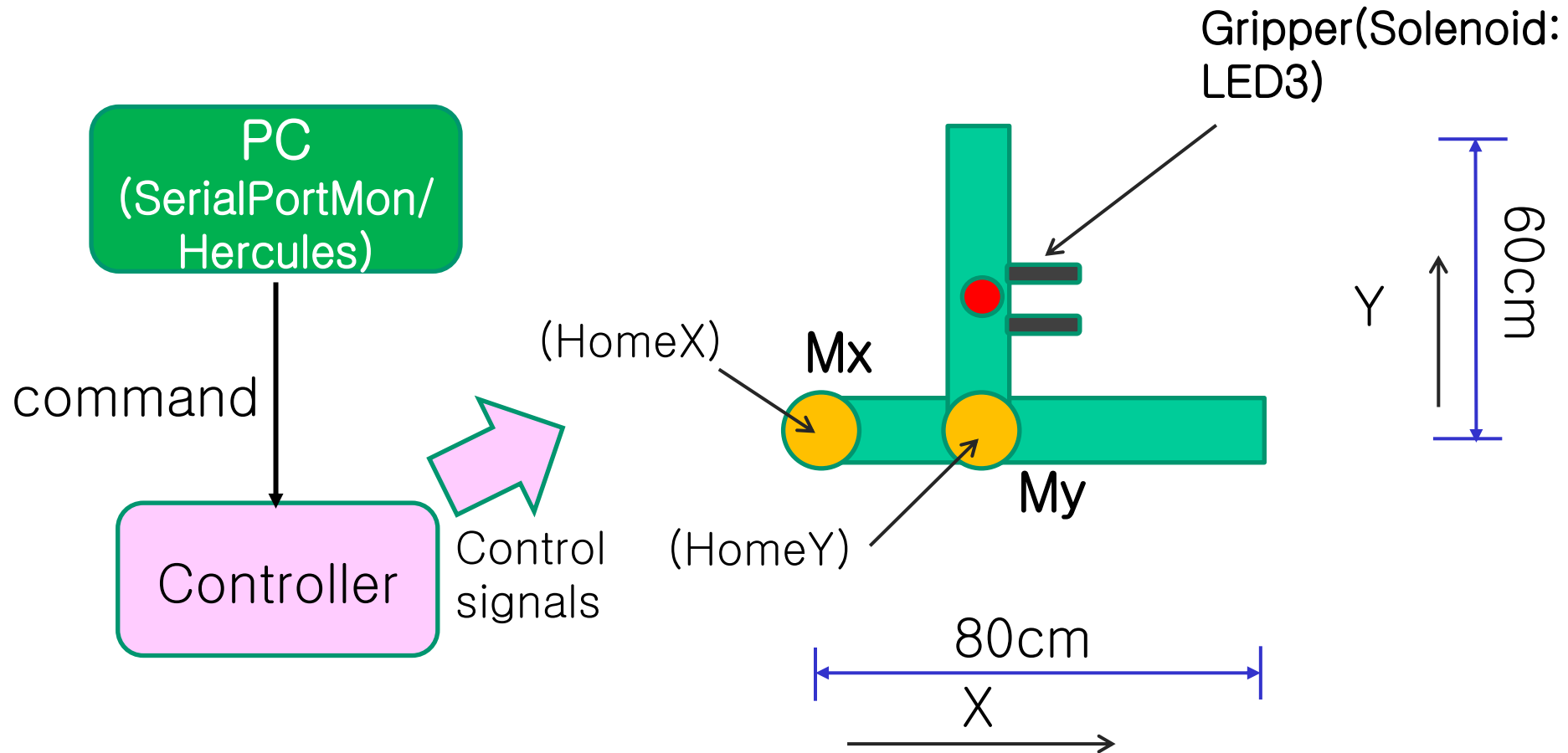


PR2. X-Y Robot Control

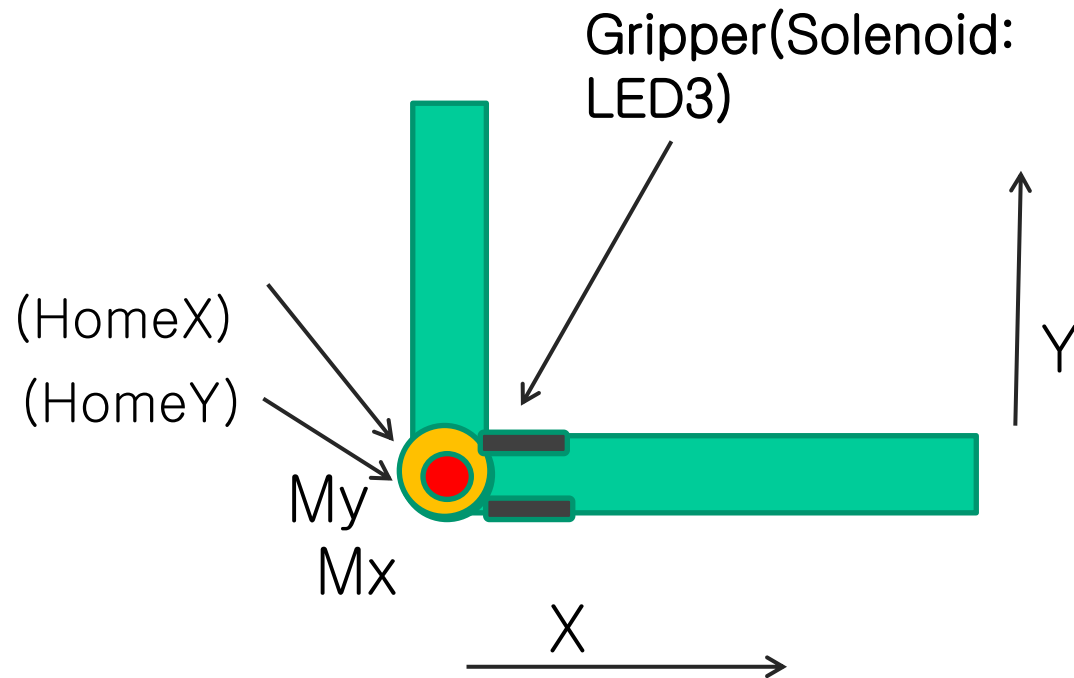
- 개요: PC에서 원격으로 보낸 Robot 동작명령(command)을 받아, 해석하고 command에 따라 로봇 동작용 제어 신호를 발생



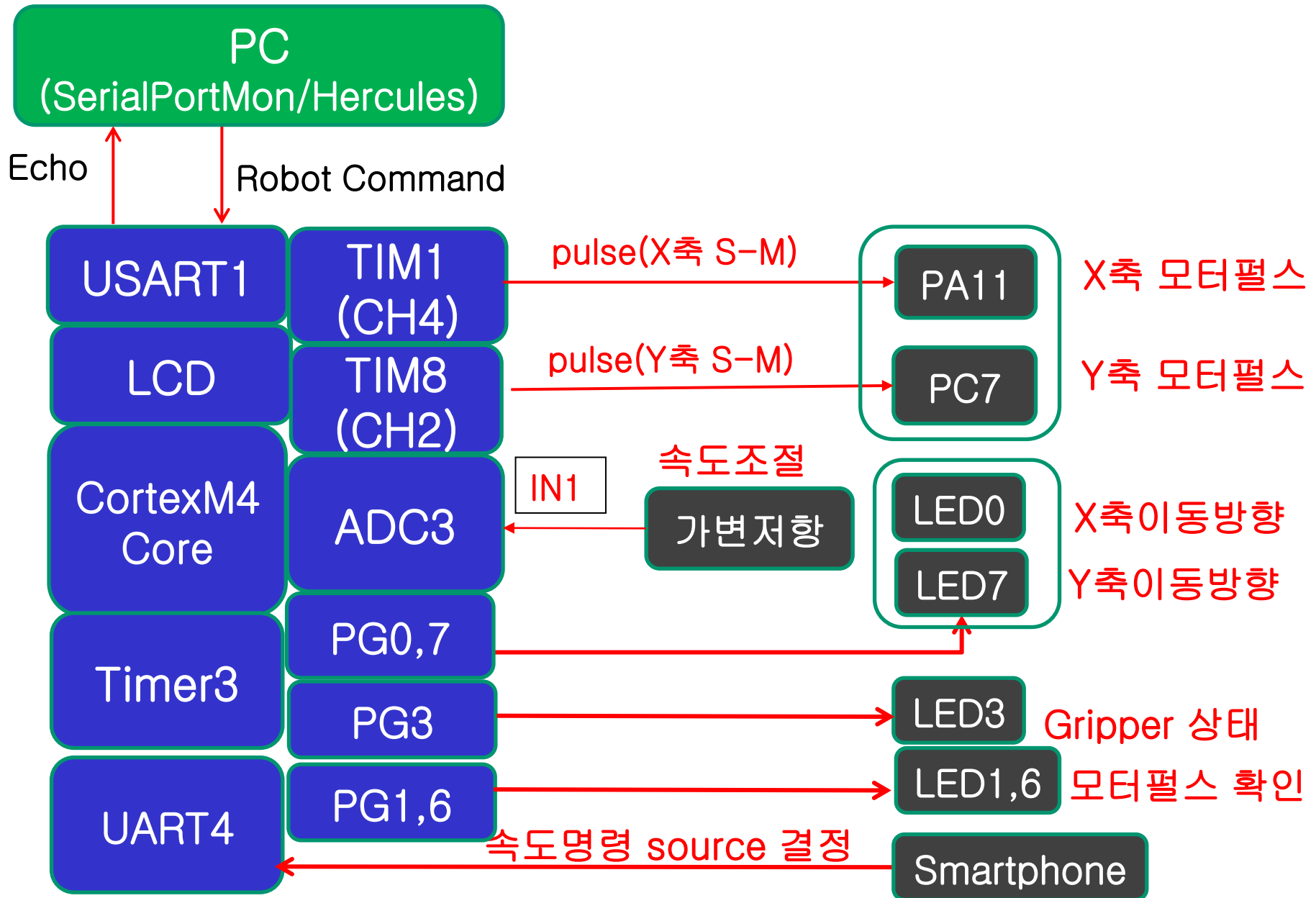
*Mx: X축 모터, My: Y축 모터

● : Gripper 위치

● 초기 위치(Reset 후 home 위치)



● H/W 구성도



● 화면구성

X-Y Robot Control : HGD

SD=2, RPM=3(PC)

GX=00, GY=00, DX= 00, DY=-00

TPX=00, TPY=00

CX=00, CY=00

CPX=00, CPY=00

CX 

CY 

*화면상 숫자색 의미: 파란색 숫자는 명령어수신때만 변경, 빨간색 숫자는 로봇동작하는 동안 계속 변경되는 수(글씨크기: 7pt)

*리셋 초기값:

-SD=2, RPM=3

-이외의 모든 값 '0',

-막대그래프도 표시없음
(CX,CY=0 이므로)

- SD: Step Deg(X,Y축 동일값), RPM: 속도(X,Y축 동일적용)
- GX,GY(목표 그리퍼 위치(좌표)) : PC로부터 수신후 표시
- DX,DY(목표거리(G-C)) : GX,GY로부터 계산 및 표시,
부호표시(DX=-?? :CCW, DY=+??: CW)
- TPX,TPY(목표 총 펄스수) : DX,DY 로부터 계산 및 표시
- CX,CY(현재 그리퍼 위치(좌표)) : 로봇 이동중 계속 업데이트(10cm 단위로)
- CPX,CPY(하나의 이동명령 실행시작부터 발생되고 있는 현재시점의 펄스수: 0~TPX(TPY)) : 로봇 이동중 계속 업데이트
- 막대 그래프: 로봇 이동중 계속 크기 변경(10cm 단위로)

● Robot Command

:PC에서 송신하는 로봇 작동 명령어(command, 2 문자의 string)로서, 프로그램에서는 명령어 문자열을 수신하고 해독하여 로봇 동작 실행

• Command 종류

- 축 이동명령: **Xn, Yn** (n: 그리퍼의 절대좌표값(n*10cm))
- Gripping 명령: **Gn** (n=0: Open, n=1: Closed)
- 속도(RPM) 명령: **Rn** (n: 1~4 rpm)
- **Xn** :그리퍼를 현재위치에서 X축 좌표 n(*10cm) 위치로 이동하는 명령(n:0~8)
 - 10*n 을 GX(목표위치) 변수에 저장, $GX = 10 * n$ ($0 \leq GX \leq 80$)
 - X목표위치(GX)-X현재위치(CX) = X이동거리(DX)를 계산하여 스텝모터 구동
 - if $DX < 0$, X축 모터를 CCW 방향(LED0 off)으로 DX만큼 이동하는 펄스 발생
 - if $DX \geq 0$, X축 모터를 CW 방향(LED0 on)으로 DX만큼 이동하는 펄스 발생
- **Yn** :그리퍼를 현재위치에서 Y축 좌표 n(*10cm) 위치로 이동하는 명령(n:0~6)
 - 10*n 을 GY(목표위치) 변수에 저장, $GY = 10 * n$ ($0 \leq GY \leq 60$)
 - Y목표위치(GY)-Y현재위치(CY) = Y이동거리(DY)를 계산하여 스텝모터 구동
 - if $DY < 0$, Y축 모터를 CCW 방향(LED7 off)으로 DY만큼 이동하는 펄스 발생
 - if $DY \geq 0$, Y축 모터를 CW 방향(LED7 on)으로 DY만큼 이동하는 펄스 발생
- **G0** : Gripper Open : LED3 OFF
- **G1** : Gripper Closed : LED3 ON
- **Rn** : 모터의 RPM(X,Y축 동일) 변경, $RPM = n$ (n:1~4)

● 스텝모터 제어 요소 규격

- SD: Step deg(deg/pulse) (*SD=2로 고정)
(예) SD=2(deg/pulse)이면, 1pulse를 입력하면 2도 회전
* 모터 1도 회전당 로봇 팔은 1cm 직선거리 이동(즉, 1pulse 입력시 2cm 이동)
- RPM: 모터의 속도(rev/60min) *RPM=n
- 목표 이동거리: DX($0 \leq Dx \leq 80$), DY($0 \leq Dy \leq 60$)
*DX=GX(목표위치(좌표)-CX(현재위치(좌표))),
*DY=GY(목표위치(좌표)-CY(현재위치(좌표)))
- 목표거리(DX,DY)를 목표펄스수(TPX,TPY)로 변경
-X축 목표펄스수(X축 스텝모터 구동용 총 발생 펄스 수):
*TPX= DX/S ($0 \leq TPX \leq 40$)
-Y축 목표펄스수(Y축 스텝모터 구동용 총 발생 펄스 수):
*TPY= DY/S ($0 \leq TPY \leq 30$)
- CPX(현재까지 발생된 X축 펄스수): pin을 통해 발생하는 실시간 X축 펄스 수
CPY(현재까지 발생된 Y축 펄스수): pin을 통해 발생하는 실시간 Y축 펄스 수
- 요약: 목표좌표값(GX,GY) 명령을 받으면, 목표거리(DX,DY)를 계산하고, 목표 RPM을 적용하여 펄스주기를 계산하고, 목표펄스수(TPX,TPY)를 계산함. 로봇 동작중 현재 펄스수(CPX,CPY) 값이 표시되고, 해당 핀을 통해 펄스가 출력되어 모터가 회전(모터회전 표시: LED1(X),LED6(Y) 점멸, 막대그래프변경)하고, 목표위치까지 도착하면 작동 중단

● 스텝모터 구동 펄스 발생 방법

- TIM1_CH4(PA11-DAC1-J9(CON40-11))에서 X pulse 발생
- TIM8_CH2(PC7-USART6_RX-J9(CON40-9))에서 Y pulse 발생
- 펄스발생은 확장커넥터의 핀에 오실로스코프나 LED를 연결하여 확인
- TIM1_CH4, TIM8_CH2 를 CC 이벤트 발생모드로 하고, 외부핀(PA11, PC7)으로 펄스 발생하도록 하고, CCI 인터럽트도 발생하도록 함
- 발생하는 펄스 주기는 다음과 같이 결정
 - CC 인터럽트 주기: $(60 \cdot SD / 2) / (360 \cdot RPM)$
- CC 인터럽트 핸들러루틴에서 펄스 수를 카운팅하여 펄스 수가 TPX,TPY개가 되면 발생을 중단. 즉, 인터럽트발생(핸들러 실행)때마다 카운팅변수로 인터럽트발생 건수를 카운팅하여 TPX,TPY개의 펄스가 되면 펄스 발생을 중단(이 클래스에 있는 실습 프로그램 참조)
- CC 인터럽트 핸들러루틴안에서 LED1(X), LED6(Y)을 toggle 함. 즉, 핀(PA11, PC7)을 통해 출력되는 펄스와 같은 펄스가 LED1,LED6에 출력되도록 함.
- CC 인터럽트 핸들러루틴이 실행될 때마다 막대 그래프도 크기 변경이 되도록 함

- PC에서의 작업 순서
 - PC(SerialPortMon/Hercules)에서 typing하여 command를 한 문장씩(예: “Xn”) 전송
 - MCU는 command를 수신하면서 한 문자씩 받을 때마다 PC로 받은 문자를 Echo
 - Command를 해독하고 로봇 동작 실행함
 - SerialPortMon/Hercules 통신 protocol: UART1, 9600 bps, 1 stop bit, Odd parity
- 막대 그래프 표시
 - CX,CY 값에 비례하여 표시(한번의 로봇 이동 종료후 변경)
 - CX=0, CY=0 : 막대 그래프 없음
 - CX=80, CY=60: 막대 그래프 최대(막대가 80cm일때 LCD 상 우측에 거의 도달하도록 크기 결정)
- LCD상에 표시되는 변수의 초기값, 변경 및 표시 방법
 - GX,GY,DX,DY,TPX,TPY: PC 명령어 수신때마다 변경하여 표시(Reset값:0)
 - CX,CY: 로봇 이동중 현재 위치 계산하여 표시. (Reset값:0)
 - CPX,CPY: 로봇 이동중 출력되는 펄스수가 증가할 때마다 증가된 값 표시,
이동시작시에는 ‘0’ 부터 증가 (TPX,TPY값까지 증가) (Reset값:0)

- 목표 RPM(목표 속도) 입력 2가지 방법
 - (1) PC에서의 속도명령
 - (2) 마이컴키트의 가변저항 입력
- 2가지 방법중 1가지를 선택하는 방법
 - :스마트폰(Bluetooth연결, UART4(9600bps, 1stop bit, no parity))에서 송신한 값에 의해 선택
 - ‘0’ 송신: PC에서 RPM 명령(“Rn”) 송신, LCD에 “RPM=3(PC)” 표시
 - ‘1’ 송신: 가변저항(ADC)으로 RPM 결정, LCD에 “RPM=3(VR)” 표시
 - * Reset후 초기선택은 PC 송신
 - * PC가 선택되면 가변저항 변경해도 RPM값 변경안됨
가변저항이 선택되면 PC에서 RPM 명령 송신해도 적용안됨
- 가변저항(ADC3 CH1) 변경에 따른 속도값 결정(소수점 한자리까지 계산)
- 다음과 같은 관계로 부터 PRM값을 구함 : 0.0~3.3V ➔ 1~4rpm
(자세히, 0.0~0.8V ➔ 1rpm, 0.9~1.6V ➔ 2rpm,
1.7~2.4V ➔ 3rpm, 2.5~3.3V ➔ 4rpm)
- RPM값을 LCD에 표시(“RPM=3(VR)”)하고, X,Y축 스텝모터 펄스발생에 동일하게 적용함
- ADC3는 SWSTART, Timer3 Overflow Int(주기500ms)마다 ADC start명령, 반드시 ADC 인터럽트 사용하여 측정

제출파일이름: N_PR2_2018xxxxxx_홍길동 (N: 화-1, 수-2, 목-3)

- main만 변경한 경우 : N_PR2_2018xxxxxx_홍길동.c

- 파일의 첫부분에 다음을 반드시 기재

```
//*****
// PR2: 제목 .....
// 제출자: 2018xxxxxx 홍길동
// 제출일: 2022.12.xx
// 과제개요: .....

```

- * 주석을 충분히 자세하게 기재 (본인이 작성한 부분만...
 - 수업시간에 준 파일의 문장을 수정했을 경우에는 주석도 수정할 것)
- * 들여쓰기 할 것