

# Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee( 이상훈 )

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

학생 – 장성환

[redmk1025@gmail.com](mailto:redmk1025@gmail.com)

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

#define exp 2.71828
#define Hz 1000
#define sampling_time 0.001

void init_res(double (*res)[], int size){
    int i;

    for(i=0; i<size; i++){
        (*res)[i] = 0;
    }
    return;
}

void print_data(double *pr, int size){

    int i;

    for(i=0; i<size; i++){
        printf("[%d data] : %1.12lf \n", i, pr[i]);
    }
}

void compare_data(double *calc_y, double* real_y, int size){

    int i;
    double x=-5;

    for(i=0; i<size; i++){
```

```

    printf("Y[%lf] | %1.12lf | %1.12lf | %3.2lf%%\n", x, calc_y[i], real_y[i], (1-(calc_y[i]/real_y[i]))*100);
    x += sampling_time;
}
}

void make_input_data(double (*y)[], int size){
    int i;
    double x=-5;

    for(i=0; i<size; i++){
        (*y)[i]= 3.0*pow(exp,(double)-pow(x,2));
        x += sampling_time;
    }
    return;
} // y = 3e^(-x^2)

void diff_func(double *y, double (*res)[], int size){
    int i;

    init_res(res, size);

    for(i=0 ;i < size - 1; i++){
        (*res)[i] = (y[i+1]-y[i])/sampling_time;
    }
    (*res)[size-1] = (*res)[size-2];
} // y' (input diff)

void integr_func(double *y, double (*res)[], int size){
    int i;

    init_res(res, size);

```

```

for(i=1 ;i < size ; i++){
    (*res)[i] = (y[i]*sampling_time)+(*res)[i-1];
}
}

void expect_func(double (*res)[], int size){
    int i;
    init_res(res, size);
    (*res)[Hz*5] = 3;
    double x = sampling_time;

    for(i=5000; i<size; i++){
        (*res)[i+1] = (*res)[i] / (2*sampling_time*x+1) ;
        x += sampling_time;
    }

    x=-sampling_time;

    for(i=5000; i>=0; i--){
        (*res)[i-1] = (*res)[i] / (2*-sampling_time*x+1) ;
        x -= sampling_time;
    }
}

int main(void){
    double y[Hz*10]={0};
    double res[Hz*10] ={0};
    double expect_origin_res[Hz*10] ={0};

    int size = sizeof(y)/sizeof(double);

```

```
make_input_data(&y, size);

//print_data(y, size);

diff_func(y, &res, size);

//print_data(res, size);

integr_func(y, &res, size);

//print_data(res, size);

expect_func(&expect_origin_res, size);

compare_data(expect_origin_res, y, size);

//print_data(expect_origin_res, size);

return 0;
}
```