

REPORT

제어공학응용및실험 작품계획서

블루투스를 이용한 컵테일 제조기



과 목	: 제어공학응용및실험
학 번	: 201920251, 201720650, 201720668
학 과	: 제어계측공학전공
성 명	: 함주희, 서상원, 서정무
제 출 일 자	: 2021. 12. 7
담 당 교 수	: 이봉섭 교수님

작품계획서

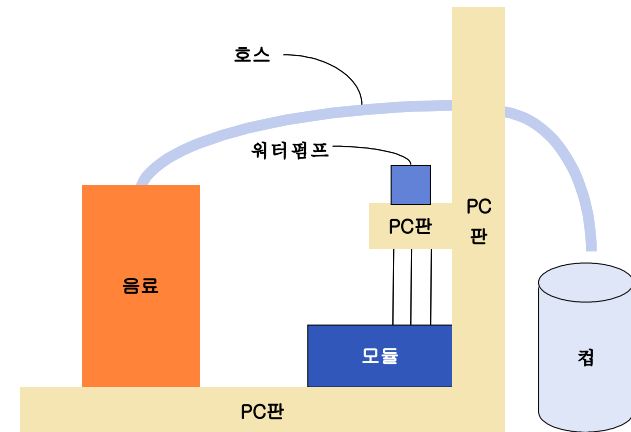
작품명 : 블루투스를 이용한 자동 컵테일 제조기

개발동기 및 목적 :

현재 코로나 19로 인한 공공시설 이용에 있어서 사람들은 접촉을 최소화하려 한다. 예를 들어 음료수 기기에서 음료를 내릴 때 손으로 눌러서 조작해야 하거나 컵테일을 제조하는 데 있어서 음료를 골라가면서 한 번에 하나씩 담아야 하는 번거로움이 있다. 시중에 나와 있는 정수기나 음료 기기를 어떻게 하면 이보다 더 편리하게 조작할 수 있을까 방법을 찾다가 블루투스를 이용한 자동 컵테일 제조기를 생각하게 되었다.

작품 구성 : 워터펌프(12V) 4개, PC(폴리카보네이트)판, 아두이노 우노, 실리콘호스(6*9mm), 릴레이모듈 (4구*2), 블루투스 모듈(HC-05), SMPS(12V), 가정용 케이블(220V), 점퍼선, 브레드보드, 저항($330\Omega \times 3$), 커패시터($1\mu F \times 4$)

작품 도안 :



소프트웨어 제작 : (아두이노 코딩)

```
cocktailgood

#include <SoftwareSerial.h>    // 소프트웨어 시리얼 라이브러리 불러옴

int BT_RXD = 12; // 아두이노측 수신부 RXD는 12번핀 (HC-06측 송신부 TXD와 연결)
int BT_TXD = 13; // 아두이노측 송신부 TXD는 13번핀 (HC-06측 수신부 RXD와 연결)

SoftwareSerial bluetooth(BT_RXD, BT_TXD);    // 소프트웨어 시리얼 bluetooth 객체 선언
//defining the pins used for each drink, you can changes this depending how you connect the pumps and the drinks.

int Whiskey = 2;
int LimeJuice = 3;
int OrangeJuice = 4;
int Sprite = 5;
int state = 0;
// how much time there is between switching to the next pump.
int TimeBetweenPours = 500;

void setup()
{

    Serial.begin(9600);
    bluetooth.begin(9600);
    Serial.println("The bluetooth gates are open.\n Connect to HC-05 from any other bluetooth device with 1234 as pairing key!");
    // Set pin 2-5 to output pins to drive the relay board
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);

    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);

}

void loop()

{

    if (bluetooth.available()) {
        state = bluetooth.read(); // Reads the data from the serial port
        Serial.println(state);
    }
    if (Serial.available()) {
        bluetooth.write(Serial.read());
    }

    // Cocktails:

    //Whiskey sprite
    if (state == 'a')
    {
        Serial.println("Whiskey Sprite in the making");
        Mix(3000, 0, 0, 5000); // Whiskey, LimeJuice, OrangeJuice, Sprite
    }

    // WhiOLa
    if (state == 'b')
    {
        Serial.println("WhiOLa in the making");
        Mix(3000, 1500, 3000, 0); // Whiskey, LimeJuice, OrangeJuice, Sprite
    }

    // WhiLaIte
    if (state == 'c')
    {
        Serial.println(" WhiLaIte in the making");
        Mix(2500, 0, 1500, 5000); // Whiskey, LimeJuice, OrangeJuice, Sprite
    }

}
```

```
// Testin and maintenance options

// Set-up Whiskey
if (state == 'm')
{
    Serial.println("Set-up Whiskey");
    Mix(1100, 0, 0, 0); // Whiskey, LimeJuice, Orange juice, Sprite
}

// Set-up Sprite
if (state == 'n')
{
    Serial.println("Set-up Sprite");
    Mix(0, 1100, 0, 0); // Whiskey, LimeJuice, Orange juice, Sprite
}

// Set-up LimeJuice
if (state == 'o')
{
    Serial.println("Set-up LimeJuice");
    Mix(0, 0, 1000, 0); // Whiskey, LimeJuice, Orange juice, Sprite
}

// Set-up Orange juice
if (state == 'p')
{
    Serial.println("Set-up Orange juice");
    Mix(0, 0, 0, 1000); // Whiskey, LimeJuice, Orange juice, Sprite
}

// Set-up all (intial first time after cleaning)
if (state == 'u')
{
    Serial.println("Set-up All");
    Mix(800, 800, 800, 800); // Whiskey, LimeJuice, Orange juice, Sprite
}

// clearing (intial first time after cleaning)
if (state == 116)
{
    Serial.println("Set-up All");
    Mix(800, 800, 800, 800); // Whiskey, LimeJuice, Orange juice, Sprite
}

}

void Mix(int DelayTimeWhiskey, int DelayTimeLimeJuice, int DelayTimeOrangeJuice, int DelayTimeSprite)
{

    state = 0;
    delay(800);
    digitalWrite(Whiskey, LOW);
    delay(DelayTimeWhiskey);
    digitalWrite(Whiskey, HIGH);
    delay(TimeBetweenPours);

    digitalWrite(LimeJuice, LOW);
    delay(DelayTimeLimeJuice);
    digitalWrite(LimeJuice, HIGH);
    delay(TimeBetweenPours);

}
```

```
digitalWrite(OrangeJuice, LOW);
delay(DelayTimeOrangeJuice);
digitalWrite(OrangeJuice, HIGH);
delay(TimeBetweenPours);

digitalWrite(Sprite, LOW);
delay(DelayTimeSprite);
digitalWrite(Sprite, HIGH);
delay(TimeBetweenPours);

Serial.println("완성!");
state = 0;
}
```

(App Inventor 코딩 사진자료)

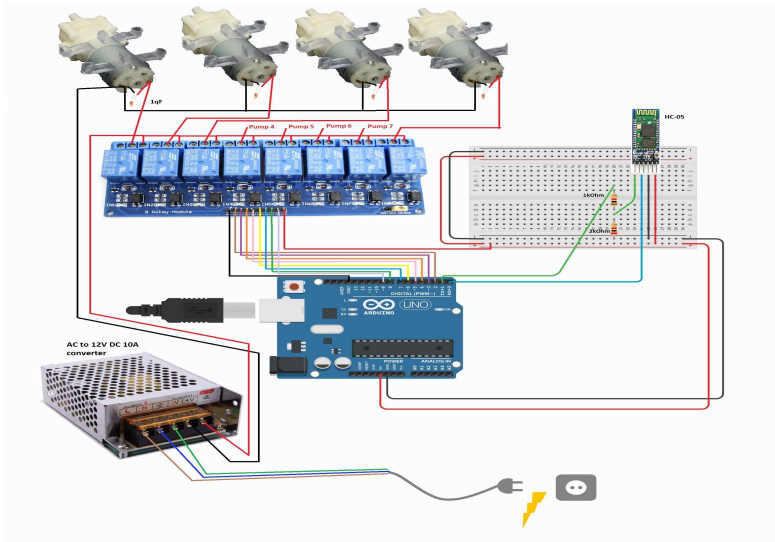


앱 화면

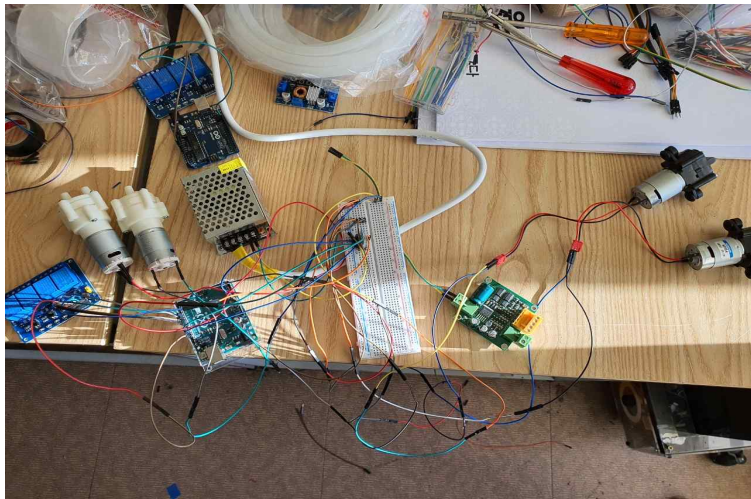


App Inventor로 제작된 앱을 핸드폰에 설치 후 실행한다.

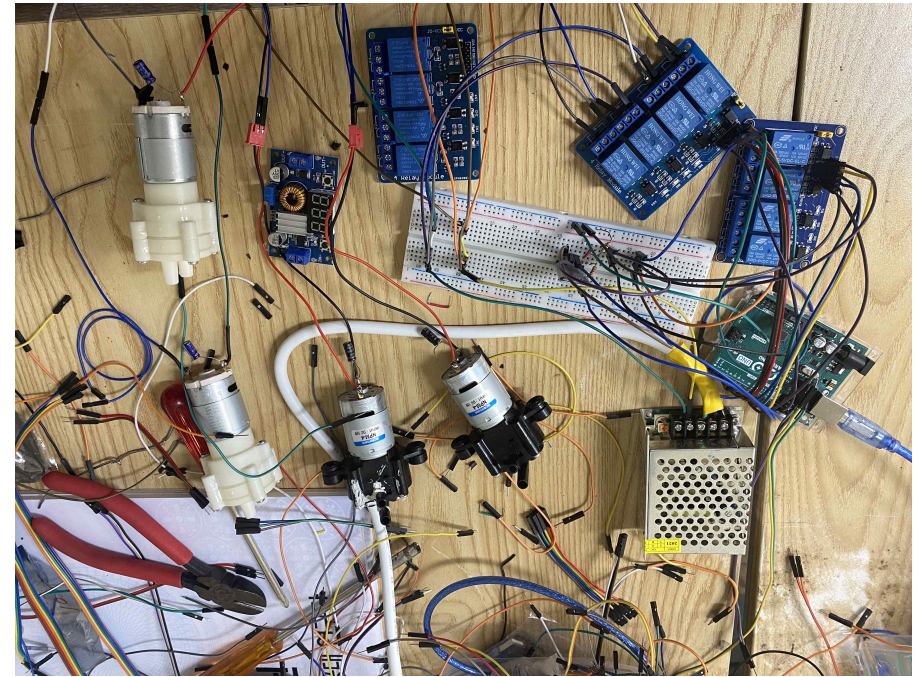
하드웨어 제작 : (모듈 설계)



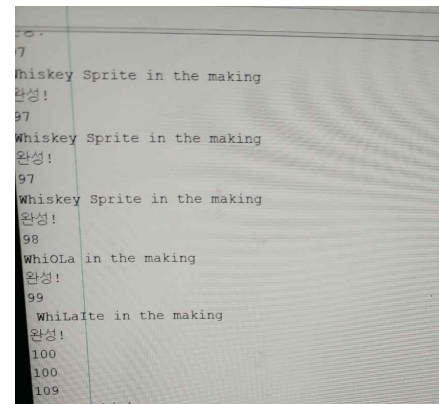
(모듈 제작)



(모듈 완성)

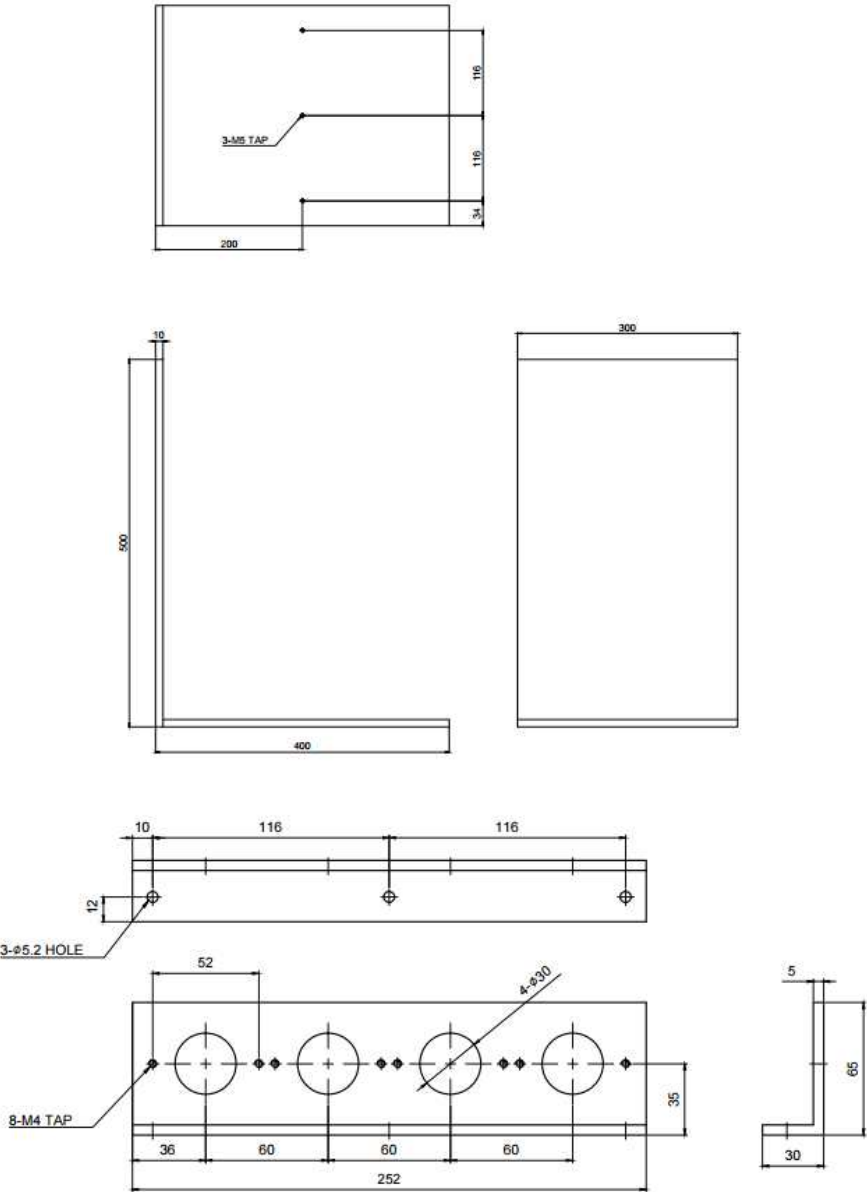


(모듈과 코딩이 잘 작동되는 것을 시리얼 모니터로 확인)



(PC판 제작)

도면 설계



작품 완성 사진



기대효과 :

스마트폰 터치만으로 컵테일이 제조되도록 블루투스 모듈을 이용하여 스마트폰 앱을 통해 원하는 컵테일을 선택해 제조할 수 있다. 또, 원하는 음료 종류와 섞는 비율을 다르게 해 다양한 조합의 음료를 설계하여 터치 조작만으로 여러 종류의 음료를 만들 수 있다. 그리고 음료를 받을 때 손으로 눌러서 조작하지 않아도 자동으로 시간 텀을 두고 음료가 나오기 때문에 간편하며 위생에 신경도 쓸 수 있다.

시행착오 :

1) 회로를 설계하면서 릴레이를 이용하여 가정용 220V를 12V로 출력을 바꿀 수 있는 SMPS를 사용하는 방법을 인터넷 자료들을 찾아보며 익혔지만, SMPS와 릴레이를 연결하는 부분에서 릴

레이가 제대로 작동하지 않았다. 하지만 많은 시간을 들여서 아두이노 모듈 회로 수정을 거듭한 결과, 다행히 릴레이 모듈이 정상적으로 잘 작동하였다.

2) 아두이노 블루투스 모듈과 App Inventor로 만든 앱이 제대로 연동이 되지 않은 경우가 잦았다. 이러한 오류를 해결하기 위해 많은 시행착오를 겪은 끝에 블루투스 모듈 연결에 문제가 있었다는 것을 알게 되었다. 회로를 설계하면서 블루투스 모듈의 TXD 핀과 RXD 핀을 헛갈리다 보니 블루투스 모듈이 서로 값을 전달하고 받지 못하는 결과를 초래하였다. 따라서 아두이노 블루투스 코딩 영상과 사용 방법을 유튜브를 통해 재차 익힌 후, 핀을 제대로 연결하여 문제를 해결하였다.

3) 아두이노 회로를 설계하면서 연결한 선이 자주 빠지고, 워터 펌프의 고장을 막기 위해 커패시터를 펌프에 달아놓는 과정에서 납땜한 선이 끊어지는 상황이 다수 발생하였다. 이를 해결하기 위해 절연 테이프를 감아 연결한 선을 고정시켰고, 납땜 과정에서 워터 펌프의 선이 끊어지지 않도록 섬세하게 작업하였다.

4) 아두이노 코딩에서 시리얼 모니터에 값이 출력되지 않는 경우가 발생하였다. 유튜브에 있는 아두이노 블루투스 모듈 제어 영상을 공부하면서 아두이노 코딩에서 시리얼 모니터에서 입력하는 값과 블루투스에서 입력하는 값을 받기 위한 함수를 제대로 선언하지 않았다는 것을 깨달았고, 이후 수정하여 시리얼 모니터에 입력값에 따른 올바른 출력값을 얻을 수 있었다.

참고자료 링크 :

https://youtu.be/7ztbMLb_dS4, <https://youtu.be/bJWsax6M8ro>, <https://youtu.be/VJnPI4i74Sg>
https://youtu.be/YJd_QDB992E