

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 : 황수정

sue100012@naver.com

사전 평가 문제

1. 단 한 번의 연산으로 대소문자 변환을 할 수 있는 연산에 대해 기술하시오

C 언어로 프로그램을 구현하시오.

> 아스키코드를 사용하는 것 같은데, 잘 모르겠습니다..ㅠ

2.Stack 및 Queue 외에 Tree 라는 자료구조가 있다.

이 중에서 Tree 는 Stack 이나 Queue 와는 다르게 어떠한 이점이 있는가?

> stack이나 queue와는 다르게 Tree는 데이터 검색과 데이터 입력과 변경이 좀 더 쉽게 이루어지는 이점이 있다.

3. 임의의 값 x가 있는데, 이를 4096 단위로 정렬하고 싶다면 어떻게 해야 할까?

C 언어로 프로그래밍 하시오.

4. int p[7] 와 int (*p)[7] 가 있다. 이 둘의 차이점에 대해 기술하시오.

> int p[7]은 int형의 값을 7개 저장할 수 있는 공간을 가진 배열을 뜻한다. 반면 int (*p)[7]은 int (*)[7]p로 풀어지며 배열포인터로 불리는 것이다. Int형 7개짜리 포인터를 뜻하는데, 이는 int형 데이터 타입의 값을 저장하는 공간 주소를 7개 저장할 수 있는 포인터를 뜻한다.

5. 다음을 C 언어로 구현하시오.

이번 문제의 힌트를 제공하자면 함수 포인터와 관련된 문제다.

아래와 같은 행렬을 생각해보자!

1 2 3

1 2 3

supply(arr, func) 이라는 함수를 만들어서 위의 행렬을 아래와 같이 바꿔보자!

supply 에 func 함수가 연산을 수행하는 함수로 만들어야 한다.

1 2 3

1 4 9

6. 다음 사항에 대해 아는대로 기술하시오.

Intel Architecture 와 ARM Architecture 의 차이점은 무엇인가?

7. 이것이 없으면 C 언어의 함수를 호출할 수 없다.

여기서 이야기하는 이것은 무엇일까?

> 메모리주소, 스택

8. 아래 내용을 C 로 구현해보도록 하자.

$3x^2 + 7x$ 를 1 ~ 2 까지 정적분하는 프로그램을 구현해보라.

예로 x 에 1 이 들어가면 $3x^2 = 9$ 가 된다.

9. Memory Hierarchy(메모리 계층 구조)에 대해 기술하시오.

> 메모리는 기본적으로 데이터를 저장하는 공간이다. HDD, RAM, Register, Cache 가 있으며, 속도는 register, cache, ram, hdd 순으로 빠르며 용량은 그 반대이다.

하드디스크(HDD, Hard Disk Drive) 상대적으로 용량당 단가가 낮아 많은용량을 구입할 수 있다. 전원이 꺼져도 데이터는 남아 있어, 반영구적으로 데이터를 소장할 수 있다. 다만 처리 속도가 느리다.

10. C 언어에서 중요시하는 메모리 구조에 대해 기술하시오.

(힌트: Stack, Heap, Data, Text 에 대해 기술하시오.)

CODE > 함수, 제어문, 상수 영역	> 가상(기능) 메모리로 그 활용과 기능에 따라 Code, Data, Stack, Heap로 구분하기도 한다. 프로그래밍 실행 중에 수시로 메모리 할당과 반환이 이루어지는 경우 동적인 영역이라고 하며, 변함없이 프로그램이 끝날 때까지 계속 살아 있는 경우 정적인 영역이라고 한다.
DATA > 전역변수	
HEAP > 동적할당	
STACK > 지역변수	

11. 파이프라인이 깨지는 경우에 대해 기술하시오.

> goto문 또는 재귀함수 호출과 같이 assembly의 jmp에 해당하는 명령문을 포함하는 경우이다. 이미 실행된 연산을 멈추고 가져오기 때문이다.

12. void (* signal(int signum, void (* handler)(int)))(int)라는 signal 함수의 프로토타입을 기술하시오. 프로토타입을 기술하라는 의미는 반환형(리턴 타입)과 함수의 이름, 그리고 인자(파라미터)가 무엇인지 기술하라는 뜻임.

> 반환타입: void

함수이름: signal

인자: int signum, void (*handler)(int)

13. goto 를 사용하는 이유에 대해 기술하시오.

> 흔히, goto문은 초보자가 다루기 힘들어 사용하지 말라고 말하는 코드이다. 그러나 System Programming에서 배우겠지만 goto를 활용하면 특정한 상황을 아주 간결하게 해결할 수 있는 경우가 많다.

if 와 break 를 조합하는 소스 코드와 goto 로 처리하는 소스코드를 가지고 있다고 가정해보자. for문 수가 많을수록, break 거는 수도 많아진다(for에 break를 걸기 때문이다). break를 걸기 위해선 flag선언이 필요하다. 그리고 for문 끝마다 if(flag){ break; }를 해주어야 break를 할 수 있다. 또한, error가 났지만 코드는 계속 진행된다. 나중에 확인하고 error가 났기 때문에 사용할 수 없어 그냥 시간을 낭비하는 경우도 있다. 그러나 goto문을 사용하면 error가 난 순간에 멈출 수 있다.

19. 아래 자료 구조를 C 언어로 구현해보시오.

Stack 자료구조를 아래와 같은 포맷에 맞춰 구현해보시오.

(힌트: 이중 포인터)

ex)

```
int main(void)
{
    stack *top = NULL;
    push(&top, 1);
    push(&top, 2);
    printf("data = %d\n", pop(&top));
}
```

20. 다음 질문에 답하시오.

Binary Tree 나 AVL Tree, Red-Black Tree와 같이 Tree 계열의 자료구조를 재귀 호출 없이 구현하고자 한다.

이 경우 반드시 필요한 것은 무엇인가 ?

21. 다음 자료 구조를 C 로 구현하시오.

Binary Tree 를 구현하시오.

초기 데이터를 입력 받은 이후 다음 값이 들어갈 때 작으면 왼쪽 크면 오른쪽으로 보내는 방식으로 구현하시오.

삭제 구현이 가능하다면 삭제도 구현하시오.

22. 다음 질문에 대해 기술하시오.

AVL 트리는 검색 속도가 빠르기로 유명하다.

Red-Black 트리도 검색 속도가 빠르지만 AVL 트리보다 느리다.

그런데 어째서 SNS 솔루션등에서는 AVL 트리가 아닌 Red-Black 트리를 사용할까 ?

> AVL 트리는 검색만 빠르다. 데이터 입출력의 속도가 느린 편으로 이는 데이터 양이 많을 경우, 그 양상을 더 나타낸다. 서버에서 몇 십만 명이 몰려서 데이터를 입력하고 삭제를 반복한다면, 감당하기 힘들 것이다. 반면 RB트리는 데이터의 삽입, 삭제 속도가 빠르다. 검색 속도 또한 준수한 편으로 대규모 데이터 처리에 적합하다.

23. 다음 자료구조를 C 로 구현하시오.

AVL 트리를 C 언어로 구현하시오.

AVL 트리는 밸런스 트리로 데이터가 1, 2, 3, 4, 5, 6, 7, 8, 9 와 같이 순서대로 쌓이는 것을 방지하기 위해 만들어진 자료구조다.

24. 다음 질문에 답하시오(주관식)

리눅스 실행 파일 포맷이 무엇인지 적으시오.

25. 다음 질문에 답하시오.

리눅스 운영체제에서 프로세스와 스레드를 구별하는 방법에 대해 기술하시오.

26. Context Switching 에 대해 기술하시오.

Multi-Tasking 의 핵심에 해당하는 기술인 Context Switching 에 대해 기술하시오.

27. $O(N)$ 과 $O(1)$ 알고리즘에 대해 고려해보자.

둘 중 무엇이 더 좋은 것인가? 이유를 기술하시오.

28. 가상 메모리란 무엇인지에 대해 기술하시오.

리눅스 커널의 task_struct 구조체를 살펴보면 mm_struct 라는 구조체가 있다. 이 구조체의 관점에서 기술하시오.

29. MMU 에 대해 기술하시오.

Cortex-M, Cortex-R, Cortex-A 등의 다양한 ARM 아키텍처가 존재한다.

이 중 Cortex-A 에는 반드시 MMU 가 탑재되는데 MMU 가 하는 일에 대해 자세히 기술하시오.

30. 소프트웨어 인터럽트에 대해 생각해보자.

리눅스 시스템에서 소프트웨어 인터럽트 번호는 0x80 에 해당한다.

소프트웨어 인터럽트가 무엇인지 기술하고 이것의 동작 메커니즘에 대해 기술하시오.

31. 운영체제 메모리 관리 기법중 페이징에 대해 기술하시오.

리눅스 커널의 task_struct, mm_struct, vm_area_struct 를 기준으로 적어도 무방하며 CP15 레지스터등의 내용등 자유롭게 기술해도 무방하다.

(힌트: 32 비트 기준 10, 10, 12)

32. 다음 디바이스 드라이버를 리눅스 상에서 구현해보시오.

Character Device Driver를 아래와 같이 동작하게 만드시오.

read(fd, 1, 10)을 동작시킬 경우 2번째 인자 ~ 3번째 인자 까지의 덧셈을 반환하도록 한다.

write(fd, 1, 5)를 동작시킬 경우 2번째 인자 ~ 5번째 인자 까지의 곱셈을 반환하도록 한다.

lseek(fd, 7, SEEK_END)를 동작시킬 경우 2번째 인자값을 반환하고 Kernel 내에서 "SEEK_END means end"를 출력하게 하라!

close(fd)를 수행하면 Kernel 내에서 "Finalize Device Driver"가 출력되게 하라!

33. 이번 멜트 다운 버그와 관련한 문제다.

OoO(Out-of-Order)인 비순차 실행에 대해 기술하라.

멜트 다운 버그가 이 OoO 기능과 관련이 있다.

그러나 해당 내용까지 기술할 필요는 없다.

34. "Hello World" 가 출력되도록 인라인 ARM 어셈블리를 활용해보자.

아래와 같은 형식을 사용해보도록 하라.

```
#include <stdio.h>
char msg[15] = "Hello World";
int main(void)
{

register unsigned char *r5 asm("r5") = NULL;
// 여기 한줄만 채우면됨
printf("r5 = %s\n", r5);

return 0;
}
```

35. 어셈블리 프로그래밍을 해보자.

아래 형식을 기반으로 배열의 내용을 "Hello Test" 로 출력되도록 어셈블리 프로그래밍 하시오.

원래 배열에는 "Hello World" 가 들어있다.

```
#include
char msg[15] = "Hello World";
int main(void)
{
register unsigned char *r5 asm("r5") = NULL;
register unsigned char r6 asm("r6") = 0;
r5 = msg;
printf("r5 = %s\n", r5);
// asm volatile 부분에 어셈블리 명령어를 채워서 기능이 동작하게 만들면됨
asm volatile("mov r6, #0x54\n"
"strb r6, [r5, #6]\n");
printf("msg = %s\n", msg);

return 0;
}
```

36. 리눅스 시스템 프로그래밍 문제로 System Call 을 활용하도록 하라.

디렉토리 내에 들어 있는 모든 File 들을 출력하는 Program을 작성하시오.

37. 리눅스 시스템 프로그래밍 기법을 활용하여 369 게임을 프로그래밍 해보시오.

2 초 내에 값을 입력하게 하시오.

박수를 쳐야 하는 경우를 Ctrl + C를 누르도록 한다.

2초내에 값을 입력하지 못할 경우 게임이 오버되게 한다.
Ctrl + C를 누르면 "Clap!"이라는 문자열이 출력되게 한다.

38. 리눅스 시스템 프로그래밍 문제다.

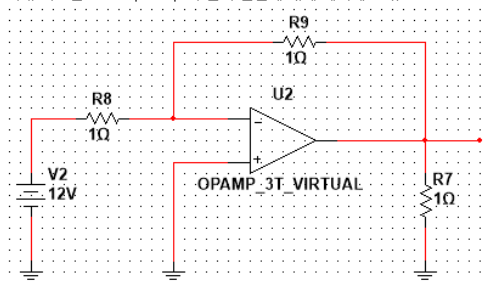
랜덤한 이름을 가지도록 디렉토리 3개를 만들고
각각의 디렉토리에 5 ~ 10개 사이의 랜덤한 이름(길이도 랜덤)을 가지도록 파일을 만들어보자!
모든 디렉토리를 순회하며 3 개의 디렉토리와 그 안의 모든 파일들의 이름 중
a, b, c 가 1개라도 들어있다면 이들을 출력하라!
출력할 때 디렉토리인지 파일인지 여부를 판별하도록 하시오.

39. 시스템 콜 관련 문제다.

ARM 에서 System Call 을 사용할 때 사용하는 레지스터를 적으시오.

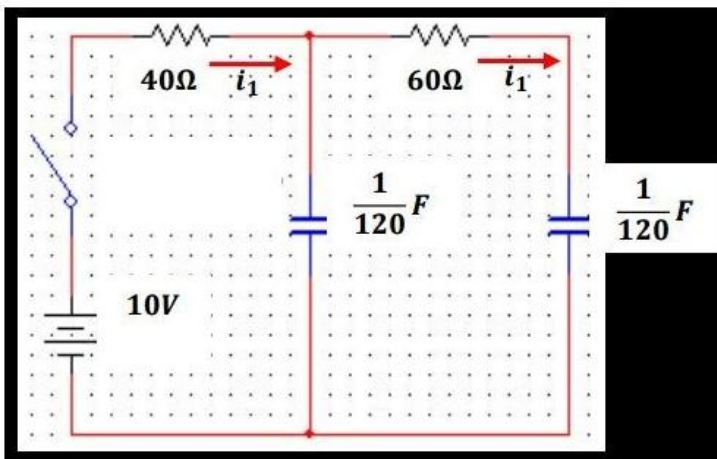
40. 다음 Op-Amp 를 해석해보자.

아래 회로를 보고 Op-Amp 의 출력 전압에 대해 해석해보자.



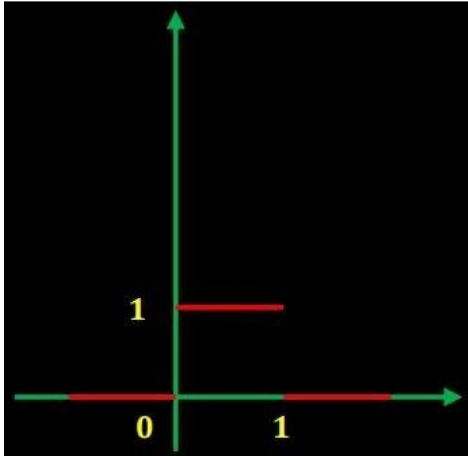
41. 아래 회로의 동작 함수에 대해 분석해보자.

아래 회로를 해석하기 위해 라플라스 변환을 활용해보길 바란다.



42. 다음 그림을 보고 문제를 푸시오.

아래 그림의 신호에 대한 푸리에 적분을 수행해보시오.



43. 다음 연립 방정식을 푸시오.

(힌트: 크래머 공식, 역 행렬 방식, determinant 등을 활용하면 쉽게 풀 수 있다)

$$3x + 6y + 9z = 60$$

$$4x + 2y + 6z = 40$$

$$3x + 2y + z = 20$$

44. 아래 그림을 보고 미분 방정식을 푸시오.

전형적인 2 계 비동차 미분 방정식 문제에 해당한다.

$$3y'' + 4y' + y = x^2 + 4$$

45. 다음 행렬의 역행렬을 구하시오.

힌트: 가장 편하게 푸는 방법은 determinant 를 활용하는 것이다.

$$\begin{pmatrix} 1 & -2 & 0 \\ 3 & 1 & 5 \\ -1 & 2 & 3 \end{pmatrix}$$

46. 레이더 범위 방정식에 대한 문제다.

여기서 도출할 수 있는 핵심적 결론은 무엇인지 기술하시오.

$$R_{max} = \left(\frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_o B F L (SNR)} \right)^{\frac{1}{4}}$$

$$SNR = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_o B F L R^4}$$

47. 공중의 방송국이라 불리는 EC-130H 에 대한 문제다.

미국과 전쟁을 하게되면 이 기체가 떠서 적국의 모든 방송국을 방송 불가능하게 재밍을 한다.

그리고 주파수를 전부 이 기체가 스니핑하여 어떤 방송에서도 이 기체가 하는 방송만을 듣게 만들 수 있다.

실제 미군이 아랍어 방송을 한 사례로 매우 유명하다.

아래 사진의 EC-130H 기체가 이와 같은 기술을 사용하는데 필요한 핵심 기술은 무엇인가 ? (힌트: 영어 단어 3 개)



48. 연립 미분 방정식을 풀어보자.

힌트: 라플라스 변환

$$x'' - 3x' + 4y' + y = 4$$

$$2y' - x' + 3y = 0$$

$$x(0) = x'(0) = y(0) = 0$$

49. 벡터 미적분학 문제다.

이 벡터 함수에 대한 Gradient, Divergence, Curl 을 구하시오.

$$\{x^2 + 2y + z \ln(x)\}\hat{i} + \{y^2z + e^z\}\hat{j} + \{xyz + 2x^2ye^z + z^3\}\hat{k}$$

50. 가우스-조르단 소거법을 C 로 구현하시오.

연립 방정식 문제를 C 가 풀 수 있게 구현하면 된다.

이것은 주로 로봇 팔의 제어등을 수행하는데 활용되는 기법이다.

51. 벡터의 사칙 연산, 내적 및 외적을 C 로 구현하시오.

힌트: 벡터의 외적은 3 by 3 행렬의 determinant 를 구하는것과 유사하다.

52. 역행렬을 구하는 프로그램을 구현하시오.

정밀 제어를 수행하기 위해 칼만 필터등을 구현하는데도 이 기능을 필수적이다.

53. 펌웨어 문제다.

간단한 LED 를 제어하고자 한다.

62. 레이더는 즉각적으로 적의 움직임을 파악할 수 있다.

레이더는 전자기파를 방출한다.

전자기파의 속도가 얼마나 빠르길래 적이 움직이자마자 속도값과 거리값을 알아채고 방향까지 알 수 있는것일까 ?

(힌트: 맥스웰이 맥스웰 방정식을 발표하고 이것을 예언했고 이로써 도출되는 결론과 관계된다)

63. VHDL 혹은 Verilog 를 사용하여 주파수 분할기를 구현해보시오.

주파수 분할기를 FPGA 를 활용하여 PWM 제어를 수행하는 작업을 수행할 때 매우 중요한 부분이다. (Vivado HLS 를 사용하는 경우라면 HLS 를 사용해도 무방하다)

64. FPGA 보드에 Petalinux 를 올리고 디바이스 드라이버를 만들어보자.

간단하게 브레드 보드에 LED 를 연결하고 이것을 FPGA 에서 디바이스 드라이버를 통해 제어해보라.

65. FPGA 보드에 Petalinux 를 올리고 PWM 제어를 수행해본다.

이를 통해서 간단한 서보 모터를 구동시켜보도록 한다.

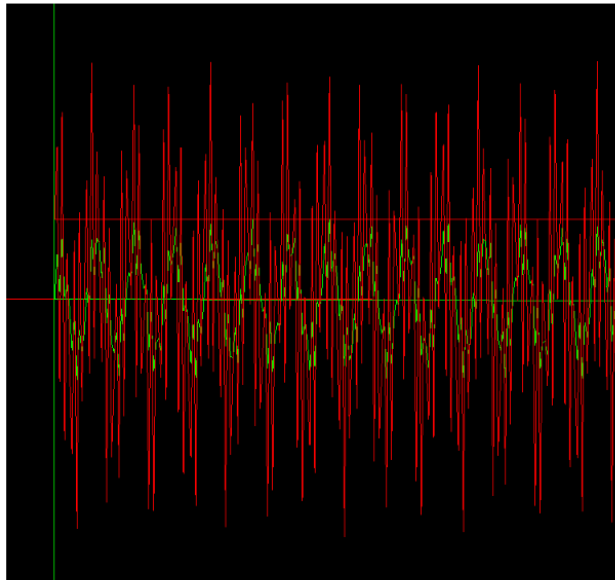
66. FPGA 보드에 petalinux 를 올리고 I2C 제어를 수행해보자.

학원에서 사용하는 대표적 I2C 장비는 Lidar 에 해당한다.

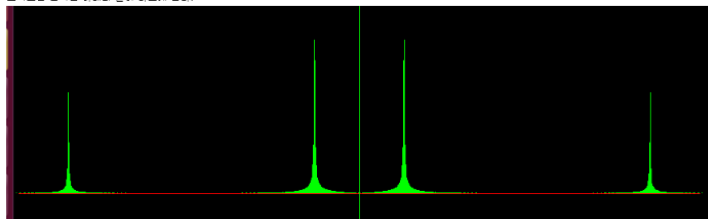
이 Lidar 를 I2C 제어를 통해 제어해보도록 하라.

67. 간단한 RC 필터를 C 로 구현하시오.

결과를 아래와 같이 출력하게 구현하시오.

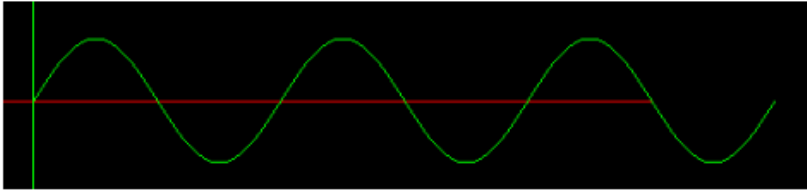


스펙트럼 분석을 아래와 같이 나오게 한다.



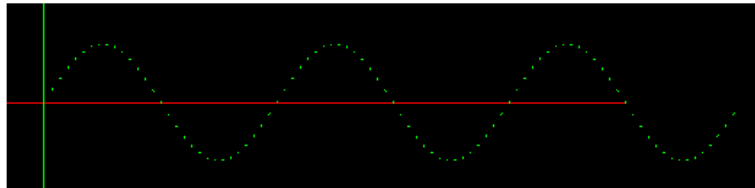
68. 간단한 sin 파형을 C 언어와 OpenGL 을 활용하여 그려보시오.

아래와 같이 출력되게 만들어보시오.



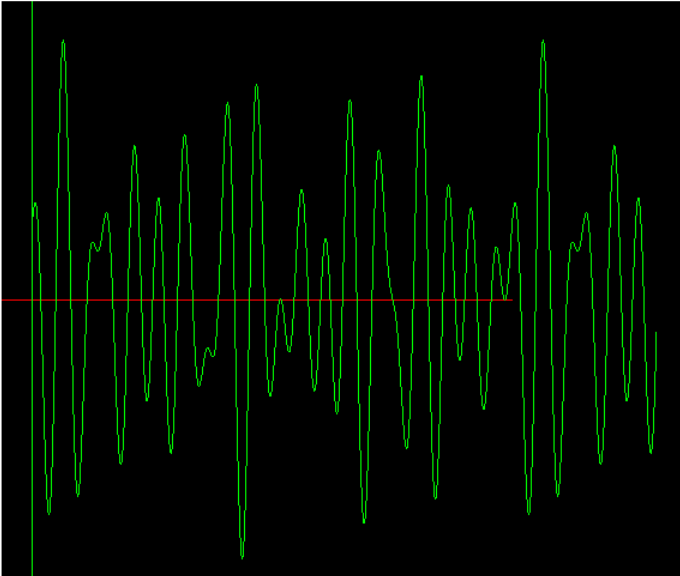
69. sine 파형을 점선으로 찍어보시오.

아래와 같이 출력되면 됨



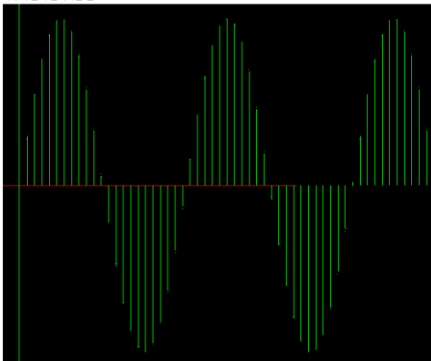
70. sine 파형을 합성하시오.

아래와 같은 합성 파형을 만들어보시오.



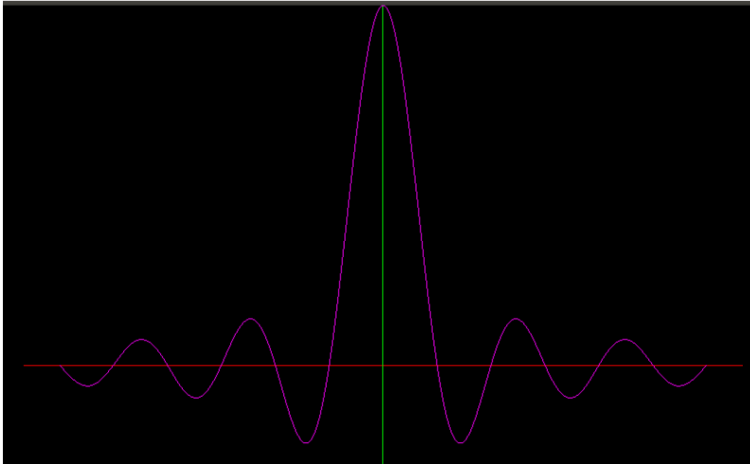
71. 이산 신호 처리를 수행할 수 있도록 신호를 디지털화 하시오.

아래와 같이 출력되면 됨



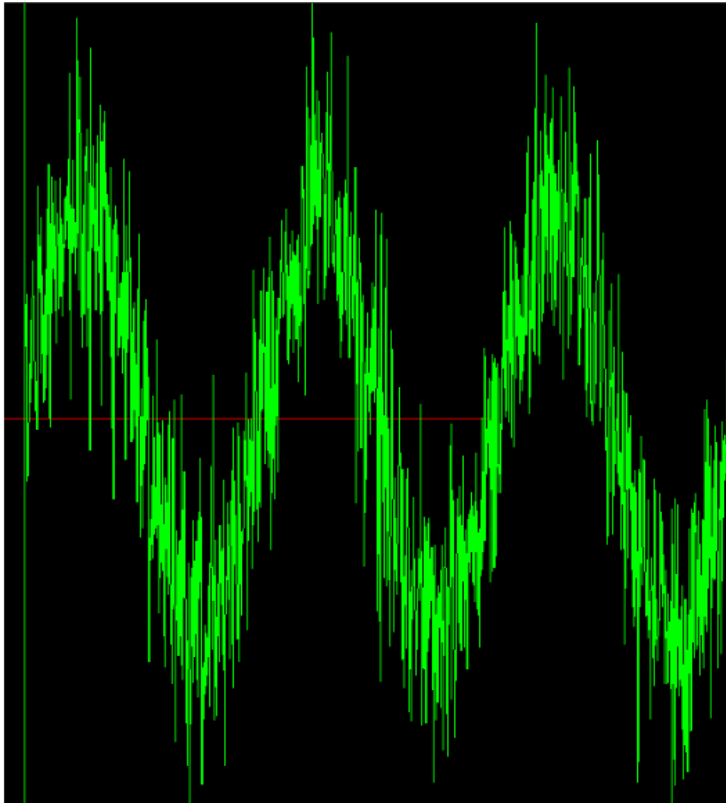
72. 수능에 자주 등장하는 sinc 함수를 구현해보자.

아래와 같이 출력되면 된다.



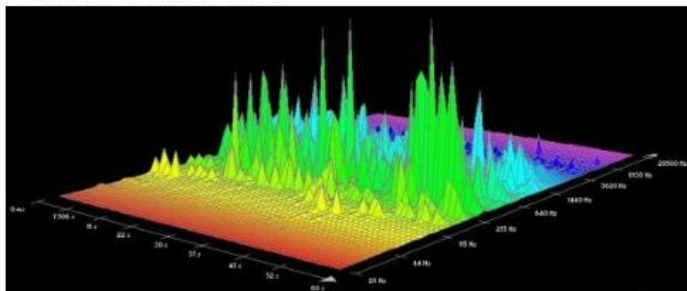
73. 가우시안 노이즈를 구현해보자.

아래와 같이 파형에 노이즈를 입력해보시오.



74. 스펙트로 그래프를 구현하시오.

임의의 신호 함수를 만들고 아래와 같이 출력되게 구현하시오.
스펙트럼, 시간, 주파수의 3D로 구성해야함



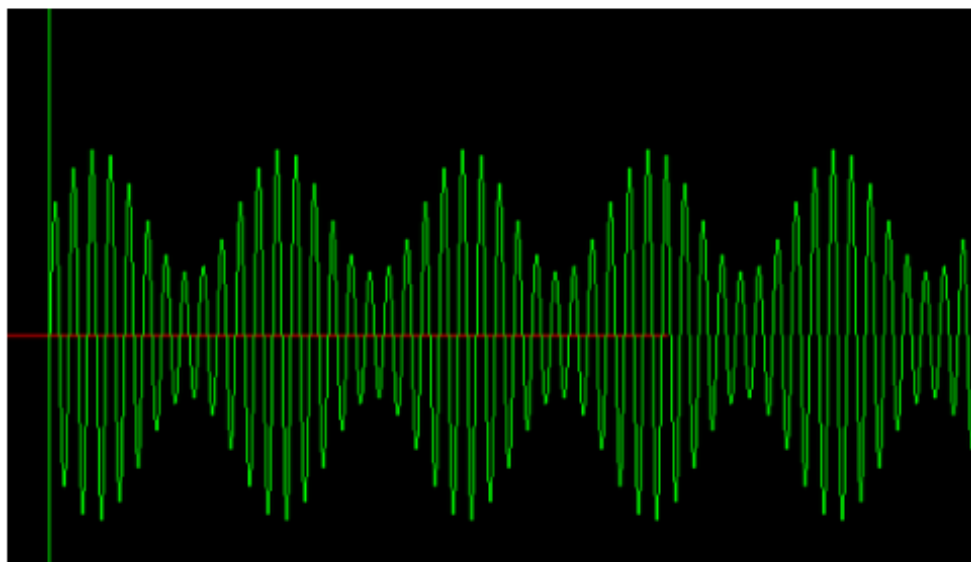
75. 네비어 스톡스 방정식에 대해 아는대로 기술하시오.

네비어 스톡스 방정식은 유체 역학 뿐만 아니라 공기 역학에서도 매우 중요한 해석 모델이다

76. AM 변조를 수행하시오.

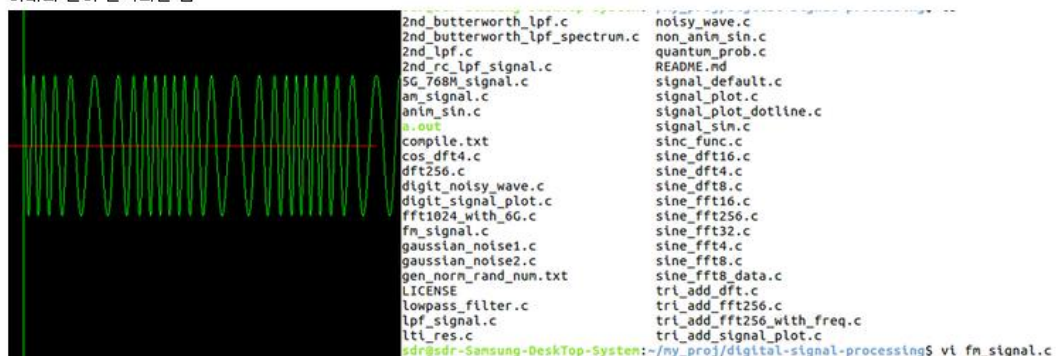
아래와 같이 출력되면 됨

아래와 같이 출력되면 됨



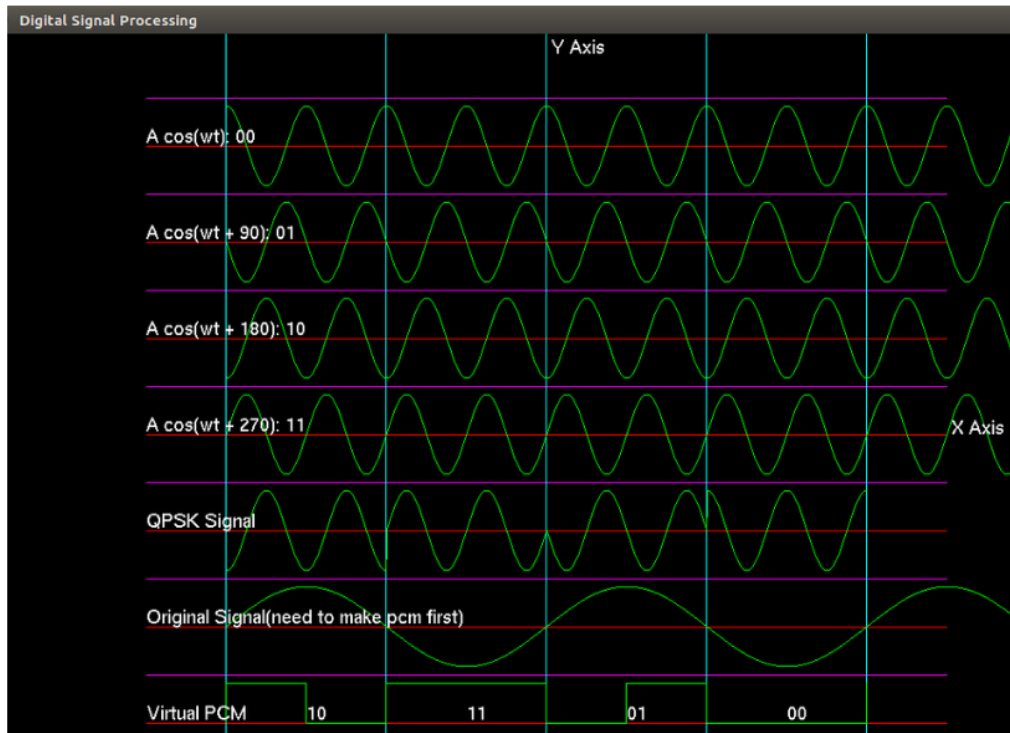
77. FM 변조를 수행해보시오.

아래와 같이 출력되면 됨



78. QPSK 신호를 만들어보시오.

아래와 같이 출력의 순



79. 쿼드콥터에 대한 운동 방정식을 기술해보자.

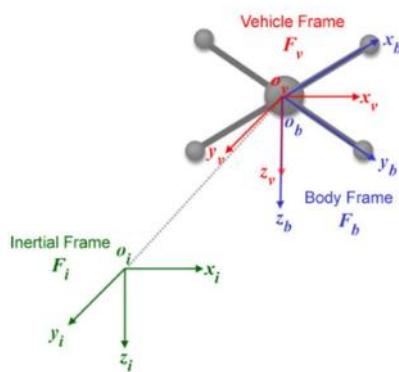
아래와 같이 관성 좌표계, 몸체 좌표계, 기체 좌표계에 입각하여 해석을 하시오.

Reference Frames

운동학 방정식과 Quadrotor의 동역학에 들어가기 전에 채택된 좌표계와 Reference Frame 을 지정하고 다른 좌표계 간의 변환이 수행되는 방법을 지정하도록 합니다.

다른 좌표 프레임의 사용은 6 DOF(6 개의 자유도)에서 Quadrotor의 위치와 고도를 식별하는데 필수적입니다. 예로 운동 방정식을 평가하기 위해 quadrotor 에 부착된 좌표계가 필요합니다. 그러나 Quadrotor 에 작용하는 힘과 모멘트는 IMU 센서 값과 함께 Body Frame 을 기준으로 평가됩니다. 마지막으로 Quadrotor의 위치와 속도는 조종기의 관성 프레임에 대해 GPS 측정을 사용하여 평가됩니다.

그러므로 아래와 같이 3 가지 주요 Reference Frame 이 채택된다:



관성 프레임 $F_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ 는 조종기를 원점으로 잡고 있는 지구의 고정 좌표계입니다. 규칙에 따라 x 축은 북쪽을 향하고 y 축은 동쪽을 향하며 z 축은 지구의 중심을 향합니다.

Body Frame $F_b = (\bar{x}_b, \bar{y}_b, \bar{z}_b)$ 은 원점이 Quadrotor의 무게중심(COG)에 존재합니다. 그리고 축은 Quadrotor 구조와 일치선으로 맞춰져서 x 축의 \bar{x}_b 는 앞 부분의 motor 방향으로 뻗어나가고 y 축의 \bar{y}_b 는 앞 부분의 motor 방향으로 뻗어나가고 z 축은 $\bar{z}_b = \bar{x}_b \times \bar{y}_b$ 과 같이 외적으로 나타납니다.

Vehicle Frame $F_v = (\bar{x}_v, \bar{y}_v, \bar{z}_v)$ 은 관성 프레임이며 원점이 Quadrotor의 무게중심(COG)에 존재합니다. Vehicle Frame 은 F_b 및 F_ϕ 라는 2 개의 변형을 가집니다. F_ϕ 은 x_v 와 y_v 가 각각 x_b 와 y_b 에 일치선으로 맞춰지도록 z 축의 z_v 를 중심으로 각도 ϕ 만큼 회전된 Vehicle Frame F_v 를 의미합니다. F_θ 은 Frame F_ϕ 가 y 축을 중심으로 y_ϕ 만큼 회전 각도 θ 만큼 회전하여 x_ϕ 와 z_ϕ 가 각각 x_b 와 z_b 에 일치선이 되도록 합니다.

80. 아래와 같이 쿼드콥터를 만들려고 한다.

이를 구현하는데 필요한 모든 기술과 각각을 본인이 어느대로 기술해보시오.



81. 맥스웰 방정식을 파헤쳐보자.

맥스웰 방정식은 무선 시스템에 있어 매우 중요한 방정식이다.

본인이 알고 있는대로 4 개의 방정식에 대해 기술해보고

Tensor 해석이 가능하다면 2 개의 방정식으로 압축하여 기술해보시오.

82. 현대 전자전에서 핵심인 ECM 에 대해 기술하시오.

(힌트: 채프나 플레어, 재밍 등등중 아무거나 하나 골라 잡고 기술하시오)

83. Larmor Formula 에 대해 기술하시오.

라모 공식이라고도 불리는 Larmor Formula 에 대해 기술하시오.

84. 창 함수에 대해 기술하시오.

창 함수에는 Rectangular, Hamming, Hanning, Kaiser 등이 존재하는데 왜 사용하는 것인가 ?

85. 기본 for 문 활용 문제다.

1 ~ 100 까지의 숫자중 홀수만 더해서 출력해보시오.

```
#include <stdio.h>
```

```
int sum(void)
```

```
{
```

```
    int a;
```

```
    int sum = 0;
```

```
    for( a = 0; a < 101; a++)
```

```
        { if( a % 2 != 0)
```

```
            sum += a;
```

```

}
printf("1~100까지 숫자 중 홀수의 합은 %d 이다.\n", sum);
return sum;
}

```

```

int main(void)
{
    sum();
    return 0;
}

```

86. 기본 배열 문제다.

1 ~ 100 까지 숫자를 모두 더해서 첫 번째 배열에 저장하고
 1 ~ 100 까지 숫자중 홀수만 더해서 두 번째 배열에 저장하고
 1 ~ 100 까지 숫자중 짝수만 더해서 세 번째 배열에 저장한다.
 다음으로 1 ~ 100 까지 숫자중 3 의 배수만 더해서 네 번째 배열에 저장한다.
 각 배열의 원소를 모두 더해서 결과값을 출력하시오.

87. 자기장에 대한 문제다.

비오-사바르 법칙에 대해 기술하시오.

88. 전자기파와 빛의 관계에 대해 기술하시오.

힌트: 이 특성 때문에 초음파등의 음파가 아닌 레이더를 사용하는 것이다.

89. 자유낙하 문제다.

물체가 자유낙하 할 때 종단 속도를 구하기 위한 방정식을 설계해보라.

(힌트: 공기 저항은 속도에 비례한다)

90. 에너지 보존 법칙에 대해 기술하시오.

힌트: 위치 에너지, 병진 운동 에너지, 회전 운동 에너지 관점에서 기술하시오.

91. 차량과 토크의 관계

실제로 차량을 개발하고자할 때 토크 선정이 매우 중요한데 그 이유가 무엇인지 상세히 기술하시오.

92. ADAS 에서 사용하는 다양한 센서를 기술하시오.

힌트: 각 센서들을 왜 이렇게 복합적으로 사용하는 것인가 ?

각각의 이유에 대해 상세히 기술하시오.