

목차





- ① 프로젝트 목적 및 참여자 역할 분배
- ② 아이디어 개요
- ③ 프로젝트 사용 센서, 모듈, 액츄에이터
- ④ 기능 소개
- ⑤ 구현 과정
- ⑥ 완성품
- ⑦ 문제상황 해결 및 한계
- ⑧ 차별성 및 기대효과
- ⑨ 마무리



① 프로젝트 목적 및 참여자 역할 분배

본 프로젝트를 통해 아두이노 플랫폼을 기반으로, 안드로이드 모바일 앱과 통신이 가능한 [스마트 유모차 시스템]을 구현하고자 한다. 기존의 유모차에서 부족했던 부분(유모차 내부 온도 측정, 아이의 배변, 브레이크기능 등등)을 아두이노를 통하여 구현을 했다.

사용한 툴 및 도구는 다음과 같다.

아두이노 IDE	앱 인벤터	아두이노	안드로이드 휴대폰
			

참가자의 역할은 다음과 같다.

이름	직책	역할
우동균	팀장	아두이노 설계 및 구현 자료조사 보고서 작성 발표
조윤찬	팀원	안드로이드 모바일 앱 프로젝트 아이디어 개요 보고서 작성 발표 PPT만들기

처음에 정한 역할은 위의 표와 같지만 작업을 진행하다 보니 서로 담당하는 역할을 도와줘야 하는 상황이 생겼다. 예를들어 앱 인벤터를 만들고 아두이노와 연결을 하는 부분에서는 코드와 앱 인벤터의 BLOCK이 처음에 설계했던 것과는 달리 서로 엮이는 부분이 있어서 이후에는 같이 진행을 하게 되었고 프로젝트 마지막쯤에는 서로의 역할이 중요하지 않게 되었다.

② 아이디어 개요

처음에는 아두이노를 아이디어가 확정되기 전에는 여러 후보가 있었다.

정수기 온도 조절	현재 나온 정수기는 온도를 디지털 화면에 알려주지만 오래된 구식 정수기의 경우는 디지털 온도 기능을 추가 하기에는 비용이 많이 들었기 때문에 아두이노를 통해서 이를 보여주려했지만 구현이 간단하다는 점에서 최종 후보에서 탈락했다.
자취방 형광등 원격 조절	앱 인벤터의 버튼을 클릭하여 침대에서 직접 일어나서 스위치를 끄러가지 않고 원격으로 집 형광등을 끄는 기능을 구현하려고 했다. 그러나 너무 흔한 아이디어였고 다이소에서 6000원에 이와 비슷한 기능을 하는 도구를 살 수 있었기 때문에 비용적인 측면에서도 이점이 없어서 최종 후보에서 탈락했다.
도어락 음성인식	집에 들어갈 때 숫자버튼을 눌러서 집에 들어가는 기능 말고 사용자의 음성을 인식해서 도어락이 열리는 아이디어였지만 위험성과 구현하기에는 외부 환경요인으로 인하여 최종 후보에서 탈락하게 되었다.
에어컨 가동	여름에 집에 들어가기전에 미리 밖에서 에어컨을 가동 시켜 집에 들어갈 때 바로 시원해질수 있게 하는 아이디어 였지만 똑같은 기능을 하는 시스템이 시장에 나와 있었기 때문에 최종 후보에서 탈락하게 되었다.
스마트 유모차	프로젝트에서 최종적으로 구현하기로 선정된 아이디어이다. 육아에 직접 관련이 되어있지 않은 사람이라면 평소에 관심이 없을 수 있는 소재이지만 육아를 하는 사람이라면 필요한 기능이라는 점과 비용, 시장경쟁력에서도 가장 우수하다고 판단이 되어 선정을 하게 되었다.

유모차의 핵심 목적은 승차감과 안정성이다. 현재 유모차 시장은 이 목적 달성을 위해 외형, 소재를 강조하고 있으며 외적인 기능으로 접이식, 분리형, 디럭스형 등을 선보이고 있다. 본 프로젝트 참여자들은 유모차 관련 안전사고를 예방하는 관점에서 유모차의 기능적 개선 필요성에 공감하고 부모(유모차 운전자)의 입장에서 사용성을 높일 수 있다고 생각하여 [i-유모차]를 고안하였다.

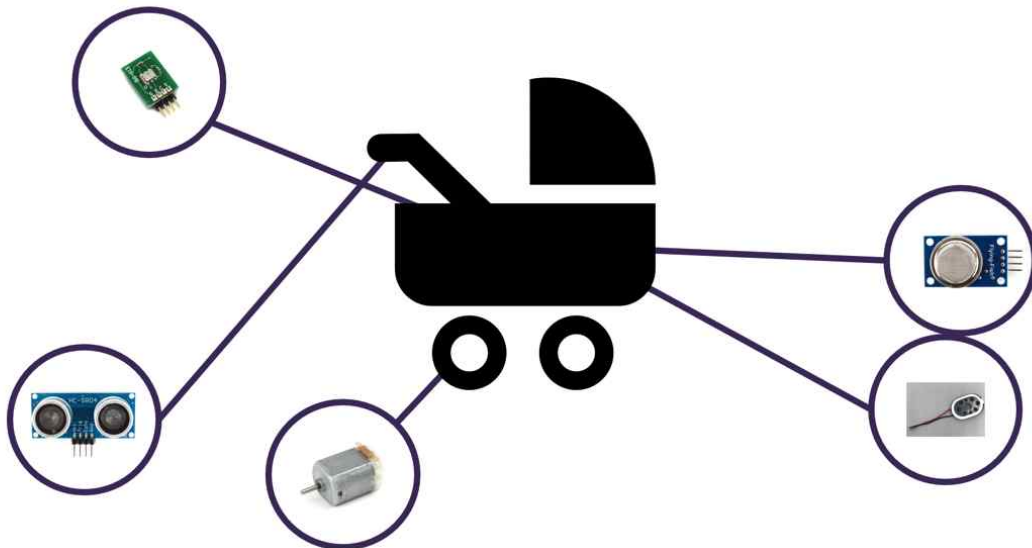
아래에 나타낸 표는 기존 유모차의 기능과 아두이노를 사용하여 만든 I-유모차의 기능을 비교한 표이다.

	기존 유모차	I-유모차
내부 온도 확인	x	o
아이 용변 확인	x	o
브레이크 기능	x	o
차양막 개방	수동	자동

위의 표를 보더라도 기존 유모차는 부모(유모차 운전자)입장에서는 고려하지 않았다는 것을 확인할 수 있다. 아두이노를 부착한 유모차를 사용하게 되면 대부분의 기능이 수동에서 자동으로 바뀌게 되고 부모 또한 육아의 수고를 덜 수 있는 장점이 생기게 된다. 또한 관련 사고 낙상, 전복, 제때 갈지 않은 기저귀로 인한 요로감염등을 예방할 수 있다.

유모차 아이디어를 구상하던 도중 직접 유모차를 구현하는 것에 대하여 잠시 고민을 하였다. 프로젝트를 만들 때 최대한 유모차로 만들고 싶었지만 비용의 문제 및 설계 문제 등등을 직면하게 되었고 용접을 하여 사실적인 유모차를 만들기는 보다는 수수깡등등의 쉽게 살 수 있는 도구를 사용하여 간의로 만들고 구현을 위주로 프로젝트를 진행하기로 하였다.


③ 프로젝트 사용 센서, 모듈, 액추에이터




처음에 구상한 기능은 다음과 같다. 유모차 아이콘에 센서, 모듈, 액추에이터를 어디에 부착하면 좋을지 대충 그려보았고 프로젝트를 진행하면서 추가된 도구는 많았지만 위의 그림에서 벗어나는 일은 크게 없었다.

다음은 프로젝트에 사용되는 센서, 모듈, 액추에이터이다. 아래 나오는 센서, 모듈, 액추에이터는 작업 과정에서 불필요하다고 생각하여 쓰지 않았던 **자이로센서와 서버모터**의 설명도 포함한다.


<사용 및 미사용 모듈>

사진	명칭	수량
	플라스틱 기어 78개입 세트 [SZH-GNP231]	1

프로젝트를 진행하는데 유모차 바퀴를 구현하기 위해서 아두이노78개입 세트를 하나 구매하였다. 따로 아두이노와 직접적인 연결은 없지만 DC모터를 통해서 바퀴가 고정되도록 한다.


사진	명칭	수량
	DC 모터	2

DC모터는 차양막을 열고 닫는것과 브레이크 기능을 사용하기 위해서 총 2개가 필요했다. DC모터는 전류의 방향을 전환하여 회전을 하는 모듈이다. 아두이노의 가장 큰 장점인 가격이 저렴하다는 점과 입력 전류에 출력이 비례를 하기 때문에 효율 또한 좋다. 손잡이에 부착한 초음파센서의 값을 받아와서 상호작용을 통하여 DC모터를 제어한다.


사진	명칭	수량
	MPU-8050 축 자이로 가속도 센서 모듈 GY-521 [SZH-EK007]	1

자이로 가속도 센서는 짐벌처럼 만들어서 길이 오르막, 내리막에 상관없이 항상 수평을 이루도록 하는 기능을 담당한다.


이후 작업 과정에서 여러 가지 요인으로 인해 짐벌은 사용하지 않게 되었다.(추후 설명)

사진	명칭	수량
	AM2302 (DHT22) 고정밀 온습도 센서 모듈 [SEN030005]	1


고정밀 온습도 센서는 유모차 내부의 온도를 측정하기 위한 기능을 담당한다. 원래는 보통의 온도센서를 사용하여 유모차의 내부 온도를 파악하려고 했지만 온도 센서만 있는 경우 정확한 온도를 파악하는 것이 어려웠기 때문에 더 정확한 고정밀 온습도 센서를 사용하여 구현하였다. 고정밀 온습도 센서에서 습도는 사용하지 않지만 아두이노에서 제공하는 라이브러리에서 온도는 값을 측정하여 LCD모니터에 출력이 되도록 구현한다.

사진	명칭	수량
	MQ-135 아두이노 유해가스/공기질 센서 모듈 [SZH-SSBH-038]	1

유해가스/공기질 센서 모듈은 아이가 앉아있는 쿠션에 설치하여 기체의 성분을 통하여 배변 유무를 측정하도록 한다. 가스센서에 들어온 값을 읽어와서 초기화를 진행한 이후 코드에 작성한 조건문(어느정도의 기준)에 들어온다면 부저를 울리거나 끄는 코드로 작동을 한다. 또한 이는 코드상에서 미니스피커와 함께 작동을 하여 알림 기능을 추가로 구현하였다.

사진	명칭	수량
	14x20mm 1W 미니스피커 [FQ-031]	1

이는 배변 유해가스/공기질 센서와 함께 배변 알림 기능을 담당한다. 가스가 측정이되면 부모(유모차 운전자)에게 알리는 기능을 한다. (부저로 변경)

사진	명칭	수량
	초음파 거리센서 모듈 HC-SR04 DM453	1

초음파 거리센서 모듈은 유모차 손잡이에 부착하고 이는 부모(유모차 운전자)와의 거리에 따라서 브레이크 기능을 담당하게 된다. 만약 설정한 거리범위에 들어 가면 브레이크 기능이 작동하지 않으며 설정한 거리에 벗어나게 된다면 브레이크가 작동하도록 한다. 초음파 센서는 물체에 반사되어 음파를 센서에 바로 진동을 시켜 일어난 전압에 대하여 측정거리를 계산한다. 즉, 반사원리를 사용하여 거리를 계산하게 되는데 이후 코드상에서 값에 따라서 부모(유모차 운전자)가 손잡이에 손을 놓은 것으로 인식하여 DC모터를 작동시키고 브레이크 기능을 하도록 구현된다.


사진	명칭	수량
	16x2 LCD I2C 모듈	1

블루투스 기능을 통하여 앱으로 온도를 확인할 수 있지만 매번 휴대폰으로 보기에는 번거로울 수 있다는 단점이 있기 때문에 LCD모듈을 활용하여 휴대폰 없이도 내부 온도를 유모차에서 확인할 수 있도록 하는 기능을 담당한다. 12개의 배선이 사용이 되며 문자 선고정밀 온습도 센서모듈과 함께 사용이 되며 유모차 내부에서 온도를 측정한 이후 코드상에서 받은 값으로 LCD에 출력이 된다. 가변저항을 사용하여 선명도에 있어서 좀 더 선명하게 구현할 수 있었지만 부품의 개수가 부족했기 때문에 이후 가변저항은 DC모터의 회전력을 제어하는데 사용하기로 하였다.


사진	명칭	수량
	서보모터 SG-90 Servo	3

자이로 센서와 더불어 짐벌을 만들 때 사용이 된다. 서보모터의 현재 각도는 가변저항에서 측정이 되는 값으로 가변저항값과 PWM신호를 비교한 이후 DC모터를 제어하며 DC모터의 회전으로 서보모터의 각도가 바뀌게 하여 짐벌을 구현을 한다.

이후 작업 과정에서 여러 가지 요인으로 인해 짐벌은 사용하지 않게 되었다.(추후 설명)

사진	명칭	수량
	블루투스 시리얼 모듈 HC-06	1

블루투스 모듈은 최종적으로 앱 인벤터를 통하여 안드로이드 폰과 연동이 되도록 하는 역할을 담당한다. 시리얼 버퍼에 저장된 데이터가 존재한다면 값을 읽고 출력을 한다. AT명령어를 사용하여 아두이노 IDE에서 확인 값을 받은 이후 안드로이드폰에서 arduino bluetooth controller 앱을 설치하고 설정한 블루투스 이름으로 연결을 요청하여 값을 보내고 받을 수 있다.

사진	명칭	수량
	L298N 모터드라이버	1

양방향으로 DC모터를 사용하기 위해 추가로 구매한 모듈이다. 모터드라이버는 12V이하의 낮은 전압에서 사용하며 아두이노 Vin에 연결을 하여 전원을 공급할 수 있는 장점이 있다. 약간의 고장이 단점이라는 것도 있지만 브레이크를 걸고 다시 해제할 때 즉, DC모터를 제어하기 위해 필요하기 때문에 사용하였다. 또한 전압을 높여 모터의 회전 속도가 빨리지고 전압을 낮추게 되면 속도가 느려지도록 조절하기 위해 모터의 속도를 조절하기 위해서 가변저항을 사용하였는데 구현을 할 때 브레이크의 경우 천천히 잡히면 안되기 때문에 가변 저항을 통하여 회전 속도를 빨리하였고 차양막의 경우는 아이의 안전을 위해서 천천히 내려가도록 구현했다.

④ 기능 소개

전체코드 설명

final | 아두이노 1.8.19

final

```
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SoftwareSerial.h>

DHT dht(3,DHT22);
LiquidCrystal_I2C lcd_1(0x3F, 16, 2);
SoftwareSerial BTSerial(5,6);

const int Pin_ECHO = 7;
const int Pin_TRIG = 8;

const int pot = A5; //가변저항 값 읽기
const int enable = 4; //L298N 속도제어용 핀
const int MotorPin1 = 12;
const int MotorPin2 = 13;
const int MotorPin3 = 10;
const int MotorPin4 = 11;

const int numTones=8;
const int tones[] = {261, 277, 294, 311, 330, 349, 370, 392};
const int AlarmPin = 9;

void setup()
{
  Serial.begin(9600); // 직렬 포트 초기화
  BTSerial.begin(9600); // 블루투스 통신 속도 지정 및 통신 시작
  pinMode(enable,OUTPUT);
  pinMode(MotorPin1, OUTPUT); //모터(차양막)
  pinMode(MotorPin2,OUTPUT); //모터(차양막)
  pinMode(AlarmPin,OUTPUT); //부저(배변알람)
  digitalWrite(enable,LOW); //일단 모터 정지
  lcd_1.begin();
  lcd_1.backlight();
  dht.begin();
}
```

- 라이브러리 불러오기

온습도 모듈을 사용하기 위해 DHT.h, I2C LCD를 사용하기 위해 LiquidCrystal_I2C, 블루투스 통신을 위해 SoftwareSerial과 Wire.h를 불러온다.

- 객체 생성

- 온습도 센서는 3번포트에서, DHT22표준을 따름을 명시한다

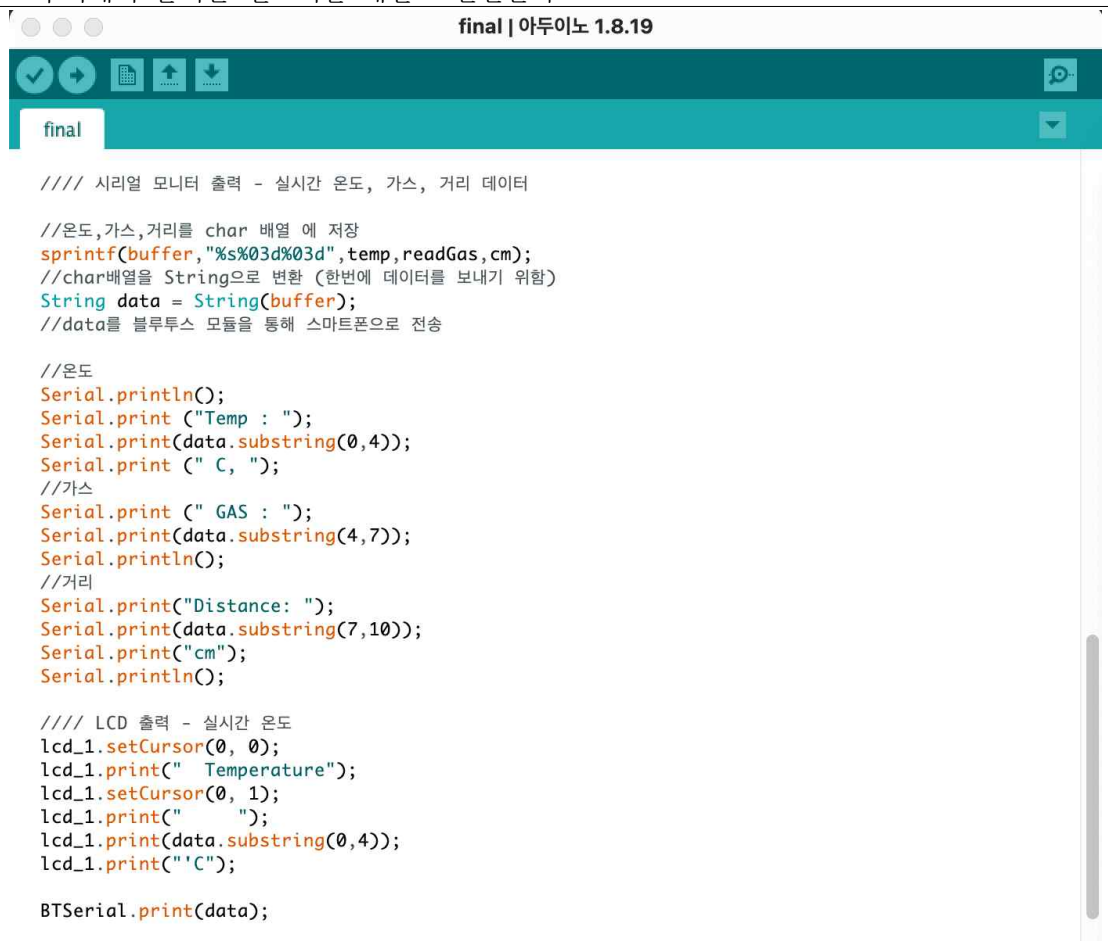
- 다른 코드에서 미리 알아온 lcd 의 주소값과 문자를 표시할 행/열 수를 초기화한다

- 블루투스 통신을 위한 포트를 설정한다. 아두이노 Mega는 Uno로 프로젝트를 진행했으므로 일반 핀을 rt,tx로 사용하기 위해 필요하다.

- 변수 및 배열 선언

- 가변저항, L298N 속도제어용, 모터, 부저를 작동시킬 핀을 설정한다

- 부저에서 울리는 멜로디를 배열로 선언한다



```
final | 아두이노 1.8.19

//// 시리얼 모니터 출력 - 실시간 온도, 가스, 거리 데이터

//온도,가스,거리를 char 배열 에 저장
sprintf(buffer, "%s%03d%03d", temp, readGas, cm);
//char배열을 String으로 변환 (한번에 데이터를 보내기 위함)
String data = String(buffer);
//data를 블루투스 모듈을 통해 스마트폰으로 전송

//온도
Serial.println();
Serial.print ("Temp : ");
Serial.print(data.substring(0,4));
Serial.print (" C, ");
//가스
Serial.print (" GAS : ");
Serial.print(data.substring(4,7));
Serial.println();
//거리
Serial.print("Distance: ");
Serial.print(data.substring(7,10));
Serial.print("cm");
Serial.println();

//// LCD 출력 - 실시간 온도
lcd_1.setCursor(0, 0);
lcd_1.print(" Temperature");
lcd_1.setCursor(0, 1);
lcd_1.print(" ");
lcd_1.print(data.substring(0,4));
lcd_1.print("'C");

BTSerial.print(data);
```

- 필요한 값이 잘 측정되고 있는지 시리얼 모니터로 확인하고자한다.
- 블루투스 모듈을 통해 스마트폰으로 데이터를 전송하기위해서, 모든 데이터를 하나의 버퍼로 묶고(sprintf) 문자열로 변환한다(data변수)
- 하나의 문자열로 저장된 데이터를 필요한 만큼 잘라서 사용한다. 첫 번째부터 4번째까지는 온도값, 5번째부터 7번째까지는 가스 값, 8번째부터 10번째까지는 거리값이다.
- 이때 가스 성분을 분석하고자 하는 것이 목적이 아니고, 정상범위에서 벗어나면 알람을 울리는 시스템이므로 가스 값은 정확한 단위로써 측정할 필요 없다. 따라서 가스센서에 부착된 가변저항 값을 적당히 돌려 인식하기 편한 값으로 설정하였다.
- lcd에 실시간 온도를 출력하고 있다. setCursor를 통해 lcd의 출력 위치를 결정할 수 있다.

```

////부저 - 알람기능, 일정 가스농도 [[[[[ 핀 9 ]]]]]
if (readGas > 100) {
  for(int i = 0; i < numTones; i++){
    tone(AlarmPin, tones[i]);
  }
}
else{
  digitalWrite(AlarmPin, LOW);
}

```

- 가스 값이 100 이상일 때, 미리 저장된 멜로디가 재생된다. 그렇지 않은 경우 재생하지 않는다.

```

////모터
////가변저항값을 읽어서 PWM 출력값으로 매핑, 모터 속도로 설
int pot_position = analogRead(pot);
int speed = map(pot_position, 0, 1024, 0,255);
Serial.println("--speed--");
Serial.println(speed);
Serial.println("-----");
// 속도제어핀에 속도 설정
analogWrite(enable, speed);
//앱에서 받아서

}

if(BTSerial.available()){
    int btn = BTSerial.read();
    switch(btn){
        case 11: digitalWrite(MotorPin1,HIGH);
                  digitalWrite(MotorPin2,LOW);
                  break;

        case 22: digitalWrite(MotorPin1,LOW);
                  digitalWrite(MotorPin2,HIGH);
                  break;

    }
}
if(cm<70){
    digitalWrite(MotorPin3,LOW);
    digitalWrite(MotorPin4,HIGH);
}
else{
    digitalWrite(MotorPin3,HIGH);
    digitalWrite(MotorPin4,LOW);
}
}
}

```

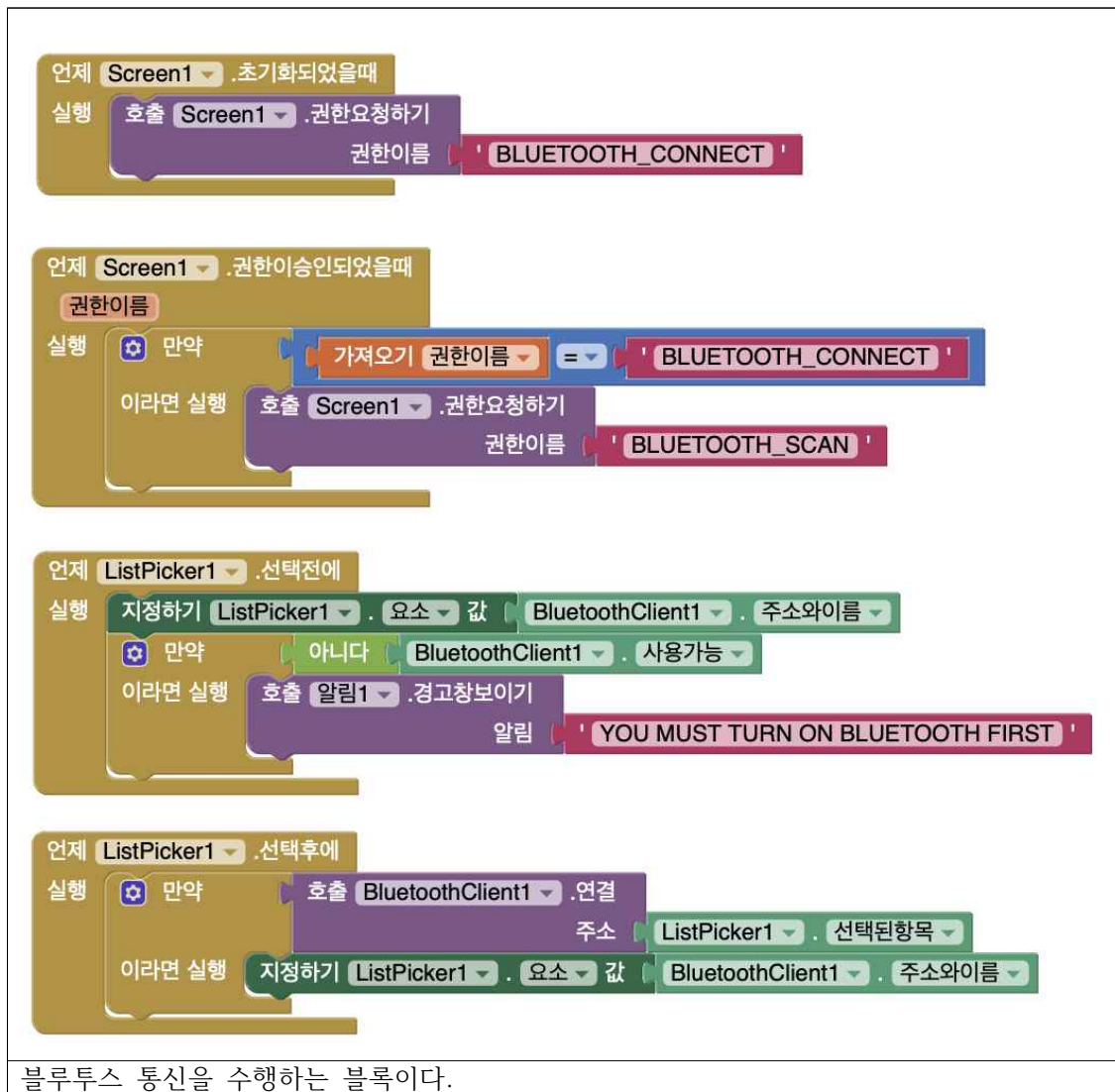
- 모터는 가변저항값을 읽어서 PWM 출력값으로 매핑하고, 이를 모터 속도로 설정하게 하였다.
- 속도값을 확인하기 위해 시리얼 모니터상에 출력하고 있다
- 속도 제어핀에 속도값을 지정해준다
- 블루투스통신이 확인되면(available),
- btn변수에 앱으로부터의 데이터를 저장한다.
- btn값이 11일 경우 차양막을 움직이는 모터를 시계방향으로 , 22일 경우 반시계방향으로 움직인다
- 거리(cm)가 70cm 미만일 때는 브레이크를 움직이는 모터를 반시계방향, 70cm이상일 경우 시계방향으로 회전시켜 브레이크 기능을 수행한다

```

//함수 - 초음파(음속)을 거리로 변환
long convertMsToCm(long ms_time) {
    return ms_time / 29 / 2;
}

```

- 거리를 측정하는 센서는 초음파센서이므로 ,음속을 실제 거리로 반환하는 함수이다.



블루투스 통신을 수행하는 블록이다.

전역변수 만들기 data 초기값 0

언제 시계1 타이머가작동할때

실행

만약 BluetoothClient1 연결여부

이러면 실행

만약 호출 BluetoothClient1 받을수있는바이트크기가져오기 > 1

이러면 실행

지정하기 전역변수 data 값 호출 BluetoothClient1 텍스트받기 바이트수 10

지정하기 Screen1 스크린설명 값 가져오기 전역변수 data

지역변수 만들기 temperature 초기값

텍스트에서 문자열 추출하기 텍스트 가져오기 전역변수 data × 1 시작위치 1 문자열길이 4

실행 지정하기 텍스트박스1 글꼴크기 값 50

지정하기 텍스트박스1 텍스트 값 가져오기 temperature

지역변수 만들기 gas 초기값

텍스트에서 문자열 추출하기 텍스트 가져오기 전역변수 data × 1 시작위치 5 문자열길이 3

실행

만약 가져오기 gas ≥ 200

이러면 실행

지정하기 레이블2 배경색 값

지정하기 레이블2 글꼴크기 값 40

지정하기 레이블2 텍스트 값 DANGER

아니라면

지정하기 레이블2 배경색 값

지정하기 레이블2 글꼴크기 값 60

지정하기 레이블2 텍스트 값 SAFE

지역변수 만들기 distance 초기값

텍스트에서 문자열 추출하기 텍스트 가져오기 전역변수 data × 1 시작위치 8 문자열길이 3

실행

만약 가져오기 distance ≥ 100

이러면 실행

지정하기 레이블3 텍스트 값 Braked

지정하기 레이블3 텍스트색상 값

지정하기 레이블3 배경색 값

아니라면

지정하기 레이블3 텍스트 값 Unbraked

지정하기 레이블3 텍스트색상 값

지정하기 레이블3 배경색 값

0

5

- 블루투스 모듈을 통해 받아온 데이터를 필요에 맞게 잘라서 사용합니다

- 앱인벤터는 자동 형변환을 지원합니다

- 15 -

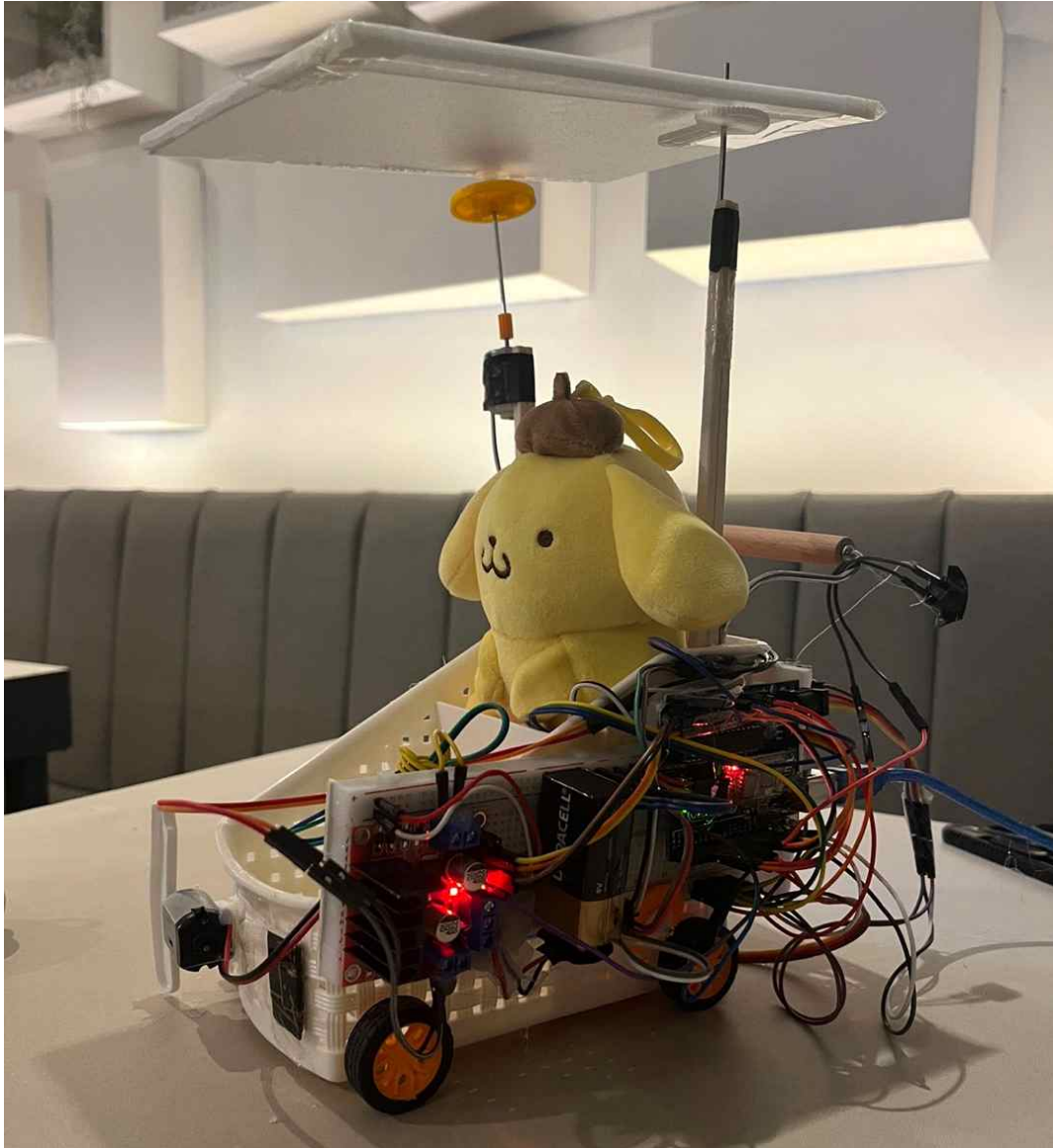


-canopyopen, close 버튼을 누르면 블루투스 모듈을 통해 11과 22 메시지를 전송합니다



- 앱인벤터상에서 디자인한 모양과 실제 어플에서 구현된 모습이 달라서 앱인벤터 상에서
는 다릅니다.

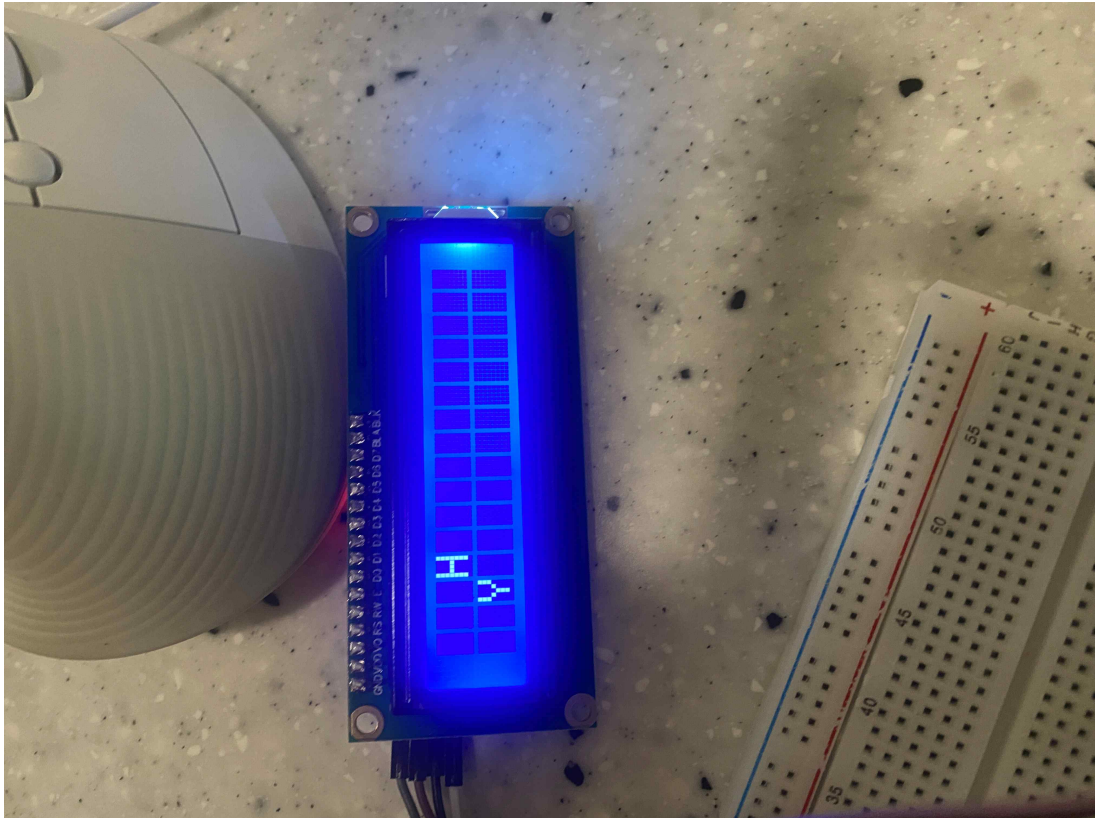
⑥ 완성품



⑦ 문제 상황 및 한계

문제 상황1

코드 구현과 관계없이 LCD에 텍스트가 하나씩만 나왔다.



실습시간에 아두이노 Mega 보드로 하였을때는 문제가 없었는데, 따로 구입한 Uno 보드는 비교적 구형일 것이므로 다른 결과를 보여주고 있다고 판단을 했다. Tinkercad 상에서 실습할때는 Adafruit_LiquidCrystal으로 하였을 때 문제없이 동작하였으나 실제 보드로 할 때는 그렇지 못했다.

해결과정1

구형인 Uno보드를 고려하여 아두이노 IDE의 구버전을 다운로드 받아서 작업을 진행하였지만 해결이 되지 않았다.

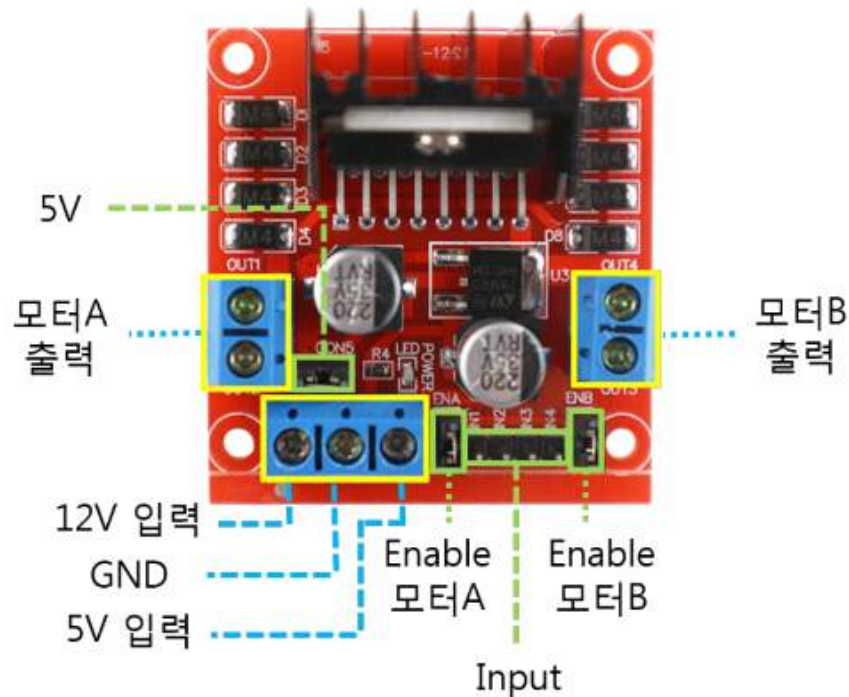
해결과정2

다른 라이브러리들(Adafruit_LiquidCrystal, jm_LiquidCrystal_i2c, 구버전 LiquidCrystal, 등)을 시도해보았다. 마지막 시도로 서드파티에서 제공하는 Arduino-LiquidCrystal-I2C-library를 다운로드받아 실행해보았고 성공적으로 작동하는 모습을 확인할 수 있었다. 사용한 라이브러리 주소는 다음과 같다.

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

문제 상황2

모터는 전원만공급하면 작동하지만 모터가 사용하는 전류가 '160[mA] +/- 40[mA]'이기 때문에 아두이노가 공급하는 '40mA'보다 많아서 아두이노 보드가 죽는 현상이 발생한다.



<모터 드라이버 2A L298N 핀배열>

해결하는 방법은 2가지가 있었다. 모터 제어 모듈(위 사진)을 구매하는 방법과 역방향 기능을 포기하는 방법이다.


해결과정1

스테핑모터 제어에 사용하는 uln2003칩을 이용해서 dc모터를 돌리기로 했지만 실패하였다.

해결과정2

역방향 제어를 위해서 l298n 모터 드라이버 모듈이 필요했다. 초반에는 이러한 문제점을 예상하지 못했기 때문에 모듈이 없었고 직접 구매를 해야하는 상황이 발생했다.

문제 상황3	
온, 습도 센서가 너무 뜨거워지는 문제가 발생했다.	
사진	
설명	
해결과정	
라이브러리를 사용하였고 저항을 연결하여 해결하였다.	

문제 상황4	
암수 케이블 개수가 부족한 상황이 있었다.	
	
해결과정	
케이블을 찢어서 피복안으로 모듈 핀을 꽂아서 해결을 하였다.	

문제 상황5

자이로 센서에서 짐벌을 구현할 때 아기를 짐벌에 태울시 안전하게 제작할 수 있을까?라는 문제점과 센서가 처음 배달이 왔을 때 핀이랑 보드가 연결이 되지 않은 상태로 배송이 왔었다.

해결과정 사진



결과 사진



해결과정

위의 사진처럼 직접 납땜을 통하여 문제를 해결하였다.

추후 작업중에 변화


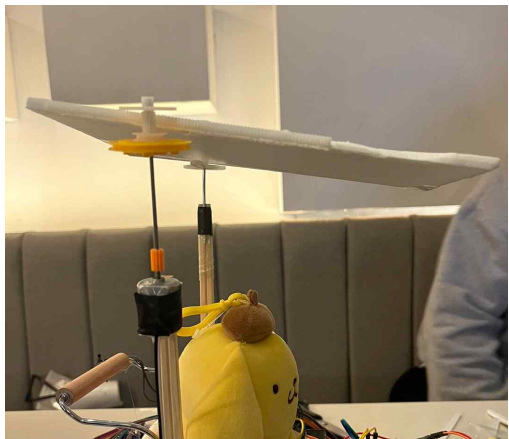
원래 계획에서는 오르막, 내리막이 많은 길거리에서 아이의 승차감을 편안하게 하기 위해서 자이로센서와 서버모터를 통하여 짐벌을 구현하려고 했지만 유모차를 만들 때 유모차 소재로 인한 안정성고정문제 때문에 짐벌을 구현하였어도 큰 변화가 없었다. 따라서 프로젝트를 진행할 때 자이로센서와 서버 모터를 이용한 짐벌은 포기해야했다.

⑧ 차별성 및 기대 효과

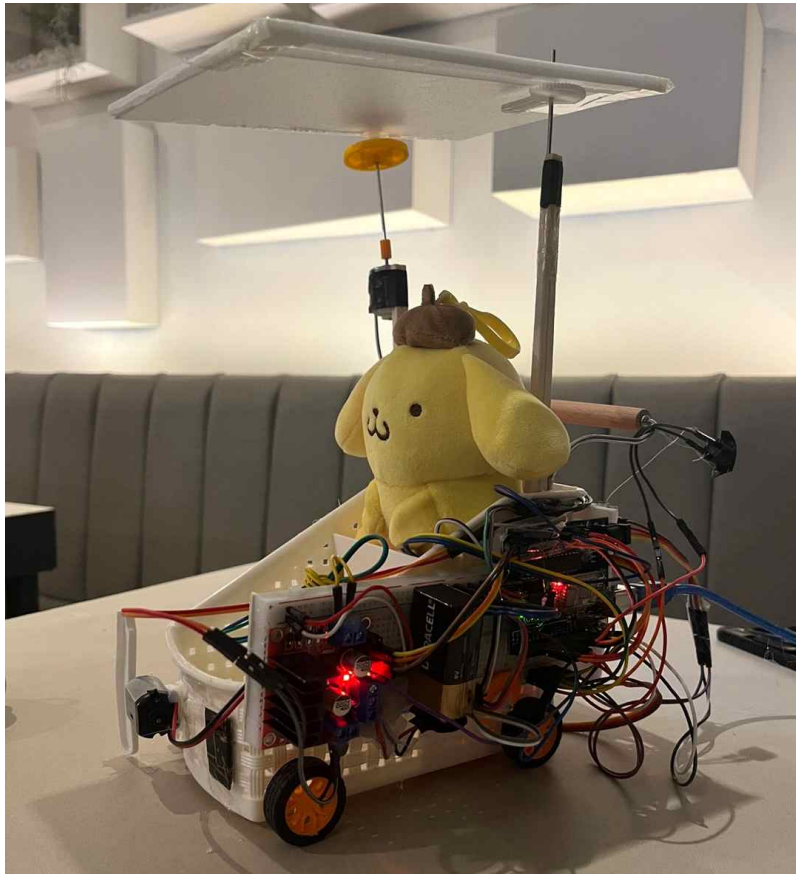
차별성1	
기존유모차	I-유모차
	
<p>기존 유모차는 부모(유모차 운전자)가 내부 온도를 잘 모르기 때문에 담요등을 둘러서 사용을 한다. 그러나 [I-유모차]의 경우 아이에게 제일 적절하다고 판단되는 온도를 토대로 내부 온도를 조절할 수 있다는 점이 있다.</p>	

차별성2	
기존유모차	I-유모차
	
<p>용변 유무를 기존 유모차는 아이의 울음등으로 부모(유모차 운전자)가 알았지만 가스농도와 소리를 통해서 수동으로 매번 확인하는 수고를 덜 수 있다.</p>	

차별성3	
기존유모차	I-유모차
	
<p>유모차 안전사고에서 제일 많이 발생하는 낙상사고 때문에 매년 문제가 발생하지만 I-유모차는 모터와 초음파 센서를 통하여 부모(유모차 운전자)가 손잡이를 잡고 있지 않을 때는 항상 브레이크가 작동이 되기 때문에 안전사고를 예방할 수 있다는 점에서 가장 큰 이점이 있다.</p>	

차별성4	
기존유모차	I-유모차
	
<p>비가 오거나 미세먼지가 있을때는 유모차는 차양막을 통해서 외부 이물질로부터 내부를 보호할 수 있다. 그러나 이것은 부모가 수동으로 차양막을 내려야 하기 때문에 수고가 많았지만 I-유모차는 자동화를 통해서 이를 해결할 수 있으며 차양막소재 또한 날카롭고 딱딱하지 않은 소재를 사용하기 때문에 안전사고도 예방할 수 있다.</p>	

기대효과 I-유모차



- ①대부분의 기등이 자동화가 되어있는 I-유모차를 통하여 부모(유모차 운전자)의 관점에서 제어를 할 수 있다.
- ②기존 유모차에서 발생하는 안전사고 또한 예방이 가능해지는 큰 이점이 있다.
- ③육아에서 제일 힘든 것이 아이의 기저귀 상태를 매번 확인해야한다는 점인데 유모차 내부에 이를 탐지하는 모듈을 통하여 부모가 확인하는 수고를 없애는 것과 동시에 발전해서 집에서도 모듈을 통하여 기저귀상태를 확인할 수 있도록 발전할 수 있다.
- ④자동화된 시스템을 통하여 육아에서도 다른 기존의 아이템들이 I-유모차와 처럼 발전 할 수 있게되는 시발점이 될 수 있다.
- ⑤아두이노를 사용하였기 때문에 비용적인 측면에서도 위와 같은 기능을 구현하기 위해서 드는 비용문제를 쉽게 해결할 수 있기 때문에 시장에서도 유리한 점이 있다.