미래자동차 모델기반 SW 캠프

TC275를 활용한 RC카 프로젝트

Team Detail guys

kyungshin

kyungshin

1. Project Background & Task Allocation

kyungshin

01. 프로젝트 소개

1) 팀 소개

강성민

개발, 서류

신재환

개발팀장

전지환

개발, 서류

kyungshin

01. 프로젝트 개요

1. 프로젝트 목표

High Tec 툴을 활용한 핸드 코딩을 하드웨어와 직접적으로 밀접한 코드를 짜는 연습을 한다.

Uart0통신을 통해 블루투스에서 TC275에 입력을 넣고 Uart3통신을 통해 TC275에서 컴퓨터 내의 putty에 출력을 보내어 키보드의 입력값에 따라 코드들이 정상적으로 구현되게끔 한다.

초음파 센서의 거리 측정으로 차량이 자율적으로 주차공간에 후진하며 들어가 주차될 수 있도록 올바른 코드 로직을 구성한다.

kyungshin

2. Process

kyungshin

2. Process

1) V모델 기반 개발

kyungshin

2. Process

2) 개발 일정 관리

구분	작업 일정(기준: 3일)		
<요구사항 도출> 업무, 자료 분석 기획 및 기획안 확정			
<시스템/아키텍쳐 설계> 요구사항에 따른 시스템 및 아키텍쳐 설계 시스템 부품연결도 설계 회로 연결도 설계			
<구현단계> 인지/판단/제어 개발 하드웨어 통합 모델 통합			
<테스트/이행 단계> 단위, 통합 테스트 인수 테스트			
<최종 산출물 정리> 완료보고서 작성 발표자료 작성 시연 동영상 제작			

kyungshin

2. Process

2) 개발 보조 툴

kyungshin

03. Requirement Analysis

kyungshin

03. Requirment Analysis

1) 요구사항

Req.ID	요구사항	중요도/긴급도	비고
SW-Logic-001	W 입력을 누르면 차량이 전진한다	중/상	
SW-Logic-002	A 입력을 누르면 차량 진행 방향을 좌측으로 회전한다.	중/상	
SW-Logic-003	S 입력을 누르면 차량의 속도가 감속하거나 후진한다.	상/상	
SW-Logic-004	D 입력을 누르면 차량 진행 방향을 우측으로 회전한다.	중/상	
SW-Logic-005	입력 A를 여러 번 누를수록 좌측으로 진행하는 속도가 빨라진다.	중/중	
SW-Logic-006	입력 D를 여러번 누를 수록 우측으로 진행하는 속도가 빨라진다.	중/중	
SW-Logic-007	R 입력을 누를 시 모터의 구동이 시작된다.	중/중	

kyungshin

03. Requirment Analysis

1) 요구사항

Req.ID	요구사항	중요도/긴급도	비고
SW-Logic-008	이외의 문자가 입력될 시 차량이 멈춘다.	상/상	
SW-Logic-009	왼쪽으로 회전하고자 하는 경우, 오른쪽 모터를 왼쪽보다 빠르게 구동한다.	중/중	
SW-Logic-010	오른쪽으로 회전하고자 하는 경우 왼쪽모터를 오른쪽보다 빠르게 구동한다.	중/중	
SW-Logic-011	P 입력을 누르면 자동 주차 모드가 된다.	중/중	
SW-Logic-012	자동 주차 모드시 키보드 입력 없이 임의로 설치한 벽 사이 공간에 차량이 90도 회전 후 후진하여 주차될 수 있도록 한다.	상/상	
SW-Logic-013	주차 과정에서 후진을 하는 중에 차량과 주차 공간 벽면 사이의 거리가 가까워 질수록 부저의 주기가 짧아진다.	상/상	

kyungshin

03. Requirment Analysis

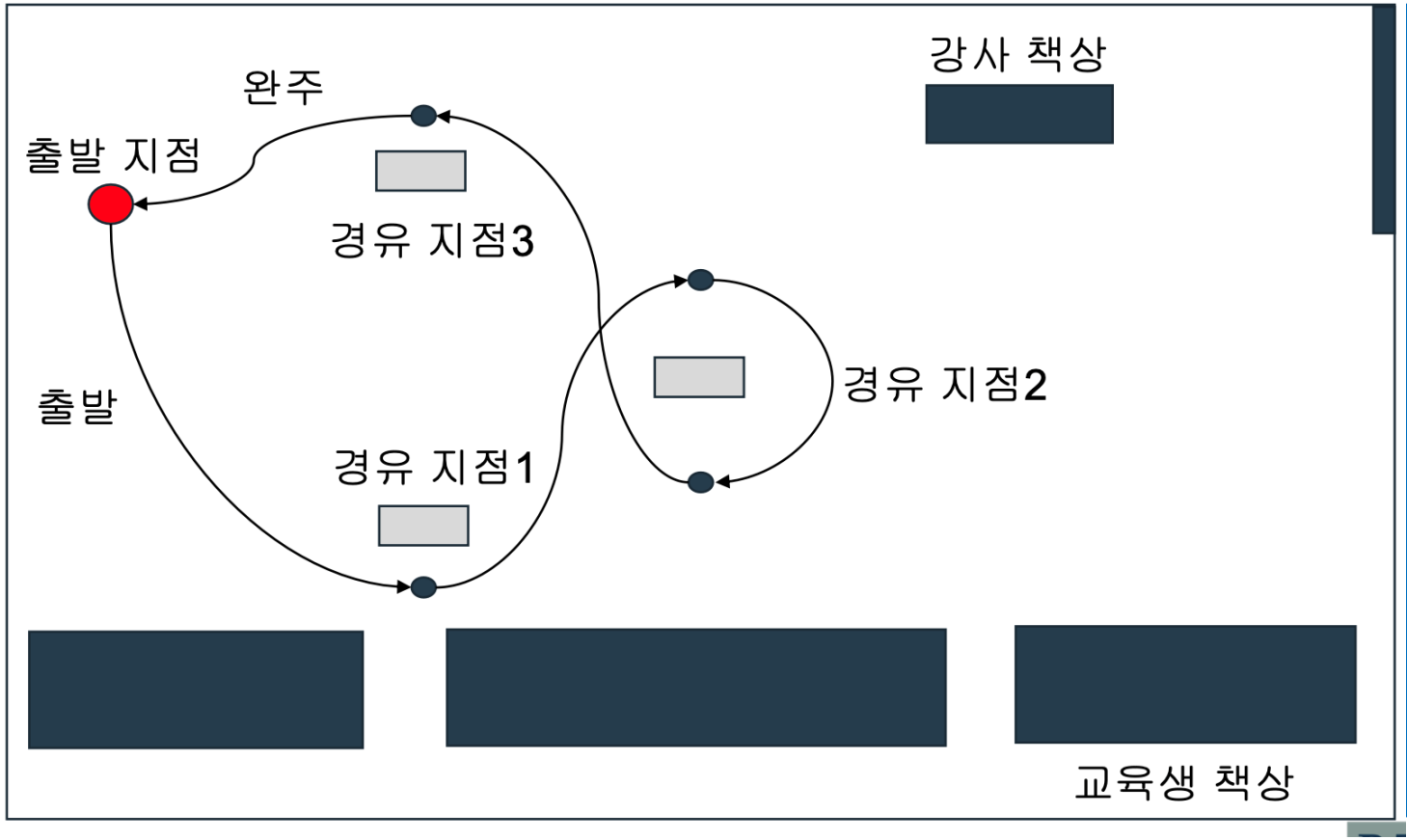
1) 요구사항

Req.ID	요구사항	중요도/긴급도	비고
SW-Logic-014	후진 도중 일정거리 이하로 다다랐을 경우 후진 제어를 더 이상 하지 못하게 멈춘다.	상/상	
SW-Logic-015	자동으로 멈춘 경우 후진이 아닐때 주행가능하다.	상/상	
SW-Logic-016	RC카의 제어는 블루투스 통신으로 한다.	중/중	
SW-Logic-017	주행 중 전방에 물체가 나타나면 긴급제동하며 알림 메세지를 출력한다.	중/중	
SW-Logic-018	긴급제동이 걸렸을 시 전진은 불가능하나 후진은 가능하다.	상/상	
SW-Logic-019	후진하여 장애물 및 벽과의 거리가 일정 이상으로 증가하면 다시 전진 주행이 가능하다.	중/상	
SW-Logic-020	듀티비가 크면 긴급제동 거리도 길고, 듀티비가 작으면 긴급제동 거리도 짧다.	상/상	

kyungshin

03. Requirment Analysis

2) 의뢰 시나리오 1

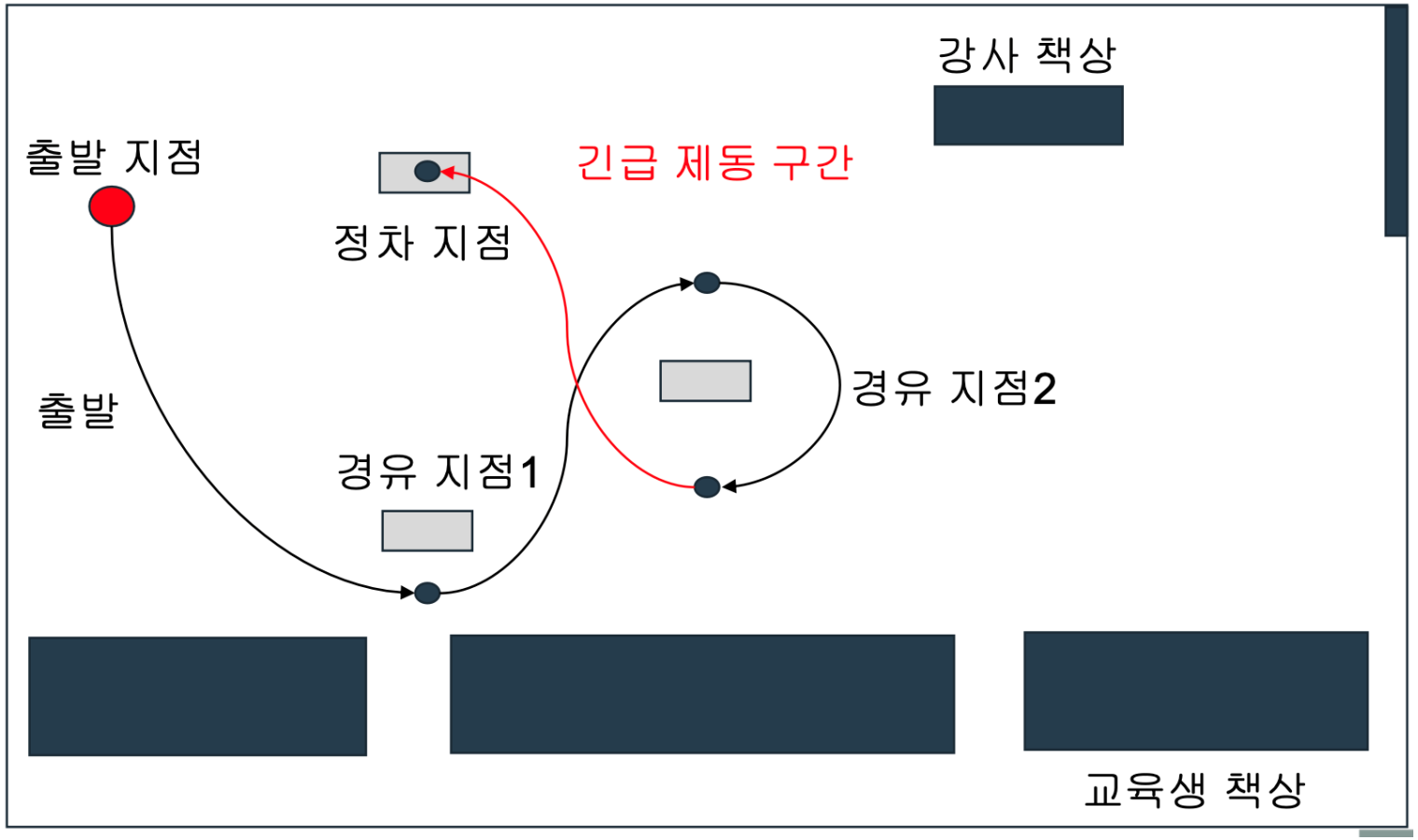


블루투스 통신 기반으로 RC카를 제어하여 다음과 같이 주행 구간을 완주하시오.

kyungshin

03. Requirment Analysis

2) 의뢰 시나리오 2

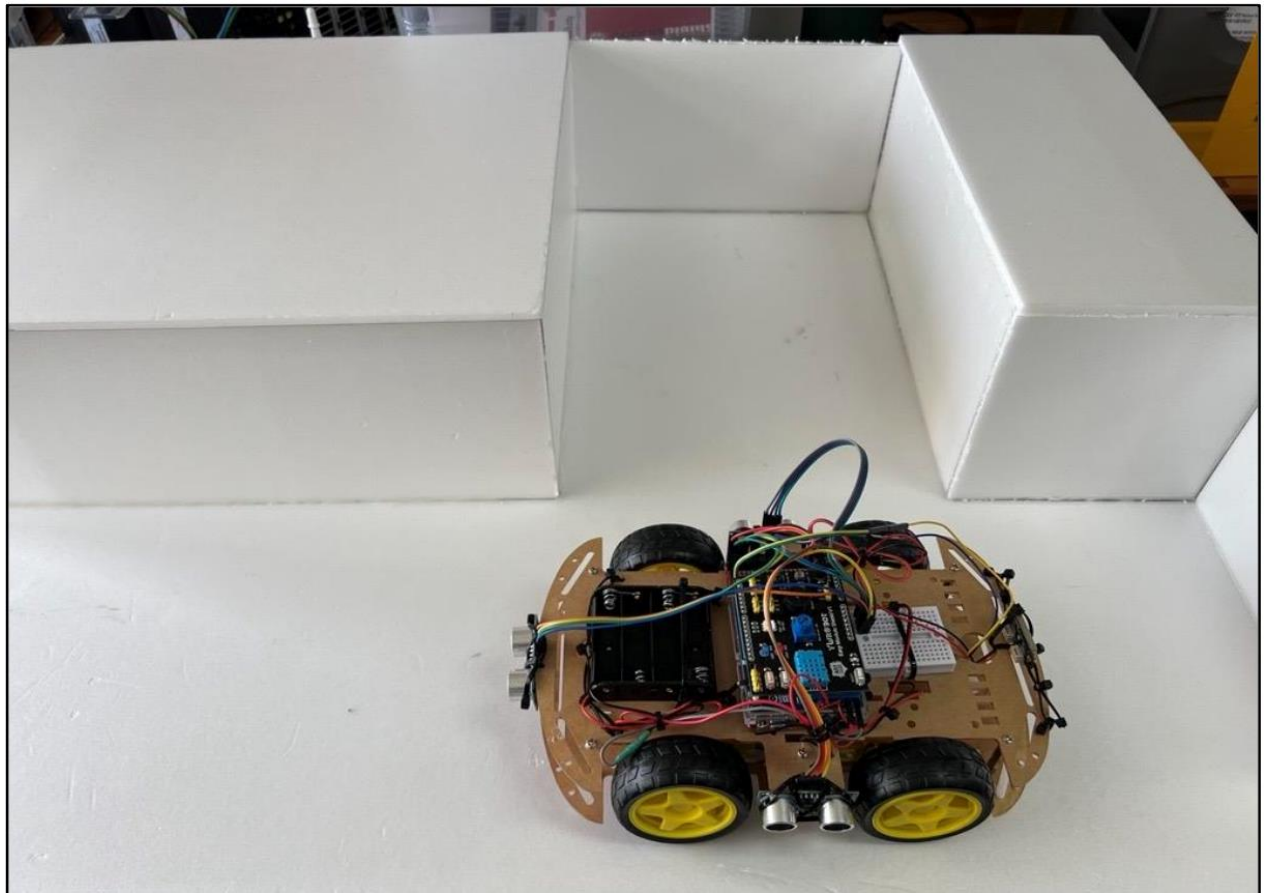


AEB 기능을 구현하여 주행 구간 도중 전방에 장애물이나 타나면 긴급제동할 수 있도록 하시오

kyungshin

03. Requirment Analysis

2) 의뢰 시나리오 3



시리얼 통신으로 **'P'**를 입력받으면 자동 주차 모드로 진입한 후 차량이 앞으로 가다가 주차공간을 인식하면 정지하도록 하라

초음파 센서로 주차공간을 인식하면 차량을 멈춤

주차 공간을 찾으면 시리얼 통신으로 "parking space detected" 출력

차량을 주차공간에 맞게 회전시킨 후 주차공간에 진입하도록 하시오

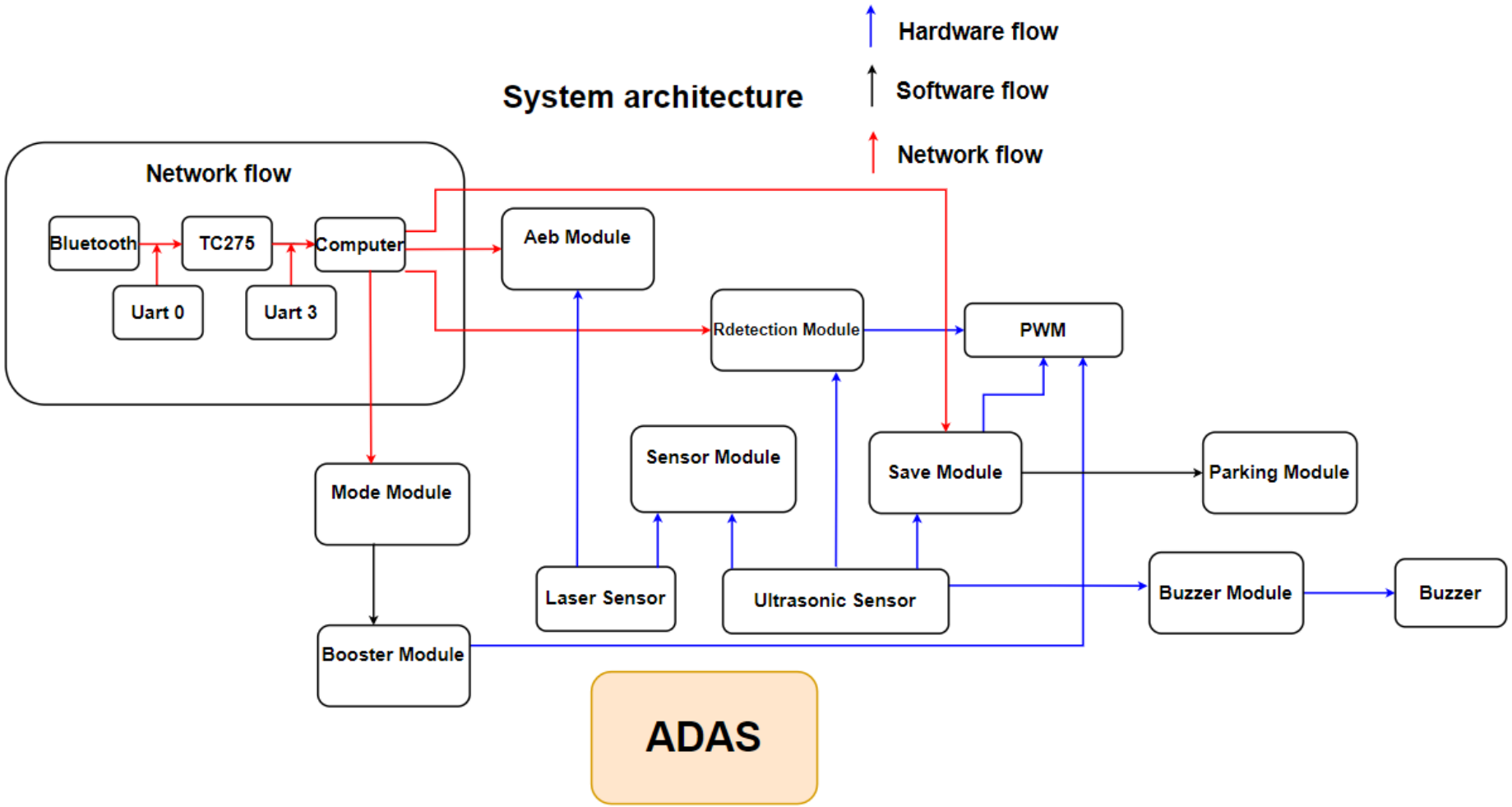
kyungshin

04. System Architecture

kyungshin

04. System Architecture

1) 시스템 아키텍처



kyungshin

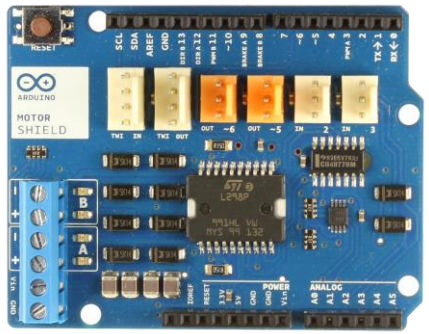
04. System Architecture

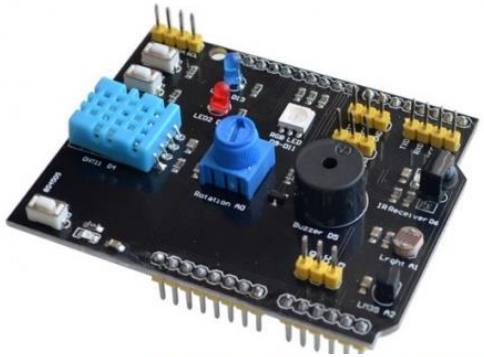
2) 기술 스택

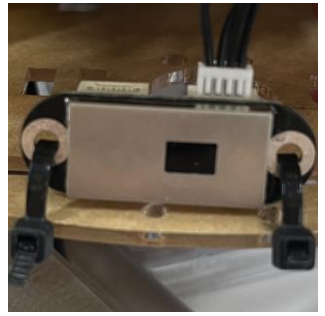
S/W

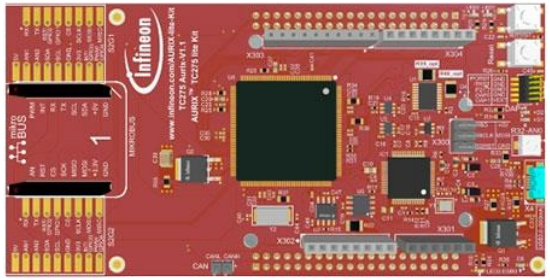
HIGHTEC

H/W









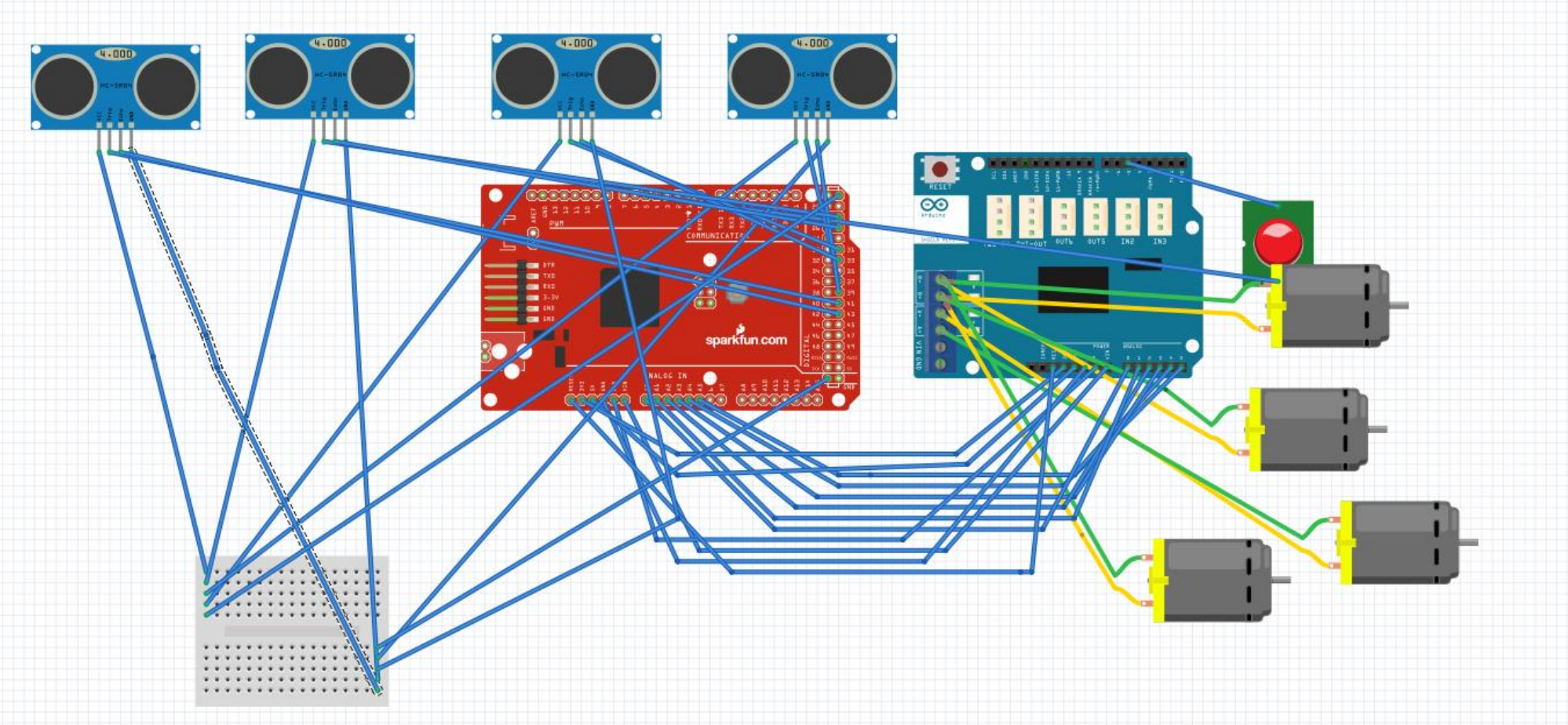
kyungshin

05. HardWare Architecture

kyungshin

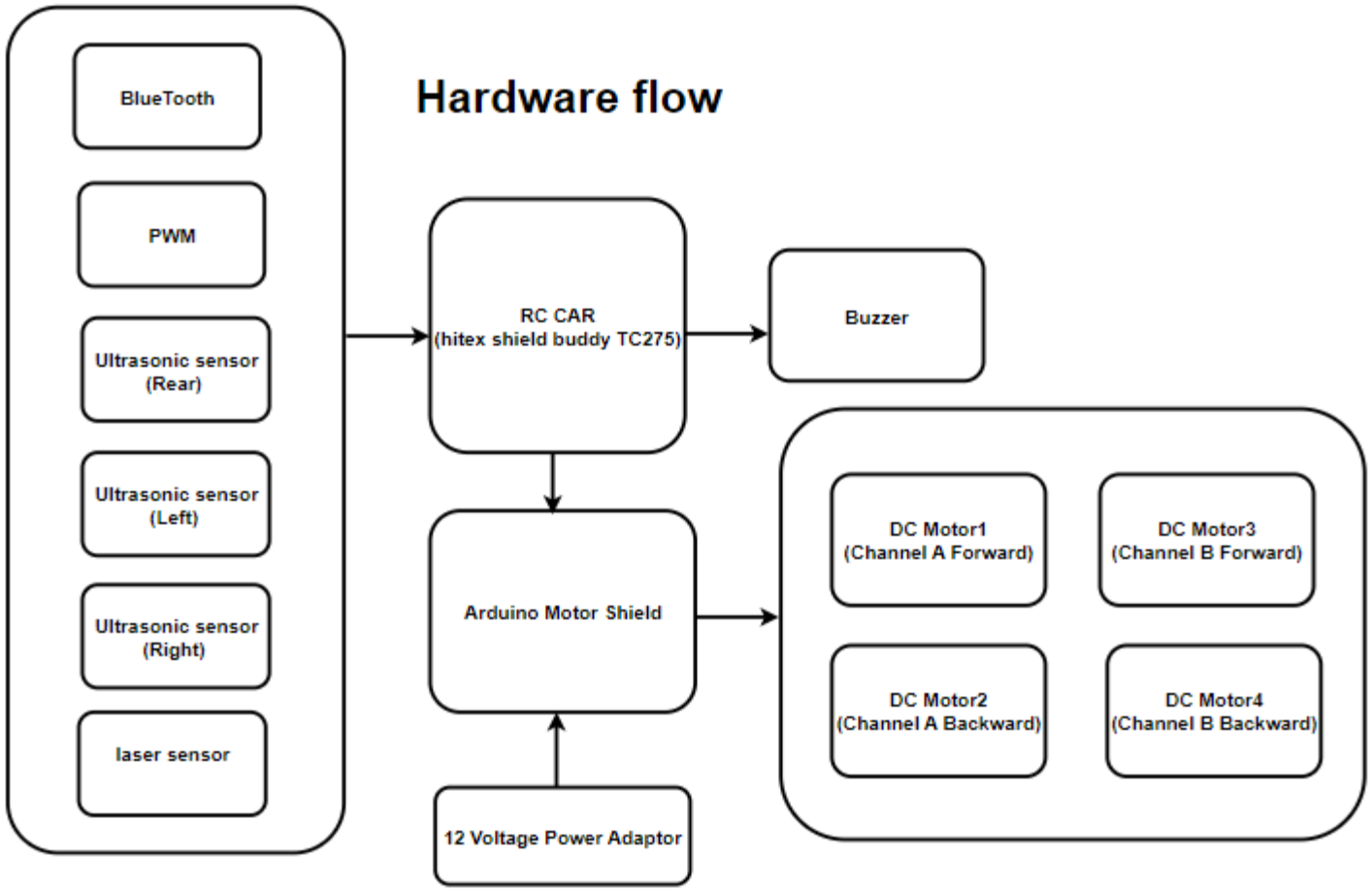
05. Hardware Architecture

1) 하드웨어 구성도



05. HardwareArchitecture

2)Hardware flow



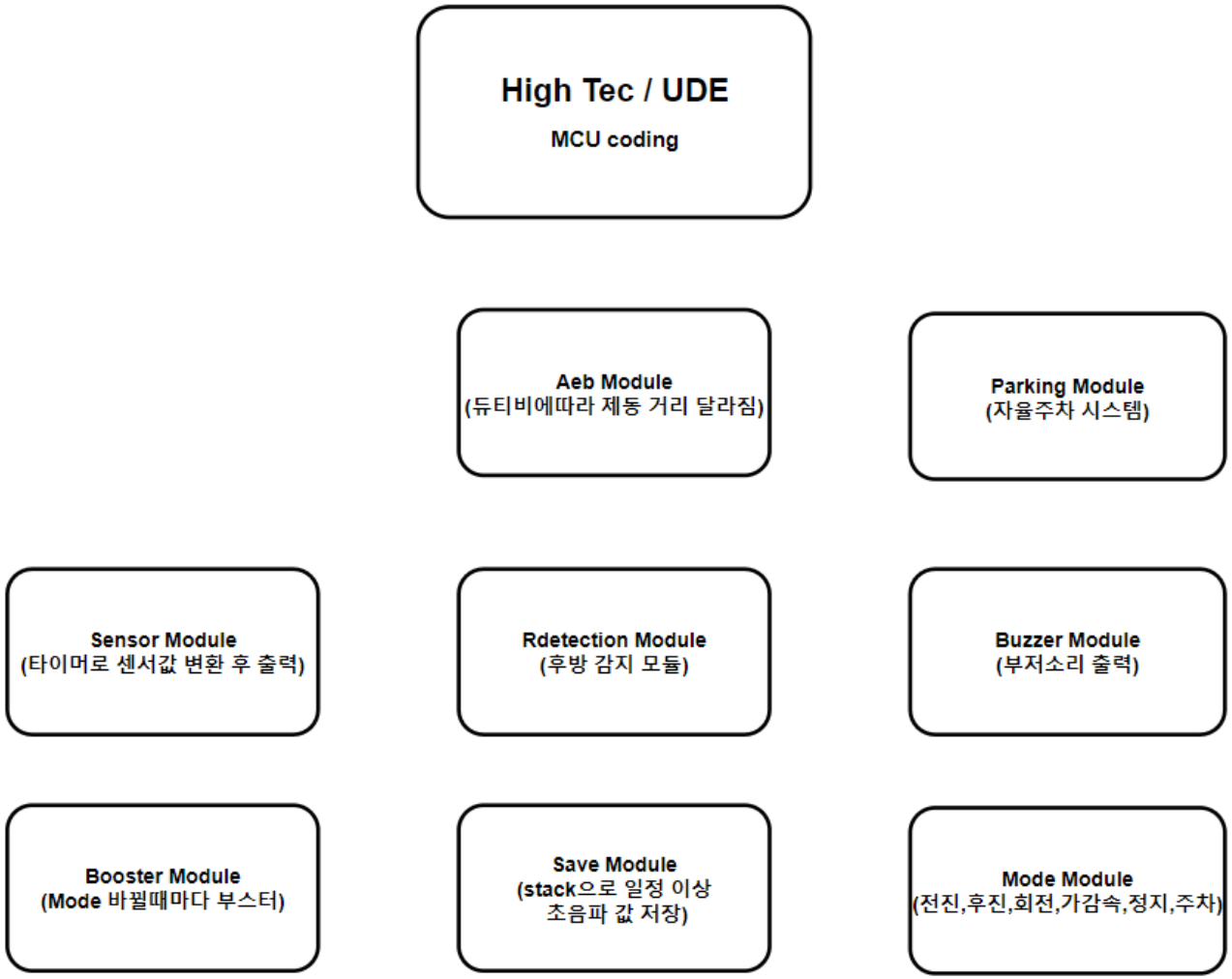
kyungshin

06. SoftWare Architecture

kyungshin

06. Software Architecture

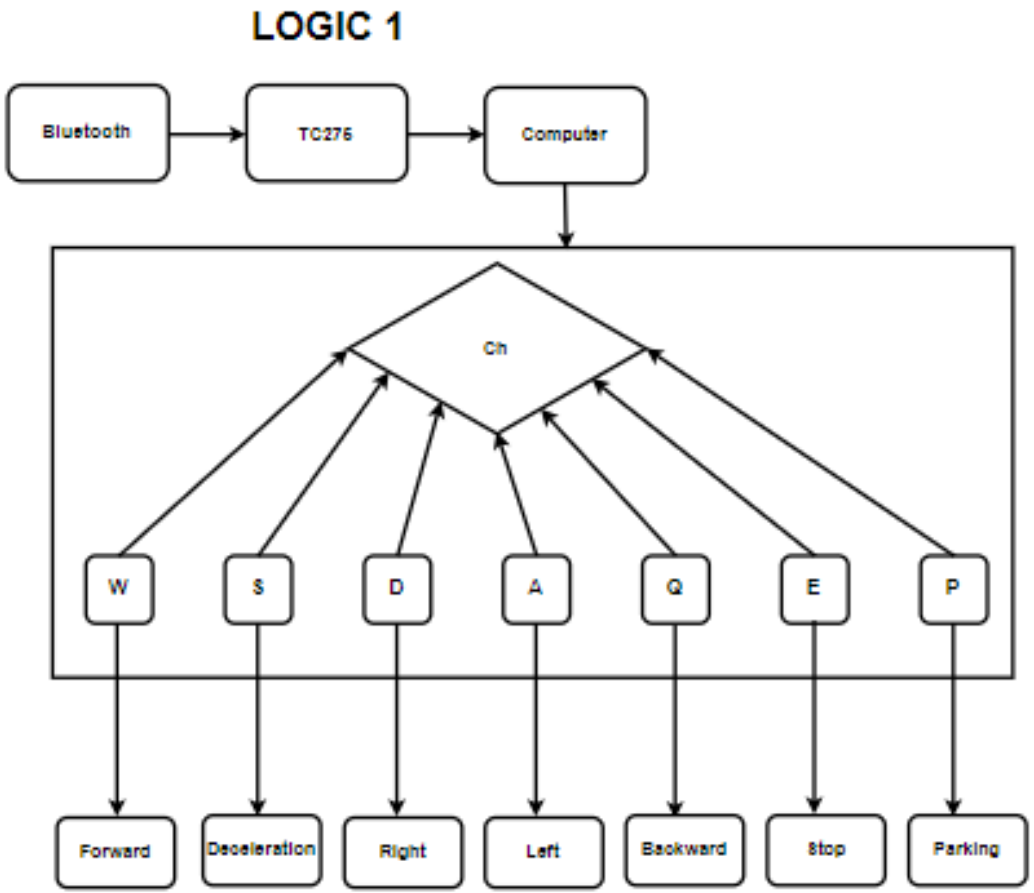
1) 모듈 구성



kyungshin

06. Software Architecture

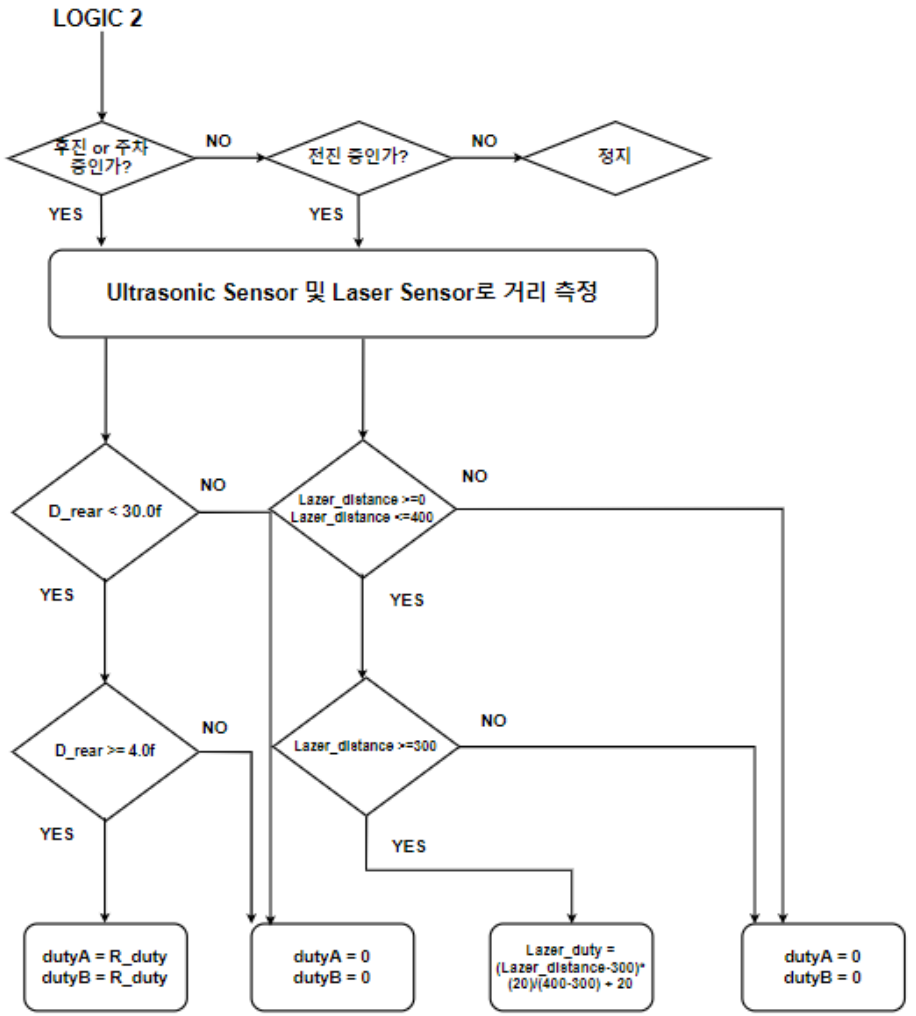
2) 순서도 logic1



kyungshin

06. Software Architecture

3) 순서도 logic2



kyungshin

6-4. SoftWare Component Detailed Design

ModeModule

```
if(ch !=-1){
 if(ch == 'w' || ch == 'W'){//전진, 가속
   state= 0;
 }
 else if(ch =='s' || ch == 'S' ){//감속
   state = 1;
 }
 else if(ch == 'd' || ch == 'D'){//우회전
   state = 2;
   change2 = 'd';
 }
 else if(ch =='a' || ch == 'A'){ //좌회전
   state = 3;
   change2 = 'a';
 }
 else if(ch == 'q' || ch =='Q'){ //후진
   state = 4;
 }
 else if(ch == 'e' || ch =='E'){ //정지
   state = 5;
 }
}
```

BoosterModule

```
switch(state){
   case 0:
     if(change2 == 'd' || change2 == 'a'){
        dutyA =0;
        dutyB= 0;
        change2=' ';
     }
      dutyA +=10*(1.2);
      dutyB +=10;
      if (dutyA > 100) dutyA = 100;
      if (dutyB > 100) dutyB = 100;
      dir = 1; //전진
      speed_flag=1; //레이저모드 킬 때 주석 해제 하면됨
      break;
```

```
//모드가 바뀔 때마다 부스터 //정지, 감속일때는 부스터 필요없음
if (change_flag == 1 && (dutyA!=0 || dutyB!=0) && slow_flag==0)
{
   movChA_PWM(90, dir);
   movChB_PWM(90, dir);
   delay_ms(20);
   change_flag = 0;
}
```

kyungshin

AebModule

```
if (((change >= 'a' && change <= 'z') && change != 's')){
 if(Lazer_distance >=0 && Lazer_distance <=400){
   if(change == 'w'){

      if(Lazer_distance >=300){
        Lazer_duty = (Lazer_distance-300)*(20)/(400-300) + 20;
        bl_printf("distance : %d(mm),  speed : %d \n",Lazer_distance, Lazer_duty);
        if(dutyA !=0){
          dutyA = Lazer_duty;
        }
        if(dutyB !=0){
          dutyB = Lazer_duty;
        }
      }
      else if(speed_flag == 1)
      {
       movChA_PWM(90, 0);
       movChB_PWM(90, 0);
       delay_ms(20);
       dutyA =0;
       dutyB =0;
       speed_flag=0;
      }
   }
  }
}
```

RdetectionModule

```
//후진 주차 시스템
if ((change == 'q' || dir == 0) && D_rear < 30.0f) //30cm 이하일때 작동
```

```
if (D_rear >= 4.0f)
{
   int R_duty = (D_rear-4)*6/(30-4) +10;

   if(dutyA !=0){
      dutyA = R_duty;
   }
   if(dutyB !=0){
      dutyB = R_duty;
   }

}
```

```
else{
    dutyA=0;
    dutyB=0;
    stopChA();
    stopChB();
}
```

BuzzerModule

```
    // 거리에 따른 부저 설정
    if (distnace > 30){
       setBeepCycle(150);
    }
    else if (distnace > 20){
       setBeepCycle(70);
    }
    else if (distnace > 10){
       setBeepCycle(30);
    }
    else{
     bl_printf("parking done\n");
      setBeepCycle(1);
    }

}
      else{
    setBeepCycle(0);
}
```

SaveModule

```
#define SZ 10000
int my_stack[SZ];
int top = -1;


void push(int data){
   if (top == SZ - 1) {
      return;
   }
   top++;
   delay_ms(30);
   my_stack[top] = data;
   return;
}

int pop(void) {
   int res = 0;
   if (top == -1) {
      return -1;
   }
   res = my_stack[top];
   delay_ms(30);
   top--;
   return res;
}
```

ParkingMoudle

```
// p 눌렀을때 자율주차
 if(stop == 0 && change == 'p'){
    dutyA = dutyB = 20;
    dir = 1;
    if(D_left > 40.0f){
       push(1);
```

```
if(top > top_standard){
   change_time = top;
   bl_printf(""\n\n parking space detected \n\n");
   stop = 1;
```

```
 else if(stop == 1 &&  change == 'p'){
    if(D_left > 40.0f){
       dutyA = dutyB = 14;
       dir =0;
       pop();
    }
```

```
if(top == change_time / 2 + 1){
   // 임시 정지
   stopChA();
   stopChB();
   delay_ms(1000);
   bl_printf("rotation\n");
```

```
stop= 0;
top = -1;
dutyA = dutyB = 30;
change = 'q';
```

kyungshin

07. Test

kyungshin

07. Test

1) 단위 모듈 테스팅 - ModeModule

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
ModeMoudle	Mtest01	r or R을 누르면 구동모드로 바뀌는지 확인함	PASS	신재환/2024.2.28	Ch = 'R'	Flag = on
ModeMoudle	Mtest02	W을 누르면 전진 모드인 State=0 으로 바뀌는지 확인	PASS	신재환/2024.2.28	Ch = 'w'	State = 0
ModeMoudle	Mtest03	S를 누르면 감속모드인 State = 1로 바뀌는지 확인함	PASS	신재환/2024.2.28	Ch = 's'	State =1;
ModeMoudle	Mtest04	D를 누르면 우회전인 State = 2로 바뀌는지 확인함.	PASS	신재환/2024.2.28	Ch = 'd'	State = 2
ModeMoudle	Mtest05	A를 누르면 좌회전인 State =3으로 바뀌는지 확인함.	PASS	신재환/2024.2.28	Ch = 'a'	State = 3
ModeMoudle	Mtest06	Q를 누르면 후진모드인 State = 4로 바뀌는지 확인.	PASS	신재환/2024.2.28	Ch = 'q'	State =4
ModeMoudle	Mtest07	e를 누르면 정지모드인 State = 5로 바뀌는지 확인.	PASS	신재환/2024.2.28	Ch = 'q'	State =4

kyungshin

07. Test

1) 단위 모듈 테스팅 - BoosterModule

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
Booster Moudle	Mtest01	앞방향으로 나아가면서 정지하지 않고 모드가 변경되었을 때 부스터가 나온다.	PASS	강성민/2024.2.28	Change _flag = 1; DutyA != 90; DutyB != 90; Slow_flag = 0;	Flag = on
Booster Moudle	Mtest02	채널 A,B의 duty값이 90이 된다.	PASS	강성민/2024.2.28	Change _flag = 1; DutyA != 90; DutyB != 90; Slow_flag = 0;	DutyA =90; DutyB =90;
Booster Moudle	Mtest03	Dir = 1 정방향으로 주행한다.	PASS	강성민/2024.2.28	Change _flag = 1; DutyA != 90; DutyB != 90; Slow_flag = 0;	Dir =1;
Booster Moudle	Mtest04	Change_flag의 값을 0으로 초기화한다.	PASS	강성민/2024.2.28	Change _flag = 1; DutyA != 90; DutyB != 90; Slow_flag = 0;	Change _flag = 0;

kyungshin

07. Test

1) 단위 모듈 테스팅 - SensorModule

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
Sensor Module	Mtest01	후면 초음파센서가 받아들이는 거리값의 단위가 어떠한지 확인	PASS	전지환/2024.2.28	거리 10cm	D_rear =10.1f
Sensor Module	Mtest02	좌측 초음파센서가 받아들이는 거리값의 단위가 어떠한지 확인	PASS	전지환/2024.2.28	거리 10cm	D_left =9.8f
Sensor Module	Mtest03	우측 초음파센서가 받아들이는 거리값의 단위가 어떠한지 확인	PASS	전지환/2024.2.28	거리 10cm	D_right =10.3f
Sensor Module	Mtest04	정면 레이저센서가 받아들이는 거리값의 단위가 어떠한지 확인	PASS	전지환/2024.2.28	거리 1cm	Lazer Distance= 1
Sensor Module	Mtest05	초음파센서가 받아들이는 거리값 최대값이 어떠한지 확인	PASS	전지환/2024.2.28	거리 4m	D_rear = 398.6f
Sensor Module	Mtest06	초음파센서가 받아들이는 거리값 최소값이 어떠한지 확인	PASS	전지환/2024.2.28	거리 0.1cm	D_rear = 0.1f
Sensor Module	Mtest07	레이저센서가 받아들이는 거리값 최대값이 어떠한지 확인	PASS	전지환/2024.2.28	거리 4cm	Lazer_Distance = 396f
Sensor Module	Mtest08	레이저센서가 받아들이는 거리값 최소값이 어떠한지 확인	PASS	전지환/2024.2.28	거리 0.1cm	Lazer_Distance = 15f

kyungshin

07. Test

1) 단위 모듈 테스팅 - BuzzerModule

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
Buzzer Module	Mtest01	부저가 후진주차시스템에서 작동하는지 확인	PASS	전지환/2024.2.28	change='q' dir = 0	Ch = 'q'
Buzzer Module	Mtest02	거리가 30cm 이상일 때 부저 경고 알람 주기 확인	PASS	전지환/2024.2.28	Distance >30	SetBeepcycle (150)
Buzzer Module	Mtest03	거리가 20cm 이상일 때 부저 경고 알람 주기 확인	PASS	전지환/2024.2.28	Distance >20	SetBeepcycle (70)
Buzzer Module	Mtest04	거리가 10cm 이상일 때 부저 경고 알람 주기 확인	PASS	전지환/2024.2.28	Distance >20	SetBeepcycle (30)
Buzzer Module	Mtest05	거리가 10cm 이하일 때 부저 경고 알람 주기 확인	PASS	전지환/2024.2.28	Distance <10	SetBeepcycle (1)
Buzzer Module	Mtest06	거리가 30cm 이상이면서 설정해둔 부저 경고 최대 거리값을 넘어섰을 때 부저 경고 알람 주기 확인	PASS	전지환/2024.2.28	Distance >40	SetBeepcycle (0)

kyungshin

07. Test

2) 단위 테스트 AebModule

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
AebModle	Atest01	LazerDistance가 mm로 넘어오는지 확인함	PASS	강성민/2024.2.28	getTofDistance();	Distance(mm)
AebModle	Atest02	감속모드인 S가 아닌 상태에서 AebModule flag가 on되는지 확인함	PASS	강성민/2024.2.28	Ch != 's'	AebModule flag = 1;
AebModle	Atest03	LazerDistance가 400mm 이하고 전진 모드인 ch = w 이면 AebModule flag가 on이 되는지 확인한다	PASS	강성민/2024.2.28	LazerDistance >= 400mm && Ch = w	AebModule flag = 1;
AebModle	Atest04	LazerDistance가 300mm 이하이면 duty가 변하는지 확인함.	PASS	강성민/2024.2.28	LazerDistance >= 300 &&Ch = w	Lazer_duty = 0~100;
AebModle	Atest05	LazerDistance가 클 수록 duty가 커지는지 확인함. Lazer_duty = (Lazer_distance-300)*(20)/(400-300) + 20;	PASS	강성민/2024.2.28	LazerDistance >= 300 && Ch =w	Lazer_duty = 20~50
AebModle	Atest06	LazerDistance에 따라 거리비례해서 모터가 도는 속도가 빨라지고 느려지는지 확인함.	PASS	강성민/2024.2.28	LazerDistance >= 300 && Ch =w	모터속도 제어
AebModle	Atest07	LazerDistance가 300mm이하이면 duty가 0이 돼서 모터가 멈추는지 확인함.	PASS	강성민/2024.2.28	LazerDistance <300 && Ch = w	Duty =0 && 모터 정지

kyungshin

07. Test

2) 단위 Save Module

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
SaveModle	Stest01	Push를 누르면 30ms 후 top이 1증가하는지 확인함.	PASS	신재환/2024.2.28	Push()	Top++
SaveModle	Stest02	Pop를 누르면 30ms 후 top이 -1감소하는지 확인함.	PASS	신재환/2024.2.28	Pop()	Top--
SaveModle	Stest03	Stack SZ인 10000이 넘어가면 더 이상 top이 증가하지 않는지 확인함.	PASS	신재환/2024.2.28	Push()	Top = 10000
SaveModle	Stest04	만약 Stack이 비어있는 top = -1인 상태에서 pop()을 하면 더 이상 top이 감소하지 않고 -1를 반환하는 지 확인함.	PASS	신재환/2024.2.28	Pop()	Top = -1
SaveModle	Stest05	초음파 거리값이 잘 나오는지 확인하고 push()를 하면 초음파 센서값이 Stack에 저장되는지 확인	PASS	신재환/2024.2.28	Push()	Distance 저장
SaveModle	Stest06	Pop()를 하면 Stack에 저장된 초음파 센서 데이터가 없어지는지 확인	PASS	신재환/2024.2.28	Pop()	Distance 데이터 delete

kyungshin

07. Test

2) 단위 테스트
ParkingModule 1

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
ParkingModle	Ptest01	Ch= p가 되고 stop flag가 0이면 자동주차 Mode flag -1이 되는지 확인	PASS	신재환/2024.2.28	Ch = p	Flag = 1
ParkingModle	Ptest02	Ch = p이면 duty = 20으로 전진하는 지 확인함.	PASS	신재환/2024.2.28	Ch = p	Duty = 20
ParkingModle	Ptest03	초음파 센서 값 40이상을 입력해주면 30ms dela y를 주면서 push해 주는지 확인.	PASS	신재환/2024.2.28	D_left >=40	Push()
ParkingModle	Ptest04	Push를 12번 이상 줘서 top이 top_standard 이상이 되면 주차공간을 찾았다는 Parking Space Det ection 문구가 나오는지 확인한다. Stop flag가 on이 되는지 확인함.	PASS	신재환/2024.2.28	Top >=12	터미널에 Parking Space Detection 출력, Stop = 1
ParkingModle	Ptest05	Top이 12미만이면 top이 -1로 초기화 되는지 확인함.	PASS	신재환/2024.2.28	Top < 12	Top = -1
ParkingModle	Ptest06	Ch = p이고 stop=1인 상태에서 입력해준 초음파 거리가 40이상이면 pop을 해주면서 후진하는지 검증	PASS	신재환/2024.2.28	Ch =p , stop= 1,	Pop()
ParkingModle	Ptest07	Pop()을 하면서 지나온거리의 중앙값에위치하는지 확인	PASS	신재환/2024.2.28	Ch p, stop = 1	Pop()

kyungshin

07. Test

2) 단위 테스트
ParkingModule 2

모듈	Test.ID	Summary	결과	Tester	예시 입력	예상 결과
ParkingModle	Ptest08	Top이 중앙값에 맞췄다면 90도 회전하는지 확인	PASS	신재환/2024.2.28	movChA_PWM(70,0); movChB_PWM(70,1); delay_ms(110);	90 rotation
ParkingModle	Ptest09	로테이션 후 duty 80으로 delay 10ms동안 후진 하는 지 확인	PASS	신재환/2024.2.28	movChA_PWM(80, 0); movChB_PWM(80,0); delay_ms(40);	Duty = 80 Dir =0
ParkingModle	Ptest010	자동으로 후진 주차 모드로 바뀌는지 확인	PASS	신재환/2024.2.28	Ch= p &&Distance >=40	Ch = q
ParkingModle	Ptest011	Top이 =-1로 초기화 되는지 확인	PASS	신재환/2024.2.28	Ch= p && Distance >=40	Top = -1
ParkingModle	Ptest012	Stop flag가 0으로 초기화되는지 확인	PASS	신재환/2024.2.28	Ch= p &&Distance >=40	Stop = 0

kyungshin

07. Test

3) 통합 테스팅

IntegTest ID	ReqSpec.ID (Traceability)	No.	Summary	Tester	결과
IntegTest-01	ReqSpec-01,02,03,04	1	W 전진, A 좌회전, S 감속, R구동, Q 후진, E 정지하는 지 확인함	신재환	PASS
IntegTest-02	ReqSpec-01,02,03,04	2	정지, 감속을 제외하고 모드가 바뀌면 10ms 동안 duty 70으로 구동하는지 확인	강성민	PASS
IntegTest-13	ReqSpec-13 ReqSpec-14	3	후진 모드상태에서 거리에 따라 부저 소리 주기가 달라지는지 확인	전지환	PASS
IntegTest-04	ReqSpec-13 ReqSpec-14	4	후진 모드에서 일정거리 이하로 다다랐을 경우 후진 제어를 더 이상 하지 못하게 멈춤.	신재환	PASS
IntegTest-05	ReqSpec-17 ReqSpec-18 ReqSpec-20	5	레이저 센서를 이용해 거리에따라 duty가 달라지는지 확인함.	강성민	PASS
IntegTest-06	ReqSpec-18 ReqSpec-19 ReqSpec-20	6	레이저 센서를 이용해 300mm이하의 거리면 duty가 0인지 검증	전지환	PASS
IntegTest-07	ReqSpec-11,12,13	7	P를 누르면 자동주차 모드가 되는지 확인	신재환	PASS
IntegTest-08	ReqSpec-11,12,13	8	자동 주차 모드시 키보드 입력 없이 임의로 설치한 벽 사이 공간에 차량이 90도 회전 후 후진하여 주차될 수 있도록 한다.	강성민	PASS
IntegTest-09	ReqSpec-08,12,13	9	자동주차 모드에서 후진을 할때 거리에 따라 부저울리는 주기가 짧아지는지 확인	전지환	PASS
IntegTest-10	ReqSpec-13,12	10	자동 주차 모드일때 top에 쌓인 반만큼의 시간동안 후진해서 중앙을 맞추는지 확인	신재환	PASS
IntegTest-11	ReqSpec-09,10,11	11	중앙값을 맞췄으면 90도 회전하는지 확인 .	강성민	PASS
IntegTest-12	ReqSpec-09,10,11	12	주차 공간을 찾지 못했으면 top이 -1되고 다시 주차 공간을 탐지하는지 확인	전지환	PASS
IntegTest-13	ReqSpec-14,15,16	13	주차공간을 찾았으면 자동으로 후진 모드로 전환되는지 확인	신재환	PASS

20개 요구사항 중 20개 항목 PASS

kyungshin

08. Review

08. Review

1) 시연 영상

kyungshin

08. Review

2) V-Cyde을 사용해 좋았던 점

좋았던 점

1. 각 단계가 명확하게 정의되어 프로젝트 추적에 용이하였다.

2. 분리되어 있던 각 기능을 통합할 때 문제점 파악이 편했다.

3. 테스트 단계에서 결함을 수정할 때 시간 절약.

4. 테스트, 개발이 직접적으로 연결되어 즉각적인 피드백.

kyungshin

08. Review

3. 아쉬웠던 점

1. Serial Bluetooth app을 통해 값을 입력해줬는데 default값으로 엔터값인 10이 들어가 센서랑 연동하면서 구현할때 어려움이 있었음.

2. while문이 센서 입력값을 처리하는 속도보다 빨라 센싱 오류가 발생해 어려움이 있었음.

kyungshin

감사합니다