EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv2 과정

제 1기

2021. 12. 17

김태훈

REDBLACK TREE

EDDI
Electronic Design Development Institute

아직 Red Black tree 완성은 못 했는데, coding하면서 찾은 공유 할만한 기법들 소개합니다.

1. POINTER의 변수명을 printf하는 define

```
#define PRINTDATA(X) (printf(#X " data : %d\n" , X->data))
#define PRINTCOLOR(X) (X->color == RED)?(printf(#X " color : %s\n" , "RED")) : (printf(#X " color : %s\n" , "BLACK"))
```

#이라는 것을 쓰면 함수의 input string을 있는 그대로 찍어 줍니다.

REDBLACK TREE

EDDI
Electronic Design Development Institute

2. TDD를 구현하기 위해서 Customized한 것

```
int main()
{
    redblack *root = NULL;

    int data[50] = { 0 };
    int len = sizeof(data) / sizeof(int);


    test1(&root, data);
    test2(&root, data);
    test3(&root, data);
    test4(&root, data);
    test5(&root, data);
    test6(&root, data);
    test7(&root, data);
    test8(&root, data);
    test9(&root, data);
    test10(&root, data);
    test11(&root, data);
    test12(&root, data);
    test13(&root, data);
    test14(&root, data);
    test15(&root, data);
    test16(&root, data);
    test17(&root, data);
    test18(&root, data);
    test19(&root, data);
    test20(&root, data);
    test21(&root, data);
    test22(&root, data);
    test_random(&root, data, len);

}
```

특정 함수를 변경할 경우 이전 test를 통과하지 못하고 현재 경우만 통과하는

협소한 변경일 수 있기 때문에

매번 실행마다 test를 통과하도록 자동화를 해주어야함.

REDBLACK TREE

2. TDD를 구현하기 위해서 Customized한 것

```c
void test_random(redblack** root, int* data, int len)
{
    printf("/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    ///////////////%s\n",__func__);
    srand(time(NULL));

    init_data(data, len);
    print_arr(data, len);



    repeat_insert_delete(root, data, len);
}
```

printf에 __func__를 인자로 넣어주고 %s를 string에 넣어주면 printf를 호출하는 함수명이 출력이 되는데, 이는 내가 현재 몇번째 test를 하고 있는지 알게 해준다.

REDBLACK TREE

EDDI
Electronic Design Development Institute

2. TDD를 구현하기 위해서 Customized한 것

```
void repeat_insert_delete(redblack** root, int* data, int len)
{
    int i;
    for (i = 0; i < len; i++)
    {
        //printf("insert data : %d\n", data[i]);
        insert_redblack(root, data[i]);
        //printf("Now root is %d\n",(*root)->data);
        // property 1 : The color of root node is BLACK
        assert(((*root) ? (*root)->color : BLACK) == BLACK);
        //print_redblack(*root);
    }
    // property 1 : The color of root node is BLACK
    assert(((*root) ? (*root)->color : BLACK) == BLACK);
    printf("The color of root node is BLACK\n");
    print_redblack(*root);

    for(i = 0; i < len; i++)
    {
        printf("delete data : %d\n", data[i]);
        delete_redblack(root, data[i]);
        recur_update_black_level(root);
        if((*root)!=NULL)
            printf("Now root is %d\n",(*root)->data);
        print_redblack(*root);
        // property 1 : The color of root node is BLACK
        assert(((*root) ? (*root)->color : BLACK) == BLACK);
    }
    assert(((*root) ? (*root)->color : BLACK) == BLACK);
    printf("The color of root node is BLACK\n");
    print_redblack(*root);
}
```

assert 함수를 이용하면
RED BLACK TREE의 특성을 위반했을 때
효과적으로 알 수 있습니다.
가령 옆의 경우는 ROOT 노드는 black이라는
특성을 늘 체크합니다.

PRINTF를 써도 되지만,
ASSERT를 쓰면 NDEBUG 매크로를 정의하면
컴파일이 안되므로 유용합니다.

REDBLACK TREE

EDDI
Electronic Design Development Institute

2. TDD를 구현하기 위해서 Customized한 것

```
void redblack_property_test(redblack **root)
{
    // 1. The color of root is black
    // NUMBER 1 condition will be checked in print function.

    // this function will be inserted in print function.
    // 2. There is no adjacent red node
    if((*root)->color == RED)
    {
        if((*root)->parent)
        {
            //PRINTDATA((*root)->parent);
            assert((*root)->parent->color == BLACK);
        }
        if((*root)->left)
        {
            assert((*root)->left->color == BLACK);
        }
        if((*root)->right)
        {
            assert((*root)->right->color == BLACK);
        }
    }
    // 3. Same black_level
    // compare left and right black level

    if((*root)->right && (*root)->left)
    {
        assert((*root)->right->black_level == (*root)->left->black_level);
    }
}
```

이번 경우는 RED 노드는 연달아서 올 수 없다는 특성과 BLACK level이 좌우 똑같아야 한다는 특성을 assert로 걸러주고 있습니다.
이를 위해서 black level이라는 변수와 함수를 정의해줄 필요는 있습니다.

REDBLACK TREE

EDDI
Electronic Design
Development Institute

2. TDD를 구현하기 위해서 Customized한 것

```
void print_redblack_for_check(redblack *tree)
{
    redblack* tmp = tree;

    if(tmp)
    {
        print_redblack_for_check(tmp->left);
        if(tmp->data == -1)
            printf("data = NIL\t");
        else
            printf("data = %4d\t", tmp->data);

        if(tmp->parent)
        {
            if(tmp->parent->data == -1)
            {
                printf("parent = NIL\t");
                printf("parent color = BLACK\t");
            }
            else
            {
                printf("parent = %4d\t", tmp->parent->data);
                if(tmp->parent->color == BLACK)
                {
                    printf("parent color = BLACK\t");
                }
                else
                {
                    printf("parent color = RED\t");
                }
            }
        }
        else
        {
            printf("parent = NULL\t");
            printf("parent color = NULL\t");
        }
        if(tmp->left)
        {
            if(tmp->left->data == -1)
                printf("left = NIL\t");
            else
                printf("left = %4d\t", tmp->left->data);
```

PRINT 함수안에 property를 체크하는 함수를 넣어주면
각 노드를 traverse할 때 매번 property가 체크됩니다.

REDBLACK TREE

EDDI
Electronic Design Development Institute

2. TDD를 구현하기 위해서 Customized한 것

```
void recur_update_black_level(redblack **root)
{
    if(root==NULL)
        return;
    if((*root)==NULL)
        return;
    if((*root)->data != -1)
    {
        recur_update_black_level(&(*root)->left);
        recur_update_black_level(&(*root)->right);
        if((*root)->color == BLACK)
            (*root)->black_level = MAX(LEVEL((*root)->left), LEVEL((*root)->right)) + 1;
        else
            (*root)->black_level = MAX(LEVEL((*root)->left), LEVEL((*root)->right));

    }
}
```

위와 같은 방법으로 black level을 update할 수 있습니다.

REDBLACK TREE

EDDI
Electronic Design Development Institute

2. TDD를 구현하기 위해서 Customized한 것

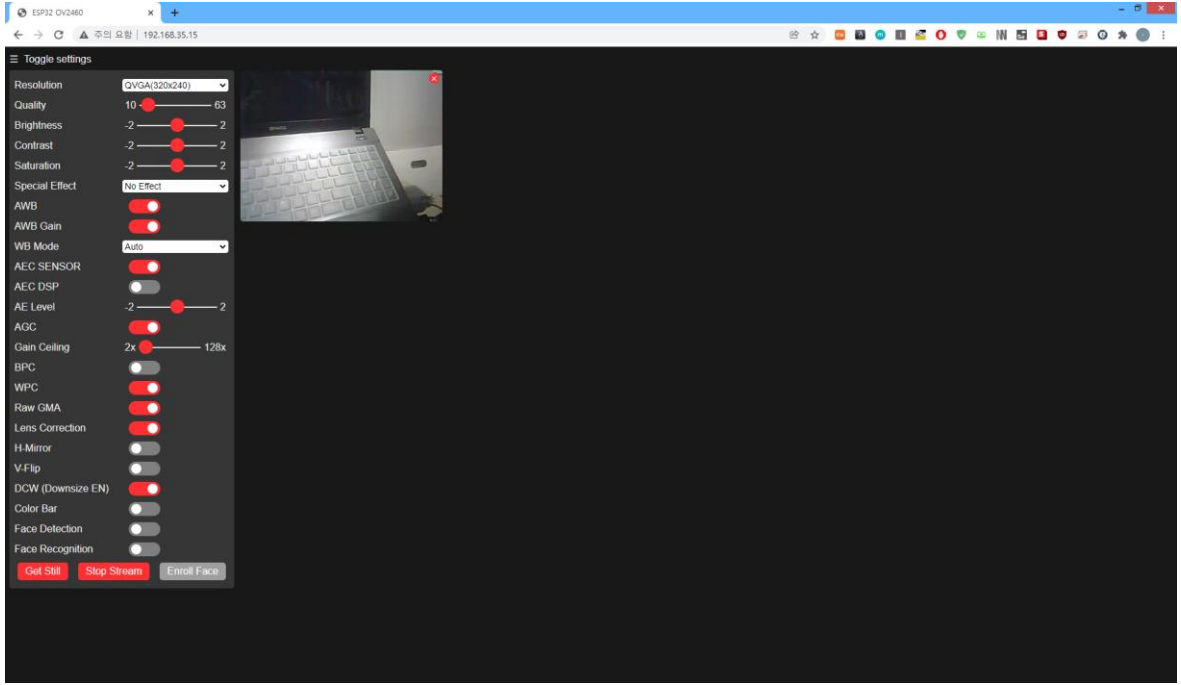
```
Now root is 14
data = NIL
data =    1     color = RED     left = NIL      right = NIL     parent =    2
parent color = BLACK    black level =    0
data = NIL
data =    2     color = BLACK   left =    1     right = NIL     parent =    6
parent color = BLACK    black level =    1
data = NIL
data =    6     color = BLACK   left =    2     right =    9    parent =   11
parent color = RED      black level =    2
data = NIL
data =    9     color = BLACK   left = NIL      right = NIL     parent =    6
parent color = BLACK    black level =    1
data = NIL
data =   11     color = RED     left =    6     right =   12    parent =   14
parent color = BLACK    black level =    2
data = NIL
data =   12     color = BLACK   left = NIL      right =   13    parent =   11
parent color = RED      black level =    1
data = NIL
data =   13     color = RED     left = NIL      right = NIL     parent =   12
parent color = BLACK    black level =    0
data = NIL
Assertion failed!

Program: C:\Users\Ted\Desktop\a.exe
File: redblack_tree.c, Line 1583

Expression: (*root)->right->black_level == (*root)->left->black_level
```

특성을 만족하지 못하게 알고리즘을 구현했을 경우
왼쪽과 같이 어떠한 경우에 그렇게 나오는지
assert를 해주므로 디버깅하기 수월합니다.

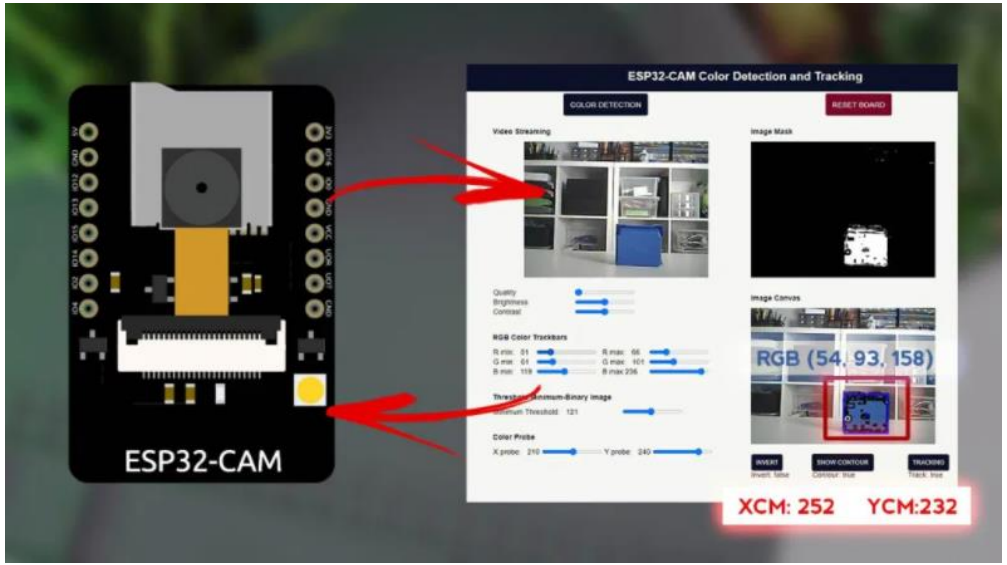
ESP32 CAM

EDDI
Electronic Design
Development Institute

ESP32를 Arduino IDE를 이용해서 확장 패키지를 설치 하여 예제를 실행한 모습입니다.
집에 있는 공유기와 연결하여 wifi로 통신하고 있습니다.



향후 계획



일단 서버 기반으로 OPENCV로 특정 Object의 좌표를 tracking하는 것을 개발하고
최종적으로는 mcu단에서 처리가 가능하도록 경량화 하는게 목표입니다.