



2019 2학기 아두이노 프로젝트 최종 보고서

## ***DCS: Drawing Circle Simulation***

작성자 : 1학년 4반 2번 이름: 김석준

### 요약

본 프로그램은 기존의 매우 그리기 힘들고 정밀도가 떨어졌던 원 그리기에 아두이노를 대입하여 보다 기술적이고 첨단적인 원 그리기에 대한 방향성을 제시하였다는 점에서 의미가 있다. 각도도 정확히 못 잴채 정성적인 분석만을 했어야 했던 기존 아날로그적 특성에서 벗어나 보다 정밀하고 색도 바꿀 수 있는 첨단 기술로서의 도입을 발화시켰다는 점에 있어서 의미가 있으며, 또한 개발자의 입장에서 자신의 기본적인 아두이노 실력을 키울 수 있었다는 점에서 큰 의미가 있었다. 각종 아두이노의 기본적 원리를 토대로 필요한 코드들의 의미를 스스로 깨우치며 알아가는 과정에서 큰 성장을 하였다고 생각한다.



## I. 프로그램 개발의 동기 및 목적

칠판용 컴퍼스를 본 적이 있는가? 원을 그리기 어려워하는 선생님들과 몇몇 학생들을 위하여 제작된 아이디어 상품으로, 지금도 많은 선생님들에 의해 애용되고 있다. 그러나, 이러한 컴퍼스를 직접 사용해보지 않은 사람은 모르겠지만, 사용법이 상당히 난해한 편이다. 특히 종이 컴퍼스처럼 바닥에 고정할 수 있는 핀도 없기에 단순히 손의 힘만으로 이를 버텨 내어야 한다. 이러한 문제점에 본 개발자는 보다 쉽게 원을, 그리고 더 나아가 정확한 각도의 부채꼴을 그릴 수 있는 상품을 아두이노를 기반으로 제작하고자 하였고, 이 제품을 토대로 칠판에서의 원 그리는 방법에 대한 새로운 방향성의 제시와 아이디어에 대한 고찰의 대상이 됨으로써 향후 더 발전된 결과를 만들어내고자 하였다.

## II. 프로그램 소개

본 프로그램은 크게 두가지, 대부채꼴(Big Sector Form)과 소부채꼴(Small Sector Form)을 그릴 수 있다. 대부채꼴은 손잡이를 중심으로 회전하는 것이고, 소부채꼴은 손잡이와 반대인 축을 기점으로 작게 회전하는 것을 말한다. 둘 중 한가지를 선택하게 되면 세부적으로 그 각도와 색깔을 묻게 되는데, 각도의 경우 서보모터의 특성상 0~180도까지 가능하며, 색깔의 경우 노란색(Yellow)과 빨간색(Red)을 선택하는 것이 가능하다.

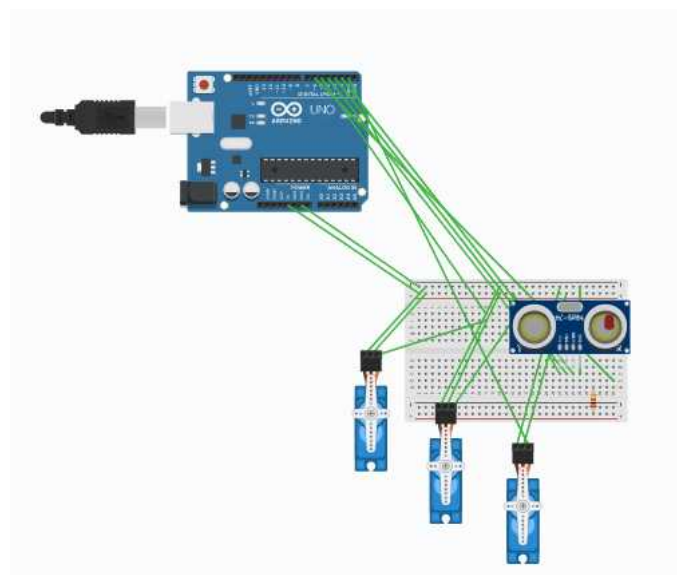
선택을 하고 나면 오토레벨링(autoleveling)을 진행하는데, 이는 연필(분필)이 바닥에 닿아 실제로 원을 유연하게 그릴 수 있는지 예측할 수 있는 척도로서, 보다 그림에 있어 정확도를 높이는데에 기여하였다. 최종적으로 그림을 그리면 서보모터가 원상복귀되고 처음부터 시작할 수 있게 된다.

## III. 하드웨어 소개

### 1 구현할 기능과 필요한 하드웨어 소개

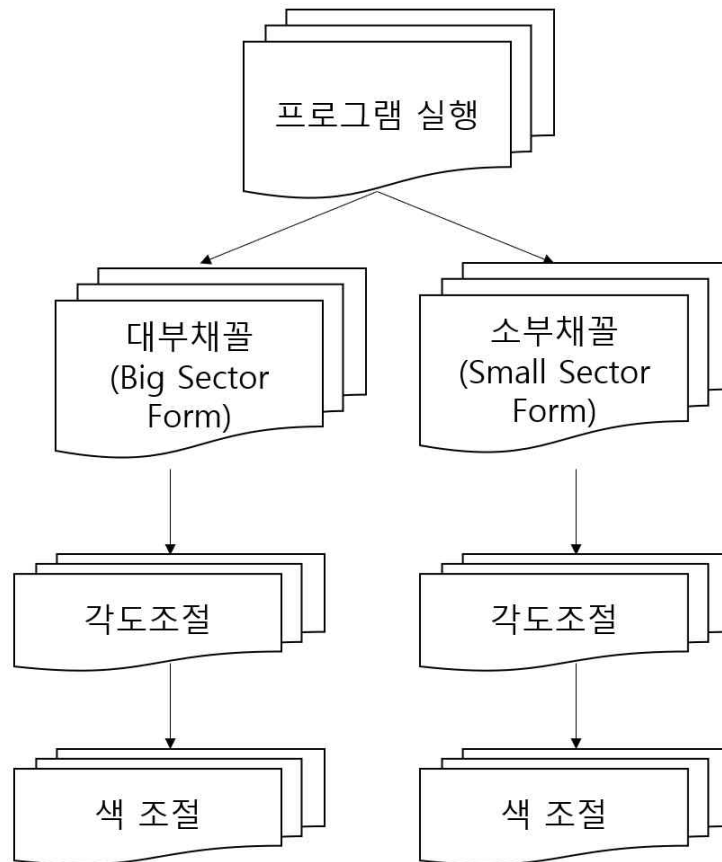
	구현할 기능	필요 하드웨어(사진첨부)
1	컴퍼스로서 회전의 역할을 담당	
2	오토레벨링 확인 작업 담당	 초음파센서
3	그림을 그리는 진행 여부 확인	

## 2. 전체 하드웨어 구성도





#### IV. 프로그램 플로우 차트(순서도)



#### V. 프로그램 사용법 및 주요 실행 결과

본 DCS 프로그램을 사용하기 위해서는 먼저 케이블을 통하여 컴퓨터와 시리얼 통신을 진행하여야한다. 케이블 연결 후 시리얼 모니터를 열게 되면 환영한다는 문구와 함께 두가지 선택지, 대부채꼴과 소부채꼴 중 하나를 선택할 수 있게 된다. 둘 중 하나를 선택하게 되면 그릴 부채꼴의 각도와 색깔을 선택하게 하는데, 이들을 모두 선택하면 오토레벨링(autoleveling)을 진행할 수 있게된다. 오토레벨링이란 연필(분필)이 바닥에 닿아 그림을 그릴 수 있는지에 대한 여부로, 초음파센서를 이용하여 바닥까지의 거리를 측정해 그 범위를 계산하여 그릴 수 있는지에 대한 여부를 알려준다. 만약 그릴 수 있다면 잠시의 여유시간 후에 본격적으로 그리기 시작하는데, 우선 LED를 켜서 그림이 그려지고 있음을 강조하고, 분필색을 결정한 후 서보모터를 입력된 각도만큼 돌려 부채꼴을 그리게 된다. 부채꼴을 전부 그리고 난 후 프로그램은 loop문을 따라 재시작된다.

#### VI. 기술적 요소

본 프로그램에서 사용된 주요 아이디어는 시리얼통신을 할때 사용자를 기다려 입력받을 수 있도록 하는 `while (Serial.available() == 0) {}` 이다. 이것을 통해 loop문을



돌지 않고 사용자를 기다리며 입력받기까지 대기가 가능하게된다. 또한, 초음파센서 관련 프로그램을 도입함으로 오토레벨링을 구현할 수 있었으며, for 문과 delay 함수를 이용한 서보모터의 제어도 본 프로그램에서 주요한 위치를 차지하고 있다.

## Ⅶ. 참고 문헌 및 웹 사이트

-[http://makeshare.org/bbs/board.php?bo\\_table=arduinosenor&wr\\_id=366](http://makeshare.org/bbs/board.php?bo_table=arduinosenor&wr_id=366) :

초음파센서 샘플 코드

-<https://codingrun.com/110> : 서보모터 제어

## Ⅷ. 개발 소감 ( 배운점, 어려웠던 점과 해결했던 내용, 아쉬운 점, 느낀점 )

개발을 하는 과정에서 부딪힌 어려움이 많았다. 특히 제품의 균형을 맞추는 것이나 서보모터를 제어하는 것 등 다양한 부분에서 어려움을 겪고 극복했으나, 그중에서도 집중을 요했던 것은 시리얼 통신상에서의 입력이었다. 시리얼통신의 입력은 loop문을 계속 순환하며 지속되는것이라 사용자의 입력을 기다려주지 않는다는 점에서 이를 제어하는 것이 쉽지 않았다. 이를 고민하고 친구와 상의해보는 과정에서 좋은 방법을 떠올리게 되었고, 이를 tinkercad 상에서 확인해봄으로써 해결하였는데 해결에 있어서 안도감과 자신감을 배울 수 있었다.

반면, 재료의 부족이 큰 아쉬움으로 남았다. 학생신분으로 사용할 수 있는 재료가 한정적이다보니 큰 강도를 요구해야하는 본 작품임에도 불구하고 나무와 글루건 정도로만 해결해야 했던 아쉬움이 있다. 추후 더 고강도의 재료로 개발자의 생각을 보다 구체적으로 구현하고자 한다.

어려움과 고된 시간이 필요했던 과제였으나, 직접 조사하는 과정에서 아두이노의 원리를 깨우치고 응용할 수 있는 기회가 되어 좋았다고 생각한다. 앞으로 더 좋은 기술과 향상된 지식을 바탕으로 더 완성도 높은 작품을 제작해보고 싶다는 생각이 든다.

## Ⅸ. 소스 코드

```
#include <Servo.h>
```

```
Servo servo1;
```

```
Servo servo2;
```

```
Servo servo3;
```

```
int angle1 = 0, angle2 = 0, angle3 = 0; //세 개의 서보모터의 각도 변수
```

```
int pin1 = 1, pin2 = 2, pin3 = 3; //세 개의 서보모터의 핀 변수
```

```
int ledPin = 6;
```

```
int trigPin = 4, echoPin = 5; //초음파 센서의 입출력 핀 변수
```

```
int height = 4; //오토 레벨링 높이(단위:cm)
```



```
int error = 1; //오토 레벨링 높이 오차(단위:cm)

void setup() {
    servo1.attach(pin1);
    servo2.attach(pin2);
    servo3.attach(pin3);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    int distance = 0; //autoleveling에 쓰일 변수
    long degree;
    char color;
    char c;
    Serial.println("Welcome to the Drawing Circle Simulation!");
    Serial.println("a. Big Sector Form");
    Serial.println("b. Small Sector Form"); //선택지
    while (Serial.available() == 0) {}
    c = Serial.read();

    if (c == 'a') {
        Serial.println("You chose a big sector form.");
        Serial.println("Please input the degree.(0~180)");
        while (Serial.available() == 0) {}
        degree = Serial.parseInt();
        Serial.println("Please input the color.");
        Serial.println("a. Red");
        Serial.println("b. Yellow");
        while (Serial.available() == 0) {}
        color = Serial.read();
        Serial.println("Setting is over. Autoleveling will start after 3 seconds.");
        while (1)
        {
            delay(3000);
            digitalWrite(trigPin, LOW);
            delayMicroseconds(2);
            digitalWrite(trigPin, HIGH);
            delayMicroseconds(10);
            digitalWrite(trigPin, LOW);
            distance = pulseIn(echoPin, HIGH) / 58.2;
        }
    }
}
```



```
Serial.print("Result of the autoleveling: ");
Serial.print(distance);
Serial.println("cm"); //오토 레벨링 결과 출력
delay(1000);
if (distance > height + error || distance < height - error) {
    Serial.println("Autoleveling failed.");
    Serial.println("a. First page");
    Serial.println("b. Restart autoleveling");
    while (Serial.available() == 0) {}
    c = Serial.read();
    if (c == 1) {
        Serial.println("Return to the first page.");
        delay(1000);
        break;
    }
    else {
        Serial.println("Restart autoleveling after 3 seconds.");
    }
}

else
{
    Serial.println("Autoleveling succeeded!");
    Serial.println("The sector form will be drawn after 3000 seconds.");
    delay(1000);
    Serial.print("3 ");
    delay(1000);
    Serial.print("2 ");
    delay(1000);
    Serial.println("1");
    delay(1000);
    analogWrite(ledPin, 255);
    if (color == 'a') {
        angle3 = 0;
        servo3.write(angle3);
    }
    else if (color == 'b') {
        angle3 = 180;
        servo3.write(angle3);
    }
    delay(1000);
    for (angle1 = 0; angle1 < degree; angle1++) {
```



```

        servo1.write(angle1);
        delay(100);
    }
    delay(1000);
    analogWrite(ledPin, 0);
    Serial.println("Succeeded in drawing.");
    Serial.println("Return to the first page.");
    servo1.write(0);
    servo2.write(0);
    servo3.write(0);
    delay(3000);
    break;
}
}
}

```

```

else if (c == 'b') {
    Serial.println("You chose a small sector form.");
    Serial.println("Please input the degree.(0~180)");
    while (Serial.available() == 0) {}
    degree = Serial.parseInt();
    Serial.println("Please input the color.");
    Serial.println("a. Red");
    Serial.println("b. Yellow");
    while (Serial.available() == 0) {}
    color = Serial.read();
    Serial.println("Setting is over. Autoleveling will start after 3 seconds.");
    while (1)
    {
        delay(3000);
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        distance = pulseIn(echoPin, HIGH) / 58.2;
        Serial.print("Result of the autoleveling: ");
        Serial.print(distance);
        Serial.println("cm"); //오토 레벨링 결과 출력
        delay(1000);
    }
}

```





```
if (distance > height + error || distance < height - error) {
  Serial.println("Autoleveling failed.");
  Serial.println("a. First page");
  Serial.println("b. Restart autoleveling");
  while (Serial.available() == 0) {}
  c = Serial.read();
  if (c == 1) {
    Serial.println("Return to the first page.");
    delay(1000);
    break;
  }
  else {
    Serial.println("Restart autoleveling after 3 seconds.");
  }
}

else
{
  Serial.println("Autoleveling succeeded!");
  Serial.println("The sector form will be drawn after 3000 seconds.");
  delay(1000);
  Serial.print("3 ");
  delay(1000);
  Serial.print("2 ");
  delay(1000);
  Serial.println("1");
  delay(1000);
  analogWrite(ledPin, 255);
  if (color == 'a') {
    angle3 = 0;
    servo3.write(angle3);
  }
  else if (color == 'b') {
    angle3 = 180;
    servo3.write(angle3);
  }
  delay(1000);
  for (angle2 = 0; angle2 < angle2; angle2++) {
    servo2.write(angle2);
    delay(100);
  }
  delay(1000);
}
```



```
    analogWrite(ledPin, 0);  
    Serial.println("Succeeded in drawing.");  
    Serial.println("Return to the first page.");  
    servo1.write(0);  
    servo2.write(0);  
    servo3.write(0);  
    delay(3000);  
    break;  
  }  
}  
}
```