



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 4기

2022. 12. 31

진동민

학습목표 & 7회차 날짜

학습목표

- 애자일 보드(칸반)를 이해하고 작성할 수 있음
- 백로그가 무엇인지 이해하고 작성할 수 있음
- 구조체 선언할때만 사용가능한 0 배열은 무엇인가

수업 날짜

2022-10-08 (토) 오후 6시~9시

목차

- 1) 업무를 가정한 문제
- 2) 애자일 보드
- 3) Backlog
- 4) 백로그 작성하는 방법
- 5) Sprint Term, In Progress, Review, Complete, Additional Work
- 6) 커밋과 백로그 연결하기
- 7) 결과물
- 8) 일정에 쫓기는 상황에서는 어떤식으로 관리가 필요한가
- 9) 0 배열
- 10) _의 의미
- 11) 참고 문서
- 12) 수업내용 사진

업무를 가정한 문제

문제은행 >

Lv1 업무 프로세스 베이스로 문제 풀어보기 [1]

링크뽐 카페메이저 1:1 채팅
2022.09.20. 12:13 조회 37

댓글 0 URL 복사

이 문제는 C언어 문제입니다.

그러므로 실제 FW를 연동할 필요는 없습니다.

아래 상황에 맞게 프로그램을 작성해주세요.

뿐만 아니라 확장성 및 유지보수성 향상을 위해 의존성을 분리하도록 합니다.

* 현재 A 장치와 B 장치가 존재합니다.

A 장치와 B 장치에서 데이터를 수집하는 데이터 수집 서버가 있습니다.

실제 서버였다면 A, B가 서로 경쟁을 하겠지만 무조건 A가 먼저, 그 다음에 B가 데이터를 주는 구조로 작성합니다.

문제를 풀면서 백로그를 작성하는 방법에 대해 학습할 예정입니다.

그러므로 A는 1 ~ 5 사이의 정수를 보내고

B는 2 ~ 4 사이의 정수를 보냅니다.

(이제 이렇게 적어봤다고 int A (void) { return rand() % (MAX - MIN + 1) + MIN; } 하는 사람은 없길 바랍니다)

실제 상황이었다면 프로토콜 기반으로 작업이 진행되지만

그러면 문제 풀이에 설명, 백로그 작성 등으로 너무나도 길어지니 축약했을 뿐입니다.

언제든지 더 복잡한 문제로 확장할 수 있도록 프로그램을 설계하는 훈련이기도 하므로 고찰이 필요합니다.

A, B에게 정보를 수신하였다면 이 정보를 가지고 작업을 처리해야 합니다.

그러나 위와 같은 이유로 단순히 A에서 받은 정보와 B에서 받은 정보를 곁해서 지속적으로 누산하겠습니다.

(실제 상황이었다면 스레드, 자료구조가 모두 활용되었겠지만 여기서는 그렇게 하지 않기 위해 제약을 걸었습니다)

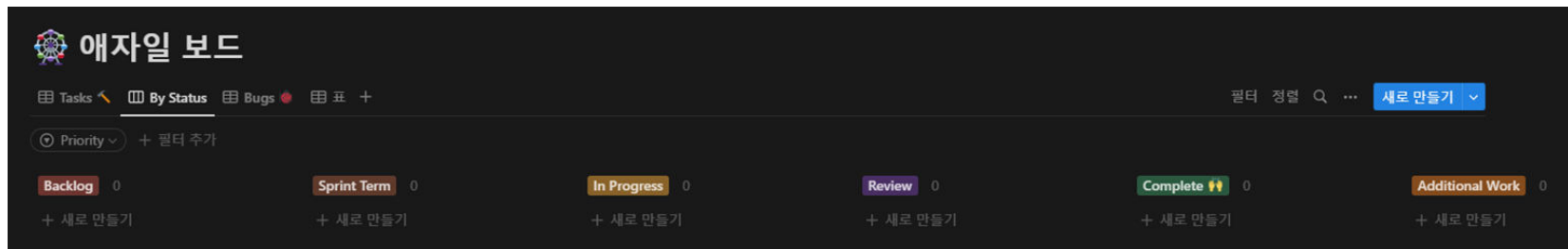
10번 누산한 결과를 출력하도록 프로그램을 작성하세요.

(의존성 분리, 백로그 작성 진행하면서 풀어보세요) <--- **꼭포수 설계로 풀지 말고요.**

왼쪽 그림에 명시된 상황의 문제를 애자일 프로세스에 기반한 칸반 및 애자일 보드를 이용하여 백로그를 작성하며 프로그래밍하는 방법을 수업에서 배웠다.

문제 링크: <https://cafe.naver.com/eddicorp/974>

애자일 보드



학원에서 사용하는 애자일 보드는 위와 같다.

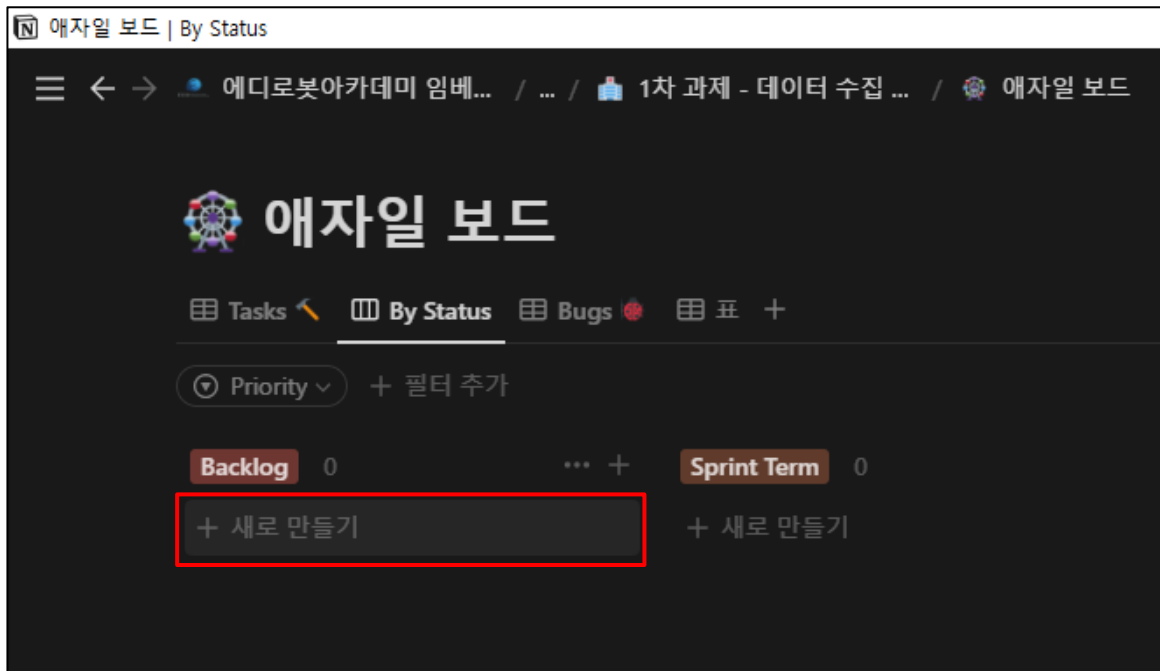
애자일 보드를 사용하여 문제를 풀기 전에 이미지에 보이는 각 용어들을 알아보자

- Backlog란
 - Backlog라는 부분은 사실 브레인스토밍이라고 봐도 무방하다.
 - 아이디어성 사항들을 이 부분에 배치하면 된다.
 - 근데 아이디어성이라는 것이 전체 Epic(다루려는 제품)을 기준으로한 아이디어들이라고 보면 된다.
- Backlog 내부에 배치할 정보
 - 아이디어니까 굉장히 상세한 스펙을 기술할 필요는 없다.
 - 물론 작업을 진행하기 위한 최소한의 요구사항을 정리할 필요는 없지만 이것이 굉장히 정확할 필요가 없다는 뜻이다.

백로그를 작성하는 방법은 다음 장에서 보자

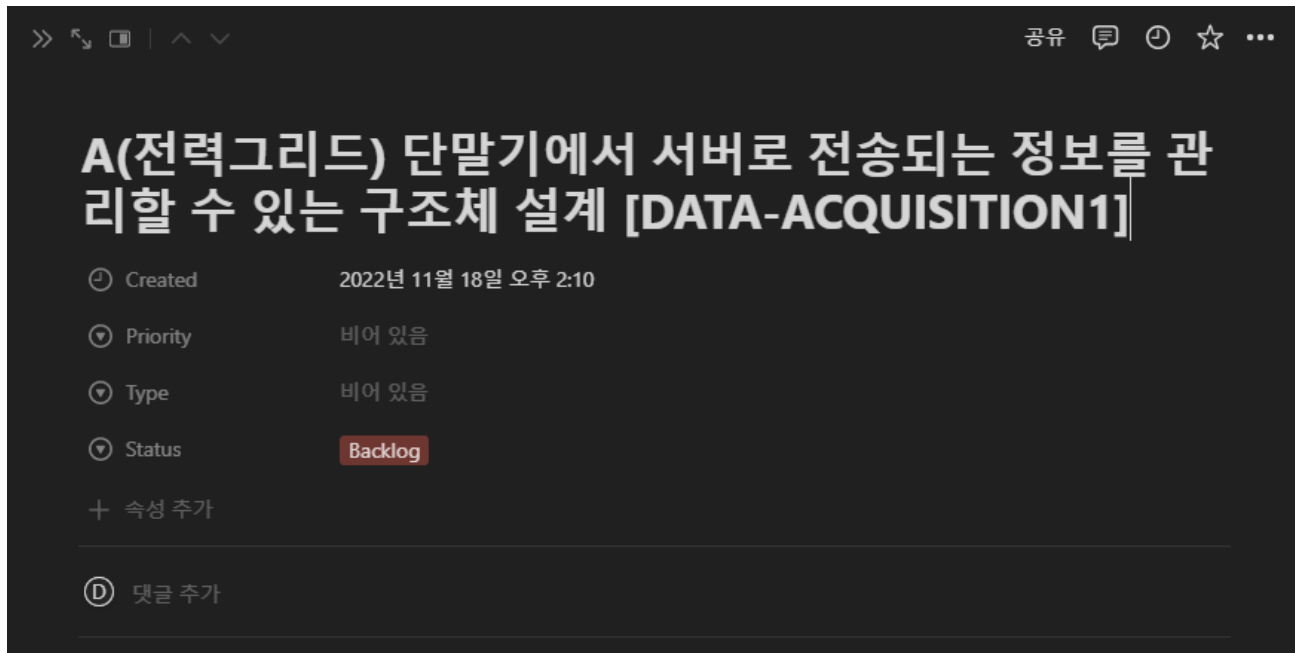
백로그 작성하는 방법

- 백로그를 만들기 위해 '새로 만들기' 버튼을 누른다.



백로그 작성하는 방법

- 백로그는 사용자 스토리와 어떤 목적으로 무엇을 하는지에 대해 작성되어야 한다.
- 백로그 이름은 다른 사람이 보더라도 이해할 수 있도록 명확하게 작성해야 한다.
- 또한, 백로그 태그는 백로그 이름 뒤에 대괄호를 이용하여 에픽 이름과 작성하고 있는 백로그가 몇 번째인지를 작성한다.

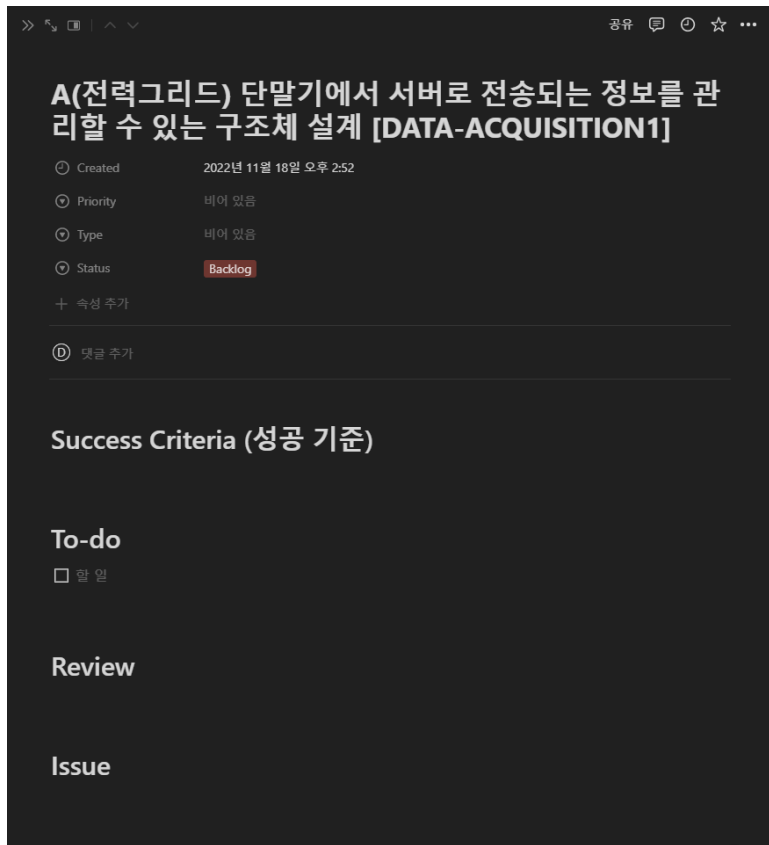


백로그 작성하는 방법

- 백로그 내부는 다음과 같다.
 - Success Criteria (성공 기준)
 - To-do
 - Review
 - Issue
- Success Criteria
 - 잘 마무리되었음을 판정하는 기준을 작성
- To-do
 - 백로그를 진행하기 위한 To-do 목록
- Review
- 프로세스 차원에서 코드 리뷰를 작성
- Issue
 - 발생한 문제를 기록

(참고)

<https://cafe.naver.com/eddicorp/839>



백로그 작성하는 방법

Success Criteria (성공 기준)

1. A에서 넘어오는 정보는 전력 그리드에서 서버로 전송하는 정보를 파악합니다.
2. 파악한 정보를 토대로 전력 그리드의 전송 정보를 관리할 구조체를 작성합니다.

To-do

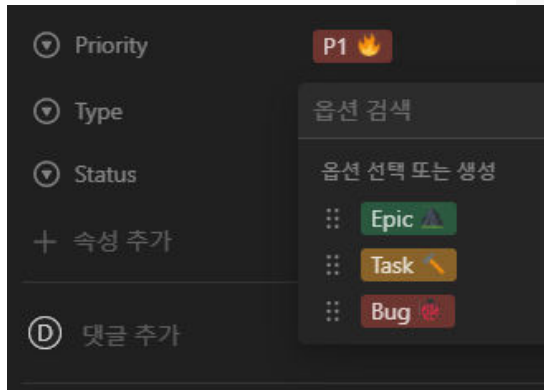
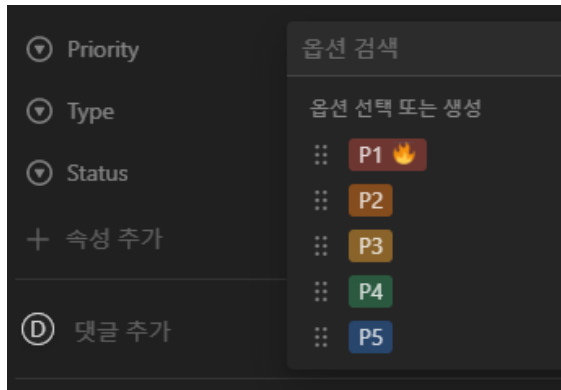
- ☐ 전력 그리드에서 서버로 전송되는 정보 파악
- ☐ 전력 그리드의 전송 정보 관리할 구조체 작성
- ☐ 이 구조체를 할당하는 함수
- ☐ 이 구조체를 해제하는 함수

(참고사항)

1. 전력 그리드에서는 전체 사용 전력 정보, 배터리 셀별 전압 정보 같은 것들이 전송됨.
(가정: 이 값의 평균값은 1 ~ 5이며 이 값을 전송하고 있음)
2. 원래라면 위의 모든 정보들이 구조체 내부에서 관리되어야 함
우리는 가정을 하였기 때문에 사실상 정수형 변수 1개가 됨.

Review

Issue



- Priority로 우선순위를 설정할 수 있다.
- 여기서는 새로운 기능을 추가하는 백로그이므로 Type은 Task로 택했다.

내부 내용을 채워넣은 백로그

(참고) Spint Term

- Sprint Term이란
 - 여기에는 스프린트 기간동안에 진행할 사항을 Backlog에서 땡겨와서 배치하는 구간이다. (주간 일정)
 - 하지만 회사가 아닌 학원이므로 사용하지 않는다.

In Progress

- In Progress란
 - Sprint Term은 1주라는 기간이 정해져 있는 것이고, In Progress는 지금 뭐하고 있는지에 해당한다.
 - 업무 처리가 빠른 사람들은 Sprint Term에 많은 양이 있고 In Progress가 자주 바뀐다.

- Review란
 - Review는 In Progress에 있었던 백로그의 작업이 끝나면 다음으로 배치되는 구간으로 현재 담당자(들)의 작업을 리뷰하는 부분이다.

Complete

- Complete란
 - Complete는 작업이 완료된 사항을 배치한다.
 - 작업이 완료되었다는 것은 실행부터 테스트까지 완벽하게 진행이 완료된 사항을 의미한다.
 - 반드시 돌아가는 코드 형식으로 완성된 상태를 Complete에 배치하도록 한다.
 - Git 으로 관리하니까 사실 큰 문제는 없을 것으로 보인다.

Additional Work

- Additional Work란
 - Sprint Term에 배치하였지만 작업을 완료하지 못했을 경우 배치하는 구간이다.
 - 스크럼 회의 이후에는 Sprint Term 혹은 In Progress로 이동시키도록 한다.
 - 만약 당장 급하지 않다면 그냥 여기에 쌓아놓고 일정여유가 있을 때 처리하도록 한다.
 - 즉, 내가 혹은 단체가 하는 작업에서 어떤 것의 우선순위가 더 높은지 결정할 필요가 있다.

커밋과 백로그 연동하기

- 작업이 끝난 백로그를 Complete에 배치하고 나면, 작성한 코드를 커밋하면 된다.
- 그리고 백로그 이름을 커밋 메시지에 포함하여 커밋한다.
 - 예) 앞에서 작성한 백로그를 기준으로 커밋한다면, 다음과 같다
 - `git commit -am "feat: A(전력그리드) 단말기에서 서버로 전송되는 정보를 관리할 수 있는 구조체 설계 [DATA-ACQUISITION1]"`
 - 커밋 메시지 맨 앞에 붙는 것은 커밋 메시지 타입이다. (참고: <https://cafe.naver.com/eddicorp/1169>)

결과물

- 위치는 Lv1 프로젝트 (4기) / 과제 관리 게시판 / 진동민 / 1차 과제에 있다.

Complete 🙌 5

📄 A(전력그리드) 단말기에서 서버로 전송되는 정보를 관리할 수 있는 구조체 설계 [DATA-ACQUISITION1]

P4

📄 B(공기조화) 단말기에서 서버로 전송되는 정보를 관리할 수 있는 구조체 설계 [DATA-ACQUISITION2]

P4

📄 A에서 서버로 전송한 정보를 수신하는 스레드 설계 [DATA-ACQUISITION3]

P4

📄 B에서 서버로 전송한 정보를 수신하는 스레드 설계 [DATA-ACQUISITION4]

P4

📄 A와 B로부터 수신한 정보들을 가지고 작업을 처리하는 메커니즘 설계 [DATA-ACQUISITION5]

P4

일정에 쫓기는 상황에서는 어떤식으로 관리가 필요한가



EDDI
Electronic Design
Development Institute

- 예를 들어 제품 출시가 임박한 상황이라면, 애자일 보드에서 가장 중요한 백로그를 Sprint Term이나 In Progress에 배치하고 그 외의 것들은 Additional Work에 배치한다.

- 만약 A(여기서는 전력그리드) 단말기에서 서버로 전송되는 정보 중에 여러 배터리 전압을 저장해야한다면 구조체를 다음과 같이 작성해야 한다.

```
1 typedef struct _power_grid_info power_grid_info;
2
3 struct _power_grid_info
4 {
5     float power;
6     int battery_max;
7     float *battery_cell_volatage;
8 };
```

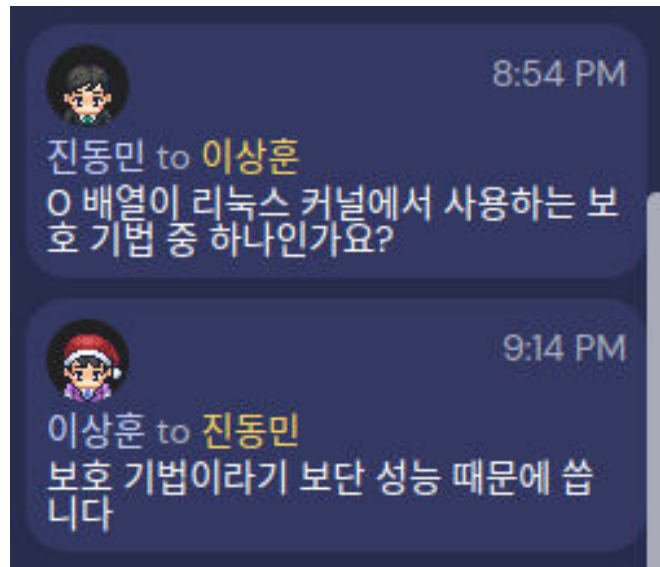
- 위 코드의 경우** 배열에 접근하기 위해 포인터 변수를 사용하는데 이 배열 길이를 알기 위해 int형 변수를 하나 더 선언해야한다는 점이다.
- 이를 해결하기 위해 **0 배열**이 탄생했다. 다음과 같이 선언하여 사용한다. 0 배열을 사용하려면 malloc으로 할당하면 된다.

```
1 typedef struct _power_grid_info power_grid_info;
2
3 struct _power_grid_info
4 {
5     float power;
6     float battery_cell_volatage[0];
7 };
```

- 0 배열은 변수 하나를 선언하고 이 변수를 가리키는 **포인터 변수**이다. (아래 그림 참고)

battery_cell_voltage → 
(float)

0 배열



_의 의미

코딩할 때 이름 앞에 붙는 _의 의미가 무엇인지 궁금해서 물어보았다.

_는 내부 기능이므로 고치려면 회의를 해야한다고 한다. 그러니 앵간하면 건들지 말라는 뜻이다.

참고 문서

구글에 '0배열' 검색

<https://hydroponicglass.tistory.com/302>

- <https://stackoverflow.com/questions/295027/array-of-zero-length>

구글에 'sizeof c-hack' 검색

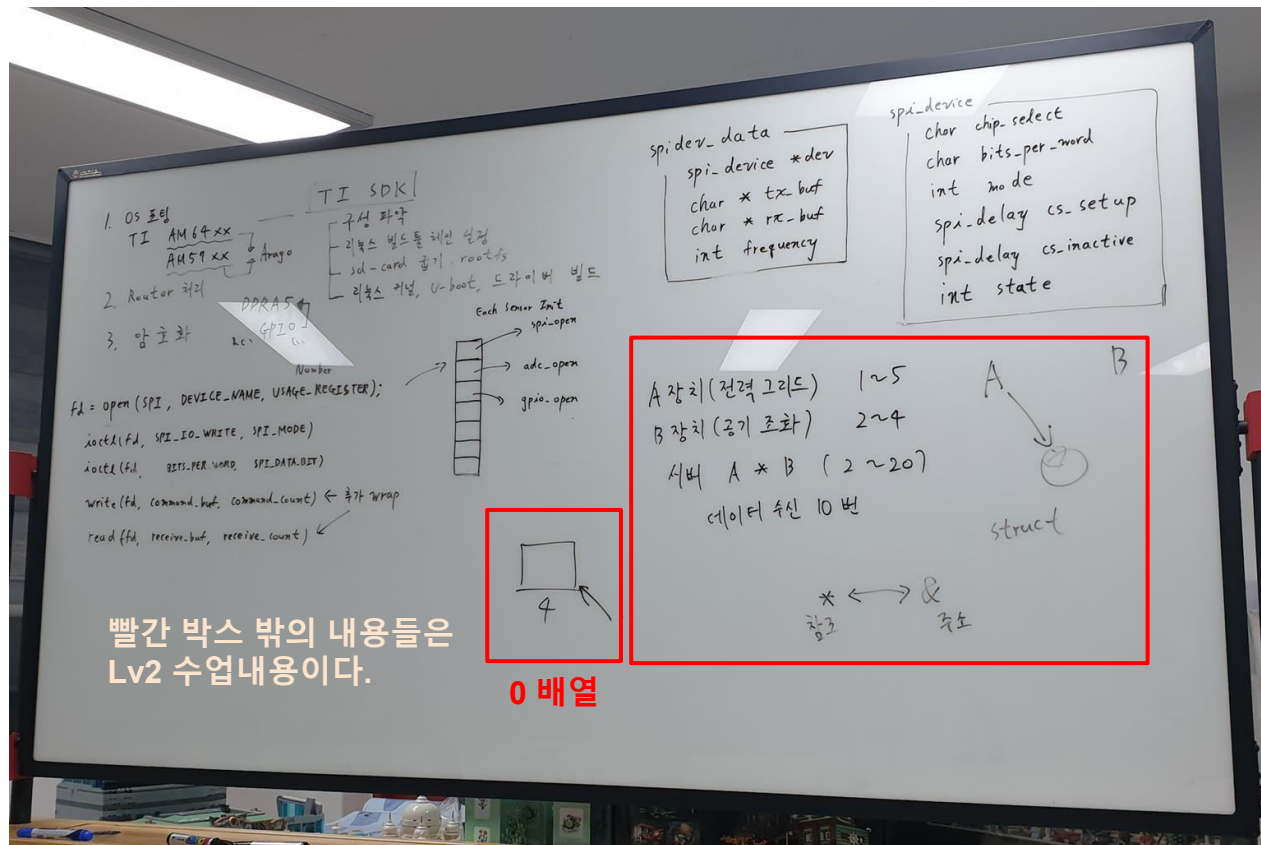
<https://stackoverflow.com/questions/295027/array-of-zero-length>

<https://gist.github.com/manjuraj/1297075>

gcc 공식 문서

<https://gcc.gnu.org/onlinedocs/gcc/Zero-Length.html>

수업내용 사진



CONTENTS

1) 형식은 자유롭게~~~

<공부 내용을 적어주세요.>