우분투리눅스로 임베디드 개발하기

이 호 민

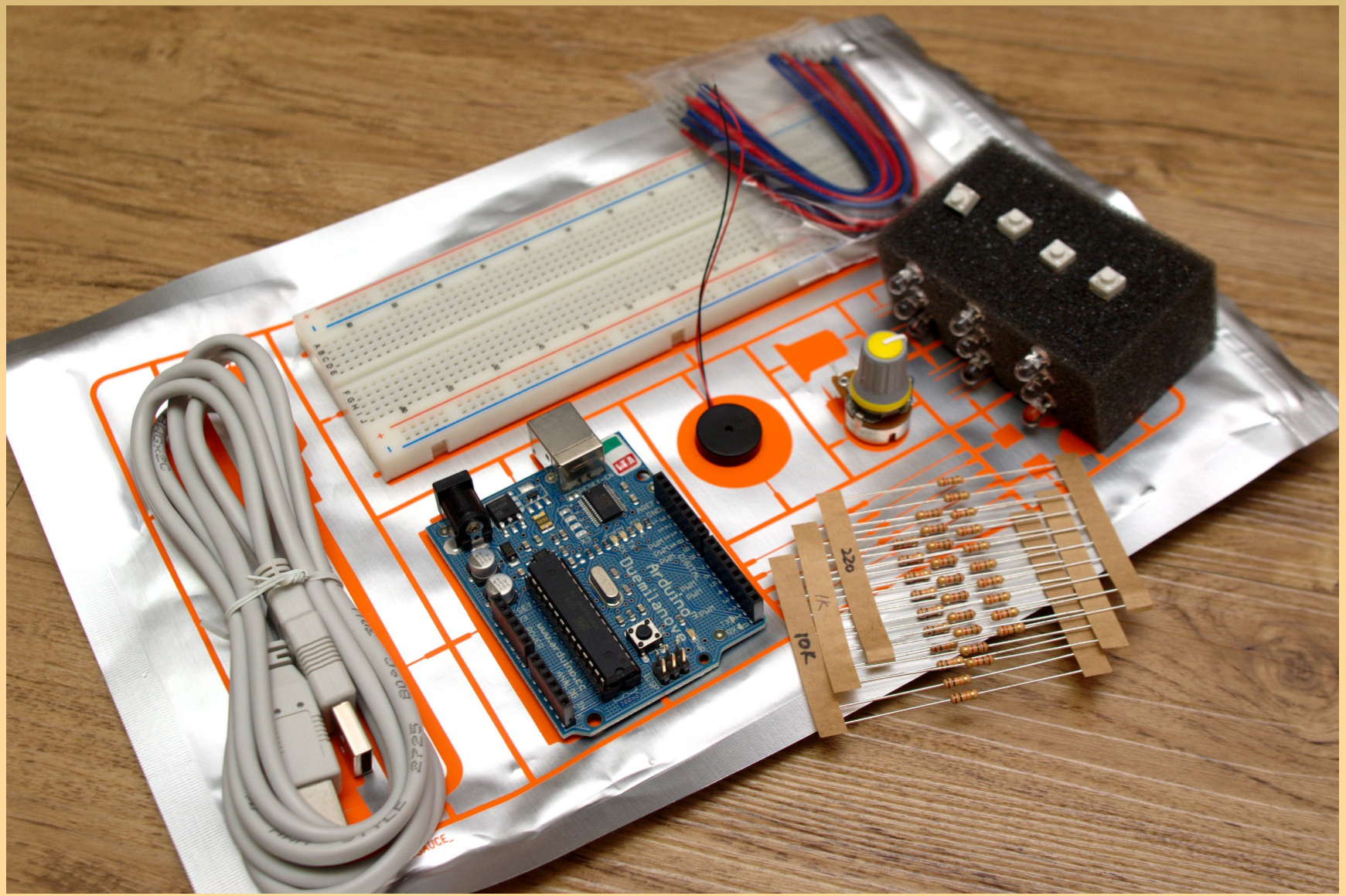
ubuntu

Host 로 우분투를 사용하는 이유

- 미려하고 편한 UI
- 진화를 멈추지 않은 cmdline – bash
- 안전한 OS
- 더 좋은 FileSystem
- 개발 도구 기본 제공
  - 컴파일러 , 툴체인 , 디버거 , etc...
- 효율적인 패키징 (deb) 시스템

ubuntu

아두이노 소개



ubuntu

아두이노 소개 ***2/2***

- AVR 프로세서를 사용한 마이컴
  - Open source HW
  - Self-Programing 을 위한 부트로더 내장
  - 프로토타입 개발에 편한 점퍼선 연결 GPIO
  - USB 케이블로 업로드 및 디버깅
- 아두이노 언어
  - C/CPP 에서 중복되는 부분을 빼서 간소화함
  - 스케치 - 아두이노 언어로 작성한 프로그램

아두이노 개발환경 설치

- $ sudo add-apt-repository ppa:arduino-ubuntu-team
  - 소스리스트에 아두이노 저장소 (ppa) 추가
- $ sudo apt-get update
  - 소스리스트 갱신
- $ sudo apt-get install arduino
  - 아두이노 패키지 설치

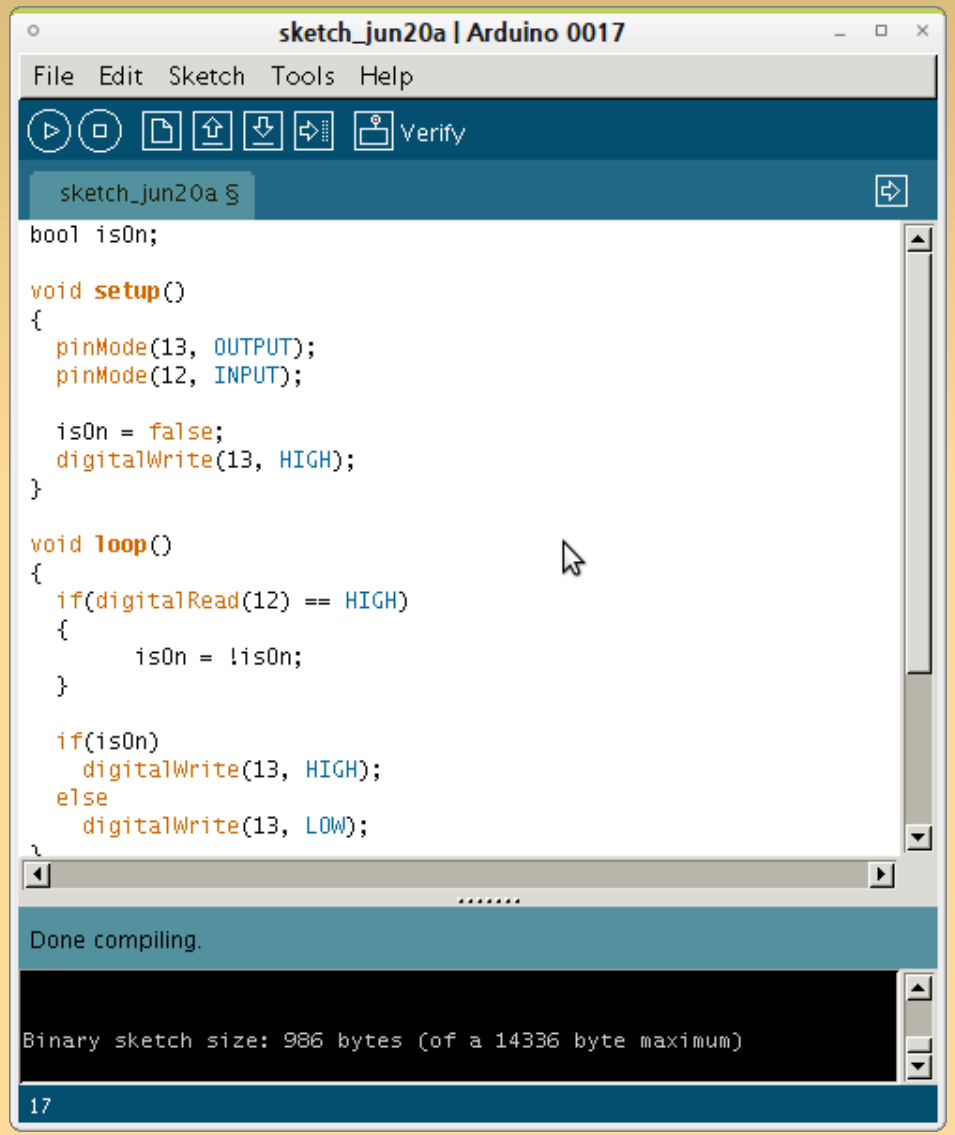
아두이노 패키지 구성

- 필요한 패지키들 (dependency) 이 같이 설치됨
  - avr-cross-toolchain
  - Java, etc...
- debian/control

```
Package: arduino
Architecture: all
Depends: ${misc:Depends}, gcc-avr, avr-libc,
   avrdude (>= 5.10-1ubuntu1),
   default-jre | java6-runtime, librxtx-java (>=2.1.7r2-4ubuntu1)
Description: The Arduino libraries and the development environment
   Arduino is an open-source electronics prototyping platform...
```

ubuntu

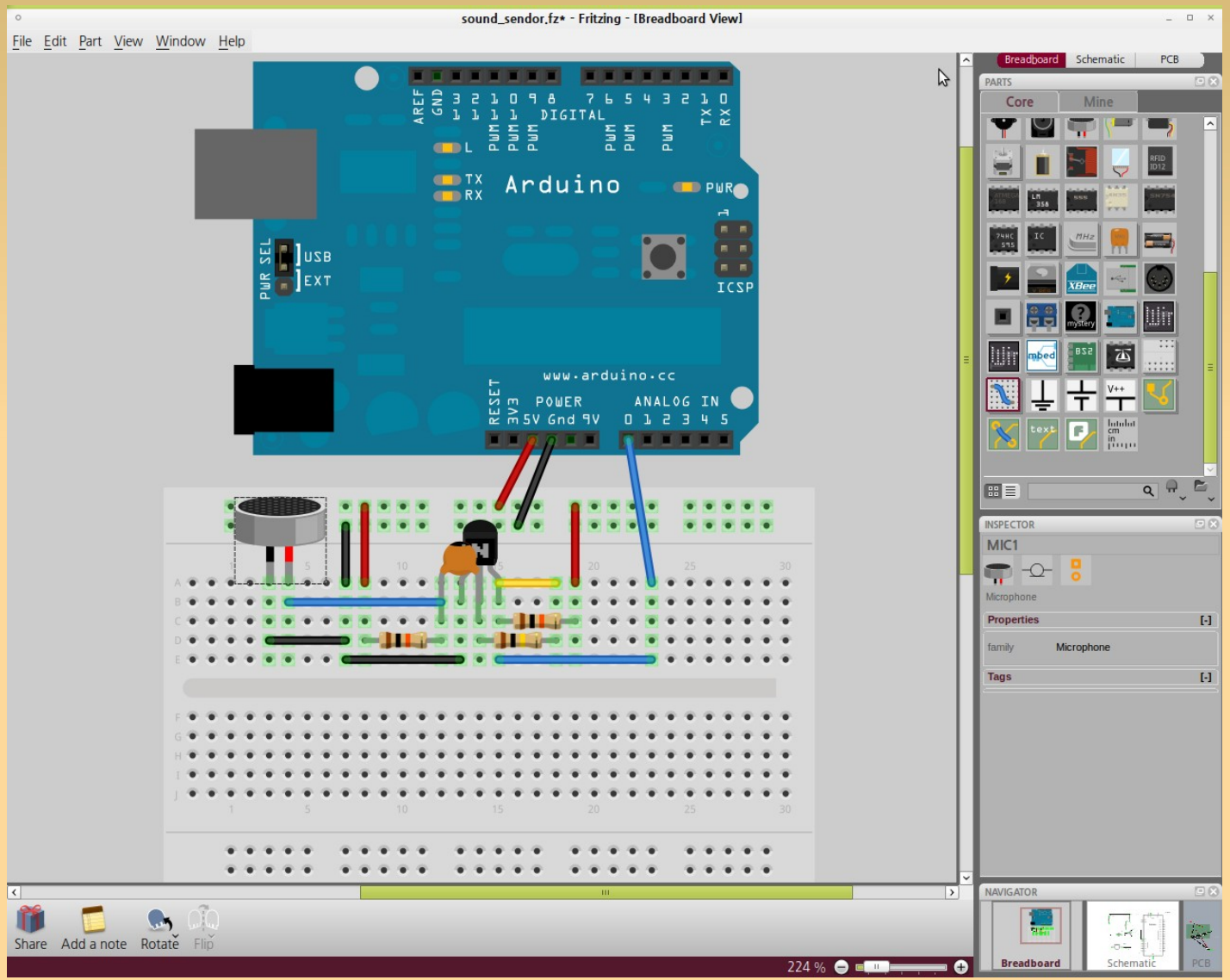
아두이노 IDE



- Verify 를 누르면 ..
  - 스케지에 살을 붙여 cpp 언어로 변경
  - avr-gcc 크로스컴파일
- Upload 를 누르면 ...
  - 이미지를 아두이노 보드에 올림

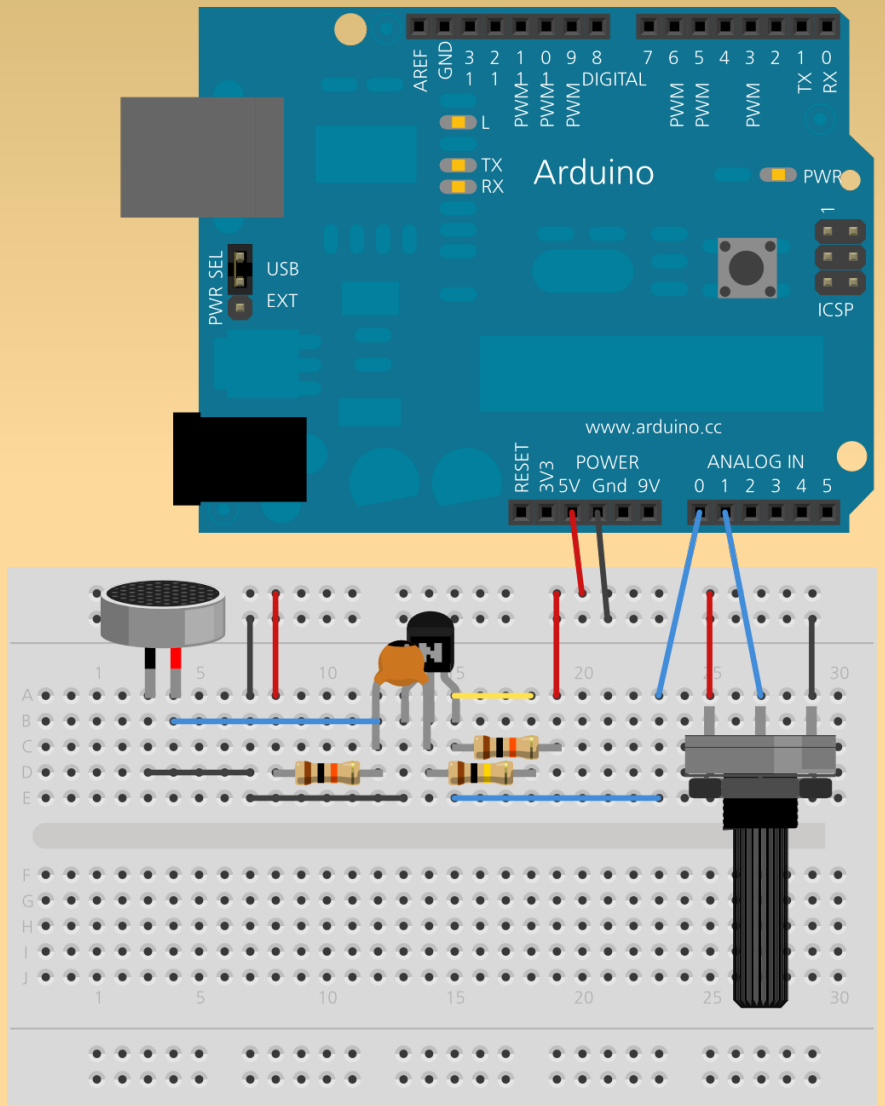
ubuntu

Frizing – 소개



ubuntu

Frizing – 브레드보드



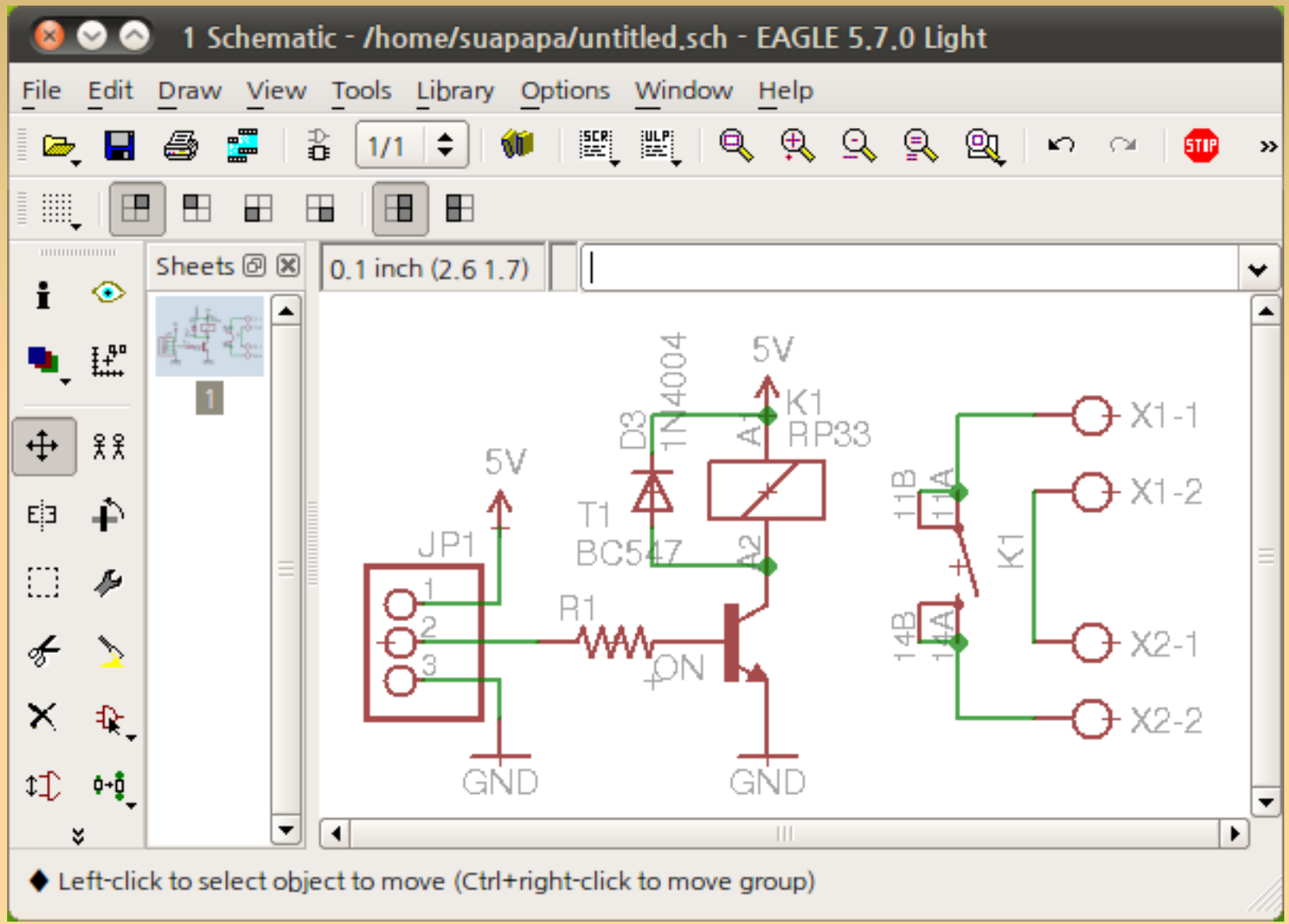
- http://fritzing.org
- 빠른 하드웨어 구성
- 적은 수정 비용
- 구멍들은 표준 간격
- 행 / 열의 구멍들은 서로 연결되어 있음
- 가운데 분리 영역 DIL 패키징의 칩 사용

ubuntu

EagleCad 1/3 - 소개

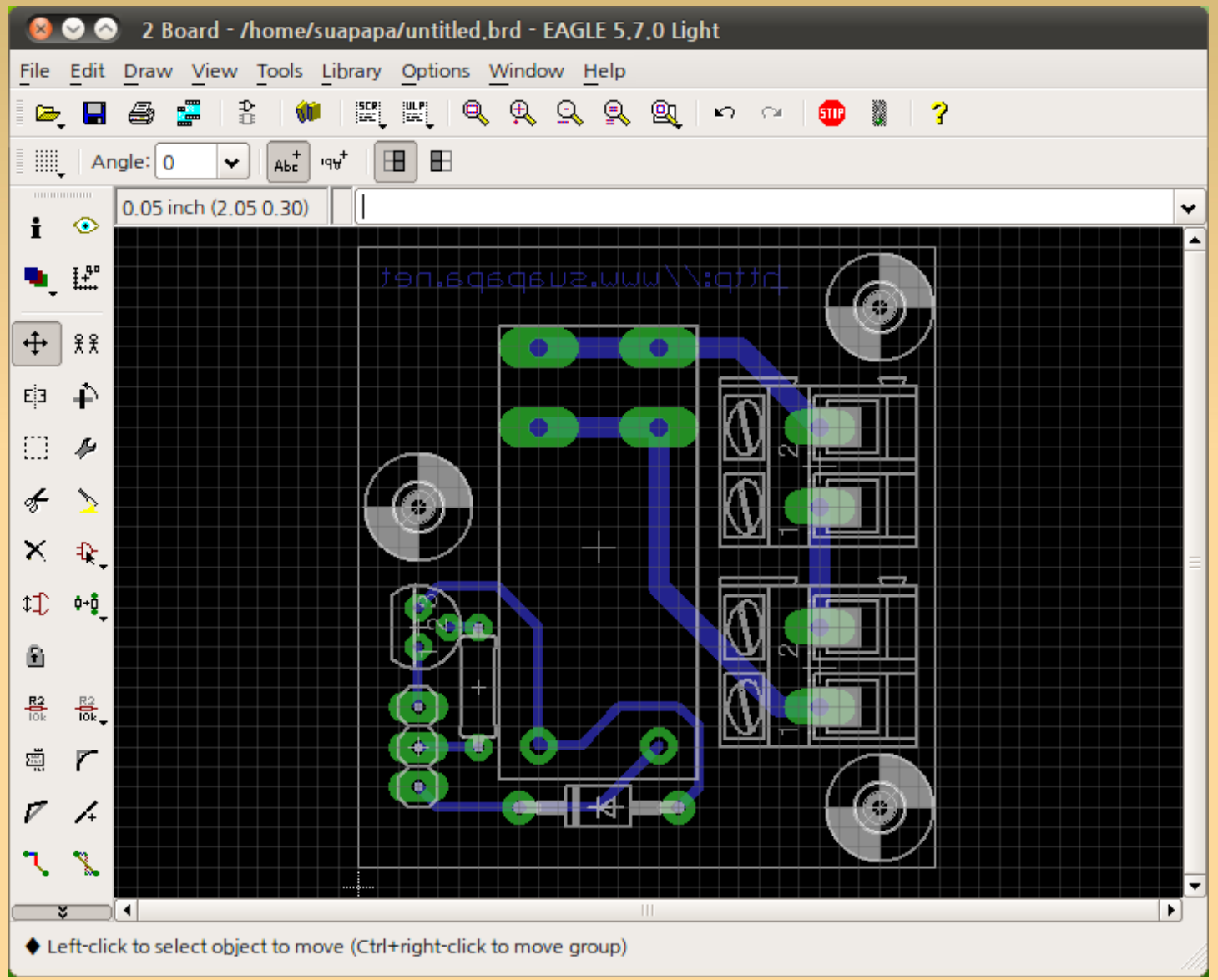
- http://www.cadsoftusa.com/
- 설치
  - $ sudo apt-get install eaglecad
- 회로설계 – sch
- 아트워크 – brd
  - gerber 파일로 출력 가능
- 라이브러리 – lbr

EagleCad 2/3 – 회로설계



ubuntu

EagleCad 3/3 – 아트워크



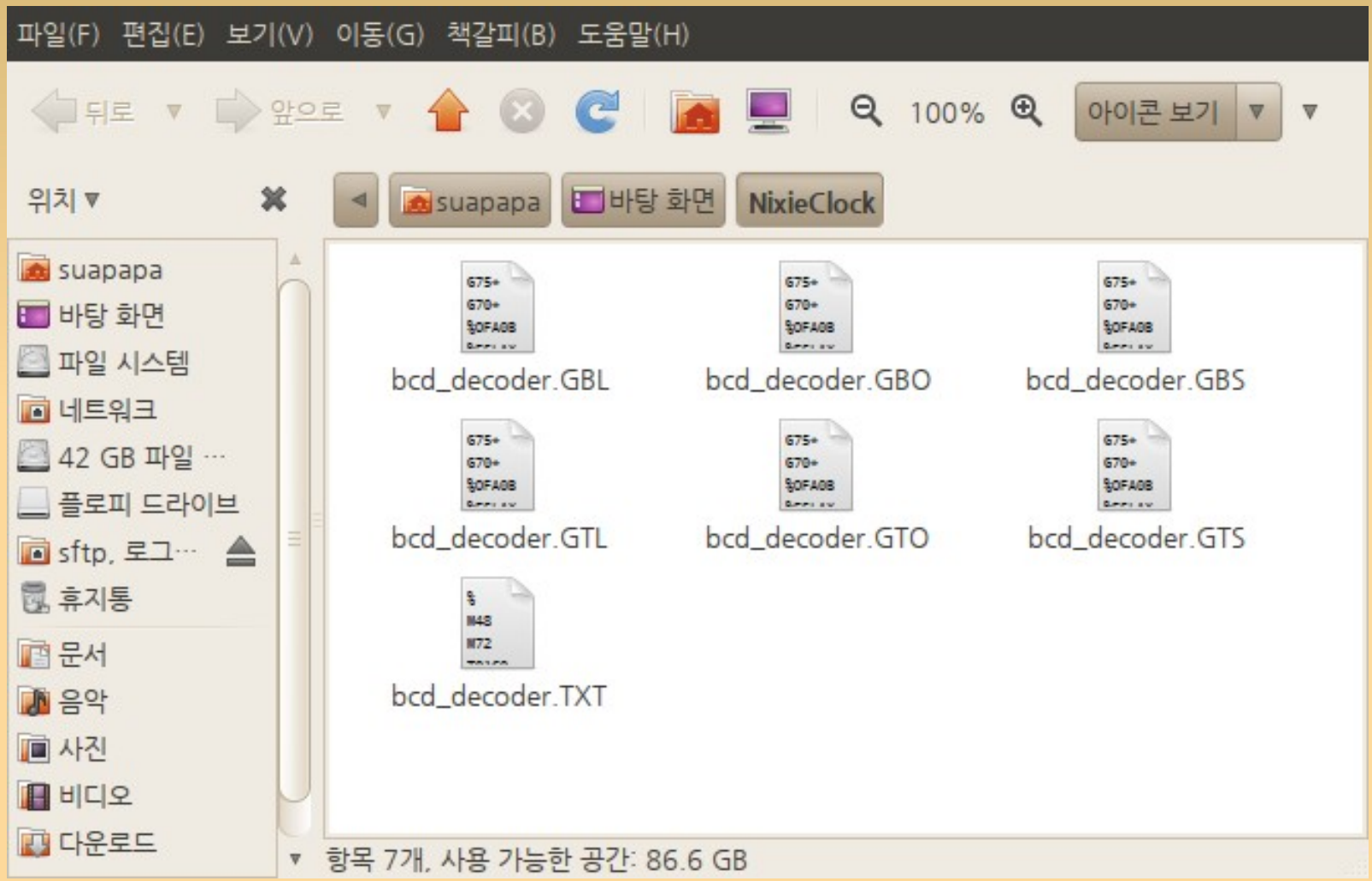
ubuntu

Gerbv 1/3 – Gerber Viewer

- Gerber 는 회로도의 PDF 파일
- 설치
  - $ sudo apt-get install gerbv
- 레이어 별로 파일 생성
  - GBL, GBO, GBS – bottom layer
  - GTL, GTO, GTS – top layer
  - TXT - drill
- 압축하여 PCB 제조 (fab) 업체에 주문

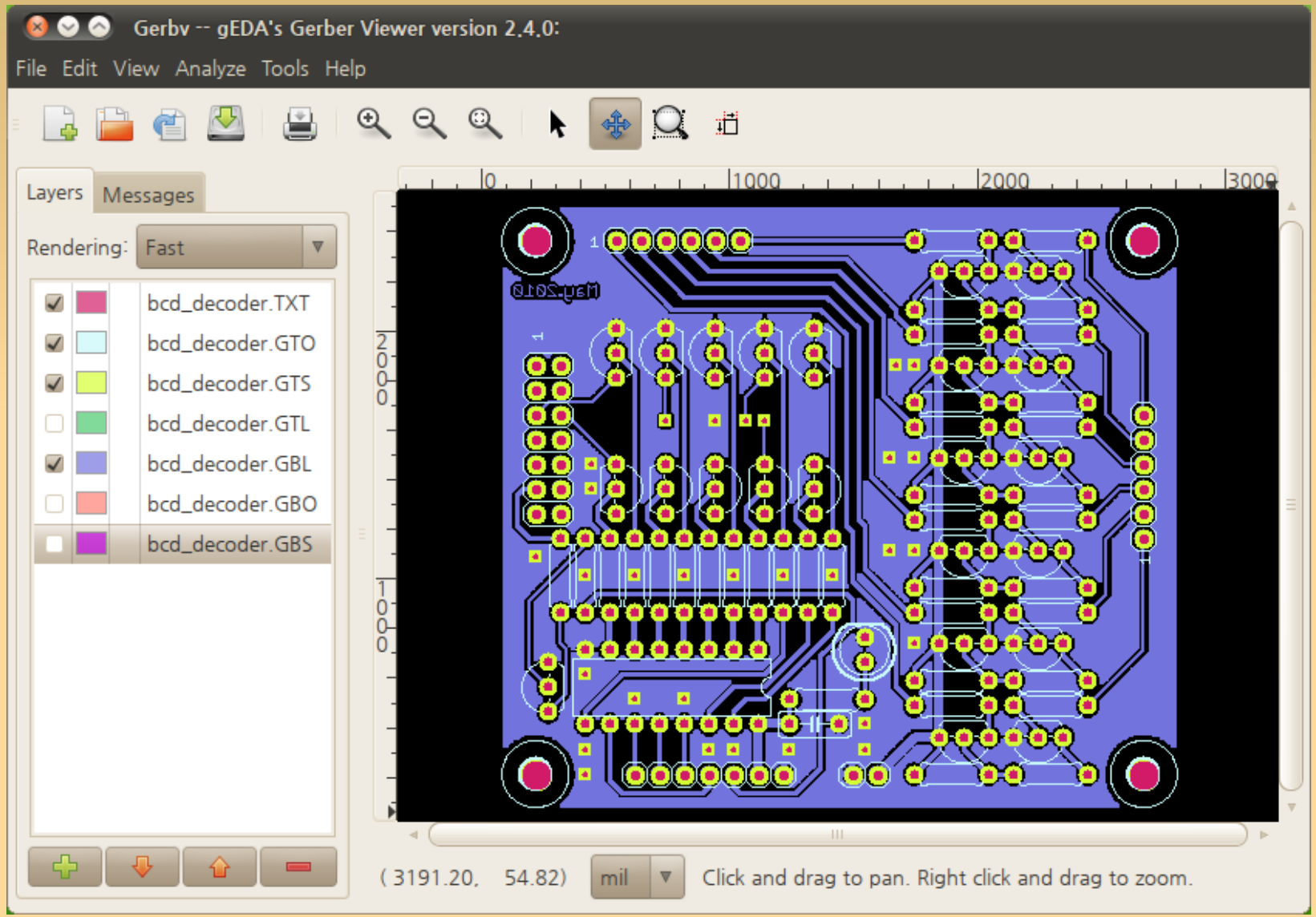
ubuntu

Gerbv 2/3



ubuntu

Gerbv 3/3



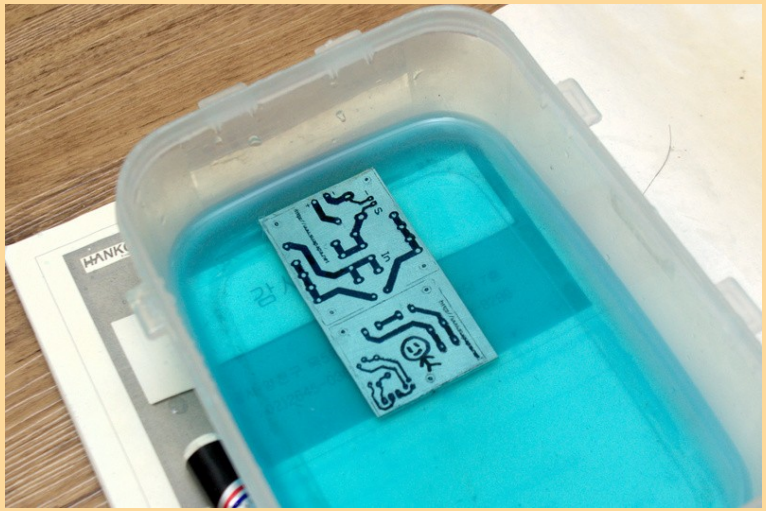
ubuntu

Home – fab 1/3

- Export layer
  - Bottom or Top Layer
  - PAD, VIA, Route, Dimention
- 색반전 후 출력
  - PressNPeel 필름
- 다리미로 동판에 패턴 인쇄

ubuntu

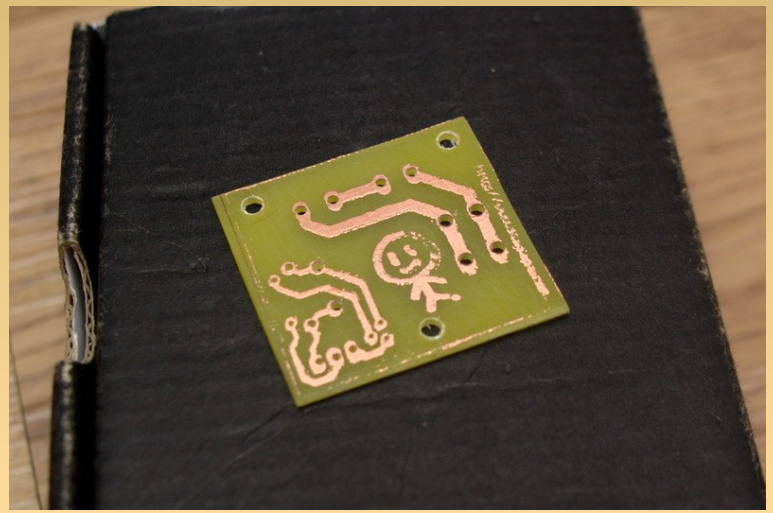
Home – fab 2/3

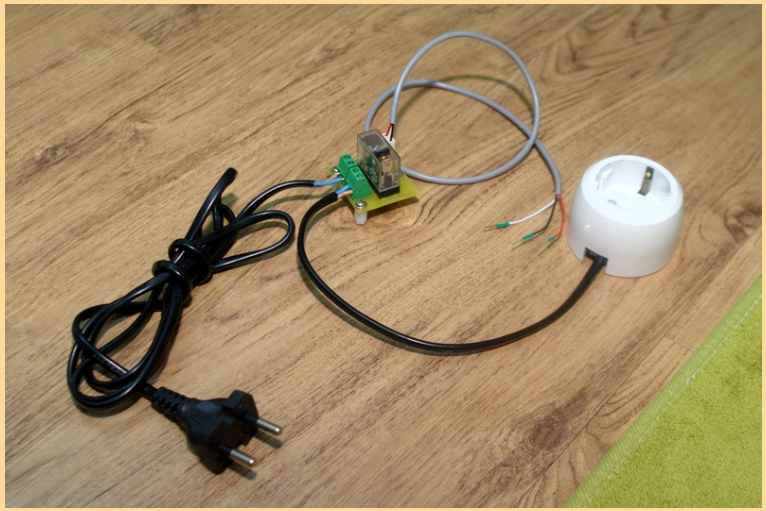


- 필름을 벗겨낸 후
- 유성팬으로 패턴 수정
- 에칭액을 사용해 에칭

ubuntu

Home – fab 3/3



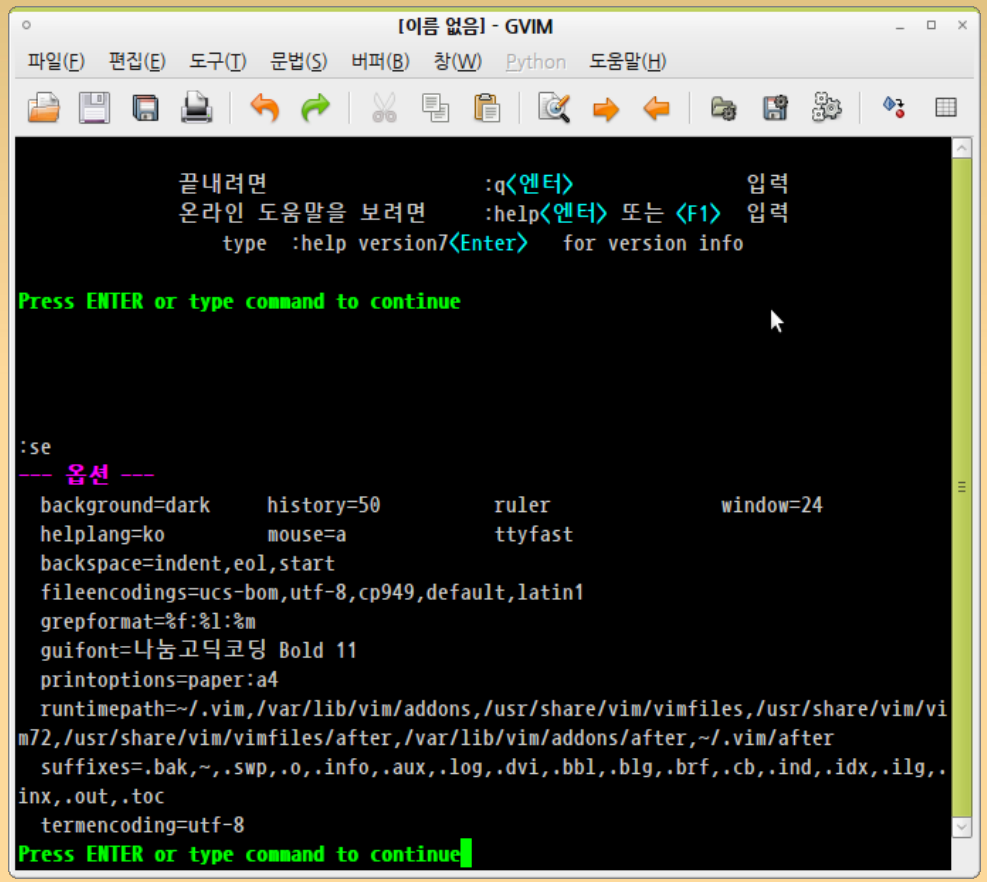


- 마스킹을 벗겨냄
- 드릴링
- 전자부품을 조립 / 완성

ubuntu

Vim – 소개

- 설치 (gvim)
  - $ sudo apt-get install vim-gnome
- GUI 메뉴로 폰트 변경 후 현재 설정 확인
  - :set
- 기본 폰트 설정
  - ~/.vimrc
  - set gfn= 나눔고딕코딩 \ Bold\ 10



ubuntu

Vim – 사용

- 내장 도움말
  - :help
- Graphical vi-vim Cheat Sheet and Tutorial
  - http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html

ctags & cscope

- ctags
  - http://ctags.sourceforge.net/
  - 소스의 심볼들을 빠르게 검색
- cscope
  - http://cscope.sourceforge.net/
  - 함수 호출 검색
  - 정규식으로 파일 검색
  - 파일을 포함한 파일 , etc...

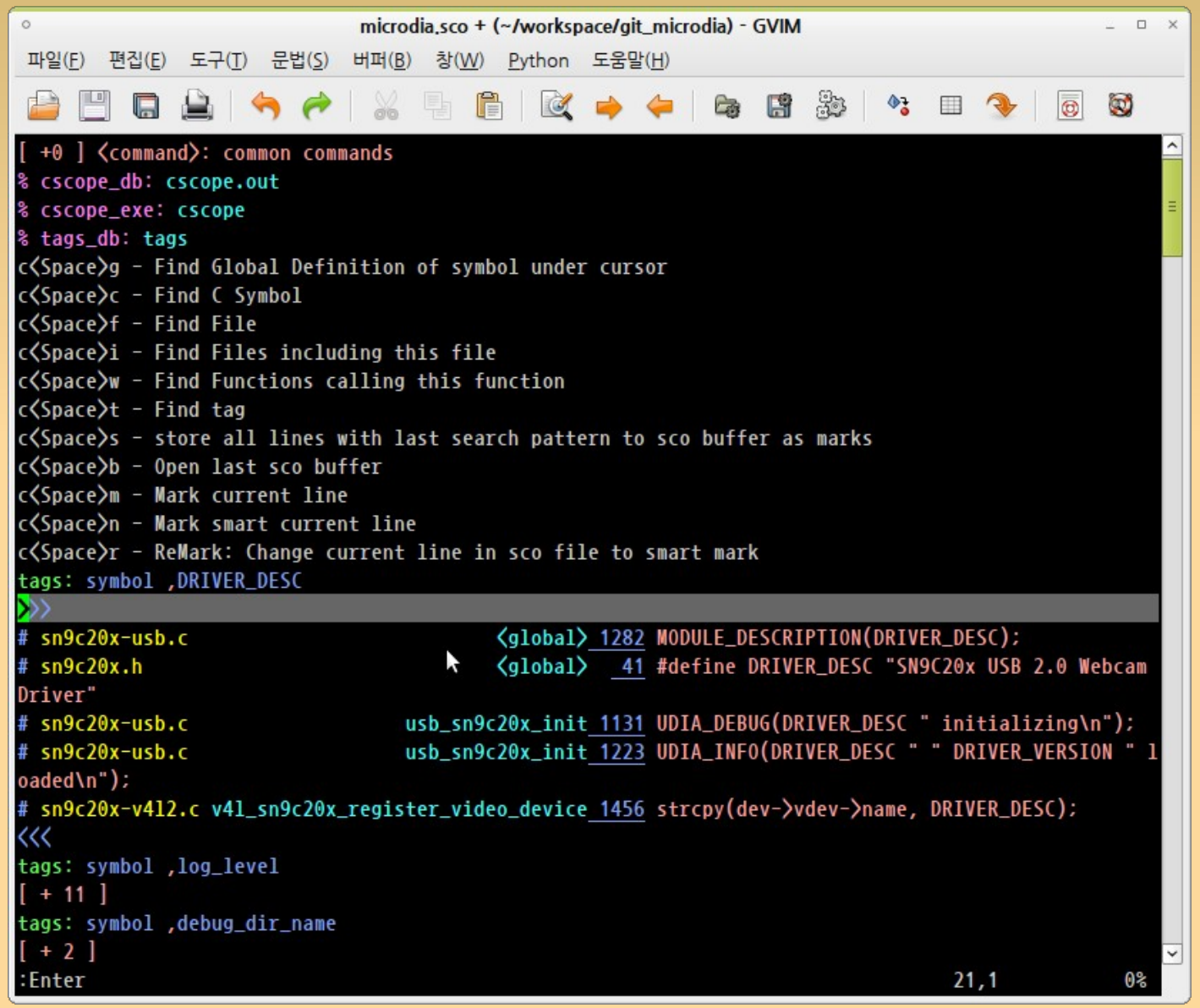
ctags & cscope – vim 연동 1/2

- SourceCodeObedience 플러그인 설치
  - http://www.vim.org/scripts/script.php?script_id=1638
  - 설치 - ~/.vim/plugin 폴더에 압축 해제
- ctags, cscope DB 생성

```
#!/bin/bash
SOURCE_LIST=source.list
rm -rf cscope.out tags
find . \( -name '*.c' -o -name '*.cpp' -o -name '*.cc' -o -name '*.h' -o
-name '*.s' -o -name '*.S' \) -print > $SOURCE_LIST
cscope -b -i $SOURCE_LIST
ctags -L $SOURCE_LIST
```

ubuntu

ctags & cscope – vim 연동 2/2



ubuntu

arm-cross-toolchain

- 크로스 툴 체인 빌드
  - http://www.kegel.com/crosstool/
- 기 빌드된 툴 체인 설치
  - CodeSourcery, etc...
  - 적당한 곳에 압축 해제 후 사용
- 다음과 같이 설치 위치 PATH 추가
  - $ vi ~/.bashrc
  - PATH=$PATH:/usr/local/arm/arm-2009q3/bin

Arm 타겟 으로 포팅하기 1/3

- 기존 target → Makefile 수정

```
$ make
cc  -c mkdosfs.c -o mkdosfs.o
cc  mkdosfs.o -o mkdosfs
$ file mkdosfs
mkdosfs: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.15, not
stripped
```

```
$ vi Makefile
CC=arm-none-linux-gnueabi-gcc
CFLAGS=--static
...
```

ubuntu

Arm 타겟으로 포팅하기 2/3

```
$ make
arm-none-linux-gnueabi-gcc --static -c mkdosfs.c -o mkdosfs.o
arm-none-linux-gnueabi-gcc  mkdosfs.o -o mkdosfs

$ file mkdosfs
mkdosfs: ELF 32-bit LSB executable, ARM, version 1 (SYSV),
dynamically
linked (uses shared libs), for GNU/Linux 2.6.16, not stripped
```

Arm 타겟으로 포팅하기 3/3

- 리눅스 커널의 Makefile 중 ...
  - CROSS_COMPILE 변수로 툴체인 설정
  - $ make ARCH=arm \
    CROSS_COMPILE="arm-none-eabi-" -j3

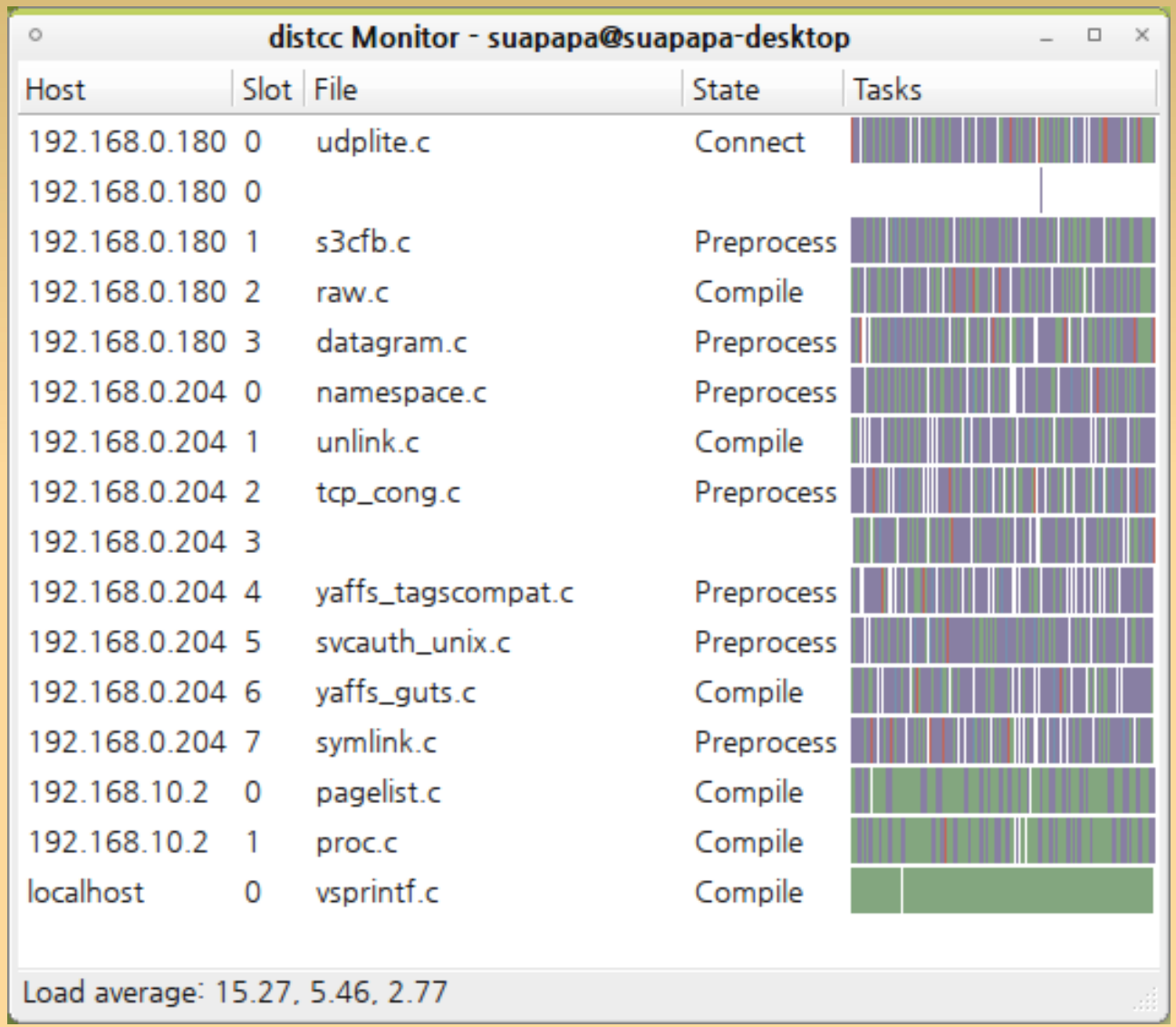
```
# Make variables (CC, etc...)
AS              = $(CROSS_COMPILE)as
LD              = $(CROSS_COMPILE)ld
CC              = $(CROSS_COMPILE)gcc
CPP             = $(CC) -E
AR              = $(CROSS_COMPILE)ar
...
```

ubuntu

Distcc - 분산 빌드 1/2

- 설치
  - $ sudo apt-get install distcc distccmon-gnome
- 서버 설정
  - /etc/default/distcc, /etc/init.d/distcc
- distcc 서버들 사용하기
  - $ DISTCC_HOSTS=”192.168.10.2/3 localhost/2” make -j 16
  - -j 옵션으로 (프로세스 개수 *3 + 1)

Distcc – 분산빌드 2/2



ubuntu

Q&A

감사합니다 .

ubuntu