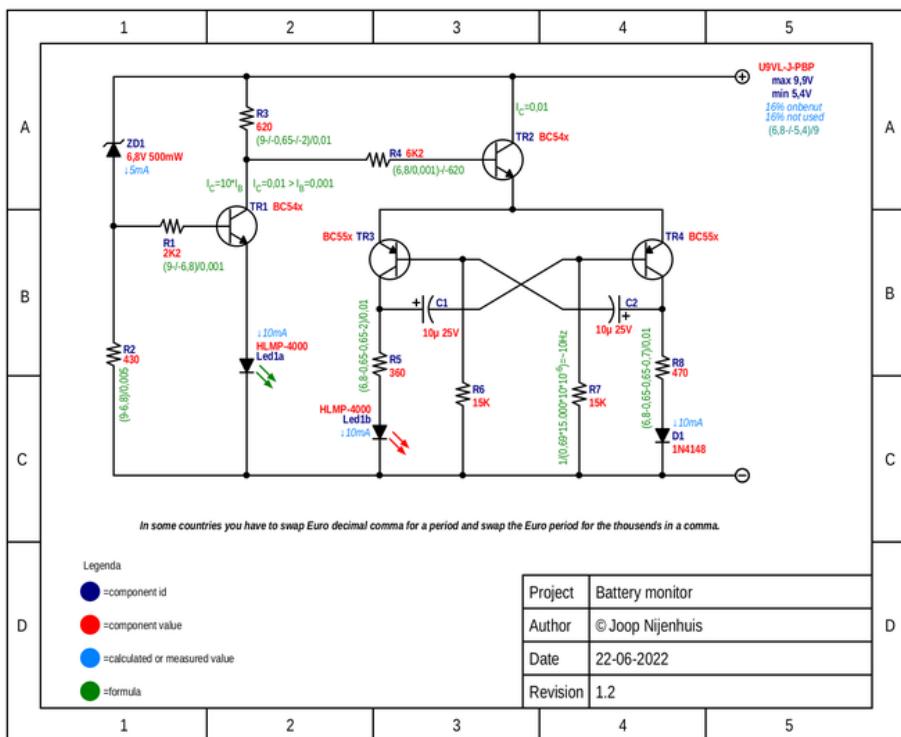
FidoCadJ voorbeelden

en hoe het ook anders kan...

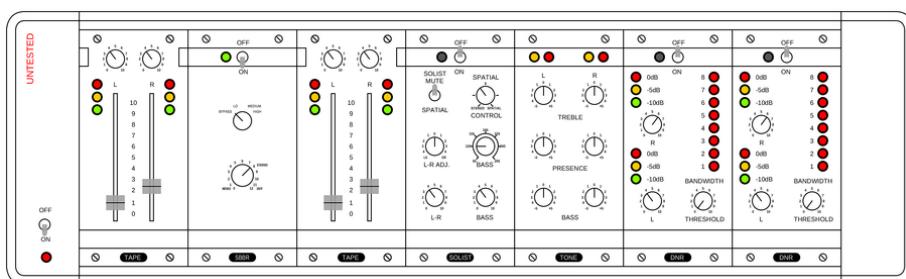


Figuur B.1: Voorbeeld van gebruiker DanteCpp (Oostenrijk)

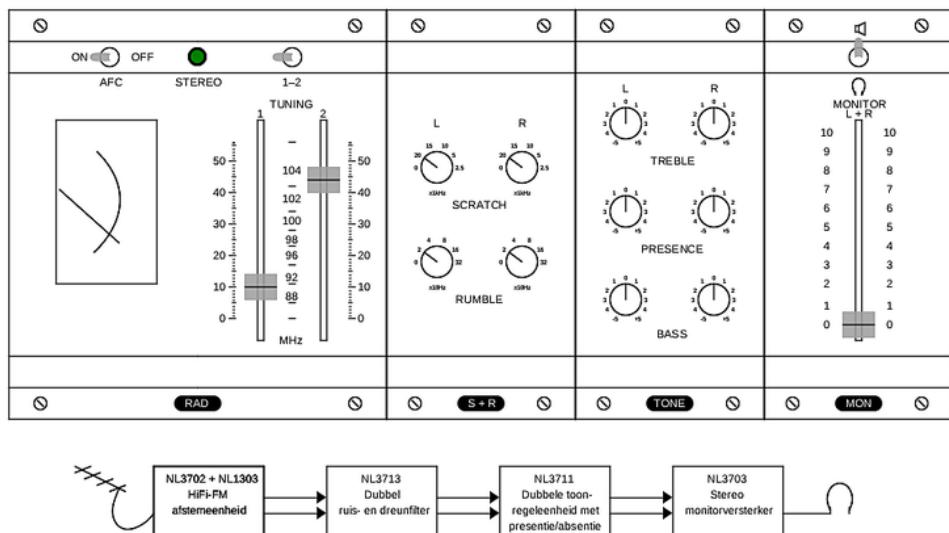
Dit voorbeeld komt uit <https://github.com/DarwinNE/FidoCadJ/issues/16> en is oorspronkelijk gemaakt in een Ubuntu Linux systeem. Ik heb de code uit de forumpost gekopieerd en in een ascii tekstverwerker geplakt en opgeslagen met een bestandsnaam en de extensie ".fcd". Daarna ingelezen in FidoCadJ op een Raspberry Pi 400 (Nederland). Het enigste wat ik moest herstellen waren de "N1" en "N2" bijschriften waarbij de getallen op de letter "N" werden gezet. Voor de rest is de tekening ongewijzigd. Vervolgens in FidoCadJ geëxporteerd naar een png-afbeelding. En dit is het resultaat.



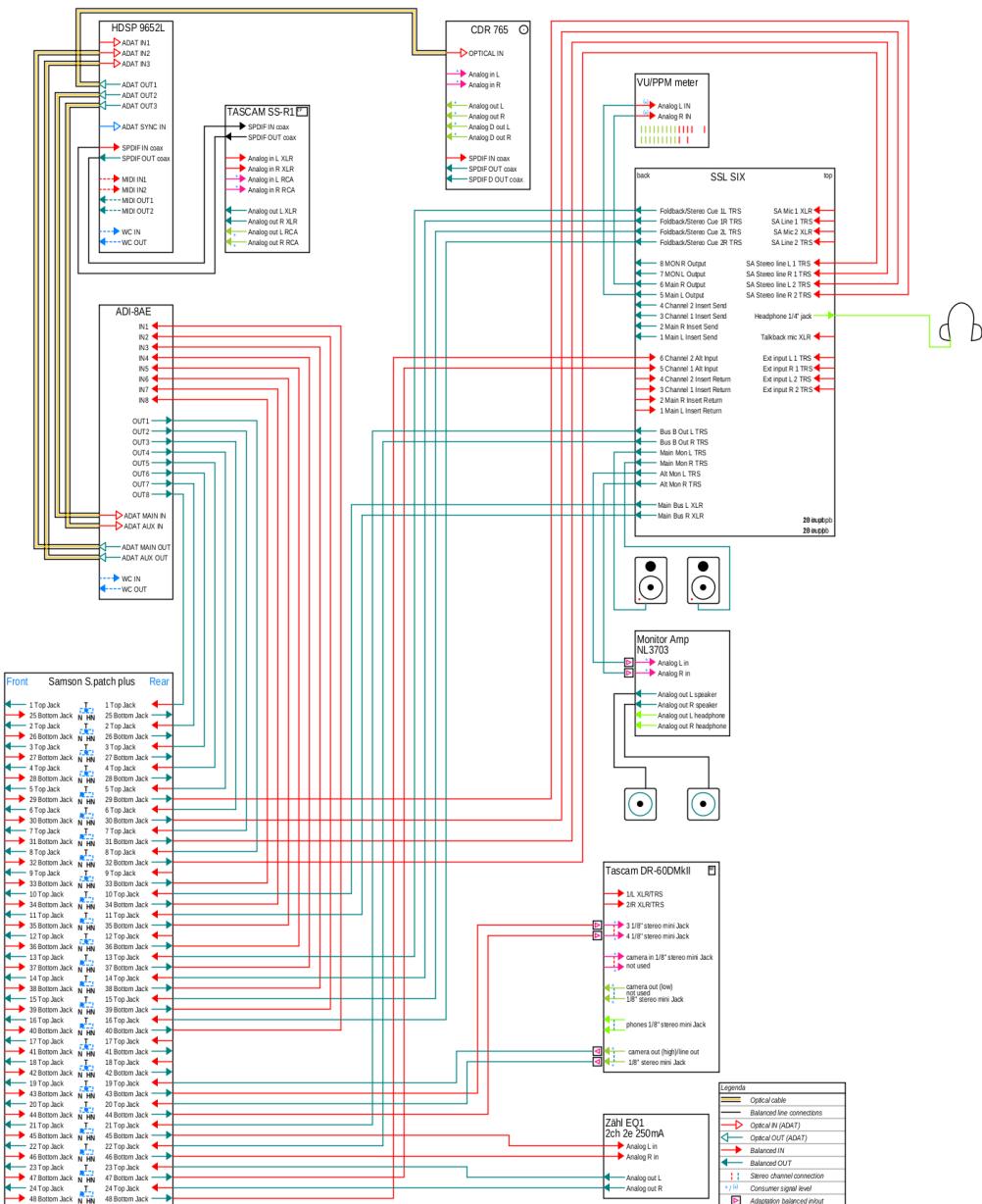
Figuur B.2: Een voorbeeld van een FidoCadJ tekening met andere inzet van de beschikbare lagen (lees kleuren).



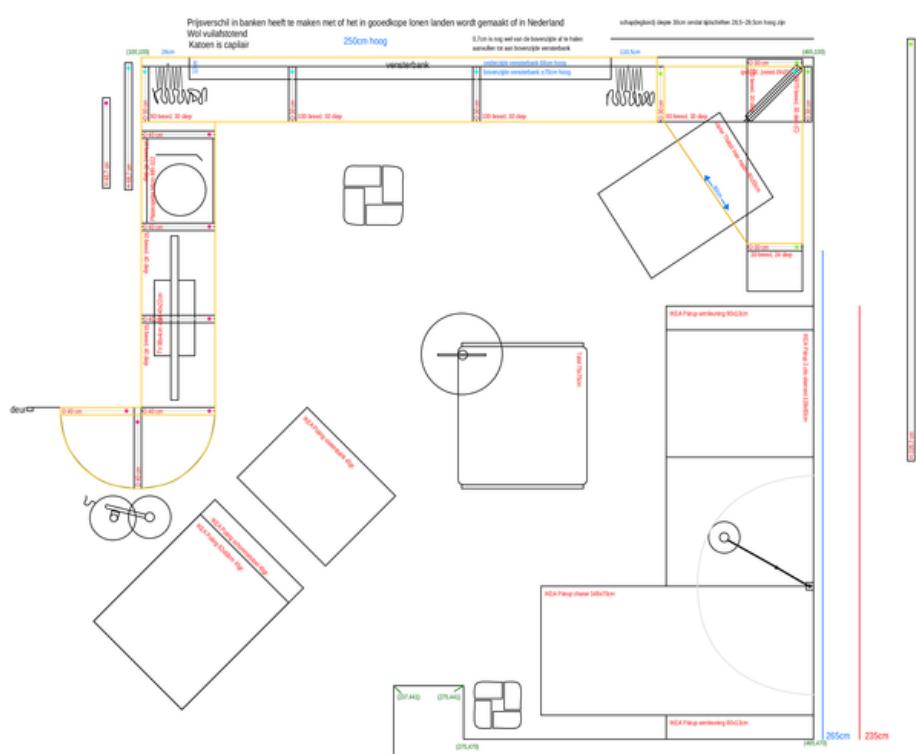
Figuur B.3: Modulair samengesteld effectrack. Door op de pagina in te zoomen (>200%) kunnen knoppen en teksten beter leesbaar worden.



Figuur B.4: Tuner samengesteld uit mengversterkereenheden van Philips. Door op de pagina in te zoomen (>200%) kunnen knoppen en teksten beter leesbaar worden.



Figuur B.5: Voorbeeld van een geluidsstudio indeling (de werkelijke opzet is veel gecompliceerder en past niet op een A4, het idee zal wel duidelijk zijn). Door op de pagina in te zoomen (>200%) kunnen teksten beter leesbaar worden.



Figuur B.6: Mijn poging voor een nieuwe indeling van de woonkamer.

Bijlage C

FidoCadJ art





Tiger, tiger, burning bright... By Igor Arbanas "elettrodomus".

Index

- ++, 37
- L^AT_EX, 25
- PDFL^AT_EX, 25
- A4, 20
- Abstract, iii
- Android, 53
- App Store, 47
- autoplacer, 15
- autorouter, 15
- Bézier, 9, 35, 38
- Bézier-curve, 38
- BE, 38, 39
- Bekijk/Lagen, 15
- Bestand/Opties, 29
- bibliotheek, 29
 - standaard bibliotheek, 29
- Bibliotheek bestandsnaam, 30
- Bibliotheek formaat, 44
- Bibliotheekbestanden gebruiken, 29
- Bibliotheken bijwerken, 30
- bitmapformaat, 24
- CAD, 15
- CadSoft, 25
- Chinees, 3
- circuit, 30, 37
- Coördinaten systeem, 34
- coördinatensysteem, 34
- code, 14
- codering, 34
- Compatibiliteit met FidoCAD, 45
- compileren, 52
- Compileren van sources, 52
- compressie met verlies, 25
- Conclusie, 46
- connectie, 9
- controlepunt, 11
- Courier New, 37
- CP, 39
- CP -1252, 41
- CV, 39
- Dankbetuigingen, iv
- De filosofie van FidoCadJ, 1
- De lagen, 15
- De liniaal gebruiken, 22
- diagonale shift, 14
- Duits, 3
- e-mail, 14
- Eagle, iv, 25
- Een eenvoudig schema, 10
- Een eenvoudige printplaat, 15
- Een symbool splitsen, 33
- Een van de grondleggers van circuit-analyse, 61
- eenvoudige tekst, 37
- Eenvoudige tekst (verouderd), 36
- Elektrisch knooppunt, 40
- elektrisch schema, 1, 8, 15, 35
- element, 6, 35
- Elementen tekenen, 35
- ellips, 9, 35
- encoding, 41
- Engels, 3
- EP, 38
- EPS, 2, 25
- EV, 38
- export, 24
- Exporteren, 24
- FCJ, 41
- FidoCAD, 1, 2, 8, 23, 30, 34–36, 39, 41–45
- FidoCadJ art, 61
- FidoCadJ en de toekomst, 4
- FidoCadJ extensies, 41
- FidoCadJ Licentie, vi
- FidoCadJ voorbeelden, 55
- FidoCadJ-bronnen, 52

- FidoCadJ-extensie, 12, 41
 FidoCadJ-opties, 26
 FidoCadJ.app, 47
 fidocadj.jar, 47
 FIDOLIB, 44
 FidoReadJ, 2
 FJC, 42, 44
 FJC A, 43
 FJC B, 43
 FJC C, 43
 FJC L, 42
 forum, 14
 fouttolerantie, 43
 Frans, 3
 Geavanceerde tekst, 37
 gereedschapbalk, 6, 8, 12
 Geschiedenis van FidoCadJ, 2
 Gevulde of lege ellips, 38
 Gevulde of lege rechthoek, 36
 Gevulde of Lege veelhoek, 38
 GIG, 22
 GitHub, 3
 Grieks, 3
 Groep, 32
 GTK+, 28
 header, 34, 44
 Hernoemen (bibliotheeksymbool), 33
 Het raster, 15
 Hoe FidoCadJ op macOS te downloaden en uit te voeren, 47
 Hoe FidoCadJ te downloaden en uit te voeren, 52
<http://www.matematicamente.it>, iv
 IHRaM-bibliotheek, 29
 Inhoudsopgave, viii
 Inkscape, 25
 Inleiding, 1
 Instelling elektrische aansluiting, 43
 interpreter, 2
 iPad, 48
 iPhone, 48
 it.hobby.elettronica, iv, 2
 it.hobby.fai-da-te, 2
 Italiaans, 3
 Japans, 3
 jar, 26
 Java, 1–3, 26, 28
 JPG, 25
 JRE, 26, 48
 KiCad, 16
 Klem aan raster, 15
 kleur, 15
 knooppunt, 35
 Kopiëren bibliotheek(symbool), 33
 Kopiëren, split de niet standaard ma-
 cros, 30
 Koptekst beschrijving, 34
 kristallen bol, 43
 kromme, 9, 35
 kruisen van sporen, 16
 laag, 15, 16, 43
 Laag instelling, 42
 Laat raster zien, 15
 LATEX, 4
 LI, 35
 Licentie, ii
 lijn, 9, 35
 Lijnbreedte, 43
 lijnstijlen, 22
 Lijst met exportbestandsindelingen, 25
 Lijst van figuren, x
 Lijst van tabellen, xi
 Linux, iv, 28, 48
 logische eenheid, 15, 22, 37
 look & feel , 28
 Lorenzo Lutti, 1, 36
 Macintosh, 7, 26
 macOS, 6, 7, 47
 macro, 8, 35, 44
 Macro-oproep, 41
 Matematicamente, 5
 maximale lengte van de tekst, 37
 maximum aantal hoekpunten, 38, 39
 MC, 41, 44
 menu balken, 6
 menubalk, 8, 10, 15, 19, 24, 29, 30
 Modulair samengesteld effecttrack, 57
 Motif, 28
 MS-DOS-prompt, 26
 Naam, 32
 Natuurlijke kubieke kromme (complexe
 kromme), 39

- Nederlands, 3
newsgroup, iv
Nieuwe indeling woonkamer, 60
Nieuwe symbolen definiëren, 30
nieuwsgroep, 14
nieuwsgroepen, 2
NOOBS, 51
- Objectgeoriënteerd programmeren, 2
ontwikkelmachine, 19
Oorsprong, 32
Op een grafisch systeem, 49
opdrachtregel, 26
Opdrachtregel opties, 26
OpenJDK, 48
Opslaan als..., split de niet standaard macros, 30
Oracle, 26, 48
- PA, 40
parameters, 35
PCB, 1, 35
 footprint, 8
PCB oog, 35, 40
PCB oog, 9
PCB Spoor, 17, 18
PCB spoor, 9
PCB-bibliotheek, 44
PCB-spoor, 35, 40
PDF, 2, 25
PGF, 2, 25
Pijl- en lijnstijlen, 22
pijlen, 22
pijlstijlen, 22
PL, 40
Plakken bibliotheek(symbool), 33
Platformspecifieke informatie, 47
PNG, 25, 27
polygoon, 16, 17
Polylijnbogen, 38
Portable installatie, 50
Postscript, 25
PP, 38
Press&Peel, 19
primitief, 6
printen, 19
printplaat, 1, 15, 18, 20
PV, 38
- Quaqua, iv, 47
- R41 transfers, 15, 17
Raspberry Pi 400, 6, 7, 51
raster, 15
rechthoek, 9, 35
rechthoek., 16
recursie, 44
repository, 52
resolutie, 15
RP, 36
RV, 36
- SA, 40
Samenvatting tekenopdrachten, 9
SCR, 25
segment, 35
selectie, 8, 9
selectiemodus, 16
slepen en neerzetten, 33
Snel zoeken, 8
Snow Leopard, 47
solderen, 15
SourceForge, 3
Spaans, 3
Spline, 39
Standaard bibliotheken, 44
standaardbibliotheek, 10, 44
sterretje, 37
streeppjesstijlen, 22
superuser, 49
SVG, 25
symbool, 10
- TE, 37
teken afmetingen, 18
Teken hulpmiddelen, 6
Tekenen met FidoCadJ, 6
Tekenformaat, macro's & bibliotheek, 34
tekst, 9, 35
 geroteerd, 37
 gespiegeld, 37
 stijl, 37
teksten spiegelen, 18
terminal, 26
Tiger, tiger, burning bright..., 62
Toets, 32
Tolerantie voor syntaxfouten, 43
transistor, 10
Tsjechisch, 3

Tuner samengesteld uit mengversterkerseenheden van Philips, 58
TY, 37

Ubuntu, 49, 50
Unix, 26
Usenet-groep, 1
UTF-8, 34, 41

vector tekening, 6
vectorformaat, 24
veelhoek, 9, 35
verouderd, 36
verschuiven, 9
Verwijderen (bibliotheeksymbool), 33
Via elk platform, vanaf de terminal, 48
vierkante haakjes, 44
Voorbeeld FidoCadJ tekening met andere inzet lagen, 57
Voorbeeld van een geluidsstudio indeling, 59
Voorbeeld van gebruiker DanteCpp (Oostenrijk), 56

Wacom tablet, 52
weerstand, 12
Werkvlakbalk, 6, 8, 15
Wijzigen sneltoets, 33
Wijzigen van bestaande symbolen, 33
Windows, 1, 26, 52
WInE, 1
www.electroyou.it, 4, 5
www.grix.it, iv, 5

zeefdruk, 15, 16
zoom, 9, 36

Deze handleiding is geschreven met behulp van PDFLATEX met TeXstudio op een Raspberry Pi 400. Lijsten zijn samengesteld met behulp van de [listings](#) pakket. Het [pgf](#) pakket is gebruikt voor de figuur [2.6](#) en ook voor de afbeelding op pagina [61](#). Zie de preamble van het tex-bestand voor de benodigde pakketten (met enkele opmerkingen). Al die pakketten zijn beschikbaar in het CTAN-archief.