

# Project – Proposal

## 지하철 좌석 확인 시스템

과 목	임베디드시스템S/W설계
담당교수	김태석 교수님
학 과	컴퓨터공학과
학 번	2015722087
성 명	김민철
날 짜	2020. 05. 11 (월)



## 1. 요약

시내 버스의 경우에는 여유, 보통, 혼잡 세 가지의 표시로 버스의 탑승객이 어느 정도 있는지 정류장 혹은 휴대폰으로 확인할 수 있게 하여 이용자로 하여금 편의성을 제공하고, 시외 버스의 경우에는 좌석이 몇 자리가 남아 있는지 확인할 수 있게 하는 편의성을 제공한다. 하지만 버스와 함께 가장 많이 이용되는 교통 수단인 지하철의 경우에는 이용자의 입장에서 제공받을 수 있는 정보가 없다는 생각이 들어 이번 프로젝트를 기획하게 되었다.

이번 프로젝트를 통해 개발하게 될 시스템에서 기존의 시내 버스, 시외 버스에서 제공하는 편의성을 지하철에서도 제공할 수 있게 하는 것이 목표이다. 지하철은 각 열차마다, 칸 마다 탑승 시에 카드를 찍지도 않기 때문에 정확한 인원 파악이 어렵고, 정확한 인원 파악을 위한 시스템을 구현하기에는 많은 비용이 소모될 것이다. 따라서 확인하는 방법으로는 좌석을 이용하는 것을 생각하였다.

좌석에 착좌 센서를 부착하여 좌석에 앉아 있는지 확인하여 좌석이 일정 수 이상 비어 있을 경우에는 여유, 좌석이 모두 이용되며 손잡이가 일정 수 이상 비어 있을 경우에는 보통, 좌석과 손잡이가 대부분 이용 중이라면 혼잡의 정보 혹은 남아있는 잔여 좌석 수를 이용자들에게 제공한다.

## 2. 본문

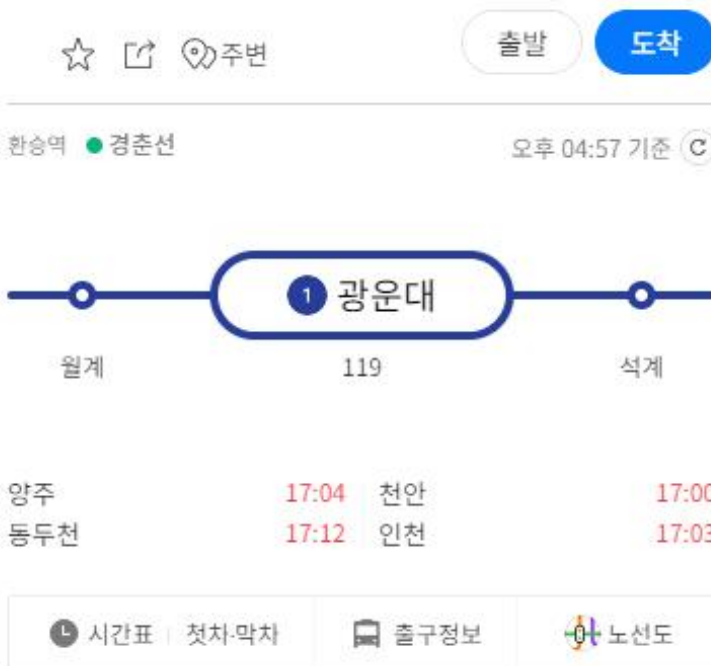
### A. 프로젝트 개요



이미 개발되어 서비스되고 있는 시내 버스와 좌석 버스의 정보이다.

### < 광운대역 1호선

수도권 지하철



그에 비해 지하철의 정보는 열차의 도착, 출발 시간 외에는 혼잡 정도에 대한 정보가 존재 하지 않는다.



열차 내부에서 사람이 혼잡하지 않을 경우에는 좌석도 비어있으며, 손잡이를 잡고 있는 이용객도 없다.

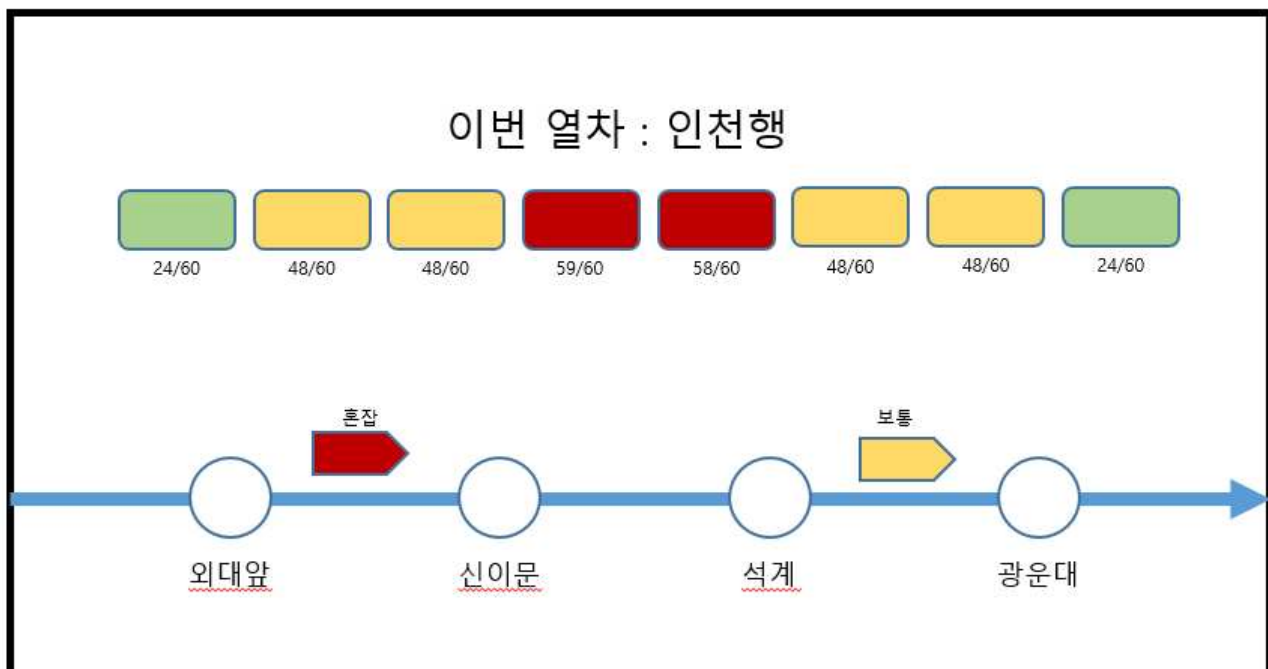


하지만 혼잡할 경우에는 좌석이 가장 먼저 가득 차며, 손잡이를 잡거나, 손잡이 밑에 서있는 사람들이 많아진다.

이를 지하철의 해당 열차를 탑승해야 알 수 있는 것이 아니라, 휴대폰 혹은 외부에서도 확인하게 하는 것이 이용자를 위한 편의라고 판단하였다. 지하철을 이용하다보면 어떤 칸은 혼잡스러운 데에 비해 또 어떤 칸은 쾌적한 경우를 볼 수가 있었다. 이런 혼잡한 정도의 정보를 이용자들에게 제공한다면 보다 많은 사람들이 쾌적하게 지하철을 이용할 수 있을 것 같다는 생각이 들었다.



-기존의 안내 화면 구성



-구현할 새로운 안내 화면 구성

구현된 프로그램이 외부로 데이터를 전송하기만 한다면, 안내 화면과 어플리케이션을 통해 사용자에게 정보를 제공하는 것은 문제가 없다.



## B. RTOS의 필요성

이동 수단에서 작동하는 시스템이기 때문에 전력 소모가 적어야하며, 실시간성이 가장 중요한 Hard RealTime System이기 때문에 RTOS를 이용해야하는 시스템이라고 생각했다. 다른 작업보다도 좌석의 상태 변화를 확인하는 것이 최우선이고, 이를 인터넷을 통해 외부로 전달하는 작업도 필요하다.

## C. 각 task의 정의

Task는 총 10개가 생성된다. 그 중에서 8개는 각 열차의 칸에서 좌석의 변화를 감지하는 역할을 담당하는 **Train\_1,2,,,,,8 task**로 좌석의 변화(event)가 발생하면 인터럽트가 발생하여 우선적으로 좌석의 변화를 수정한다. 또 하나는 Main Loop의 역할을 수행할 **Manager task**이다. 프로그램이 동작하는 동안 계속해서 좌석의 착좌 센서를 통해 좌석의 상태를 확인하고 사용 중인 좌석의 개수를 종합하는 역할을 수행한다. 마지막 하나는 데이터를 인터넷을 통해 외부로 전송하는 **Sender task** 이다. Manager에서 종합한 좌석의 개수를 인터넷을 통해 외부로 전송하는 역할을 수행한다.

## D. task간 semaphore와 message queue의 활용방안

Semaphore를 이용하여 작업하게 될 공유 메모리에는 열차에서 사용되고 있는 모든 좌석의 수를 기록하려고 한다. 각 열차의 칸마다의 event 처리 task에서 공유 메모리에 좌석 수를 수정할 때, semaphore를 사용하여 다른 task와 동시 수정이 발생하지 않도록 처리한다.

Message Queue를 이용하는 것은 Manager에서 종합한 칸 별 좌석의 개수와 열차 전체의 좌석의 개수를 Sender에게 보낼 때 사용한다.

## E. task 동작 결과

```
int main(void)
{
    OSInit();

    OSTaskCreate(Manager, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 10);

    OSStart();

    return 0;
}
```

- main 함수의 코드이다. Manager Task 하나만을 생성하여 OS를 Start 하였다.

```
void Manager(void *data)
{
    printf("Hi, This Task's name is MANAGER.\n");

    INT16U i;
    INT16U idx;

    for(i = 0; i < 24; i++)
    {
        idx = i % 3 + 1;

        switch(idx)
        {
            case 1:
                OSTaskCreate(Train_1, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 1);
                break;

            case 2:
                OSTaskCreate(Train_2, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 2);
                break;

            case 3:
                OSTaskCreate(Train_3, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 3);
                break;

            case 4:
                OSTaskCreate(Train_4, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 4);
                break;

            default:
                break;
        }
    }

    OSTaskCreate(Sender, (void *)0, &TaskStk[TASK_STK_SIZE - 1], 5);

    exit(0);
}
```

- Manager Task의 동작 코드이다. 각 열차의 좌석 개수를 체크하는 역할을 수행하는 Train Task들을 실행하도록 하였고 4번씩 실행하고 끝나면 Sender Task를 생성하여 실행하도록 하였다.

```
Microsoft(R) Program Maintenance Utility 버전 14.22.27905.0
Copyright (c) Microsoft Corporation. All rights reserved.

test.c
ucos_ii.c
pc.c
os_cpu_c.c
c:\software\ucos-ii\ports\80x86\win32\vc\src\os_cpu_c.c(120): warning C4477: 'prin
' 형식의 인수가 필요하지만 variadic 인수 1의 형식이 'OS_TCB *'입니다.
c:\software\ucos-ii\ports\80x86\win32\vc\src\os_cpu_c.c(123): warning C4477: 'prin
' 형식의 인수가 필요하지만 variadic 인수 1의 형식이 'OS_TCB *'입니다.
코드를 생성하고 있습니다...

C:\SOFTWARE\PROJECT>test.exe
Hi, This Task's name is MANAGER.
I'm No.1 of Train.
I'm No.2 of Train.
I'm No.3 of Train.
I'm No.4 of Train.
I'm No.1 of Train.
I'm No.2 of Train.
I'm No.3 of Train.
I'm No.4 of Train.
I'm No.1 of Train.
I'm No.2 of Train.
I'm No.3 of Train.
I'm No.4 of Train.
I'm Sender lol.
```

- 올바른 결과가 나온 것 같다. 간단하게 텍스트만을 출력하게 했음에도 처음으로 다뤄보는 환경에서의 코드 작성이라 어려움이 있었고, 배울점도 많았다.