커피 HOWTO(COFFEE-HOWTO)

Fotis Georgatos

gef (at) hellug.gr

Annie Pinder

ajp (at) leguin.org.uk

도운이

전혜진

linuxsingle (at) hanmail.net

v0.8, 2000-11-30

오랫동안, 컴퓨터로 커피를 끓일 수 있다는 사실은 놀라움이었다. 사람들이 장시간동안 컴퓨터 앞에서 깨어서 작업을 하기 위해 커피는 필요했다. 다들 알다시피, 코딩이라는 것은 밤에 더 잘 되는 법이니까. (역자 주: 녹차와 박카스도 만만치 않다고 개인적으로 생각한다.... ^^)

내가 들었던 소프트웨어에 대한 이야기 중 가장 기억에 남는 것은 바로 그것으로 커피를 끓일 수 있는 것이었다! 커피는 석유 다음가는 세계의 필수품이다. (My GOD....) 우리는 리눅스로 커피를 끓일 수 있으며, 특히 그 맛은 일품이다!!

가장 중요한 문제는 역시 어떻게 컴퓨터로 커피머신을 조작하는지에 관한 것이었고, 이 점은 소프트웨어를 사용한 제어를 하기로 한다. 간단히 말하자면, 커피 머신의 전원을 제어하는 회로를 사용하여 켜고 끄는 것을 하겠다는 것이다.

이 HOWTO는 프로젝트화 되어 진행중인데, http://coffee.sourceforge.net/ 자세한 내용은 이 곳을 참고하자.

1. 커피 메뉴판

1.1. 프렌치 커피(French)

별 신경 쓰지 않아도 쉽게 마실 수 있기 때문에 프로그래머들이 널리 마시는 커피이다. 이 맛은 수많은 프로그래머들에게 영감을 주어 놀라운 소프트웨어를 만들어내는데 공헌하였으며 하루를 일찍 시작하도록 해 주었다. 예컨대 Windows 조차 아침 5시에 커피의 도움으로 작성되었던 것이다! 결과는 보장한다.

1.2. 네스카페(Nescafe)

네스카페는 좀 진한 커피로, 약간의 물과 설탕으로 혼합된 커피에 끓는 물을 부어 만든다. 대개는 커피 1스푼에 설탕 1스푼을 넣고 물을 조금 넣어 섞는다. 물을 느긋하게 시간을 두어 끓이다가 충분히 뜨거워 졌을 때 이 모든 것을 섞고 경우에 따라 우유를 조금 붓는다. 커피머신보다 간단한 방법은 있지만, 아마 시간이 더 걸릴 것이다.

1.3. 프라페(Frappe)

가장 널리 알려진, 변형된 커피라 볼 수 있는데, 이것을 만드는 데는 커피 머신은 필요가 없다. 냉장고 속의 찬물과 얼음 조각들이 필요할 뿐이다.

1.4. 프레도(Freddo)

이 녀석은 좀 어려운 건데, coffee-faq (참고 문헌에 나와 있다.)를 참고하도록. ^^

1.5. 에스프레소(Espresso)

에스프레소는 매우 진한, 이탈리아 식의 커피이다. 이 커피는 작은 잔에 각설탕과 함께 나온다. 왜 그러는지 궁금하다면 뒤에 나오는 "남용시의 부작용"을 참고하도록. 맛있는 에스프레소를 만들기 위해서는 신선한 커피 원두와 물, 각설탕과 특별한 기구가 필요하다. 이 기구는 물을 끓여, 높은 온도의 수증기로 갈아진 커피 원두를 통과시킨다. 자동으로 이 과정을 해 주는 기계는 꽤 비싸지만, 저가형의 기계 역시 사용할 만 하다.

자, 기계에 물을 채우고는 끓을 때 까지 덥힌다. 기계의 필터핸들에 커피 가루 1스푼을 넣고는 아래로 내려보낸다. 이것만으로는 충분치 않다. 적당한 온도가 되면 필터핸들을 기계에 끼우고 작동시킨다. 30초 정도 지나면 섬세하고 뜨거운 에스프레소가 나오기 시작한다. 이것은 좋은 식사를 마친 후 마시면 좋다. 기분이 좋아지고, 몇 시간은 더 코딩을 할 수 있다.

1.6. 카푸치노(Cappuccino)

에스프레소에 대한 부분을 참고하고..... 만약에 더 빼어난 커피머신을 갖고 있다면, 이것을 사용하여 몽글몽글한 우유 거품을 얹을 수 있다. 커피에 얹을 크림과 같은 것이 필요한데, 이것들은 마련하기 간단하다. 커피머신을 열고 우유의 크림같은 거품을 에스프레소 위에 얹어 넣는다. 초콜릿 가루 등을 위에 얹어 장식하면 된다.

2. 회로 만들기

일반적인 다이어그램은 다음과 같다:

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

발상의 초점은, 중계기 혹은 3극관 교류 스위치를 사용하여 컴퓨터로 전압량을 조절하는 데 있다.

커피 머신이 200W 이상을 사용하고 있다면 중계기 회로를 사용해야 할 것이다. 커피 머신이 높은 출력을 필요로 하지 않는다면 교류 스위치를 사용해도 상관 없겠지만.

여기서 설명하는 모든 회로는 테스트를 거친 것이지만, 실천해서 생기는 문제의 책임은 실천한 사람에게 있는 것이다. 전기적인 지식이 없다면 비극적인 결과가 초래될 수 있으므로 실행에 옮기지 않기를 바란다.

220V를 사용하는 실험은 위험할 수 있으므로 조심해야 한다. 적당한 퓨즈를 사용하는 것을 추천하고 싶다.

2.1. 컴퓨터에서 0-5V의 전압 조절하기

병렬 포트를 이용하여 컴퓨터에서 0-5V의 전압을 조잘하는 간단한 예제이다.

```
      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     /
         \______________________________________________________/
```

1번 핀은 스트로보이다. (반대 논리이다.)

2-9번 핀은 데이터 버스의 신호로, 엄밀하게 말하면 OUTB 명령과 함께 병렬 포트의 빗장으로 쓰이는 셈이다.

10번 핀은 알림 신호(ACK)로, 제어하는 것이다. 그래서 우리는 CPU로의 인터럽트를 만들 수 있다.

18-25번 핀은 바탕(GND)이 되는 짧은 회로이다.

세부 내용 :

```
<= in   DB25    Cent    Name of         Reg
=> out  pin     pin     Signal          Bit     Function Notes
------  ----    ----    --------        ---     --------------------------
---
=>        1       1     -Strobe         C0-     보내지는 >0.5 uS 의 낮은 펄
스 셋팅
=>        2       2     Data 0          D0      의미있는 최소한의 데이터 셋팅
=>        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      의미있는 최대한의 데이터 셋팅
<=       10      10     -Ack            S6+ IRQ 용인된 이후의 낮은 펄스 ~ 5 uS
<=       11      11     +Busy           S7-     Busy/Offline/Error를 위한 높
은 펄스
<=       12      12     +PaperEnd       S5+     논지 외의 것을 위한 높은 펄스
<=       13      13     +SelectIn       S4+     프린터를 위한 높은 펄스
=>       14      14     -AutoFd         C1-     한 가닥에서의 자동반응을 위
한 낮은 펄스 셋팅
<=       15      32     -Error          S3+     Error/Offline/PaperEnd를 위
한 낮은 펄스
=>       16      31     -Init           C2+     init에 대한 > 50uS 의 낮은 펄
스
=>       17      36     -Select         C3-     지정된 프린터를 위한 낮은 펄
스 셋팅
==       18-25   19-30, Ground
```

2.2. 중계기로 제어하기

가장 간단한 회로는 다음과 같이 구성되어 있다.:

```
                            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
```

중계기 타입과 같은 전압으로 Vcc 를 연결한다. (일반적으로 5 나 12V를 사용한다) 중계기의 명세는 커피 머신과 동일하게 하는 것을 권한다.

트랜지스터의, 집전기(collector) (C) 핀 대신 이미터 (E) 핀 다음에 중계기를 놓는다. 이것은 트랜지스터가 나빠지는 경향이 있고 아마도 그 결과로 커피도 별 볼일이 없을 수 있으므로 별로 좋은 방법은 아니다. :-) 다이오드 1N4002는 중계기와의 연결에서 트랜지스터를 보호할 수 있으므로 더 유용하다. 트랜지스터를 사용하지 않는다면, 아마 시커멓게 되며 탄 내가.....

2.3. TRIAC의 제어 #1

간단한 회로를 원할 뿐이라면, 일반적인 기능의 SC141D와 비슷한 TRIAC인 모토로라에서 나온 트리악 드라이버 MOC301[012]를 사용할 수 있다. 이것은 당신이 여분의 파워 서플라이를 사용하지 않는다면 모험적인 일이다.

반응 로드가 없는 회로:

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

220 볼트에서 사용할 생각이라면, 3021을 입수하도록 하자. 반응 로드는 **Motorola Application Note AN-780**를 보면 알 수 있듯이 콘덴서 바이패스와 연결되는 것이 보통이다. 커피머신은 저항 로드를 사용하지 모터와 같은 반응을 사용하지 않지만, 당신의 것이 그렇게 할지 또 누가 알겠는가.

2.4. TRIAC의 제어 #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
             |
            ---
           ///
220V를 위해 레지스터를 변환할 수 있다.
```

회로 설명 :

MC3032은 TRIAC의 광음향절연체이다. 180 옴의 저항기를 광음향절연체의 LED 이미터에 일반적으로 맞추어 조절한다. 필요하다면 저항기의 값을 바꾼다. (e.g., 15 mA).

로드 없이 실험할 수는 없다는 것을 잊지 말자. TRIAC은 AC 전압과 연결되지 않으면 작동되지 않으며, AC 와 로드 없이는 단순한 실험도 할 수 없다. 500 볼트에 최고 .01의 세액이 붙는다는 것도.

3. 소프트웨어

3.1. 소프트웨어

이것이 작동하도록 하기 위해 다음의 실행을 가능하게 해야 한다.:

- 커널을 호출하여 **ioperm** 명령을 통해 I/O 주소 공간의 퍼미션을 얻는다. : eg **ioperm(BASE, range ,1);**
- 지시대로 일을 끝냈다면, 병렬 포트를 0-5V 로 맞춘다. eg **outb(1, BASE);**
- 커피가 끓을 때 까지 충분한 시간을 기다린다. 명령행을 바라보는 동안이라면 가장 좋을 것이다.
- 그리고 나면 커피머신은 꺼져야 한다. : outb(0 , BASE);
- 모든 것이 끝나기 전에 병렬 포트에 **ioperm(BASE, range, 0);** 명령으로 사용 포기 신호를 보낸다.

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

setuid 프로그램으로 만들면 모든 사람들이 커피를 마실 수 있게 되므로 더 유용하다!

3.2. 디바이스 드라이버

커널 해커 가이드 (http://www.redhat.com:8080/HyperNews/get/khg.html)를 참고하여 디바이스 드라이버에 손대자. (내 생각으로는 사용자 공간이 필요한 것이다.) 이것을 모듈로 만들면 커널을 새로 업할 때 마다 매번 컴파일할 필요도 없어진다. 그리고 아래와 같이 입력하자.:

echo cappuccino >/dev/coffee

그리고 당신이 뜨거운 커피를 순식간에 만들어내기를 원한다면, /dev/coffee가 루트만이 손댈 수 있는지 아니면 여타의 사용자도 손댈 수 있는지에 대해 정확히 권한을 주어야 한다.

이 방식의 뛰어난 점은 병렬 포트의 ACK를 통해 커피 머신과 소통하는 것을 지원한다는 것이다. 우수한 커피 머신은 인터럽트를 만들어 낼 수도 있을 것이다.

숙제로 해 보시길!!!

3.3. 인터넷으로의 연결

만약 당신이 C 프로그램을 할 수 있다면, 당신은 간단한 CGI고 커피 머신을 켜고 끄는 것을 제어할 수도 있다. 쓸만한 웹페이지를 만들어서 어떻게 커피를 만드는지에 대해 떠들고는 아파치 웹 서버 상에 올려 버려도 좋을 것이다.

4. 남용시의 부작용

왜 이런 것이 있는지는 번역하면서도 이해가 안 가기는 하지만..... (커피를 마시라는 건지 말라는 건지 모르겠는 말이군요. ^^;;;;

- 흥분
- 신경과민
- 불면증
- 심박급속증 혹은 심장에의 영향
- 불안감
- 빛에 대한 민감증
- 환청으로 인한 두통
- 위장장애

5. 잔소리

아래의 내용은 우리가 내놓은 의견들이다. :

- 여기에 논의된 모든 하드웨어와 소프트웨어를 이용하여 토스트, 고소고기, 애플파이 등등을 지원하도록 확장하는 것 - 이 점은 커피머신 하우투에 언급되어 있다. : 역자 주.
- 8개의 커피머신을 제어하기. 당신은 제일 먼저 끓는 커피로 마실 수 있다. 물론 이것은 어떤 것이 먼저 덥혀지는가의 문제일 것이다.
- 병렬 벡터 커피머신은 추후에 나올 예정이다.
- 더욱 자동화된 것을 원한다면 물이 담기는 것과 온도, 커피의 품질 등등을 조절하기 위한 회로와 센서들이 더 필요할 것이다.
- 가까운 미래에 우리는 SNMP의 특징을 충족시킬 수 있을 것이다.
- 115Kbps의 속도를 내는 시리얼 커피 머신.

6. 참고자료

- http://daisy.uwaterloo.ca/~alopez-o/caffaq.html : 이것은 인터넷 상에서**Coffee-FAQ**라는 이름으로 잘 알려져 있다.
- http://faculty.washington.edu/chudler/caff.html : 카페인과 신경계의 현상
- http://www.gardfoods.com/coffee/coffee.coffee.htm : 약간 광범위한 문서. 예를 들면 인간과 커피의 관계와 같은 내용의 문서이다. 100,000년 전에 이미 사람들은 커피 열매를 씹어 왔다. 커피는 525AD에 예맨 지방에서 자라났지만, 그 곳이 원산지는 아니었다.
- http://www.faqs.org/rfcs/rfc2324.html : RFC2324: 하이퍼텍스트 커피포트 제어 프로토콜 (HTCPCP/1.0)
- http://www.faqs.org/rfcs/rfc2325.html : RFC2325: SMIv2를 사용하는 커피 포트(원문 : 물방울을 똑똑 떨어뜨리며 액체를 데우는 기구) 관리의 정의
- http://sunsite.unc.edu/LDP/HOWTO/mini/IO-Port-Programming : 일반적으로 운영되는 리눅스 시스템에서의 I/O 포트 프로그래밍
- http://lonestar.texas.net/~andrew/f_pc_.htm : 아스키 문서로 된 회로에 관한 설명. 이것들 중 일부는 병렬 포트에 관한 것이다.
- http://shell.rmi.net/~hisys/parport.html : 질문하는 것을 귀찮아하지만 않는다면 병렬 포트에 관해 원하는 대로 배울 수 있는 곳.
- http://www.redhat.com:8080/HyperNews/get/khg.html : 자신만의 디바이스 드라이버 작성하기.

- http://www.hut.fi/Misc/Electronics/circuits/parallel_output.html 전자공학에 취미가 있는 사람이라면 꼭 가봐야 하는 Tomi Engdahl의 홈페이지
- http://dir.yahoo.com/Computers_and_Internet/Internet/Devices_Connected_to_the_Internet/Coffee_Machines/커피머신 온라인. 불행히도 검증된 것은 아니다.
- http://www.cs.su.oz.au/~bob/coffee.html카푸치노 전용의 커피머신. 개선되어야만 할 것이다.
- http://einstein.et.tudelft.nl/~janssen/ 네덜란드의 핫 커피.
- http://www.cl.cam.ac.uk/coffee/coffee.html
- http://www.cs.su.oz.au/~bob/Coffee/index.html
- http://www.menet.umn.edu/coffeecam/

7. etc

7.1. 서문

이 문서는 리눅스의 괴짜 유저들 사이의, 리눅스로 커피를 끓일 수 있느냐 없느냐에 대한 작은 논쟁을 문서화 한 것이다. 이 내용은 우리가 사랑하는 리눅스에 대한 온라인 잡지 magaz (http://www.linux.gr/magaz)에 실리기도 했다. magaz 는 당신과 마찬가지로 알아들을 수 없는 소리를 늘어놓는다는 사실을 기억하라.

7.2. 저자들의 한마디

내 이름은 Fotis Georgatos (http://students.ceid.upatras.gr/~gef)이고, 무슨 소리인지 알아들을 수 없는 문서들과 웹 페이지 관리로 바쁜 사람이다. 독자들이 더 이상의 수정이 필요 없다고 생각할 때 까지 이 HOWTO에 대한 제언을 환영할 것이다.

나는 애니 파인더. 커피 팬입니다. 영국에 살고 있죠. 이 문서를 제가 번역했습니다. 현재 영국에서 고등학교 마지막 학년을 보내고 있습니다.

전혜진입니다. ^^ 번역하는 저도 짧게 주저리~를 하겠습니다. ^^ 혹시 번역이 잘못된 내용이 있다면, 그것은 제가 커피를 상당히 싫어하기 때문에 이해 부족으로 일어난 일일 가능성이 아주 크겠습니다. 저는 커피를 마시면 맥박이 빨라져서, 커피를 좋아하지 않거든요. 전 주로 생수병에 녹차를 담가 먹거나 코코아를 마시곤 합니다만..... 커피 마시는 모습은 어른스러워 보여서 부럽다는 생각은 간혹 합니다.

7.3. Copyrights

격식을 갖추지 않은 채 저작권에 관해 말하자면 리눅스에 관한 모든 것과 동일하다. 이것에 관해 찾으려면 아무 HOWTO나 붙들고 읽어 보면 될 것이다. 한 가지 다른 점은, 무단 복제는 사양하겠다는 것이다.

7.4. 도와 주신 분들

- 에티오피아-커피의 기원으로 알려진 곳. 많은 사람들이 아메리카 발견시 콜롬버스가 커피를 가져온 것이 기원이라는 말을 정정해 주었다. 아메리카 발견 훨씬 이전에도 중동에서는 커피를 마시는 것이 일반적이었으니까. 그리고, 우리는 유럽까지 담배와 초콜릿을 보내 주신 Cristobal Colon (http://sunsite.unc.edu/expo/1492.exhibit/c-Columbus/columbus.html)님을 잊지 못할 것이다.
- 리눅스로 커피를 못 끓이겠다고 말씀해 주신 Kostas Lialiambis 님.
- 흥미있고 재미있는 내용을 보내 주신 Panagiotis Vrionis, Yannakopoulos Haralambos 님.
- 추가와 수정에 도움을 주신 수많은 네티즌 여러분.