

# 반려동물 자동 급식기

제어계측공학전공 최영천  
졸업작품 지도교수 김효성

## 제1장 작품의 개요

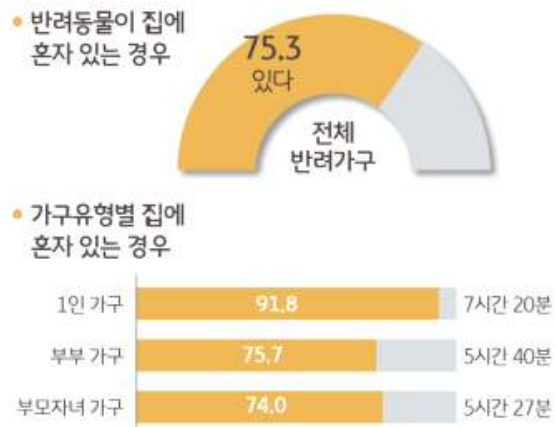
kb 경영연구소 2021 한국 반려동물 보고서에 따르면 2020년 말 기준 반려동물을 부양하는 가구가 604만 가구로 한국 전체 가구의 29.7%를 차지하고 반려인은 1448만 명으로 한국인 4명 중 1명 이상이 반려동물과 함께 살아가고 있다고 한다. 그리고 <그림 1> 농림축산식품부 동물등록정보 데이터를 가공한 요약정보를 보면 해마다 동물과 함께 살아가는 사람의 수가 늘어나고 있다는 것을 알 수 있다. 사실 인간과 동물은 이미 오래전부터 서로 상호작용을 통해 우리의 사회적 욕구를 만족시켜주고 끊임없이 영감을 주고받으며 문학과 예술의 발전에 영향을 끼쳤는데 최근 사회가 급속도로 발달하는 와중에 개인주의가 심화되면서 각박해진 많은 사람들은 정서적 교감을 위해 반려동물을 키우는 것이 더 흔해졌다. 이러한 흐름에서 기존 법률상 애완동물은 아직 물건으로 취급하고 있지만 사람들의 인식은 더 이상 물건이 아닌 가족으로 생각하는 사람들이 많아지고 있다. 반려동물이 사람처럼 대우받는 문화로 인간과 닮아간다는 뜻인 펫 휴머니제이션(pet humanization)이라는 신조어가 생기기도 하고 이 현상은 계속 확대되는 추세이다. 그런 만큼 각종 애로사항도 빈번히 발생하고 있는데 <그림 2>를 보면 반려동물이 집에 혼자 있는 경우가 전체 반려 가구의 75.3%이고 그 시간은 하루 평균 5시간 40분이다. 반려동물을 가족처럼 여겨 직접 돌보고 싶은 마음이 큰 반려인들이지만 바쁜 현대 사회에서는 마냥 쉽지 않은 일임을 알 수 있다. 그중 동물들과 떨어져 있을 때 급식을 원활하게 제공할 수 없다는 문제점이 반려인들이 많이 걱정하는 부분이라고 생각하였고 이를 해결하고자 사람이 설정한 주기에 자동으로 사료를 배식하는 반려동물 자동 급식기로 작품을 선정하였다.



<그림 1> 반려견 등록건수 및 증가율

### 반려동물이 집에 혼자 있는 시간은 하루 평균 5시간 40분

[단위: %]

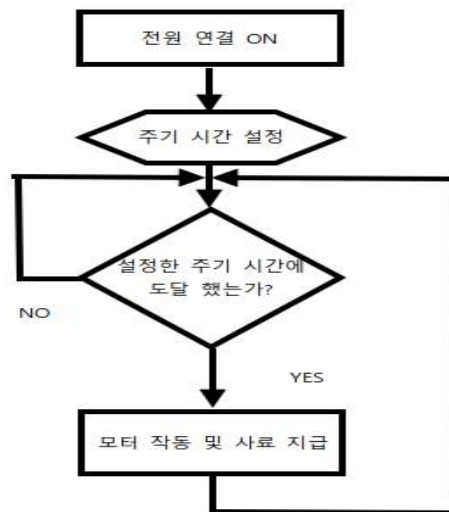


<그림 2> 반려동물 집에 혼자 있는 시간

## 제2장 작품의 원리 및 구성

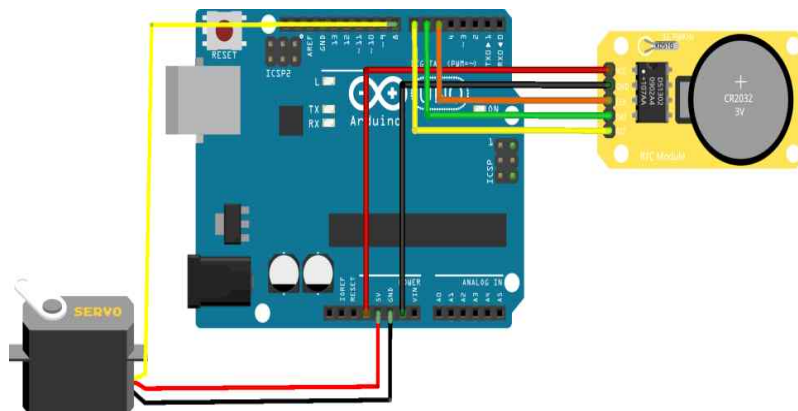
### 1. 작품의 원리

본 작품은 DS1302 라이브러리를 따로 불러오고 기준이 되는 초기 시간을 먼저 설정한다. 아두이노 업로드와 동시에 초기 시간을 DS1302 모듈로 들어가게 하여 기억시키면 설정한 초기 시간을 기준으로 시간이 흐르기 시작한다. 이후 원하는 사료 공급 주기를 설정하여 작품에 전원 인가 시 그 주기마다 서보모터를 동작시켜 사료 공급 마개를 회전 시킨 후 일정 시간이 지나면 원위치 시킨다. 이때 코딩을 통해 모터의 회전 각도와 다시 원위치 시키기까지의 지연 시간을 원하는 대로 제어할 수 있다.



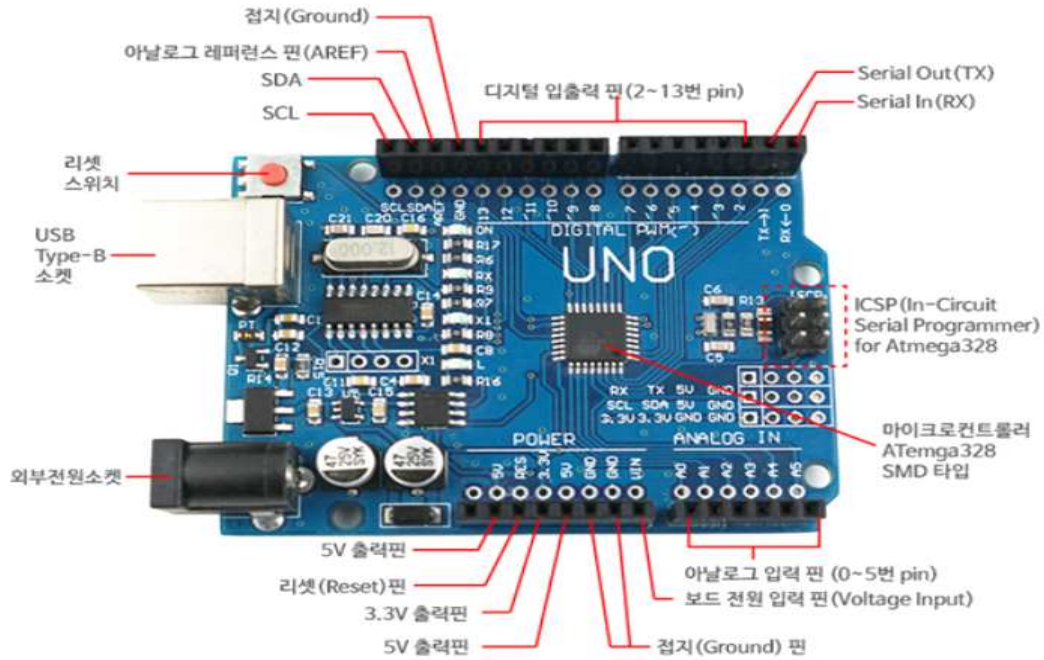
<그림 3> 반려동물 자동급식기 순서도

### 2. 작품의 구성



<그림 4> 작품의 회로도

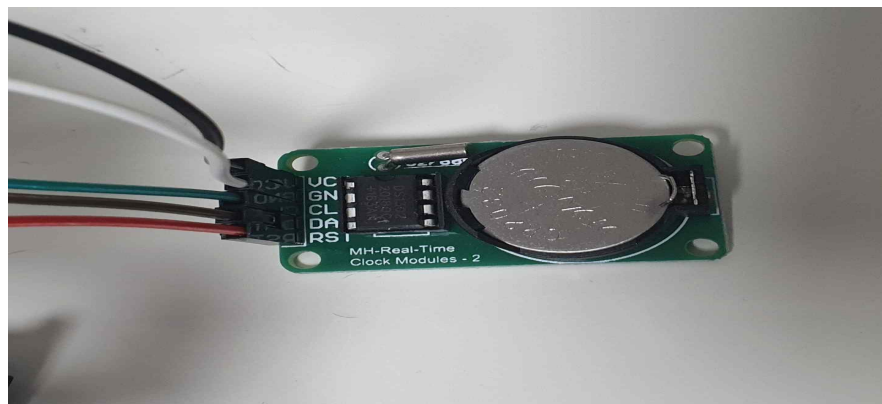
(1) 아두이노 우노



<그림 5> 아두이노 우노

아두이노는 대중적으로 가장 많이 사용되고 있는 오픈 소스 하드웨어로 다양한 웨어러블 컴퓨터 프로토타입 개발 및 차세대 디지털 기기 발명에 활용되고 있다. 하드웨어에 익숙지 않은 사람들도 손쉽게 제어할 수 있도록 설계되어 있어 원하는 제품을 만들 때 상당히 유용한 장치이다. 아두이노는 다양한 스위치나 센서로부터 입력 값을 받아들여 LED나 모터와 같은 전자 장치들로 출력을 제어함으로써 환경과 상호작용이 가능한 제품을 만들어 낼 수 있다. 또한 보드 설계도 오픈소스이기에 누구나 설계도에 따라 제품을 직접 제작할 수 있다.

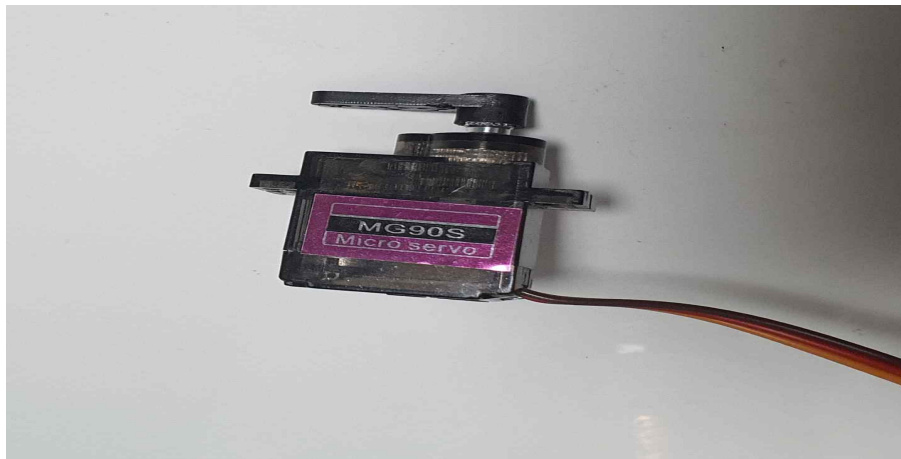
(2) DS1302 RTC 모듈



<그림 6> DS1302 RTC 모듈

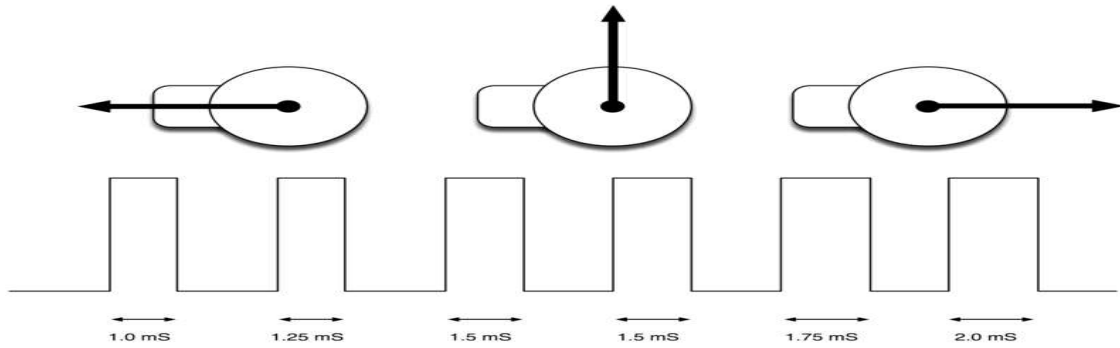
DS1302 RTC 모듈이란 DS1302라는 칩을 통해 현재 날짜, 시간, 요일 등을 설정하고 실시간으로 알아올 수 있는 모듈을 말한다. 총 5개의 핀으로 구성되어 있는데 각각 VCC, GND, CLK, DAT, RST이다. 이 모듈은 아두이노에 공급되는 전원과 상관없이 독립적으로 계속 작동되어야 하기 때문에 CR2032 3V 배터리를 필요로 한다. 소스 코드에서 초기 시간을 설정할 때 처음에 주석을 제거하고 업로드하고 이후 주석 처리하고 다시 업로드해야 한다. 그 이유는 설정한 코드에 주석 처리를 하지 않게 되면 아두이노가 켜질 때마다 초기 설정한 시간대로 다시 초기화 시키기 때문이다. 시간을 맞춰줄 때만 주석을 풀고 업로드한다.

### (3) MG90S 서보모터



<그림 7> MG90S 서보모터

서보모터란 일반 모터와는 달리 움직임을 지정하면 제어 계측 회로에 의해 입력된 만큼만 정확하게 움직일 수 있는 모터이다. 갈색 선은 접지, 빨간색 선은 전원, 주황색 선은 신호로 3개의 선이 구성되어 있다. 서보모터의 회전 각도는 PWM 방식으로 제어 즉 펄스의 폭 변조를 사용하여 정보를 보낸다는 것을 의미한다. 디지털 신호는 1과 0으로 이루어져 있는데 이 1과 0 신호를 얼마 동안 보내는지에 따라 서보모터의 위치가 결정되는 방식이다. 본 작품에서는 MG90S 서보모터를 사용하였는데 PWM 주기가 20ms로 이 단위로 제어 신호를 처리한다. 이 서보는 20ms 주기마다 펄스를 받게 되는데 만약 펄스가 1ms 동안 high 이면 각은 -90도로 왼쪽 끝까지에 위치하고 펄스가 1.5ms 동안 high 이면 각도가 0도로 중간에 위치하고 펄스가 2ms 동안 high 이면 각도가 90도로 오른쪽 끝까지에 위치한다. PWM은 이렇게 듀티 사이클을 조정해서 변조하는 방식이고 이 서보모터의 듀티 사이클은 1~2ms이다. MG90S 서보모터의 사양을 살펴보면 무게가 13.4g으로 작고 가볍기 때문에 좁은 장소에 잘 맞고 약 180도 회전할 수 있는 모터이다. 작동 전압은 4.8V~6.0V 사이이며 스톱 토크 즉 순간적인 최대 토크는 1.8kgf.cm(4.8V), 2.2kgf.cm(6V)이고 작동 속도는 0.1초/60도(4.8V), 0.08초/60도(6V)이다. 본 작품의 서보모터는 아두이노에 내장되어 있는 Servo.h라는 라이브러리를 사용하여 제어하였다. 본래 PWM의 듀티 사이클을 이용해 각도 조절을 해야 하지만 얼마만큼의 듀티 사이클을 적용해야 몇 도 움직이는지 구분하기 쉽지 않은 애로사항을 서보 라이브러리의 write() 함수를 사용해 각도 값을 매개변수로 넣어 조절을 가능하게 했다.



<그림 8> 펄스에 의한 모터의 위치 제어



#### MG90S servo, Metal gear with one bearing

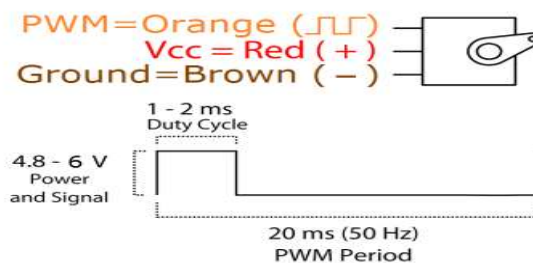
Tiny and lightweight with high output power, this tiny servo is perfect for RC Airplane, Helicopter, Quadcopter or Robot. This servo has *metal gears* for added strength and durability.

Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

#### Specifications

- Weight: 13.4 g
- Dimension: 22.5 x 12 x 35.5 mm approx.
- Stall torque: 1.8 kgf·cm (4.8 V), 2.2 kgf·cm (6 V)
- Operating speed: 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V)
- Operating voltage: 4.8 V - 6.0 V
- Dead band width: 5  $\mu$ s

<그림 9> MG90S 서보모터 데이터 시트 1



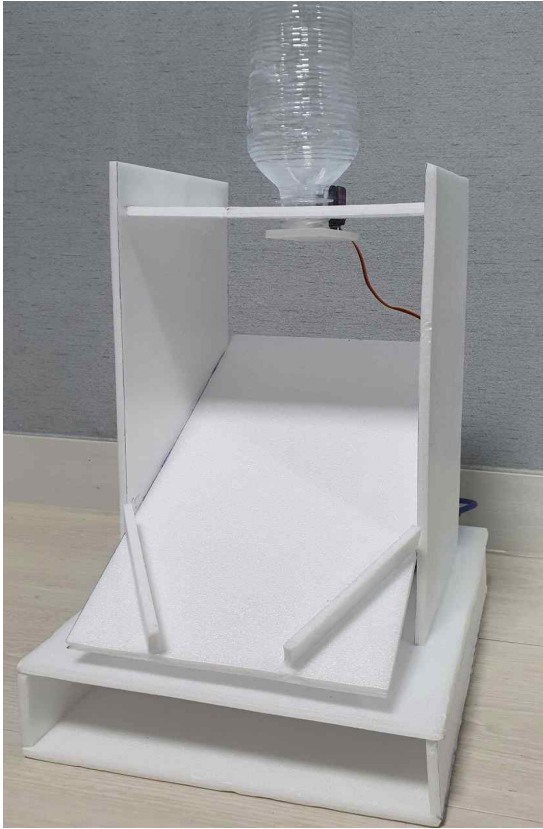
Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

<그림 10> MG90S 서보모터 데이터 시트 2

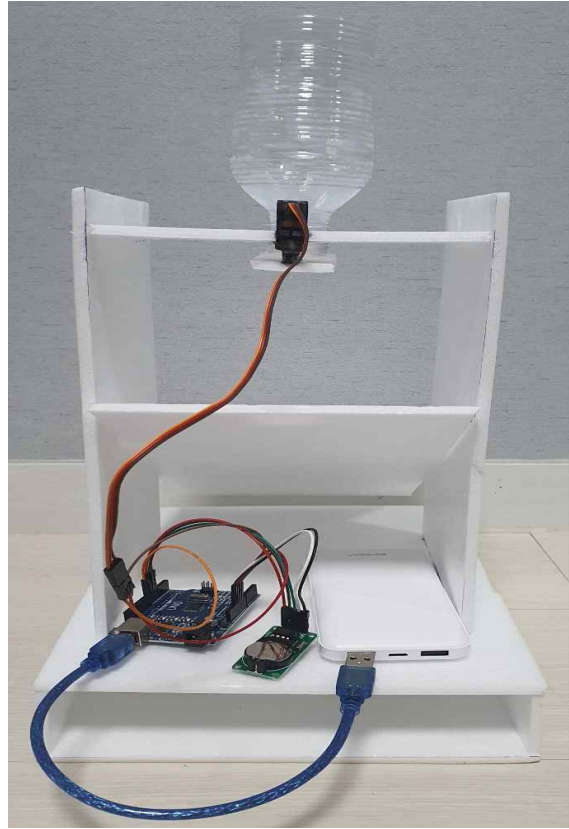


### 3. 모형제작

자동 급식기의 뼈대를 설계할 때 사료가 바깥으로 잘 빠져나올 수 있도록 하는 것과 서보모터에 달린 사료 마개가 페트병의 입구와 밀착할 수 있도록 두 가지를 주의하여 설계했다.



<그림 11> 자동 급식기 모형 (앞)



<그림 12> 자동 급식기 모형 (뒤)

## 제3장 결론 및 작품제작 후기

### 1. 결론

본 작품은 반려 가구가 증가하는 추세 속 바쁜 현대인들이 반려동물에게 원활한 사료 지급을 어려워하는 문제점을 해결하고자 사용자가 설정한 주기에 맞춰 서보모터를 작동시켜 사료가 지급되게끔 하였고 RTC 모듈과 CR2032 배터리를 이용해 메인 시스템인 아두이노의 전원이 차단되더라도 초기 설정 시간을 기억시켜 다음 전원이 들어왔을 때도 서보모터에 정확한 시간을 전달할 수 있도록 프로그래밍하여 제작하였다.

### 2. 작품제작 후기

작품을 만들기 전 이런 경험이 처음이기에 무엇부터 시작해야 할지 막막하고 두려웠다.

그렇지만 작품을 설계하고 부품을 직접 찾아 구입하여 하나하나씩 제작하는 과정을 몸소 느끼니깐 재밌었다. 물론 온전히 순탄했던 건 아닌데 작품을 만드는 도중에 뼈대 제작이 생각보다 설계대로 안되어서 2 번의 시행착오가 있었고 서보모터를 동작하는 과정에서 처음 사용한 SG90S 서보모터의 기어 부분이 불량이라 오차가 생겼던걸 코딩 문제로 착각하여 해매었다. 결국 메탈기어 서보모터인 MG90S로 교체하여 문제를 해결하였다. 작품을 제작하면서 아쉬운 부분은 아두이노 우노가 만들고 싶은 것을 만드는데 상당히 확장성이 있다는 것을 깨달았는데 코딩 소스를 작성하는 기술이 많이 미흡하여 여러 가지 기능을 좀 더 추가하지 못한 게 아쉬웠다. 비록 미흡하게 느껴지는 작품 제작이었지만 생각한 대로 구현을 해나갔던 과정이 값진 경험이었다고 생각한다.

## 부록

### 1. 아두이노 코딩

```
#include <Servo.h> // 서보모터 라이브러리 불러오기
#include <DS1302.h> //DS 1302 라이브러리

const int RST = 7; // DS 1302의 핀 설정
const int DATA = 6;
const int CLOCK = 5;

DS1302 rtc(RST, DATA, CLOCK); // DS1302 모듈을 rtc 오브젝트로 설정

Servo servo; // 서보모터 이름 설정

int servoMotorPin = 8; // 서보모터 신호 핀 설정

void setup() {
    /* 아래 1~3번의 작업은 초기 날짜를 설정하고 업로드 한후,
    주석 처리하여 다시 업로드 한다.
    이유는 아두이노가 off->on 될 때 마다 시간을 초기화 하기 때문이다.*/

    /*1. rtc에 쓰기방지 OFF, 정지기능 OFF*/
    //rtc.writeProtect(false);
    //rtc.halt(false);

    /*2. 새로운 시간과 날짜를 세팅
    초기 시간을 2022년 10월 09일 20시 40분 00초 일요일로 설정.*/
    //Time t(2022, 11, 07, 17, 50, 00, Time::kMonday);

    /*3. 현재 시간을 DS 1302에 입력.*/
    //rtc.time(t);

    // 서보모터 8번 핀 사용
    servo.attach(servoMotorPin);
}

void loop() {
    // DS 1302로 부터 현재 시간을 읽어온다.
    Time t = rtc.time();

    // Ex) if(t.hr%3==0 && t.min==0 && t.sec==0) 시, 분, 초는 배수를 이용해 원하는 주기 시간을 설정할 수 있다.
    if(t.sec%5==0) // 초가 5의 배수이므로 5초 간격으로 동작 설정
    {
        // 서보모터 동작
        servo.write(33); // 33도 회전 후 반대로 33도 회전 원위치
        delay(1000);
        servo.write(-33);
        delay(1000);
        servo.write(0);
        delay(1000);
    }
}
```

<그림 13> 아두이노 코딩 소스

## 참고문헌

- [1] kb 경영연구소 2021 한국 반려동물보고서,  
<https://www.kbfg.com/kbresearch/report/reportView.do?reportId=2000160>
- [2] 에듀로봇, <https://blog.naver.com/helloroborts/222602257364>
- [3] 디바이스마트, [https://blog.naver.com/no1\\_devicemart/221623220179](https://blog.naver.com/no1_devicemart/221623220179)
- [4] INFOINNO 디지털 세상 탐방기, <https://blog.naver.com/infoinno1010/222781011087>
- [5] 브런치, <https://brunch.co.kr/@font/121>