

# TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee (이상훈)  
[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)  
학생 - 김형주  
[mihaelkel@naver.com](mailto:mihaelkel@naver.com)

vector.h

```
1  #ifndef __VECTOR_H__
2  #define __VECTOR_H__
3  #include <stdio.h>
4  #include <math.h>
5  void print_vector(double* x);
6  void vector_sum(double* res, double* x, double* y);
7  void vector_sub(double* res, double* x, double* y);
8  void vector_scalar_mul(double* res, double* x, double scalar);
9  double vector_inner_product(double* x, double* y);
10 void vector_cross_product(double* res, double* x, double* y);
11 double vector_abs_val(double* x);
12 void vector_gso_normalization(double* x0, double* x1, double* x2, double* w0, double* w1, double* w2);
13 #endif
```

vector.c

```
1  #include "vector.h"
2  #include <stdio.h>
3  #include <math.h>
4  void print_vector(double* x)
5  {
6      printf("%lf, %lf, %lf\n", x[0], x[1], x[2]);
7  }
8  void vector_sum(double* res, double* x, double* y){
9      int i;
10     for(i=0; i<3; i++)
11         res[i] = x[i] + y[i];
12 }
13 void vector_sub(double* res, double* x, double* y){
14     int i;
15     for(i=0; i<3; i++)
16         res[i] = x[i] - y[i];
17 }
18 void vector_scalar_mul(double* res, double* x, double scalar)
19 {
20     int i;
21     for(i=0; i<3; i++)
22         res[i] = x[i]*scalar;
23 }
24 double vector_inner_product(double* x, double* y)
25 {
26     int i;
27     double res = 0;
28     for(i=0; i<3; i++)
29         res += x[i]*y[i];
30     return res;
31 }
32 void vector_cross_product(double* res, double* x, double* y)
33 {
34     int i;
35     for(i=0; i<3; i++)
36         res[i] = x[(i+1)%3]*y[(i+2)%3]-x[(i+2)%3]*y[(i+1)%3];
37 }
38 }
39 double vector_abs_val(double* x)
```

```

40 {
41     int i;
42     double res = 0;
43     for(i=0;i<3;i++)
44         res += x[i]*x[i];
45     res = sqrt(res);
46     return res;
47 }
48 void vector_gso_normalization(double* x0, double* x1, double* x2, double* w0, double* w1, double* w2)
49 {
50     int i;
51     for(i=0;i<3;i++)
52         w0[i] = x0[i];
53
54     for(i=0;i<3;i++)
55     {
56         w1[i] = x1[i] - ((vector_inner_product(x1,w0))/pow(vector_abs_val(w0),2))*w0[i];
57     }
58
59     for(i=0;i<3;i++)
60     {
61         w2[i] = x2[i] - ((vector_inner_product(x2,w0))/pow(vector_abs_val(w0),2))*w0[i]
62             - ((vector_inner_product(x2,w1))/pow(vector_abs_val(w1),2))*w1[i];
63     }
64
65
66 }
67

```

```

1  #include "vector.h"
2
3  int main(void)
4  {
5      double v0[3] = {0.0, 4.0, 0.0};
6      double v1[3] = {2.0, 2.0, 1.0};
7      double v2[3] = {1.0, 1.0, 1.0};
8
9      double w0[3];
10     double w1[3];
11     double w2[3];
12     vector_gso_normalization(v0,v1,v2,w0,w1,w2);
13
14     print_vector(w0);
15     print_vector(w1);
16     print_vector(w2);
17 }

```