COFFEE−HOWTO

Table of Contents

COFFEE−HOWTO

Georgatos Photis, < gef@ceid.upatras.gr>

v0.5, 15 January 1998

One of the most bothering remarks on software, I have ever heard, is weather this or that thing can make coffee. So, Linux DOES make coffee. And it tastes good, instead!

For a long time, humanity was wondering how could a computer make coffee... People need coffee to get awake and stay asleep in front of the computer for a long time. Everyone knows that coding is better at night...

The main problem is how to control the coffee machine with the computer, so that it will be controlled by software. This generally means an ON/OFF switch implemented as a circuit which controls the coffee−machine's power supply.

1.Menu

2.Electronic circuit

3.Software

4.Overdose symptoms

5.Expansions

6.References

7.etc

1.Menu

1.1 French

Popular coffe among programmers because doesn't need a lot of care, like all commercial software. Its exciting taste has inspired thousands of programmers in writing incredible software, written in the very first ours of a day. Windows for example are written at 5:00 o'clock in the morning, favouring to this coffee! Result is guaranteed.

1.2 Nescaffe

Nescaffe is a rather strong coffee, made by pouring hot water in a mixture of coffee, sugar and some water. You usually take 1 spoon of coffee and 1 spoon of sugar with just a bit of water, to mix it. In the meantime you should have the water boiling. As soon as the water is hot enough, you mix them all together and preferably add milk. Although you can use something simpler than a coffee−machine to boil the water, I have seen this scheme a lot of times...

1.3 frappe

Popular variation of the above mentioned coffee. Actually, it doesn't need any coffee−machine, rather a refridgerator to get cold water and ice−cubes.

1.4 freddo

This is a difficult one, read coffee−faq (see references)

1.5 cappuccino (To be added)

1.6 espresso (To be added)

2.Electronic circuit

A general diagram is like this:

```
--------- 0-5V  --------- ~220V  ----------------
|  PC    |===>===|Circuit|========|Coffee-Machine|
---------       ---------        ----------------
```

The concept is that we take a controling voltage from the computer, which drives an electrically isolated circuit with Relay or Triac.

You must choose a Relay circuit, if you have a big coffee−machine (greater than 200W or so), otherwise you can use a triac−based one.

All circuits presented are at least once tested, but it's YOURS RESPONSIBILITY the results. If you have no experience with electronics you should NOT try with these, otherwise you may get a bad one...

You should be very careful while experimenting with 220V, and there is no obsolence in using an appropriate fuse.

2.1 Driving voltage 0−5V from the computer

Here is a simple example to get a voltage 0−5V from the parallel port of the computer.

```
        Back View           -----     Pin 10 - ACK
        Male DB-25          |   |     Pin  9 - D7
        Connector           |   |                             Pin 2 - D0
                            v   v                             v   Pin 1 -
  ~Strobe
         ____________________________________________________________
        /                                                            \
        \     13  12  11  10   9   8   7   6   5   4   3   2   1     /
         \                                                          /
          \     25  24  23  22  21  20  19  18  17  16  15  14     /
           \______________________________________________________/
```

Pin 1 is Strobe (inverse logic)

Pins 2−9 is DATA BUS's signals, exactly what was written to the parallel port's latches with an OUTB command.

Pin 10 is the acknowledge signal (ACK), controlled by you, so that you can produce an interrupt to the CPU.

Pins 18−25 are short−circuited and this is the ground (GND).

In detail:

```
  <= in   DB25    Cent    Name of         Reg
  => out  pin     pin     Signal          Bit     Function Notes
  ------  ----    ----    --------        ---     -----------------------------
  =>        1       1     -Strobe         C0-     Set Low pulse >0.5 us to send
  =>        2       2     Data 0          D0      Set to least significant data
  =>        3       3     Data 1          D1      ...
  =>        4       4     Data 2          D2      ...
  =>        5       5     Data 3          D3      ...
  =>        6       6     Data 4          D4      ...
  =>        7       7     Data 5          D5      ...
  =>        8       8     Data 6          D6      ...
  =>        9       9     Data 7          D7      Set to most significant data
  <=       10      10     -Ack            S6+ IRQ Low Pulse ~ 5 uS, after accept
  <=       11      11     +Busy           S7-     High for Busy/Offline/Error
  <=       12      12     +PaperEnd       S5+     High for out of paper
  <=       13      13     +SelectIn       S4+     High for printer selected
  =>       14      14     -AutoFd         C1-     Set Low to autofeed one line
  <=       15      32     -Error          S3+     Low for Error/Offline/PaperEnd
  =>       16      31     -Init           C2+     Set Low pulse > 50uS to init
  =>       17      36     -Select         C3-     Set Low to select printer
  ==       18-25   19-30, Ground
```

2.2 Controlling with a Relay

The simplest circuit that somebody can build is:

```
                            Vcc
                             |
                             +------+
                             |    __|__
                           Relay   /^\  Diode 1N4002
                            Coil  /---\
                             |      |
                             +------+
                             |
                          | /
                 4.7K    B |/  C
parallel port >-\/\/\/\/---|        NPN Transistor: BC547A or 2N2222A
data pi                    |\  E
                          | V
                             |
parallel port >--------------+
ground pin                   |
                          Ground
```

Connect Vcc with the same voltage as the relay type (usually 5 or 12V). Obviously, the relay's specifications should be reasonable for your coffee−machine.

Barmen, usually, tend to put the relay AFTER the transistor, at the emitter (E) pin instead of the collector (C) pin. This is a bad practice because it biases the transistor badly, and may result in bad coffee :−). Diode 1N4002 is useful to protect the transistor from the relay's currents. If you don't use it the transistor will become darker and smelling...

2.3 Controlling with TRIAC #1

If you only want a simple circuit, you can use Motorola's triac driver MOC301[012], together with a general purpose TRIAC like SC141D. This method has the advantage that you don't need an extra power supply.

For non−inductive loads this is the circuitry:

```
       270     1 +-------+ 6    180
+5v -VAVAVA-----+        +----VAVAVA-----+-------------- Line Hot
               2 |  MOC  |               |
TTL in ---------+ 3012  +nc            VA  SC141D
                 |       | 4           / |
               nc+       +------------/  |
                 +-------+               +----\/\/\/---- Line Neutral
                                              LOAD
```

If you are going to work with 220V, prefer a 3021. Inductive loads should be used in conjuction with bypass capacitors, see *Motorola Application Note AN−780*. Coffee−machines are mainly resistive loads and not inductive (like a motor), but who knows what's yours...

2.4 Controlling with TRIAC #2

```
+5VDC
|    180                      180            2.2k
+---/\/\/\----+-----+   +----/\/\/-+--/\/\/\---+-------> 120V
              |    1|   |6         |           |         Hot
              |    +=====+         |           | MT1
              |    | MC  | TRIAC   |          +-+
              |    | 3032| Driver  |        G | | TRIAC
              |    +=====+         |         /| |
              \    2|   |4         |        / +-+
       2N3904  |----+   |          |        |  | MT2
              /     |   +--------- | -------+  |
             V      \              |        |  |
             |      /              |        \  |
             |      \ 43    .01u  ---   10k /  |
             |      /       500V  ---       \  |
             |      |              |        /  |
             +------+              |        |  |            Neutral
             |                     +--------+--+---o    o--> 120V
             /                                      load
 >-/\/\--|  2N3904
             \
              V
              |
             ---
            ///
You should change resistors accordingly for 220V.
```

Circuit description:

The MC3032 is an optoisolator TRIAC driver. The 180−ohm resistor sets the current for the LED emitter in the optoisolator. Change the value of this resistor − if necessary − to get reasonable current (e.g., 15 mA).

Note that you cannot test this circuit without a load. The TRIAC will not switch unless connected to an AC voltage source, so you can't test it for simple switching w/o applying AC and a load. Note the 500V rating on the .01 cap.

3.Software

3.1 Software

You will have to build an executable that will perform like this:

- Get permissions to use I/O address space, by calling kernel, with the command **ioperm**: eg *ioperm(BASE, range ,1);*
- Perform an out request instruction, to set the 0−5V voltage to the parallel port, eg *outb(1, BASE);*
- Wait for enough time so that coffee gets ready. It would be nice if that time is read by looking at the command line.
- Then it will turn off the coffee−machine: outb(0 , BASE);
- Before ending it should give back the paraller port with a *ioperm(BASE, range, 0);*

Change BASE = 0x3bc for /dev/lp0, 0x378 for /dev/lp1, and 0x278 for /dev/lp2, range=8.

It would be nice if you had that program setuid, so that everybody can drink coffee!

3.2 Device driver

Just read kernel hacker's guide, implement a device driver (it could even be user space i think). Please, compile it as a module, so that we won't need a kernel compile in every update. Then write:

```
echo cappuccino >/dev/coffee
```

And you will have a hot cup of coffee in 1 minute. Remember to give the right permission to /dev/coffee, depending on whether you want only root making coffee or not.

The advantage of this method is that it supports feedback from the coffee−machine by using the ACK of parallel port, so that smart coffee−machines could produce an interrupt.

Do it as homework.

3.3 Connecting with the Internet

If you have implemented the C programme (see above), you just have to write down a simple CGI script to turn ON and OFF the coffee−machine. You should write some nice webpages, explaining how to make coffee, and put them at an **apache** web server...

4.Overdose symptoms

- excitement
- nervousness
- insomnia
- tachycardia or cardiac arhythmia
- gastrointestinal disturbance
- restlessness

5.Expansions

These are our ideas:

- All hardware and software described here, can be expanded so that it will support toast, beaf, applepies, etc.
- Cluster with 8 coffee−machines. This will let you have coffee even when the one gets off. Of course there will be a perfomance hit.
- Parallel vector coffee−machine will be a future release.
- If you want the maximum automation you'll need more circuits and censors, so that you can control water flow, temperature, coffee quantity etc.
- In the near future we will implement SNMP features.
- Serial coffee−machine at 115Kbps.

6.References

- http://daisy.uwaterloo.ca/~alopez−o/caffaq.html This is Internet's **Coffee−FAQ**
- http://lonestar.texas.net/~andrew/f_pc_.htm A lot of circuits in ASCII. Some of them are for parallel port.
- http://shell.rmi.net/~hisys/parport.html Whatever you wanted to learn about a parallel port and didn't dare to ask.
- http://sunsite.unc.edu/LDP/HOWTO/mini/IO−Port−Programming Programming of I/O ports under popular operating system Linux.
- http://www.redhat.com:8080/HyperNews/get/khg.html How to write your own device drivers. Come on, do it!
- http://www.hut.fi/Misc/Electronics/circuits/parallel_output.html Tomi Engdahl's web page is a *must see* for everyone who likes electronics.
- http://www.yahoo.com/Computers_and_Internet/Internet/Entertainment/Interesting_Devices_Connected_to_the_Net/Coffee_Machines/ Coffee−machines on−line. Unfortunatelly, there are no tests.

- http://www.cs.su.oz.au/~bob/coffee.html This coffee−machine offers only cappuccino. It should be upgraded!
- http://einstein.et.tudelft.nl/~janssen/ Hot coffee from Netherlands.
- http://circe.chinalake.navy.mil/cgi−bin/spion/snapit.cgi
- http://www.cl.cam.ac.uk/coffee/coffee.html
- http://www.menet.umn.edu/coffeecam/

7.etc

7.1 preface

This document was initially written in hellenic (greek), originating from a small debate in the linux−greek−users list , whether linux can make coffee or not. It has been an article in our beautiful online magazine on Linux called "magaz": http://www.linux.gr/magaz. Remember, that magaz is greek.

7.2 Authorship and maintenance

My name is Georgatos Photis, still−yet−foralongmaybe student at Computer Engineering and Informatics department in University of Patras. Usually, I am pretty busy with the greek documentation and webpages maintainance, but I can accept submissions to this HOWTO, as long as you don't be anxious on the changes.

7.3 Copyrights

The casual copyright with everything you get with linux... To find it, you'll have to read all of them and count the most common. Otherwise, no, you can't copy it.

7.4 Credits

- Christofer Colombus. His real name is Cristobal Colon. He was the 1st one who brought coffee from America, which was for first discovered (among Europeans) by him in 1492.
- Kostas Lialiambis is the one who dared saying that he can't make coffee with his Linux box!
- Panagiotis Vrionis, Yannakopoulos Haralambos, for giving me interesting and humoristing notes.
