caps Accurate Capacitance V1.01.006
Installation and Use

Jed Marti

KI7NNP

November 15, 2023

The *caps* program accepts a file of capacitance values as measured by your capacitance meter and attempts to provide a circuit with 1, 2, 3, or 4 capacitors to give you the nearest to what you want. The program is command line driven, no graphics - what's the point?

1 Installation

You've downloaded *caps.tgz*. First expand this somewhere, then move the executable wherever you want it.
For Linux:

```
tar xvfz caps.tgz
sudo mv dist/caps /usr/local/bin
```

For Windows this puts the executables in c:\local, you can put it wherever you want[1]

```
tar xvfz caps.tgz
copy dist\caps.exe c:\local
```

You'll find these files:

00README Just in case.

64caps.dat A file of capacitor selections for collections of up to 64 capacitors.

caps The Linux executable.

rcaps The Raspberry PI executable.

caps.c The main program.

caps.exe The windows executable (just in case)

caps32 A selection of my capacitors for testing. Make your own.

caps.h Prototypes and other stuff for the program.

[1] Windows 11 appears to come with `tar` if not, you'll have to download `tar.exe` from the Internet.

Makefile To create executable program, editable for Linux or Windows.

manual.pdf This manual.

solutions.c The circuit solutions.

utils.c Miscellaneous utility functions.

To build the Linux program from sources do the following:

```
cd dist
make clean
make
sudo make install
```

For Windows you'll need the `gcc` compiler, if you want Microsoft C you're on your own. First, modify `Makefile` to substitute Windows for the Linux build. It should look like:

```
    .
    .
    .
# Linux 64 style.
#CC = cc
#CFLAGS = -Ofast
#EXE =
#O = o
#RM = rm -rf
#CP = cp
#BIN = /usr/local/bin
#LFLAGS = -lm

# Windows GCC - uncomment the next lines for GCC - tested with Windows 11.
CC = gcc
CFLAGS = -O1 -DWINDOWS
EXE = .exe
O = o
RM = del /Q
CP = copy
BIN = /local
LFLAGS = -lm
DEL = del
    .
    .
    .
```

Then:

```
cd dist
make
make install
```

2 Creating Capacitance Files

The accuracy of the result depends much upon the accuracy of your measurements. The simple multimeter capacitance checker is probably not accurate enough for picofarad level. In any case, select a number of capacitors to help create your result. Some near the value, twice the value and perhaps some small ones. Measure them, attach a tape or sticky note with the measured value and set aside. Create a file with one line for each capacitor measurement in integer picofarads. Blank lines will be ignored, a # in column 1 indicates a comment. Any characters after the first integer are ignored. Only one capacitor per line. See *dist/caps32* for an example.

3 Execution

Usage:

caps *cap-file-name permutation-file resultf target-pf*

cap-file-name The capacitor values to check against (see above on creating such files).

permutation-file The permutation file name. For example: **dist/64caps.dat** or wherever you put it.

resultf Where to put the results with the following options.

- **-** Output to *stdout*.
- ***integer*** Output goes to a file named **res***integer*. This consists of 3 integers separated by commas. The first is the target value, the second the closest value, and the third the number of capacitors.
- ***file-name*** Output to the named file.

For example:

```
caps dist/caps32 dist/64caps.dat - 2308
```

. . . should result in:

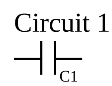
```
caps KI7NNP V1.01.004
Read 32 values from dist/caps32
Circuit 1 with capacitance 1009 PF
    C1 = 1009 PF

Circuit 2b with capacitance 2307 PF
    C1 = 4535 PF
    C2 = 4695 PF

Circuit 3b with capacitance 2309 PF
    C1 = 4535 PF
    C2 = 4695 PF
    C3 = 11 PF
```

```
Circuit 4j with capacitance 2308 PF
    C1 = 219 PF
    C2 = 92 PF
    C3 = 988 PF
    C4 = 1009 PF
```

4 The Circuits

Results will reference a circuit and the capacitor values. From the above, Circuit 4j referenced above appears on page 9.[2]

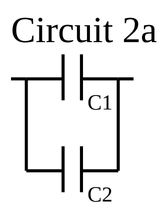
$$C = C_1 \tag{1}$$



4.1 Two Capacitors

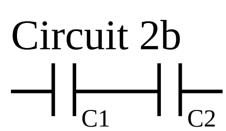
Two in parallel.

$$C = C_1 + C_2 \tag{2}$$



Two in series.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} \tag{3}$$

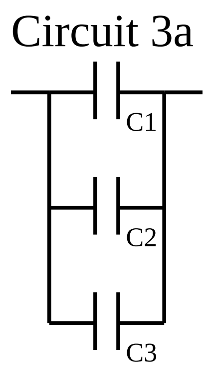


[2]Thanks to David Luker for Circuit 4i and Sean Oleary for paper comments.

4.2 Three Capacitors

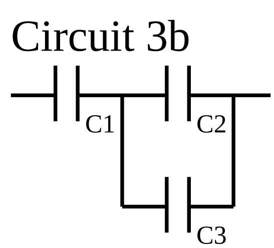
There are 4 possibilities.
Three in parallel.

$$C = C_1 + C_2 + C_3 \tag{4}$$



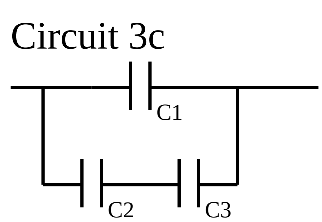
Two in series with one parallel to the last one.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2+C_3}} \tag{5}$$



Two in series with one parallel to both[3].

$$C = C_1 + \frac{1}{\frac{1}{C_2} + \frac{1}{C_3}} \tag{6}$$



[3]Thanks to Mark Adler AD1M for correcting this formula.

And finally, three in series.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}} \tag{7}$$

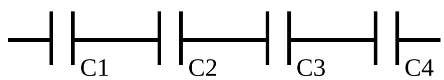
Circuit 3d

4.3 Four Capacitors

There are 10 different combinations that aren't symmetric.
Four in series.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4}} \tag{8}$$

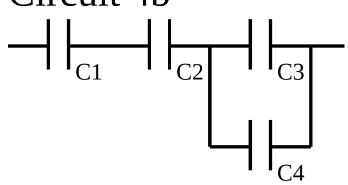
Circuit 4a



Three in series, one additional in parallel.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3 + C_4}} \tag{9}$$

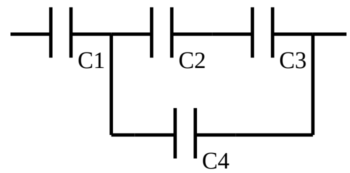
Circuit 4b



Three in series with one in parallel with two of them.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{c4 + \frac{1}{\frac{1}{C_2} + \frac{1}{C_3}}}} \tag{10}$$

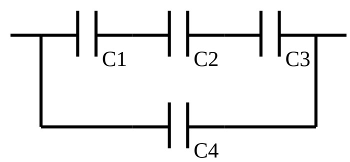
Circuit 4c



Three in series, one additional in parallel to all three.

$$C = C_4 + \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}} \tag{11}$$

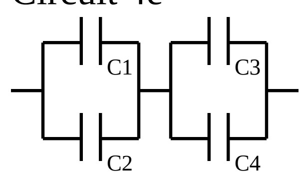
Circuit 4d



Two sets of two in parallel, parallel sets in series.

$$C = \frac{1}{\frac{1}{C_1+C_2} + \frac{1}{C_3+C_4}} \tag{12}$$

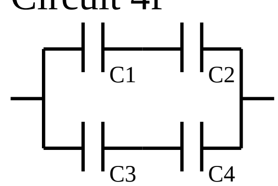
Circuit 4e



Two sets in series, the sets in parallel.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + \frac{1}{\frac{1}{C_3} + \frac{1}{C_4}} \tag{13}$$

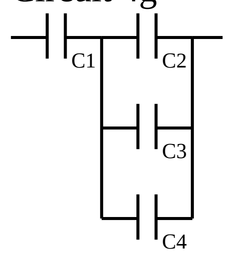
Circuit 4f



Three in parallel in series with one.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2+C_3+C_4}} \tag{14}$$

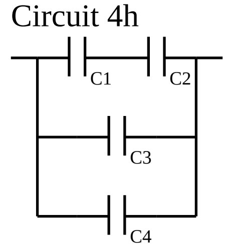
Circuit 4g



Two in series with two more in parallel with them.

$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + C_3 + C_4 \tag{15}$$

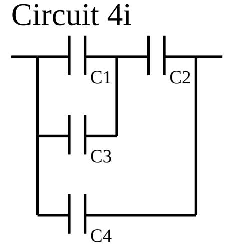
Circuit 4h



Hard to describe.

$$C = \frac{1}{\frac{1}{C_1+C_3} + \frac{1}{C_2}} + C_4 \tag{16}$$

Circuit 4i



4 in parallel.

$$C = C_1 + C_2 + C_3 + C_4 \tag{17}$$

Circuit 4j