< 프로그램 제출시 의무 작성 사항 >

(1) 프로그램 상단에 다음과 같은 주석을 작성함

```
////////////////////////////////////////////////////////////
// PR1: 자동차 추종 시스템
// 제출자: 20XX13XXXX 홍길동
// 주요 내용
//  -거리센서: ADC2_IN1(PA1) 이용
//  - 추적차 엔진 : XXXXXXXXXXXXXXXXXXXXXXX
//  - YYYYY : ZZZZZZZZZZZZ
////////////////////////////////////////////////////////////
```

(2) 주석

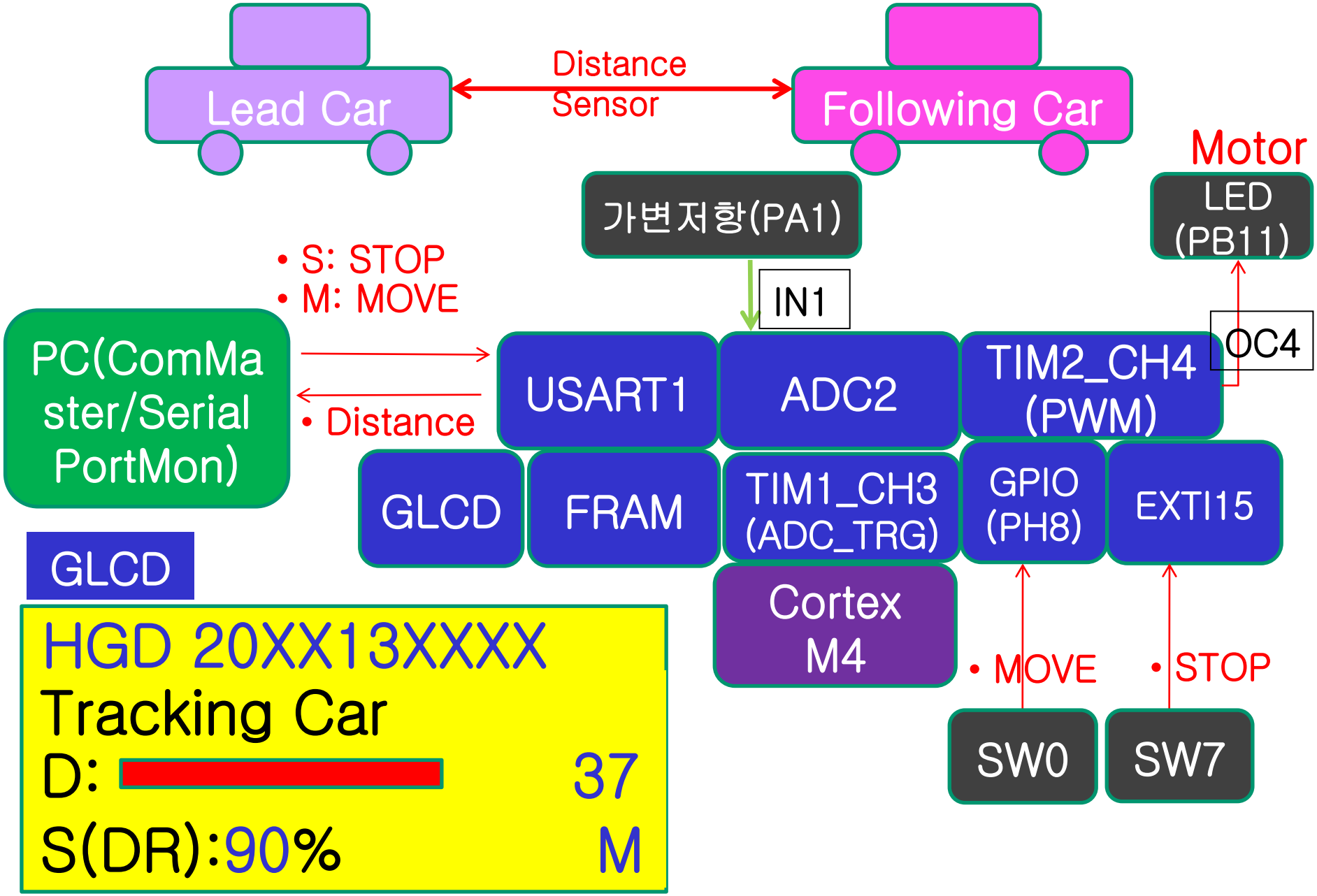
-학생이 새롭게 추가한 문장에 주석 기재

-실습시간에 배포된 프로그램의 문장을 수정하면 해당 주석도 변경

(3) 제출 파일명:

- 화요일반: 1_PR1_20XX13XXXX_홍길동.c
- 수요일반: 2_PR1_20XX13XXXX_홍길동.c
- 목요일반: 3_PR1_20XX13XXXX_홍길동.c

TermPr-1: 자동차 추종 시스템 (1/6)



(2/6)

● 개요

• 자동차 2대가 길을 따라 운행하고 있음. 선도차(Lead car)는 random하게 운행하고, 추종차(Following car)는 선도차와의 거리를 계속 측정하면서 속도를 조절하며 선도차를 따라가야 함. 추종차 시스템을 프로그래밍 해야 함.

• 주요 요소

–거리센서: ADC2_IN1(PA1)

–추종차 엔진: TIM2_CH4(PWM)(PB11, LED) * LED: 추가 부착한 LED

–추종차 시동: Move-key(SW0(GPIO, PH8)), Stop-key (SW7(EXTI15))

–원격(PC 통신프로그램) 추종차 시동: Move-key('M'), Stop-key('S')

–거리값 표시: GLCD(D), PC

–추종차 속도 표시: GLCD(V)

● GLCD 화면설명

• 3rd line: D(거리)는 선도차와 추종차 간 거리(거리센서 측정값), '37'은 거리값

• 4th line: S(속도)는 DR(듀티비) 표시, 'M'은 추종차 운행상태 ('M' or 'S')

(3/6) 선도차 거리 측정

● 선도차와의 거리측정

- 가변저항(ADC2_IN1(PA1))의 변경이 거리의 변경을 의미.
- 측정주기: 300ms(TIM1_CH3(PE13)(CC 이벤트)로 start trigger 신호발생)
- ADC EOC 인터럽트를 사용함
- 인터럽트 핸들러에서 획득한 ADC 결과값을 전압으로 환산하고 이를 거리로 환산함.
- 거리와 전압과의 관계: 거리(정수형)=전압*10 + 5 (예:0.0V일때 5m, 2.5V일때 30m)
- GLCD 3rd line 끝에 거리 표시(예: 5m: '5', 37m: '37' 표시)
- GLCD 3rd line에 거리에 비례하는 길이의 막대표시 (5~38m)(그림 참조)

예) 거리=5m 경우: 가장 짧은 막대 ……

거리=38m 경우: 가장 긴 막대

- PC(ComMaster or SerialPortMon)에 300ms마다 거리값 전송(전송포맷: 37m 일때 "37m " 마지막에 스페이스 추가 또는 라인변경문자(line feed+carriage return) 도 가능)

(4/6) 추종차 속도 제어

● 추종차는 거리에 따라 속도를 변경함(TIM2_CH4(PB11), PWM)

- 측정거리에 따라 DR을 결정함

  측정거리 5~8m이면 DR=10%,

  측정거리 9~12m이면 DR=20%, (13~16),……(33~36),

  측정거리 37~38m이면 DR=90%

- TIM2_CH4(PB11)를 이용해 PWM 신호 발생: 주파수 0.2Hz(주기:5sec), 분주비 8400

  (예) DR=10%이면 High 구간 0.5s, Low구간 4.5s, 총 5s PWM 파형 발생

-실제로 모터 구동 않음. LED(PB11) 상태로 PWM 신호가 잘 발생되는지 확인

-GLCD 4th line에 DR값 표시

(5/6) 자동차 출발 / 정지 명령

(1) PC에서의 명령: MOVE: 'M' or STOP: 'S'

• 위 명령을 수신 받으면 4th line 끝에 'M'이나 'S'를 표시

• STOP 상태: PWM DR=00%(막대 없음, PWM 출력신호 LOW 상태로 중단), MOVE 상태: 측정거리에 따른 DR값 표기 및 PWM 구동

(2) key 입력으로의 명령: MOVE: 'M' or STOP: 'S'

• MOVE Key: SW0(GPIO PH8), STOP key: SW7(EXTI15)

• 위 명령 입력되면 4th line 끝에 'M'이나 'S'를 표시

• STOP 상태: PWM DR=00%(막대 없음, PWM 출력신호 LOW 상태로 중단), MOVE 상태: 측정거리에 따른 DR값 표기 및 PWM 구동

● 시스템 초기상태

• DR= 0%, 거리값: '0', 운행상태: STOP, PC상의 거리표시 :0m

HGD 20XX13XXXX
Tracking Car
D: 0
S(DR):00% S

(6/6)

● 리셋후 부팅되면 리셋전 상태 유지: FRAM 이용('S'나 'M'을 FRAM (1126번지)에 저장)

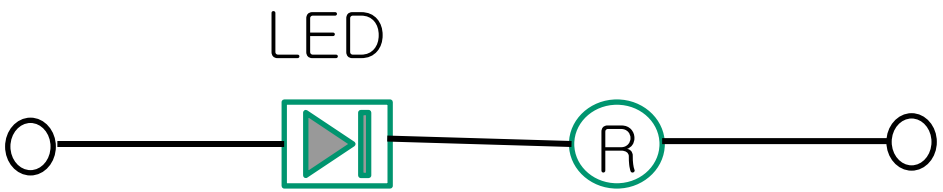
- 리셋전 상태가 'STOP' 상태이면, 리셋후 부팅시 FRAM에서 주행상태(S or M)를 로딩하여 다음과 같이 표시하고 자동차 상태도 'STOP' 상태로 셋업

HGD 20XX13XXXX
Tracking Car
D: 0
S(DR):00% S

- 리셋전 상태가 'MOVE' 상태이면, 리셋후 부팅시 FRAM에서 주행상태(S or M)를 로딩하여 다음과 같이 표시하고 자동차 상태도 'MOVE' 상태로 셋업
  또한, 현재의 거리를 측정해서 표시하고 그에 따른 속도도 표시함. PWM도 발생시켜 LED에 표시되도록 함

HGD 20XX13XXXX
Tracking Car
D: 37
S(DR):90% M

* 참조: LED 제작 및 KIT 연결 방법



+3.3V (40핀 헤더박스의 1번핀)

220~330 Ω

PB11(40핀 헤더박스의 32번핀)