

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정



2018.05.15

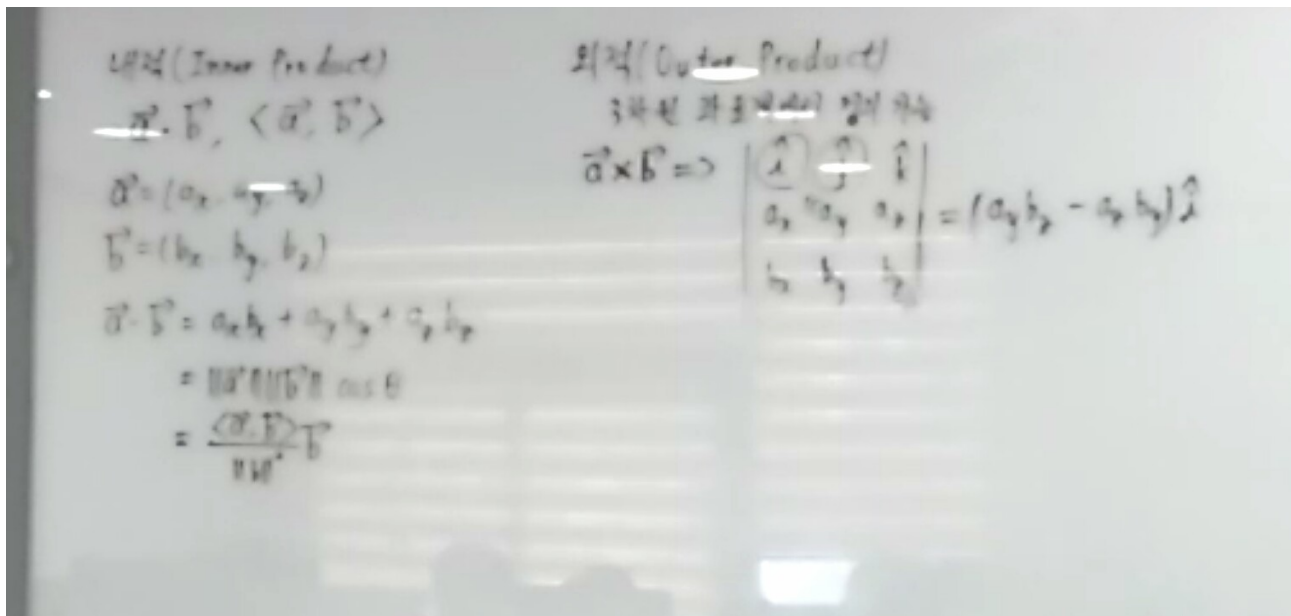
54 일차

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 신민철

akrn33@naver.com



```
#include "vector_3d.h"
#include <stdio.h>

int main(void)
{
    vec3 A = {3, 2, 1};
    vec3 B = {1, 1, 1};
    vec3 X = {1, 0, 0};
    vec3 Y = {0, 1, 0};
    vec3 v[3] = {{0, 4, 0}, {2, 2, 1}, {1, 1, 1}};
    vec3 w[3] = {};
    vec3 R = {0, 0, 0,
              vec3_add, vec3_sub, vec3_scale,
              vec3_dot, vec3_cross, print_vec3,
              gramschmidt_normalization};

    R.add(A, B, &R);
    R.print(R);

    R.sub(A, B, &R);
    R.print(R);

    R.scale(3, R, &R);
    R.print(R);

    printf("A dot B = %f\n", R.dot(A, B));

    R.cross(X, Y, &R);
    R.print(R);

    R.gramschmidt(v, w, R);

    return 0;
}
```

Files

```
#ifndef VECTOR_3D_H_
#define VECTOR_3D_H_

#include <stdio.h>
#include <math.h>

typedef struct vector3d vec3;

struct vector3d
{
    float x;
    float y;
    float z;

    void (* add)(vec3, vec3, vec3 *);
    void (* sub)(vec3, vec3, vec3 *);
    void (* scale)(float, vec3, vec3 *);
    float (* dot)(vec3, vec3);
    void (* cross)(vec3, vec3, vec3 *);
    void (* print)(vec3);

    void (* gramschmidt)(vec3 *, vec3 *, vec3);
};

void vec3_add(vec3 a, vec3 b, vec3 *r)
{
    r->x = a.x + b.x;
    r->y = a.y + b.y;
    r->z = a.z + b.z;
}

void vec3_sub(vec3 a, vec3 b, vec3 *r)
{
    r->x = a.x - b.x;
    r->y = a.y - b.y;
    r->z = a.z - b.z;
}

void vec3_scale(float factor, vec3 a, vec3 *r)
{
    r->x = a.x * factor;
    r->y = a.y * factor;
    r->z = a.z * factor;
}
```

```

float vec3_dot(vec3 a, vec3 b)
{
    return a.x * b.x + a.y * b.y + a.z * b.z;
}

void vec3_cross(vec3 a, vec3 b, vec3 *r)
{
    r->x = a.y * b.z - a.z * b.y;
    r->y = a.z * b.x - a.x * b.z;
    r->z = a.x * b.y - a.y * b.x;
}

void print_vec3(vec3 r)
{
    printf("x = %f, y = %f, z = %f\n", r.x, r.y, r.z);
}

float magnitude(vec3 v)
{
    return sqrt(v.x * v.x + v.y * v.y + v.z * v.z);
}

void gramscmidt_normalization(vec3 *arr, vec3 *res, vec3 r)
{
    vec3 scale1 = {};
    float dot1, mag1;

    mag1 = magnitude(arr[0]);
    r.scale(1.0 / mag1, arr[0], &res[0]);
    r.print(res[0]);

    mag1 = magnitude(res[0]);
    dot1 = r.dot(arr[1], res[0]);
    r.scale(dot1 * (1.0 / mag1), res[0], &scale1);
    r.sub(arr[1], scale1, &res[1]);
    r.print(res[1]);
}

#endif

```