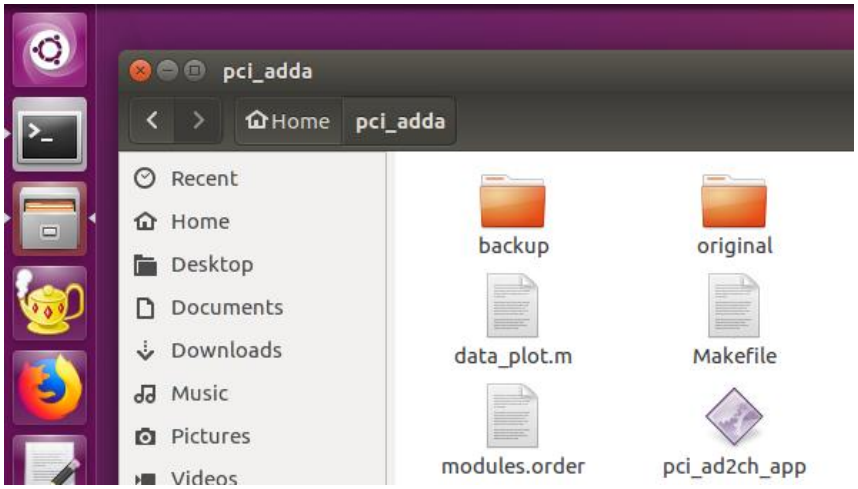
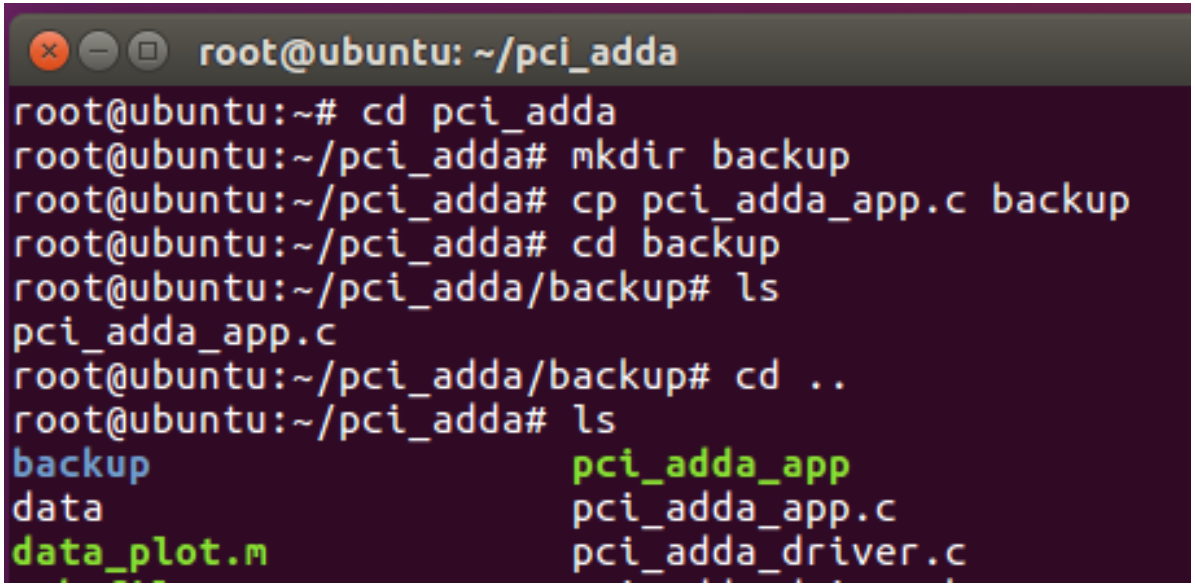
Lab 2

Simple AD/DA Conversion Programming on Linux PC

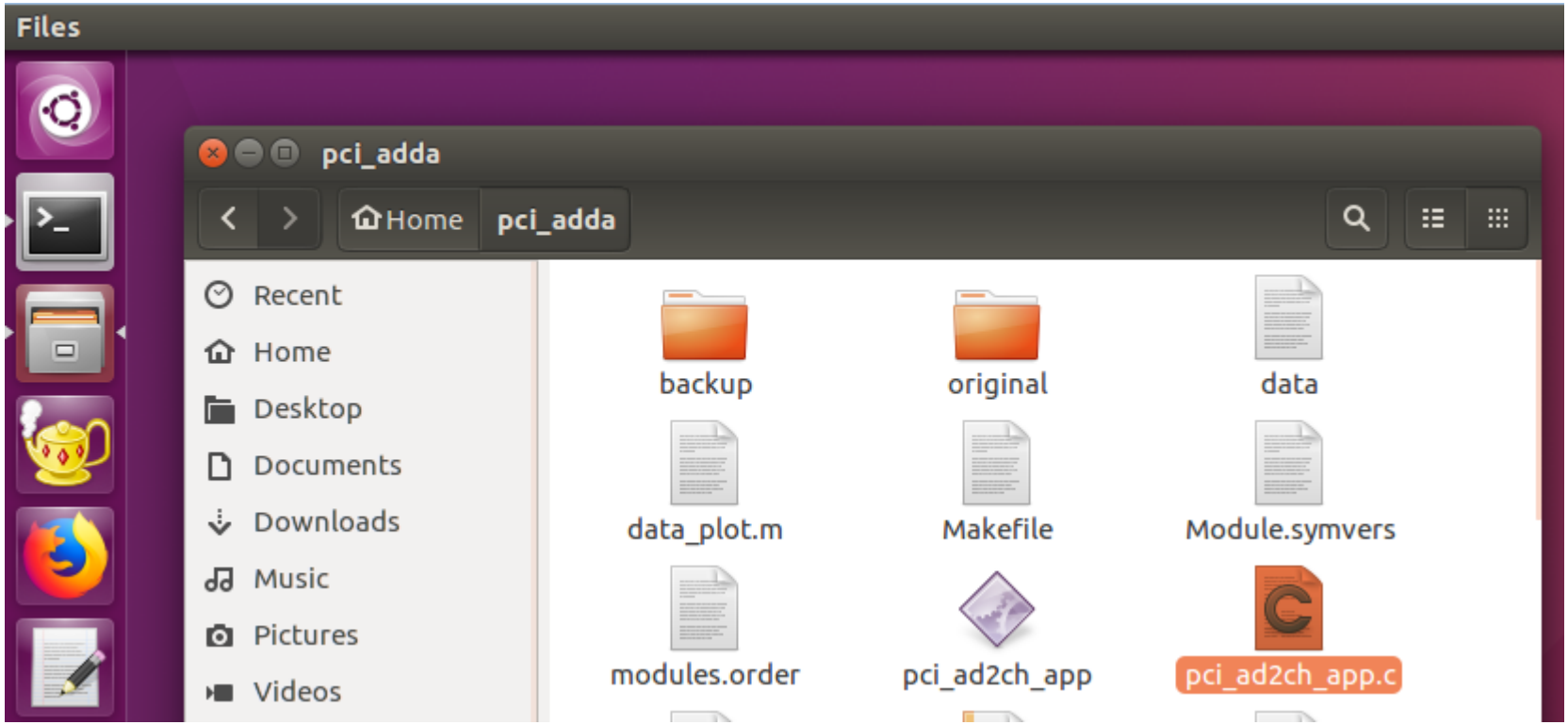
File preparation

```
# cd pci_adda
# mkdir backup
# cp pci_adda_app.c backup
# cd backup
# ls
# cd ..
# ls
# make
# ./pci_adda_app
```



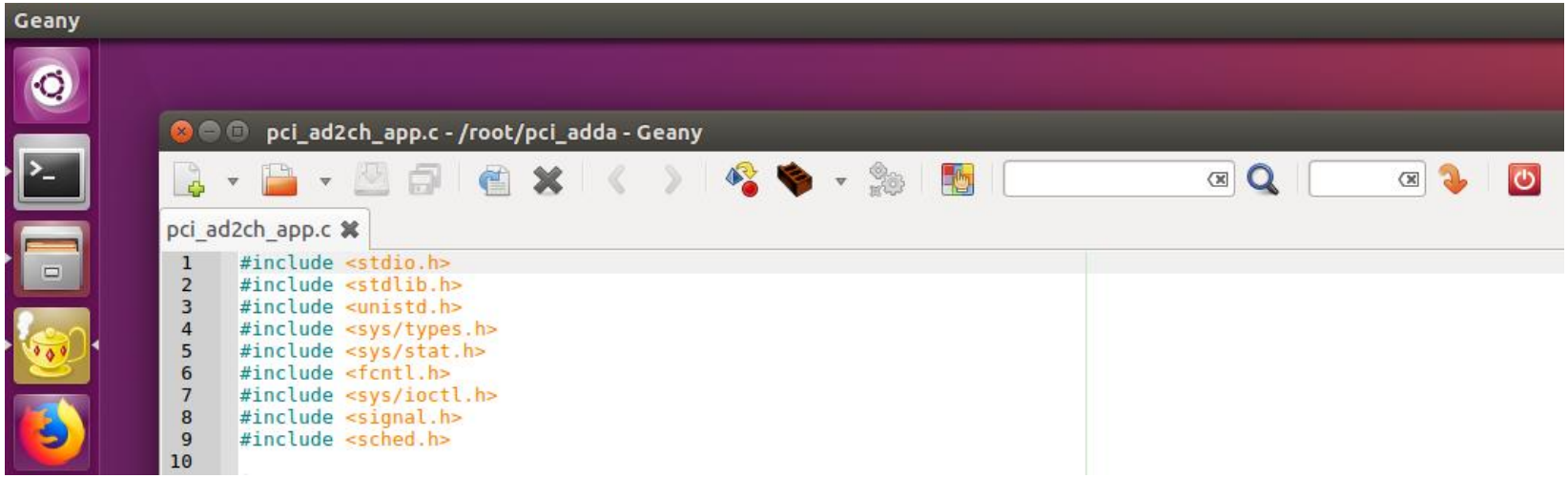


- 마우스 포인터를 메뉴 바에 가져가면 활성화된 윈도우에 대한 메뉴가 나옴.



Linux Editor

- vi – text based editor
- gedit – GUI editor
- geany – GUI editor for programming
- *.c 파일을 더블 클릭 할 경우 geany 로 파일이 열리도록 설정
- 또는 해당 파일을 선택하여 오른쪽 마우스 버튼을 누른 후 Open With 메뉴를 통해서 에디터 선택 가능



Exercise 1

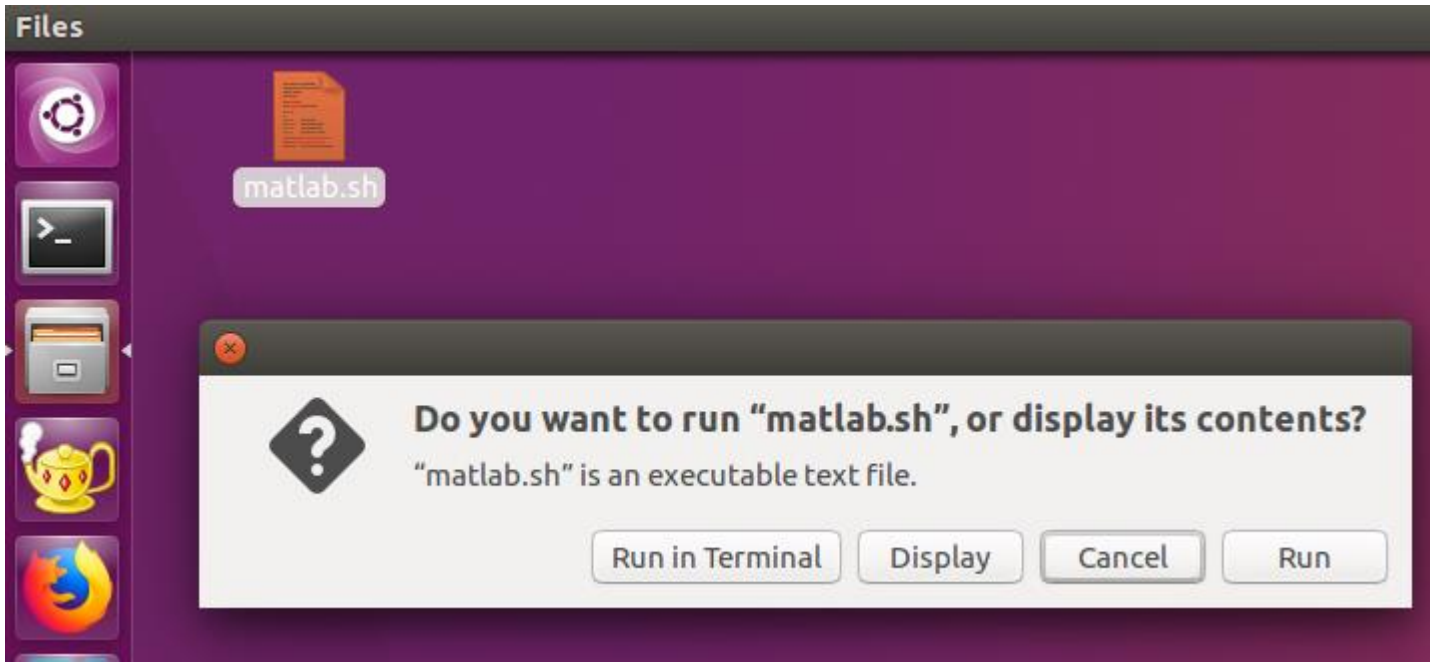
- 주파수가 100Hz, 10Hz, 1Hz인 square wave(0 volt-1 volt)를 발생 시켜서 DA converter로 출력한다.
- 주어진 program에서는 sampling frequency로 인터럽트가 발생되어, 인터럽트 서비스 루틴이 동작된다. 따라서 global 변수로서 interrupt counter 변수를 정의한 후, 이 변수를 적절히 사용하여 위의 주파수를 갖는 square wave를 발생시켜 oscilloscope로 확인한다.
- Sampling frequency = 5000 Hz

Exercise 2

- 함수 발생기 로부터 주파수가 10 Hz인 sine wave를 발생 시켜서 이를 AD converter로 입력한 후, 0.1 초 구간 동안(500 sample, 정현파 1주기)의 입력 값을 저장한 후, 이를 plotting 하여 sine wave의 그래프를 확인한다.
- 이때 AD converter로 입력된 데이터를 저장하기 위해서는 저장 하려는 데이터의 수 만큼 global 변수 array 로서 정의한 후, Interrupt Service Function 내에서 AD converter 입력 값을 변수에 저장하고, 이 변수 array가 모두 채워지면, 파일에 저장 하도록 한다. Interrupt Service Function 내에서는 파일 저장을 수행해서는 안 된다.
- Sampling frequency = 5000 Hz

MATLAB

- Double click matlab.sh on Desktop
- Click Run in Terminal
- Be patient. MATLAB is very slow.



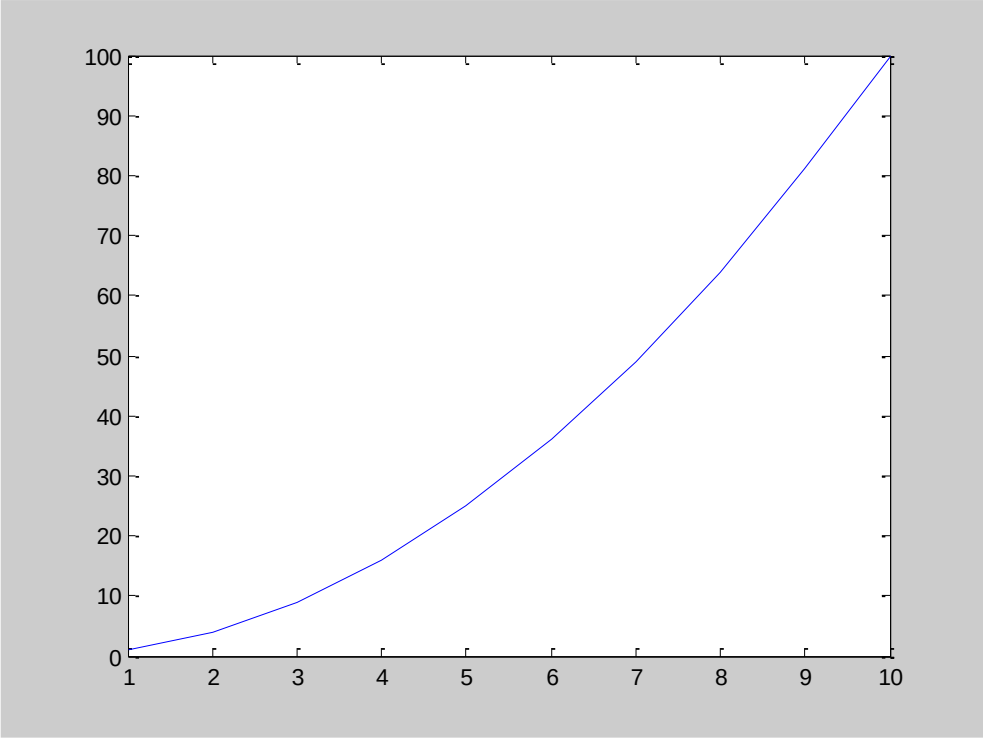
MATLAB command

```
>> load data
>> data

data =

     1     1
     2     4
     3     9
     4    16
     5    25
     6    36
     7    49
     8    64
     9    81
    10   100

>> plot(data(:,1),data(:,2))
```



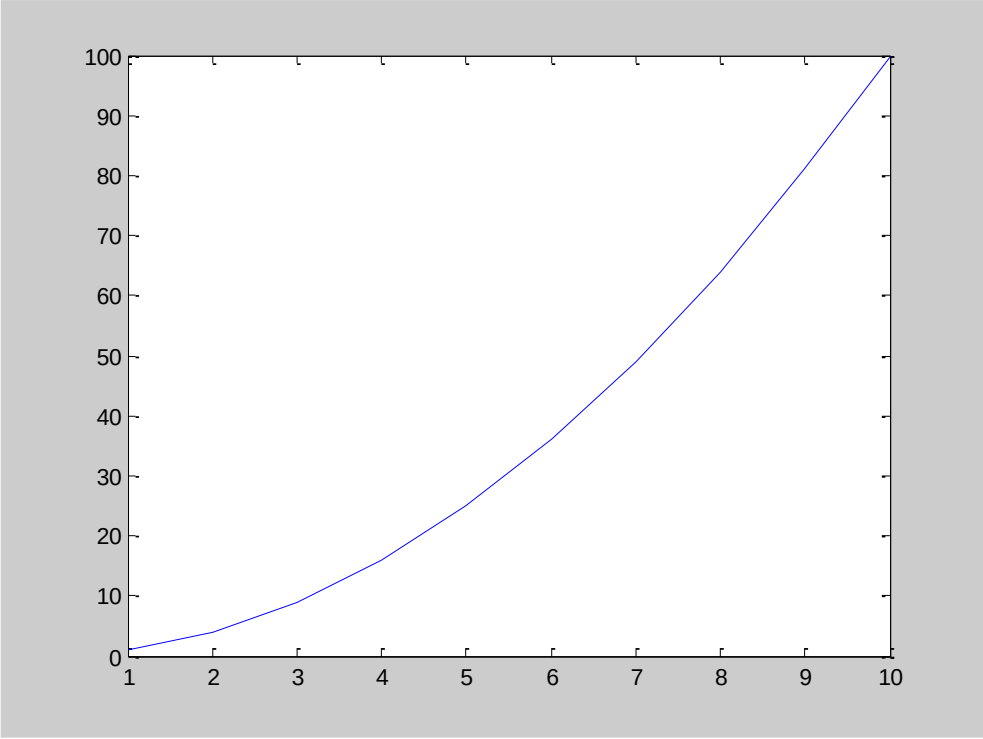
MATLAB command

```
>> s=load('data','-ascii')

s =

     1     1
     2     4
     3     9
     4    16
     5    25
     6    36
     7    49
     8    64
     9    81
    10   100

>> plot(s(:,1),s(:,2))
```



pci_adda_app.c(1)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <sched.h>

int fd;
unsigned long counter = 0;int toggle=0;
long timer_value,sf;

void usrsignal(int sig)
{
    short data[2];
    read(fd,data,1);
    write(fd,&data[0],1);
}
```

pci_adda_app.c(2)

```
 int main(void)
{
    unsigned long id;
    unsigned char c;    struct sched_param param, new_param;

    printf("start policy = %d\n", sched_getscheduler(0));
    /* 0 <- SCHED_OTHER, 1 <- SCHED_FIFO, 2 <- SCHED_RR */    param.sched_priority=59;
    printf("max priority = %d, min priority = %d, my priority = %d\n",
        sched_get_priority_max(SCHED_FIFO), sched_get_priority_min(SCHED_FIFO),
param.sched_priority);
    if (sched_setscheduler(0, SCHED_FIFO, &param) != 0){
        perror("sched_setscheduler failed\n");
        return 0;
    }
    fd = open("/dev/pci_adda",O_RDWR);
    if (fd < 0){
        printf("/dev/pci_adda open error\n");
        exit(1);
    }
```

pci_adda_app.c(3)

```
    sf=1000;
    printf("Sampling Frequency = %ld Hz \n",sf);
    timer_value=10000000/(sf*100);
    ioctl(fd,1,&timer_value,1);
    (void)signal(SIGUSR1,usrsignal);
    id = getpid();
    ioctl(fd,0,&id,1);
    printf ("Type Sampling Frequency(Maximum 50000) and press ENTER: \n");
    scanf("%ld",&sf);
    printf("Sampling Frequency = %ld Hz \n",sf);
    timer_value=10000000/(sf*100);
    ioctl(fd,1,&timer_value,1);

    printf("Type Ctrl-C to stop\n");
    for(;;) {
        sleep(1);
    }
    close(fd);
}
```

Two's complement

입력 전압	입력 값	Two's complement (short integer:2 bytes)	
10*(2047/2048)	0xfff	0x07ff	0000 0111 1111 1111
…	…	…	
10*(1/2048)	0x801	0x0001	0000 0000 0000 0001
0	0x800	0	0000 0000 0000 0000
−10*(1/2048)	0x7ff	0xffff	1111 1111 1111 1111
…	…	…	
−10	0x0	0xf800	1111 1000 000 0000

In C Programming

- char: -128~127
- unsigned char: 0~255
- short: -2^15 ~ 2^15-1
- unsigned short: 0~2^16-1
- long
- unsigned long

File open, write, close

```
FILE *fp;
 int i, number_of_data;
 fp=fopen("data","w+b");

 for (i=0;i<number_of_data;i++) {
     fprintf(fp,"%d %d\n",i,data[i]);
 }
 fclose(fp);
```