

2020년도 창의공학설계입문  
**팀프로젝트 최종보고서**

(프로젝트 명 : 다양한 센서를 활용한 자율주행 자동차 제작)

2020. 06. 18.

팀 명 : 살려조  
 팀원명단 : 정 민 규  
황 진 주

구 분	작 성 방 법	비고
o 최종제출물	- 최종결과보고서 파일 - 최종결과물 구동 소스파일	

## 1. 프로젝트 목표 및 내용

### 가. 프로젝트 목표(개요) 및 변경내용

구 분	당초계획	변경내용	변경사유
목표(개요)	QTI 센서를 이용한 라인 주행.  IR센서 리모컨을 이용한 자율주행.	블루투스를 이용한 자율주행	블루투스의 빠른 반응 속도와 편한 조작감.

### 나. 프로젝트 세부(혹은 단계별) 목표 및 추진내용

세부 목표	추진 내용
Servo모터의 속도제어를 이용한 방향제어	주행 방향, 모드에 따라 모터의 속도를 조절하여 보다 빠르고 안정감있는 주행을 하도록 하였다.
IR센서 리모컨을 이용한 자율주행	리모컨의 전, 후, 좌, 우, 모드,멈춤 버튼을 활용하여 모드변경과, 아두이노 차체를 움직였다.
QT1센서를 이용한 라인트랙 주행	2개의 QT1센서를 이용하여 기기가 향하고 있는 방향을 알아낸다. 이를 이용하여 검은색 라인을 따라 주행하도록 한다.
블루투스를 이용한 자율 주행	블루투스 모듈을 사용하여 Android 애플리케이션을 통해 아두이노를 제어한다.

## 2. 프로젝트수행 결과

### 가. 추진계획 대비 실적

		추진 일정		
		2020. 6월 1주	2020. 6월 2주	2020. 6월 3주
추진 내용	계획	IR센서 제어	QT1센서 제어	블루투스 제어
	실적	리모컨을 이용하여 Servo모터 움직임 제어 (전, 후, 좌, 우, 정지)	라인을 따라 주행 가능	애플리케이션을 통한 전, 후, 좌, 우 움직임 제어
비고				

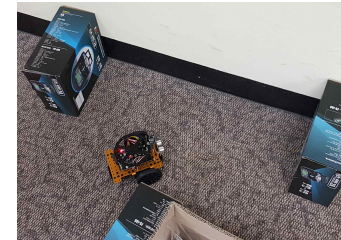
※ 추진계획 및 실적은 프로젝트 목표 및 내용과 일관성을 갖도록 작성요망

※ 비고에는 계획 대비 실적이 다른 경우, 그 이유를 언급할 것

## 나. 프로젝트 결과물

### 1) 결과물 동작 상태 확인용 사진

#### 1. 자율주행(블루투스)



#### 2. 라인을 따라 주행(IR)



#### 3. 주행하는 모습



### 2) 결과물 구동 소스 코드

```

/*****
* File Name: Final.ino
*
* Description: 마지막 공학 설계 RC제어 코드
* Created: 6/18/2020 11:31:41 AM
* Author: Jeong MinGyu
*
* Extern File
* Name Description
* =====
* boarddefs.h IR 센서 사용
* ir_Lego_PF_BitStreamEncoder.h IR 센서 사용
* IRremoteInt.h IR 센서 사용
* IRremote.h IR 센서 사용
* Servo.h 서보 모터(바퀴) 구동
*
* Extern Variables
* Name Type Description
* =====
* None
*
* Global Variables
* Name Type Description
* =====
* Dir enum 주행 방향
* leftStandard int 좌측 바퀴 정지 위치
* rightStandard int 우측 바퀴 정지 위치
* mode int 주행 모드(1-조작주행 0-자율주행)
*****/

#include <ir_Lego_PF_BitStreamEncoder.h>
#include <IRremoteInt.h>
#include <IRremote.h>
#include <boarddefs.h>

```

```

#include <Servo.h>

// 모터구동 핀
#define MOTORL_PIN 5
#define MOTORR_PIN 9

// 초음파 센서
#define TRIG_PIN 3
#define ECHO_PIN 4

// QTI센서
#define qtiL 11
#define qtiR 12

// 리모컨 제어
#define FRONT 0xFF02FD
#define BACK 0xFF9867
#define RIGHT 0xFF906F
#define LEFT 0xFFE01F
#define STOP 0xFFA857
#define MODE 0xFF22DD

// 서보 모터
Servo LServo;
Servo RServo;

// IR센서
IRrecv irrecv(2);
decode_results results;

// 방향
enum Dir
{
    forward = 0,
    backward = 1,
    left = 2,
    right = 3
};

int leftStandard = 90;    // 좌측 바퀴 정지 값
int rightStandard = 90;   // 우측 바퀴 정지 값
int mode = 1;             // 조작 모드 (1-조작주행 0-자율주행)

/*****
* Function Name: setup
* Description:
* the setup function runs once when you press reset or power the board
*
* @return: void
* @param: void
* Name      Type      Description
* =====
*
* Author:      Jeong MinGyu
* Creat Date:   2020.06.18.
* Last Edit:    2020.06.18.
*
*****/

```

```

* Global Variables
* Name      Type      Description
* =====
* None
*
* Algorithm :
*****/
void setup() {
    Serial.begin(9600);

    // QTI 센서
    pinMode(qtiL, INPUT);
    pinMode(qtiR, INPUT);

    // IR 센서
    Serial.println("Enabling IRin");
    irrecv.enableIRIn();
    Serial.println("Enabled IRin");

    // 초음파 센서
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

    // 모터 설정
    LServo.attach(MOTORL_PIN);
    RServo.attach(MOTORR_PIN);
}

/*****
* Function Name: SonarDistance
* Description:
* 벽과의 거리를 초음파 센서를 통해 측정한다.
*
* @return: int (벽과의 거리)
* @param: void
* Name      Type      Description
* =====
*
* Author:      Jeong MinGyu
* Creat Date:   2020.06.18.
* Last Edit:    2020.06.18.
*
* Global Variables
* Name      Type      Description
* =====
* None
*
* Algorithm :
*****/
int SonarDistance() {
    int sonarPwn;

    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
}

```

```

    sonarPwn = pulseIn(ECHO_PIN, 1, 18500L);

    return sonarPwn / 60;
}

/*****
* Function Name: Move
* Description:
* 자체가 이동하도록 바퀴를 동작합니다.
*
* @return: void
* @param: int
* Name      Type      Description
* =====
* speed     int       값이 클수록 이동(전환) 속도 빠름
* mode      Dir       이동 방향을 지정합니다
*
* Author:      Jeong MinGyu
* Creat Date:  2020.06.18.
* Last Edit:   2020.06.18.
*
* Global Variables
* Name      Type      Description
* =====
* leftStandard int     좌측 바퀴가 멈추는 값
* rightStandard int    우측 바퀴가 정지하는 값
*
* Algorithm :
* num  dir 좌 우 %3  %2
* 0    전  + -  0    0
* 1    후  - +  1    1
* 2    좌  - -  2    0
* 3    우  + +  0    1
*****/
void Move(int speed, Dir mode) {
    int LSpeed = mode % 3 ? -speed : speed;
    int RSpeed = mode % 2 ? speed : -speed;

    LServo.write(leftStandard - speed);    // +is Front -is Back
    RServo.write(rightStandard - speed);    // -is Front +is Back
}

/*****
* Function Name: loop
* Description:
* the loop function runs over and over again until power down or reset
*
* @return: void
* @param: void
* Name      Type      Description
* =====
*
* Author:      Jeong MinGyu
* Creat Date:  2020.06.18.
* Last Edit:   2020.06.18.
*
*****/

```

```

* Global Variables
* Name      Type      Description
* =====
* mode      int       주행 모드(1-조작주행 0-자율주행): 10이 기본값
*
* Algorithm :
*****/
void loop() {
    // 조작시 기본 속도
    int speed = 15;

    // 블루투스 조작
    if (Serial.available()) {
        char value = Serial.read();
        Serial.println(value);

        switch (value)
        {
            //회전시 조작이 편하도록 속도를 절반으로
            case '1':
                Move(0.5 * speed, left); break;
            case '2':
                Move(speed, forward); break;
            case '3':
                Move(0.5 * speed, right); break;
            case '4':
                Move(speed, backward); break;
            case '5':
                Move(0, forward); break;

            // 모드 변경 토글
            case '6':
                mode = mode ? 0 : 1; break;
            default:
                break;
        }
    }

    // IR 센서 제어
    if (irrecv.decode(&results)) {
        int count = results.rawlen;

        switch (results.value)
        {
            case FRONT: Move(speed, forward); delay(30); break;
            case BACK: Move(speed, backward); delay(30); break;
            case RIGHT: Move(0.5 * speed, right); delay(30); break;
            case LEFT: Move(0.5 * speed, left); delay(30); break;
            case STOP: Move(0, forward); delay(30); break;
            // 모드 변경 토글
            case MODE: mode = mode ? 0 : 1; break;

            default:
                break;
        }

        irrecv.resume();
    }
}

```

```
// QTI센서를 이용한 자율 주행
if (mode==0) {
    int LeftQti = digitalRead(qtiL);    // black 1 else 0
    int RightQti = digitalRead(qtiR);

    Serial.println(LeftQti);
    Serial.println(RightQti);

    // 벽 피하는 기능 추가시 사용 속도를 위해 비활성화
    // int distance = SonarDistance();
    // Serial.print("distance : ");
    // Serial.println(distance);

    // 좌측이 검은색이면 좌측 방향으로 이동
    if (LeftQti) {
        Move(speed-5, left);
        delay(3);
    }
    // 우측이 검은색이면 우측방향으로 이동
    else if (RightQti) {
        Move(speed-5, right);
        delay(3);
    }
    // 이외는 전진 주행
    else
    {
        Move(speed, forward);
    }
}
}
```

다. 참여 인원별 역할 및 수행 소감

성 명	역할 구분	수행 역할	수행 소감
황진주	팀장	코드 리뷰 기기 조작	여러 가지 변수가 많이 존재하여 어려웠지만 어려움을 즐기는 성격이기에 즐거웠다.
정민규	팀원	코드 작성 환경 구축	처음 해보는 하드웨어 제어였기에 어색하면서 어려웠던 점이 많았다. 기존 알던 지식이 적용되지 않는 예가 많아 코드 작성의 어려움이 많았다.

라. 활용방안

활용분야	활용방안	비고
오락, 유희	RC카를 직접 제작해보고, 여러 가지 트랙에서 주행을 해본다. 자율주행 대회, 라인 따라가기 등 여러 종목으로 나누어 대회를 진행한다면 즐거울 것이다.	현재

※ 비고란에는 현재까지 산출된 수행 결과이면 “현재”, 향후 예상되는 결과이면 “향후” 하고 표시함