

밥 줘, 시바



| | |
|---------|------------------------------|
| M D P 명 | 정보통신 프로젝트 실무 네트워크 프로젝트 실무 |
| 학 과 학 년 | 정보통신기기과 3학년 |
| 학 생 성 명 | 허성진, 문은서, 정성운, 윤수화 |
| 지 도 교 사 | 김진섭, 최보람, 최윤희 |



전자통신분야 마이스터고등학교
인천전자마이스터고등학교
<http://intec.icehs.kr/>





Creative Commons

저작자표시(by)-비영리(nc)-변경금지(nd) 3.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

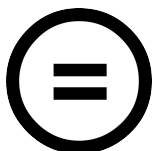
다음과 같은 조건을 따라야 합니다.



저작자표시: 귀하는 원저작자를 표시하여야 합니다.



비영리: 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지: 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들을 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

- <https://creativecommons.org/licenses/by-nc-nd/3.0/deed.ko>

2019학년도 제 10회 MDP 과제발표회 작품보고서

밥 쥐, 시바

1. 연구 윤리 서약서

2019학년도 정보통신기기과 8기 학생으로서 이 작품보고서 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교사의 지도를 받아 정직하고 엄정한 연구를 수행하여 작품보고서를 작성한다.

둘째, 작품보고서 작성 시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

2019년 11월 20일

학과: 정보통신기기과

지도교사: 김진섭, 최보람, 최윤희

성명: 허성진

문은서

정성운

윤수화

2. 작품보고서 제출

이 보고서를 제 10회 MDP 과제발표회 밥 쥐, 시바 작품보고서로 제출합니다.

2019년 11월

인천전자마이스터고등학교

정보통신기기과

허성진, 문은서, 정성운, 윤수화

이 보고서를 제 10회 MDP 과제발표회 밥 쥐, 시바 작품보고서로 인준함.

2019년 11월

학과장: 조 우 현 (성명)

교 사: 김 진 섭 (성명)



목 차

| | |
|-----------------------|-----------|
| 목 차 | i |
| 표 목 차 | ii |
| 그림목차 | iii |
| 국문요지 | iv |
| | |
| I. 서론 | |
| 1. 연구 배경 | 1 |
| 2. 연구 방법 | 1 |
| 3. 관련 전공 | 2 |
| | |
| II. 설계 및 구현 | |
| 1. 기구물 | 3 |
| 2. 하드웨어 | 5 |
| 3. 소프트웨어 | 10 |
| 4. 서버 | 22 |
| 5. 웹페이지 | 26 |
| | |
| III. 결론 | |
| 1. 실무결과 | 30 |
| 2. 시사점 | 31 |
| 3. 느낀점 | 31 |
| | |
| IV. 활동사진 | 33 |

※ 참고문헌





표 목 차

| | |
|-------------------------------|---|
| <표 I -1> 작품계획 | 1 |
| <표 I -2> 학과 교육과정 | 2 |
| <표 I -3> 학과교육과정 외 관련 전공 | 2 |
| <표 II -1> 부품 및 용량 | 6 |



그림 목 차

| | |
|--|----|
| [그림 I -1] 밥 줍, 시바 플랫폼 | 3 |
| [그림 II -2] 밥 줍, 시바 설계도 | 3 |
| [그림 II -3] 기구물 사진 | 4 |
| [그림 II -4] 블록도 | 5 |
| [그림 II -5] LM2576을 이용한 5V 전원회로 회로도 | 5 |
| [그림 II -6] LM2576을 이용한 5V 전원회로 PCB도면 | 6 |
| [그림 II -7] dsm42wm61a | 7 |
| [그림 II -8] fl42sth33-0956a | 7 |
| [그림 II -9] MG90 | 8 |
| [그림 II -10] DRV8825 | 8 |
| [그림 II -11] 마이크로 스위치 | 9 |
| [그림 II -12] 순서도 | 10 |
| [그림 II -13] 상속관계 | 11 |
| [그림 II -14] 안드로이드 메인 화면 | 21 |
| [그림 II -15] 안드로이드 마켓 화면 | 22 |
| [그림 II -16] 네트워크 구축 | 22 |
| [그림 II -17] DB 구성 | 23 |
| [그림 II -18] 사료 마켓 database | 23 |
| [그림 II -19] 간식 마켓 database | 24 |
| [그림 II -20] 간식 준 시간 database | 24 |
| [그림 II -21] 클라우드 컴퓨팅 | 25 |
| [그림 II -22] 협력 관계 | 26 |
| [그림 II -23] 웹 페이지 | 26 |
| [그림 II -24] 웹 페이지 메인 화면 | 27 |
| [그림 II -25] 웹 페이지 솔루션 화면 | 27 |
| [그림 II -26] 웹 페이지 실시간영상 화면 | 28 |
| [그림 II -27] 웹 페이지 사진 화면 | 28 |
| [그림 II -28] 웹 페이지 배급상태 화면 | 29 |
| [그림 IV -1] 조원 활동사진 | 33 |

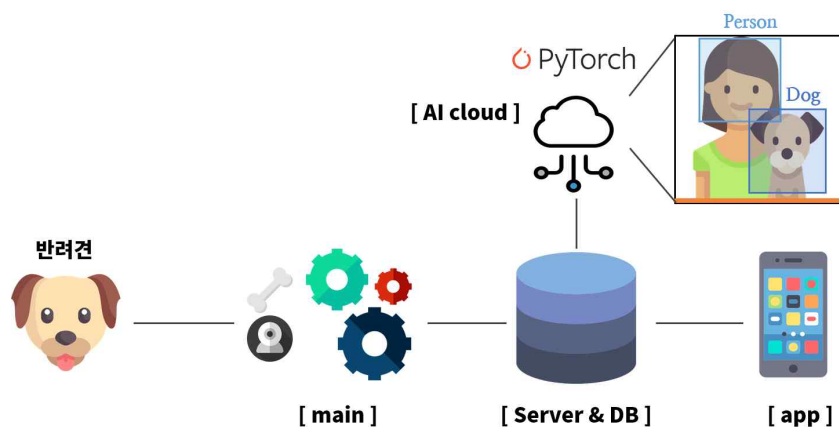
국문요지

1. 작품설명

가. 개요

국민 1000만 명 이상이 반려동물을 키운다. 그만큼 우리 사회에는 많은 반려동물, 특히 반려견들과 함께 살고 있다. 하지만 많은 반려견이 분리불안에 힘들어한다. 집에 혼자있는 반려견을 위해 AI(Pytorch)와 모터, 스트리밍 서비스를 활용하여 사료를 던져주고 CCTV로 조회할 수 있는 스마트 통합 플랫폼을 제작하고자 했다.

나. 구성



2. 제작 과정

가. 프로젝트 기획 및 계획서 작성

나. 소프트웨어 제작 및 전원회로 등의 기본 회로 제작

다. 라즈베리파이3를 이용해 모터, 스위치 구동 테스트

라. 각종 모터와 아크릴, 프로파일, 포맥스를 이용하여 하드웨어 제작

마. AI기능 클라우드 서비스화 및 SW+HW 결합

바. 최종 동작 및 보완

3. 동작 설명

가. “밥 줘, 시바” 어플을 실행

나. 어플을 통해 CCTV 조회 및 신호를 서버로 전송

다. 서버의 신호를 받아 그에 맞는 동작 실행

라. [자동 모드]를 실행하면 AI클라우드로 사진을 보내 인식

마. 반려견을 향해 사료 한 알을 던져준다

4. 기대 효과

가. 스마트 펫 통합 플랫폼을 구현함으로써 반려견과 주인 모두에게 편리한 삶을 제공한다.

나. 반려견의 분리불안을 해소, 주인이 반려견을 향해 갖는 걱정을 통합플랫폼의 AI기능을 이용하여 편리하게 해소한다.

I. 서론

1 연구 배경

지금 우리나라 인구 중 1000만 명이 반려동물을 키운다. 그렇기에 반려동물, 특히 반려견은 우리 사회에서 중요한 가족의 역할이 되었다. 하지만 직장인의 경우 반려견을 집에 홀로 두는 경우가 대부분이다. 반려견이 집에 혼자 오래 있으면 분리불안에 걸릴 확률이 높아진다. 이러한 문제를 해결하기 위해서 집에 혼자 있는 반려견을 놀아주고 관리해줄 스마트 통합 플랫폼을 만들면 좋겠다고 생각을 하여 제작하게 되었다.

2 연구 방법

<표 I -1> 작품계획

| 내 용 | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 | 10월 | 11월 |
|-----------------|----|----|----|----|----|----|-----|-----|
| 작품 계획 | | | | | | | | |
| 부품 조사 | | | | | | | | |
| H/W | | | | | | | | |
| S/W | | | | | | | | |
| 외형 제작 | | | | | | | | |
| H/W 와 S/W 연동 | | | | | | | | |
| 문제점 수정 | | | | | | | | |
| PPT 제작 | | | | | | | | |
| 보고서 작성 | | | | | | | | |

3 관련 전공

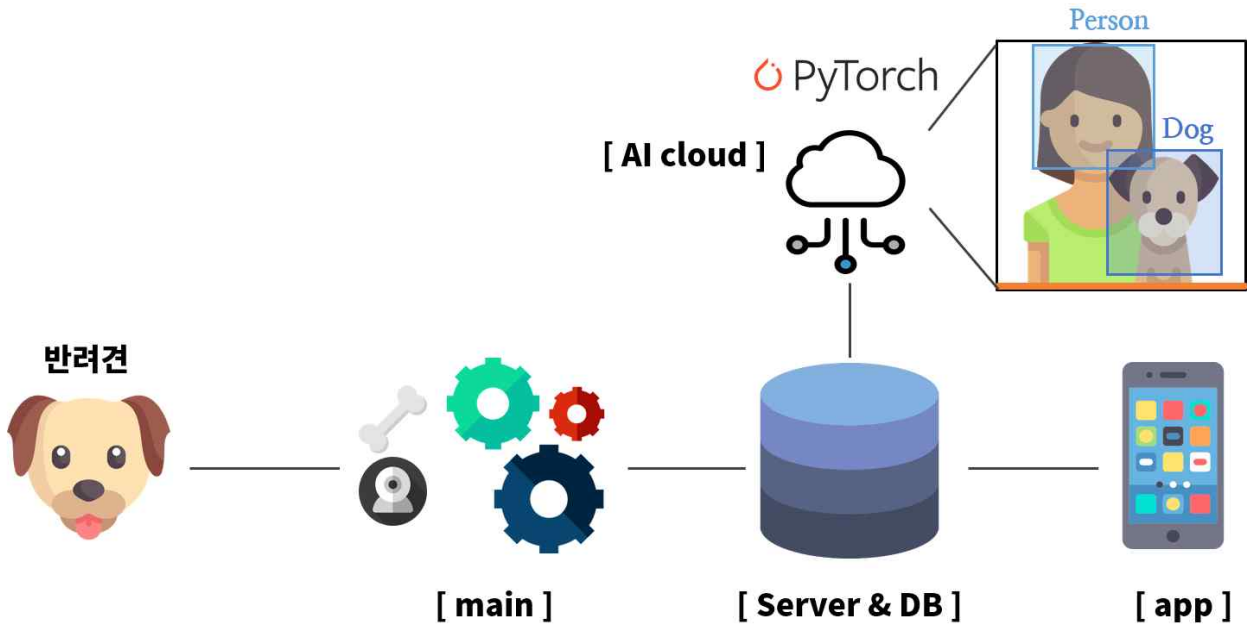
<표 I -2> 학과 교육과정

| 학년 | 과 목 | | 내 용 | 비고 |
|----|-----------------|---------------|--------------------------------|----|
| 1 | 전자 통신 기초 | | · 기초적인 회로해석을 배움 | |
| | 프로그래밍(JAVA) | | · 기초적인 C언어, JAVA를 배움 | |
| 2 | 마이크로프로세서 기초 | MPU | · 모터 구성법을 배워 활용 | |
| | | Android | · 앱 개발 능력을 통해 어플 제작 | |
| | 정보통신 | 통신망분배 | · 선 정리 및 케이블 구성능력 활용 | |
| | | Packet Tracer | · 서버 통신 등 네트워크 구축능력을 활용하여 서버구축 | |
| | 정보통신 기기 설계 | | · 회로 해석 및 회로 설계를 활용하여 전원회로 제작 | |
| 3 | 정보통신기기 소프트웨어 개발 | | · 자바스크립트를 활용하여 웹 개발 | |
| | 네트워크 구축 | | · 서버 통신 등 네트워크 구축능력을 활용하여 서버구축 | |
| | 마이크로프로세서 제어 | | · 모터 구성법을 배워 활용 | |
| | 정보통신기기 | | · 프로젝트를 통한 작품 제작 | |

<표 I -3> 학과교육과정 외 관련 전공

| 제 목 | 내 용 | 비고 |
|---------|---|----|
| node.js | 이벤트 기반 비동식 방식의 고성능 네트워크 서버 | |
| Pytorch | Python을 위한 오픈소스 머신 러닝 라이브러리로 orch를 기반으로 하며, 자연어 처리와 같은 애플리케이션을 위해 사용 | |
| FFmpeg | 디지털 음성 스트림과 영상 스트림에 대해서 다양한 종류의 형태로 기록하고 변환하는 컴퓨터 프로그램 | |
| Yolo | 실시간 개체검출(Real-Time Object Detection)의 최첨단 기술 딥러닝 라이브러리 | |
| Python | 다양한 플랫폼에서 쓸 수 있고, 라이브러리가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가 | |
| 라즈베리파이 | 마이크로프로세서, 메모리, 입출력 연결단자 등을 하나의 회로로 구성한 초소형 컴퓨터 | |

II. 설계 및 구현

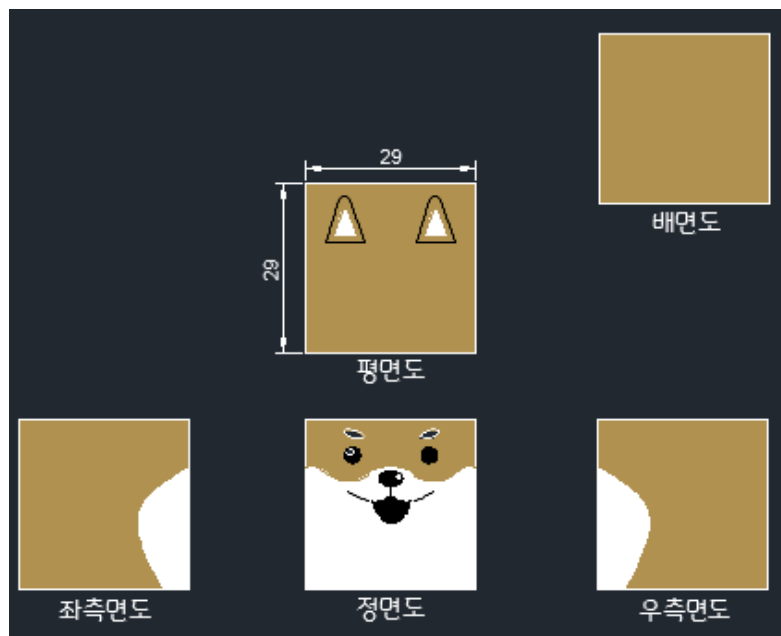


[그림 II-1] 밥 쥐, 시바 플랫폼

1 기구물

1. 설계도

미국의 오토데스크(Autodesk) 사에서 개발한 소프트웨어로, 2차원과 3차원 디자인 제도를 위한 CAD응용 소프트웨어이다.



[그림 II-2] 밥 쥐, 시바 설계도

밥 줘, 시바

“밥 줘, 시바” 기구물은 정사각형 모양으로 잡았으며, 기구물 상하좌우에 시바를 그려 반려견이 조금이라도 친숙하게 다가갈 수 있게 설계를 하였다.

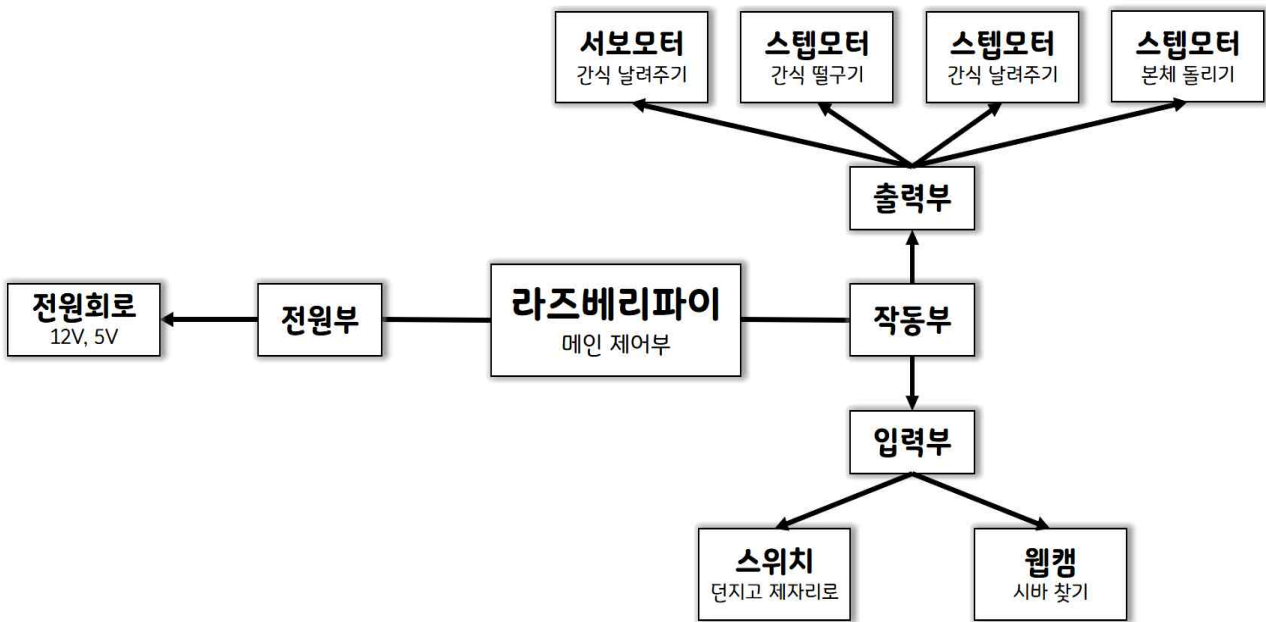
2. 기구물 사진



[그림 II-3] 기구물 사진

2 하드웨어

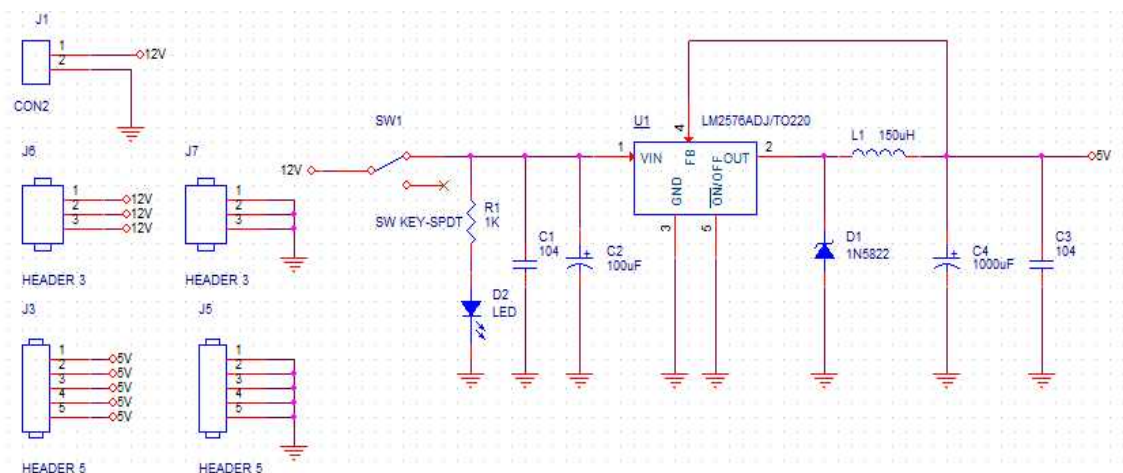
1. 블록도



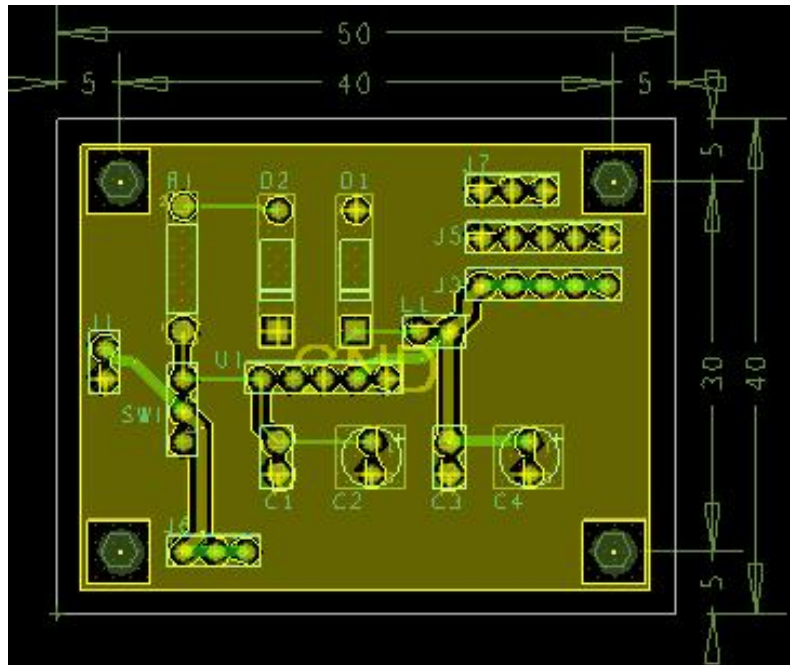
[그림 II-4] 블록도

2. 회로도

OrCAD는 캐이던스디자인시스템 사의 데스크톱용 전자설계 자동화 소프트웨어 도구이다. 전자 회로설계와 회로 시뮬레이션, 회로기관 설계 등이 쉽게 이루어져 강력한 기능에 비해 사용이 간편하여 널리 이용되고 있는 전자회로설계 전용 프로그램이다. 주로 전자회로도 및 인쇄회로기판 등을 설계하기 위해 전자회로 설계 공학자나 전자공학 기술자들에 의해 사용된다.



[그림 II-5] LM2576을 이용한 5V 전원회로 회로도



[그림 II-6] LM2576을 이용한 5V 전원회로 PCB도면

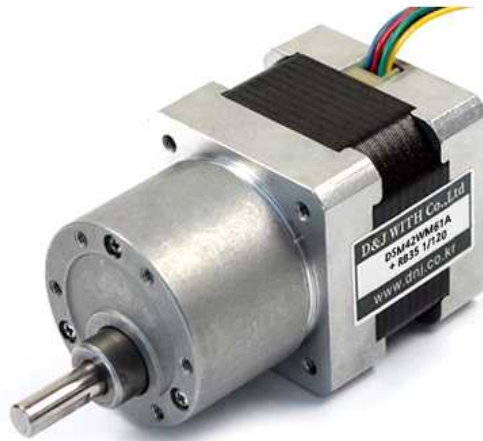
<표 II-1> 부품 및 용량

| 부품 | 사용 부품/용량 |
|----------|---------------------------|
| IC | LM2576 |
| 초트키 다이오드 | 1N5822 |
| 전해 콘덴서 | 1000[μF] (2) |
| 마일러 콘덴서 | C104 / 0.1[μF] (2) |
| 코일 | 100[μF] |
| LED | 빨강 |
| 저항 | 1[k Ω] |

3. 사용된 모듈

가. 스텝모터

1) dsm42wm61a



[그림 II-7] dsm42wm61a

dsm42wm61a는 기어드스텝모터이다. 일반 스텝모터에 비해 속도는 낮지만, 그만큼 강한 토크를 보여준다. “밥 쥐, 시바”의 본체 회전을 위해서는 강한 토크가 필요하기에 기어드 스텝모터를 찾아서 사용하게 되었다.

2) fl42sth33-0956a



[그림 II-8] fl42sth33-0956a

fl42sht33-0956a는 하이브리드 42각 2상 하이브리드 스텝모터이다. 작은 사이즈에 1.8도의 세밀한 분할각을 가지고 있어 범용적으로 동작에 사용할 수 있어서 이 모터를 사용하게 되었다.

밥 쥐, 시바

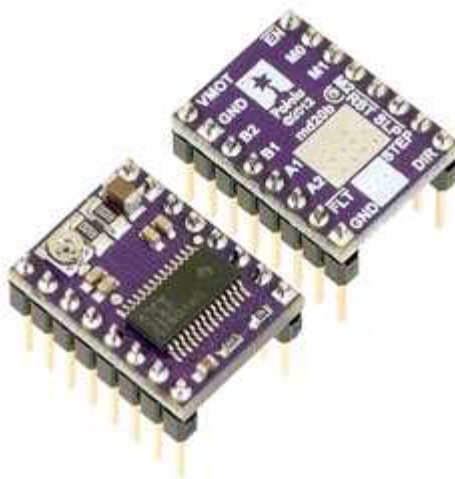
나. 서보모터



[그림 II-9] MG90

서보 모터는 일반적으로 3개의 연결선(VCC, GND, 그리고 DATA(제어선))을 가진다. 전원을 모터에 연결하고 제어선으로 위치를 조절하는 구조이므로 간단히 위치를 제어할 수 있다. 투석기를 지탱하기 위해 높은 토크를 가진 MG90을 사용했다.

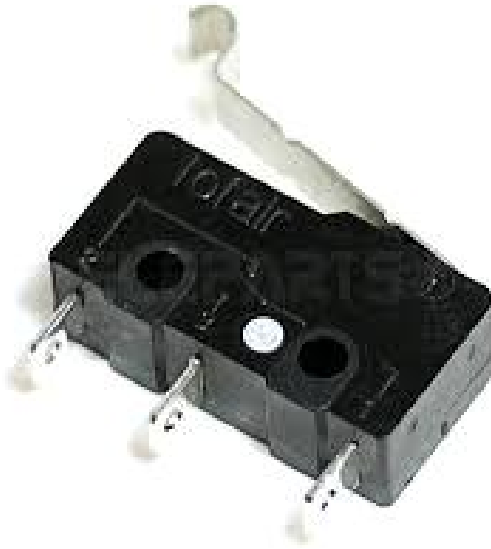
다. 스텝모터 드라이버



[그림 II-10] DRV8825

작품에서는 DRV8825를 사용하였는데, 이유는 출력 전류이다. 최대전류와 최대전압이 2.5A와 45V로 높은 수치이다. 그리고 안정적인 동작을 하기에 사용하게 되었다. “밥 쥐, 시바” 프로젝트에서 사용하는 모든 스텝 모터 드라이버는 MS1값을 High로 입력하여 Half Step으로 사용한다. 즉, 스텝각(분할각)을 0.9도로 사용하여 400번의 step이 360도를 회전하게 된다.

다. 마이크로 스위치

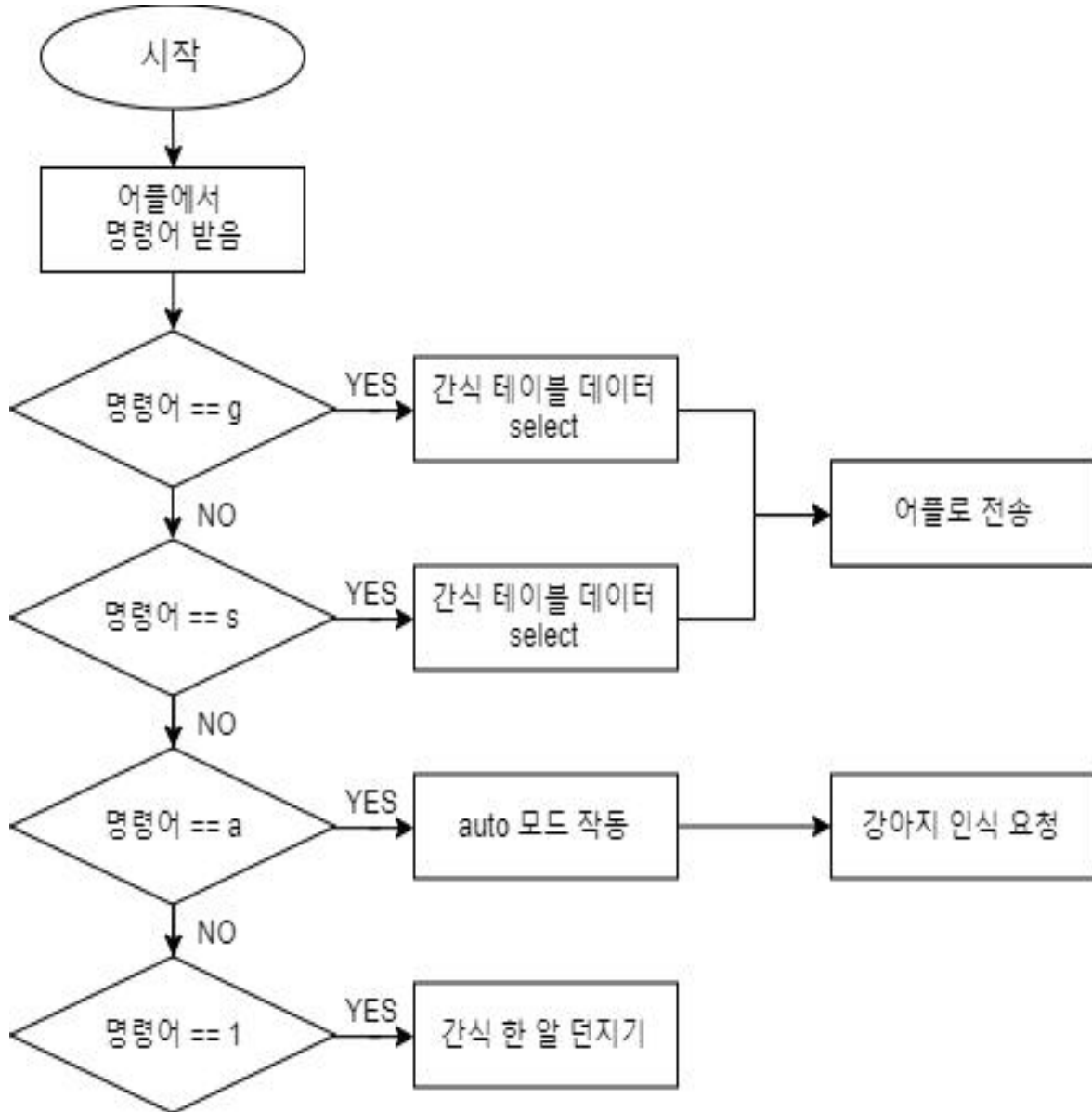


[그림 II-11] 마이크로 스위치

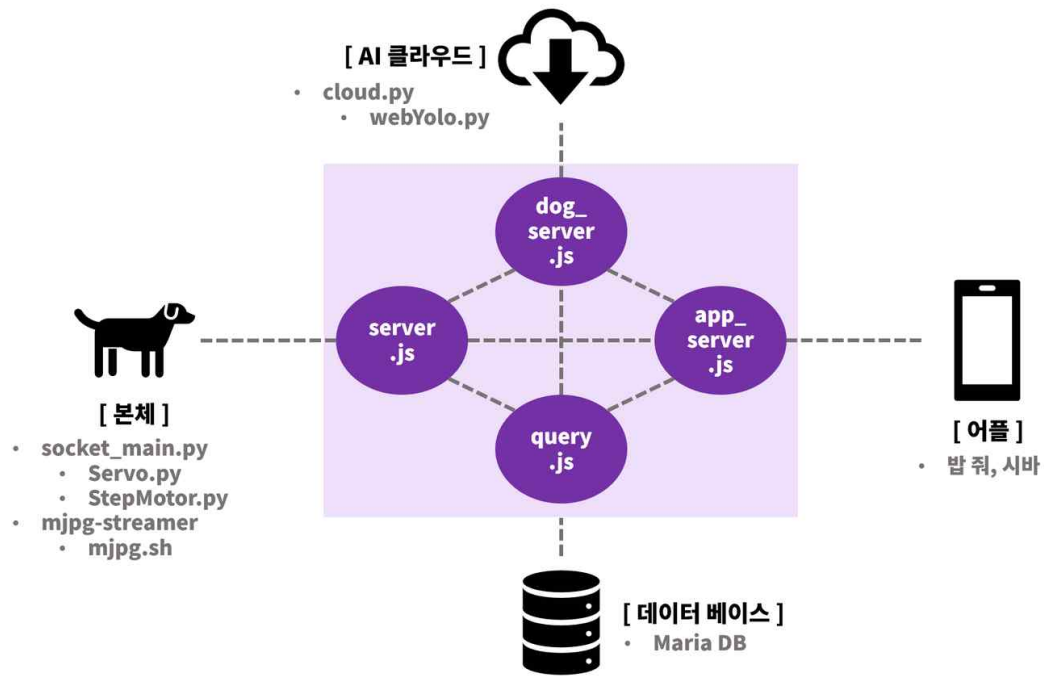
리미트 스위치라고도 불리는 이 마이크로 스위치는 산업 현장에서 널리 사용되는 전기 부품이다. 마이크로 스위치는 일반적인 스위치와 동작 원리가 다르지 않다. 일반적인 스위치는 인간의 손에 의해 on/off 되지만 리미트 혹은 마이크로 스위치는 기계의 움직임에 의하여 일정한 장소(위치)에 이르면 작동한다. 엘리베이터도 이 마이크로 스위치를 사용한다.

3 소프트웨어

1. 알고리즘



[그림 II-12] 순서도



[그림 II -13] 상속관계

밥 줘, 시바

2. 코드

1) 시바 인식 - WebYolo.py

```
from __future__ import division
import time
import torch
import torch.nn as nn
from torch.autograd import Variable
import numpy as np
import cv2
from util import *
from darknet import Darknet
from preprocess import prep_image, inp_to_image
import pandas as pd
import random
import argparse
import pickle as pkl

class PytorchYolo:

    label_list = list()
    classes = ""
    color = 0
    web_address = ""
    detect_object = ""

    def __init__(self, web_address, detect_object):
        self.web_address = web_address
        self.detect_object = detect_object

        print("Pytorch is Ready!")

    def __del__(self):
        print("")

    def prep_image(self, img, inp_dim):
        """
        Prepare image for inputting to the neural network.

        Returns a Variable
        """
```

```

orig_im = img
dim = orig_im.shape[1], orig_im.shape[0]
img = cv2.resize(orig_im, (inp_dim, inp_dim))
img_ = img[:, :, ::-1].transpose((2, 0, 1)).copy()
img_ = torch.from_numpy(img_).float().div(255.0).unsqueeze(0)
return img_, orig_im, dim

def write(self, x, img):

    c1 = tuple(x[1:3].int())
    c2 = tuple(x[3:5].int())
    cls = int(x[-1])
    label = "{0}".format(str(self.classes[cls]))    #str을 통해 변수 크기 확장
    self.label_list.append(label)
    color = random.choice(self.colors)
    cv2.rectangle(img, c1, c2, color, 1)
    t_size = cv2.getTextSize(label, cv2.FONT_HERSHEY_PLAIN, 1, 1)[0]
    c2 = c1[0] + t_size[0] + 3, c1[1] + t_size[1] + 4
    cv2.rectangle(img, c1, c2, color, -1)
    cv2.putText(img, label, (c1[0], c1[1] + t_size[1] + 4),
cv2.FONT_HERSHEY_PLAIN, 1, [225, 255, 255], 1);
    return img

def arg_parse(self):
    """
    Parse arguments to the detect module
    """
    parser = argparse.ArgumentParser(description='YOLO v3')
    parser.add_argument("--confidence", dest="confidence", help="Object
Confidence to filter predictions", default=0.25)
    parser.add_argument("--nms_thresh", dest="nms_thresh", help="NMS
Threshold", default=0.4)
    parser.add_argument("--reso", dest='reso', help=
    "Input resolution of the network. Increase to increase accuracy.
Decrease to increase speed",
                        default="160", type=str)
    return parser.parse_args()

```

```
def yolov3(self):
    cfgfile = "cfg/yolov3.cfg" # config파일 선언
    weightsfile = "yolov3.weights" # weight파일 선언
    num_classes = 80 # class개수 정의

    args = self.arg_parse() # argparse를 이용해 명령행을 파싱해오도록 함수
실행
    confidence = float(args.confidence) # confidence 변수에
--confidence값을 할당
    nms_thesh = float(args.nms_thresh)
    start = 0 # start는 0
    CUDA = torch.cuda.is_available() # cuda가 사용가능한 상황인지

    num_classes = 80 # 암튼 80
    bbox_attrs = 5 + num_classes # Bouding Box 속성

    model = Darknet(cfgfile) # Darknet
    model.load_weights(weightsfile) # Model에 weighs파일을 load해준다

    model.net_info["height"] = args.reso
    inp_dim = int(model.net_info["height"])

    assert inp_dim % 32 == 0
    assert inp_dim > 32
    """
    if CUDA:
        model.cuda() # Cuda를 사용중이면 model.cuda()
    """

    model.eval() # 모델 평가?

    # cap = cv2.VideoCapture(0) #videoCapture(0) >>
video 캡처변수 선언
    cap = cv2.VideoCapture(self.web_address)
    # videoCapture("주소") >> video 캡처변수 선언

    assert cap.isOpened(), 'Cannot capture source'
    # assert는 가정설정문, 뒤의 조건이 True가 아니면 AssertionError를 발생시킨다.
```

```

start = time.time() # 시간을 측정해주는 함수
while cap.isOpened(): # cap이 초기화가 잘 되어 있는지 확인

    ret, frame = cap.read()
    origin_frame = frame

    #####
    frame = cv2.flip(frame, 1)
    # cap.read()는 재생되는 비디오의 한 프레임씩 읽는다.
    # 제대로 읽었다면 ret값이 True가 되고, 실패하면 False.
    # 읽은 프레임은 frame에 입력이 된다.

    if ret: # ret이 true라면, 제대로 읽었다면

        img, orig_im, dim = self.prep_image(frame, inp_dim)
        #

        #             im_dim = torch.FloatTensor(dim).repeat(1,2)
        if CUDA:
            im_dim = im_dim.cuda()
            img = img.cuda()

        output = model(Variable(img), CUDA)
        output = write_results(output, confidence, num_classes,
nms=True, nms_conf=nms_thesh)

        if type(output) == int:
            print("Time = {:.2f}[sec].format(time.time() - start))
            #cv2.imshow("frame", orig_im)
            key = cv2.waitKey(1)
            if key & 0xFF == ord('q'):
                break
            continue

    output[:, 1:5] = torch.clamp(output[:, 1:5], 0.0, float(inp_dim)) / inp_dim
    #             im_dim = im_dim.repeat(output.size(0), 1)

```

```

output[:, [1, 3]] *= frame.shape[1]
    output[:, [2, 4]] *= frame.shape[0]

    self.classes = load_classes('data/coco.names')
    self.colors = pkl.load(open("palette", "rb"))

    list(map(lambda x: self.write(x, orig_im), output))

    #cv2.imshow("frame", orig_im)

    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break
    print("Time = {:.2f}[sec]".format(time.time() - start))

    #####
    if self.label_list.count(self.detect_object) >= 1:
        print(">> "+self.detect_object+" is detected")
        del self.label_list[:] # 리스트를 초기화 해줍니다
        return "y"
    else:
        print(">> "+self.detect_object+" is not detected")
        del self.label_list[:] # 리스트를 초기화 해줍니다
        return "n"
    #####

    else:
        break

if __name__ == '__main__':
    detect = PytorchYolo("http://192.168.0.54:8409/?action=snapshot", "dog")
    #detect = PytorchYolo(0, "dog")
    for i in range(0, 10):
        print(str(i+1) + " : " + detect.yolov3())

```


2) 소켓 main.py

- 소켓 수신

```
socket.on('data', function(data){
    console.log(time()+data);
    socket.write(data);    })
```

위의 코드는 소켓이 오면 콘솔로 소켓의 데이터를 띄우고 socket.write()를 통해 데이터를 다시 송신해주는 코드이다. 이 코드를 기반으로 전체 코드를 작성할 수 있었다.

- 다른 포트 간의 송수신

```
var rb_server = net.createServer(function(socket){
    console.log(time()+"Pi connected");           //Pi 연결
    exports.appsend = function(data){             //send 함수가 호출되면
        console.log(time()+"AppServer --> Pi : "+ data);           //콘솔에 app에서 Pi로
보내는 데이터 출력
        app_rb_data = data;                               //데이터 옮겨담기
        socket.write(app_rb_data);                     //rb에 데이터 전송
    }
    exports.dogsend = function(data){
        console.log(time()+"DogServer --> Pi : "+data);
        dog_rb_data = data;
        socket.write(dog_rb_data);
    }
    socket.on('data', function(data){               //Pi가 데이터를 보내면
        console.log(time()+"Pi --> Server : " + data);
        if(data == "d"){
            dog.send(data);
        }
        else if(data == "auto on"){
            app.send(data);
        }
    })
    socket.on('end', function(data){
        console.log(time()+"Pi disconnected.");
    });
    socket.on('error', function(err){               // 소켓 연결에 에러
이벤트가 왔을 때 콜백함수 실행
        console.log(time()+err);                     // 에러를 콘솔창에
출력
    });
})
```

3) app_server.js

```

var net = require('net'); // 소켓 서버를 위한 모듈
var que = require('./query.js');
rb = require('./server.js');
var moment = require('moment');

// 앱 서버 생성
var app_server = net.createServer(function(socket){ //어플 연결
    console.log(rb.time()+'App connected');
    exports.send = function(data){ //어플로 데이터 보내기 위한 사용자 정의 함수
        console.log(rb.time()+"Server --> App : " + data);
        rb_dog_data = data
        socket.write(rb_dog_data);
    }
    socket.setEncoding("utf8"); //데이터 UTF-8로 인코딩
    socket.on('data', function(data){ //데이터를 받았을때
        console.log(rb.time()+"App --> AppServer:" + data);
        if(data == "s"){ //사료 데이터(테이블)를 요청하는 명령어가 오면
            que.select('feed', function(db){ //select문 실행
                console.log(db);
                socket.write(db); //app에 데이터값 전송
            })
        }else if(data == "g"){ //간식 데이터(테이블)를 요청하는 명령어가 오면
            que.select('snack', function(db){ //select문 실행
                console.log(db);
                socket.write(db); //app에 데이터값 전송
            })
        }else{ //그 외 명령어가 오면
            try{
                rb.appsend(data); //server.js의 send 함수로 data 전송
                if(data=="1"){
                    socket.write("Shooting");
                    que.insert_time() //시간 저장
                }else
                    socket.write(data); //전송이 되었음을 app에 알림
            }catch(err){ //에러가 나면, 연결이 안되는 것이다
                console.error(rb.time()+err)
                socket.write("Pi POWER OFF") //클라이언트가 Off되어있음으로 판단하고 넘긴다
            }
        }
    });
    socket.on('end', function(data){ // 소켓 연결 종료 시 콜백함수 실행
        console.log(rb.time()+"App disconnected."); // 콘솔창에 연결이 종료되었다고 출력
    });

```

```

        socket.on('error', function(err){                // 소켓 연결에 에러 발생 시 콜백함수 실행
            console.log(rb.time()+err);                    // 에러를 콘솔창에 출력
        });
    })
    app_server.listen(8402, function(){                  //포트 8402 열기
        console.log('listening on App(8402)...');
    })

```

4)안드로이드_소켓.java

```

package com.example.shiba;

import android.os.Bundle;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.WindowManager;

import java.io.IOException;
import java.net.InetSocketAddress;
import java.nio.ByteBuffer;
import java.nio.channels.SocketChannel;
import java.nio.charset.Charset;

import static android.os.StrictMode.setThreadPolicy;

public class socket extends AppCompatActivity {

    SocketChannel socketChannel = null;

    protected void onCreate(Bundle savedInstanceState) {
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.pet_activity_saryo);

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        setThreadPolicy(policy);
    }
}

```

```
socketConnect("192.168.13.17", 8402);    //소켓 연결
    sendData("Shiba connected!");
}

public void sendData(String data){        //데이터 발신
    Log.d("Send", data);

    try {
        ByteBuffer buffer = null;
        Charset charset = Charset.forName("UTF-8");
        buffer = charset.encode(data);

        socketChannel.write(buffer);        //데이터 전송

        buffer = ByteBuffer.allocate(15000);
        int len = socketChannel.read(buffer);    //데이터 받기
        buffer.flip();

        String recive = charset.decode(buffer).toString();

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void socketConnect(String host, int port){        //소켓 연결
    //연결 부분
    try{
        socketChannel = SocketChannel.open();
        socketChannel.configureBlocking(true);
        socketChannel.connect(new InetSocketAddress(host, port));

    } catch (Exception e){
        e.printStackTrace();
    }
}
```

```
@Override
public void finish() {           //소켓 종료
    if(socketChannel.isOpen()){
        try {
            socketChannel.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    super.finish();
}
}
```

3. UI/UX



[그림 II-14] 안드로이드 메인 화면

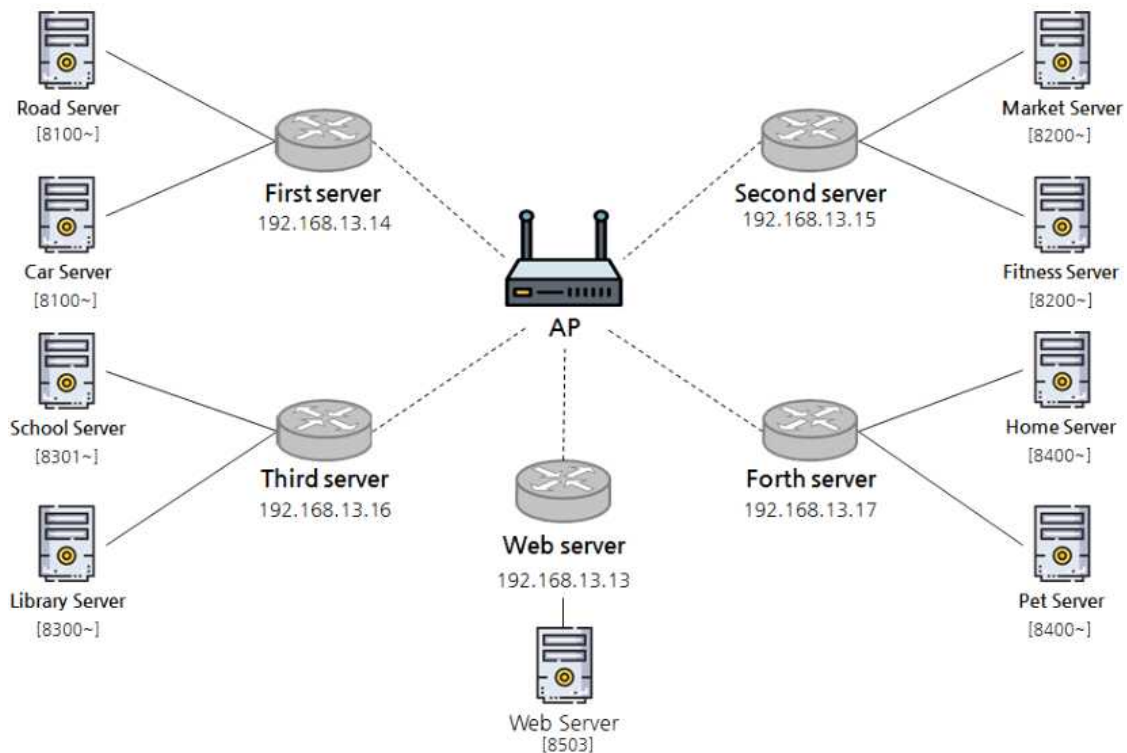
밥 줘, 시바



[그림 II-15] 안드로이드 마켓 화면

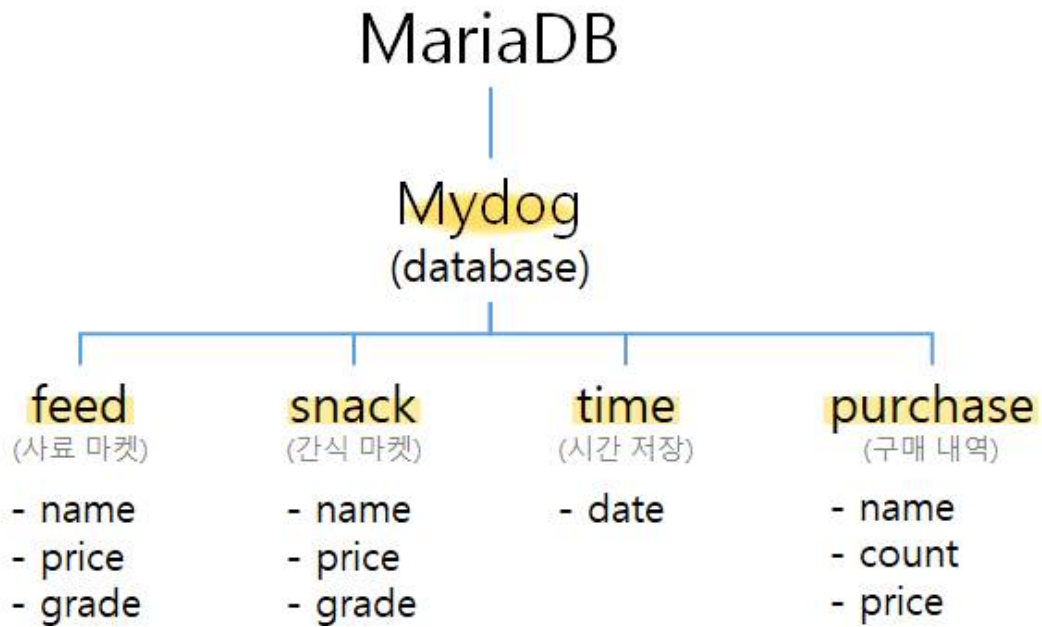
4 서버

1. 네트워크 시스템



[그림 II-16] 네트워크 구축

2. Data Base



[그림 II-17] DB 구성

```

pi@raspberrypi: ~
MariaDB [mydog]> select * from feed;
+-----+-----+-----+
| name                                     | price | grade |
+-----+-----+-----+
| 주식 토크 쇠고기, 100g, 24개 입         | 27,300 | 4      |
| 양고기 애견사료, 5.2kg                 | 59,600 | 1      |
| 강아지 사료 연어 레시피, 3kg           | 15,500 | 3      |
| 유기농 닭강아지사료, 4kg              | 48,000 | 3      |
| 피부에 좋은 사료, 2.2kg               | 28,800 | 5      |
| 치킨 & 야채 슈퍼프리미엄, 7.2kg       | 27,400 | 2      |
| 전연령 오리베네 애견사료, 6.4kg       | 34,000 | 5      |
| 강아지 사료 빛나는 피드맛, 2kg         | 18,400 | 4      |
| 생선 건강관린 애견사료, 3.9kg         | 28,800 | 4      |
| 유기농 6Free 연어, 6kg                | 37,000 | 2      |
| 그레인프리 스톱브리드 어덜트, 2kg     | 27,200 | 4      |
| 홀리스틱 네추럴바이츠, 2.27kg         | 22,900 | 5      |
| 미니 인도어 어덜트, 8.7kg              | 67,900 | 2      |
| 말랑말랑 소프트밥 양고기, 3kg         | 33,600 | 5      |
| 프리바이오틱스 애견사료              | 14,900 | 1      |
| 어덜트 스톱브리드, 11.3kg             | 99,000 | 3      |
| 류리나 퍼피 자우, 14.5kg              | 36,500 | 3      |
| 가스트로인테스티널로우렛, 1.5kg      | 23,600 | 2      |
| 팜스코 히트 강아지사료, 2kg           | 17,100 | 5      |
| 홍살 홀리스틱 연어, 5.2kg             | 59,000 | 2      |
+-----+-----+-----+
20 rows in set (0.01 sec)

MariaDB [mydog]>
  
```

[그림 II-18] 사료 마켓 database

밥 줘, 시바

```
pi@raspberrypi: ~  
MariaDB [mydog]> select * from snack;  
+-----+-----+-----+  
| name                                     | price | grade |  
+-----+-----+-----+  
| 저녁에 오리버거 100g                   | 450   | 3     |  
| 하루유산균 혼합 12개 입               | 3,600 | 4     |  
| 정말오메가3연어 180g                  | 4,900 | 4     |  
| 연달힐인트합 12개 입                 | 1,300 | 2     |  
| 카리얼 연달깎 6개 입                  | 1,800 | 5     |  
| 친구 닭고치                           | 3,500 | 1     |  
| 네이처 소고기 큐브 300g               | 2,200 | 3     |  
| 와우와우 혼합간식 150g                | 1,800 | 1     |  
| 치킨추릴고깎 5개 입                  | 1,100 | 4     |  
| 클라겐 활타스틱 30개 입              | 5,200 | 3     |  
| 도기쓰리저키숯불잡비빔 120g          | 2,000 | 3     |  
| 예쓰프렌드치킨소시지 8개 입           | 1,000 | 4     |  
| 인천우유마이렛밀크 180ml              | 2,500 | 2     |  
| 도그맨고구마별곡케익 100g            | 1,800 | 5     |  
| 미스죽연어슬라이스 300g              | 2,800 | 4     |  
| 르베도비치즈소세지대장 11개 입        | 1,300 | 5     |  
| ENF밀크스틱 27개 입                   | 4,300 | 5     |  
| 리틀렛비타칼슘비스켓 100g            | 1,000 | 1     |  
| 구트스밀크깎 5개 입                  | 900   | 4     |  
| 마들마들분특한소고기 120g            | 4,700 | 3     |  
+-----+-----+-----+  
20 rows in set (0.00 sec)  
MariaDB [mydog]>
```

[그림 II-19] 간식 마켓 database

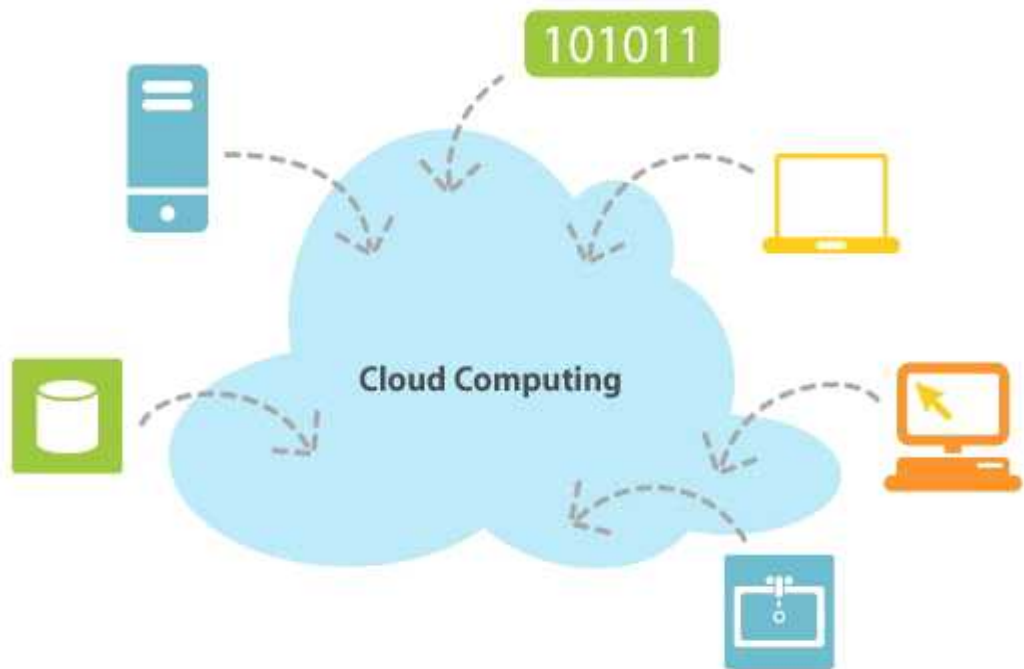
```
pi@raspberrypi: ~  
MariaDB [mydog]> select * from user  
-> ;  
+-----+  
| date |  
+-----+  
| 2019-07-05 12:26:07 |  
| 2019-07-05 12:26:11 |  
| 2019-07-05 12:26:23 |  
| 2019-10-17 19:11:01 |  
| 2019-10-17 19:11:03 |  
| 2019-10-17 19:11:05 |  
| 2019-10-17 19:11:06 |  
| 2019-10-23 15:31:23 |  
| 2019-10-23 15:55:23 |  
| 2019-10-23 16:10:28 |  
| 2019-10-23 16:11:02 |  
| 2019-10-23 16:26:42 |  
| 2019-10-23 16:27:00 |  
| 2019-10-23 17:00:20 |  
| 2019-10-23 17:00:45 |  
| 2019-10-23 17:01:02 |  
| 2019-10-24 20:20:11 |  
| 2019-10-24 20:20:34 |  
| 2019-10-24 20:27:05 |  
| 2019-10-24 20:27:46 |  
| 2019-10-24 20:32:52 |  
| 2019-10-24 20:33:47 |  
| 2019-10-24 20:39:08 |  
| 2019-10-24 20:39:26 |  
| 2019-10-24 20:39:41 |  
+-----+  
31 rows in set (0.00 sec)
```

[그림 II-20] 간식 준 시간 database

통신 방법은 TCP/IP로 인터넷 네트워크의 핵심 프로토콜이다. 전송 조절 프로토콜인 TCP는 전송할 파일을 좀 더 작은 패킷들로 나누어 인터넷 네트워크를 통해 전송하는 일과 수신된 패킷

을 원래의 메시지로 조립하는 일을 담당한다. IP는 각 패킷의 주소 부분을 처리해 패킷들이 목적지에 정확하게 도달하도록 하는 기능을 수행한다. TCP/IP는 인터넷에서 전송되는 정보나 파일들이 일정한 크기의 패킷들로 나뉘어 네트워크상 수많은 노드들의 조합으로 생성되는 경로들을 거쳐 분산적으로 전송되고, 수신지에 도착한 패킷들이 원래의 정보나 파일로 재조립되도록 하는 것이 기능이다.

3. Cloud



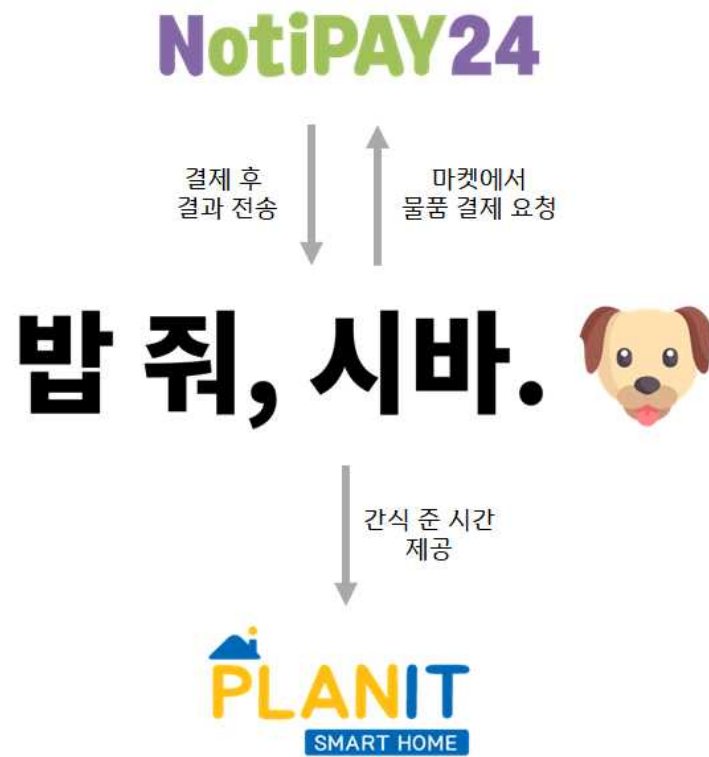
[그림 II-21] 클라우드 컴퓨팅

클라우드 컴퓨팅(영어: cloud computing)은 클라우드(인터넷)을 통해 가상화된 컴퓨터의 시스템리소스(IT 리소스)를 제공하는 것이다. 인터넷 기반 컴퓨팅의 일종으로 정보를 자신의 컴퓨터가 아닌 클라우드(인터넷)에 연결된 다른 컴퓨터로 처리하는 기술을 의미한다.

“밥 줘, 시바”의 클라우드 컴퓨팅 도입 목적은 라즈베리파이가 실시간 객체인식과 하드웨어 제어를 동시에 할 때에 기능 저하를 보였기 때문이다. 객체 인식 처리를 클라우드가 대신 처리해 줌으로서 기능의 분담화를 통해 성능의 개선을 얻고자 했다.

“밥 줘, 시바”에서 반려견에게 간식을 던져주면 그 시간을 기록해 1조 “PLANIT”에 보내주며 간식이나 사료를 살 때 결제를 위해 6조 “NotiPAY24”에 결제할 계좌와 가격을 보내면 결제 후 결과 값을 전송해준다.

밥 줘, 시바

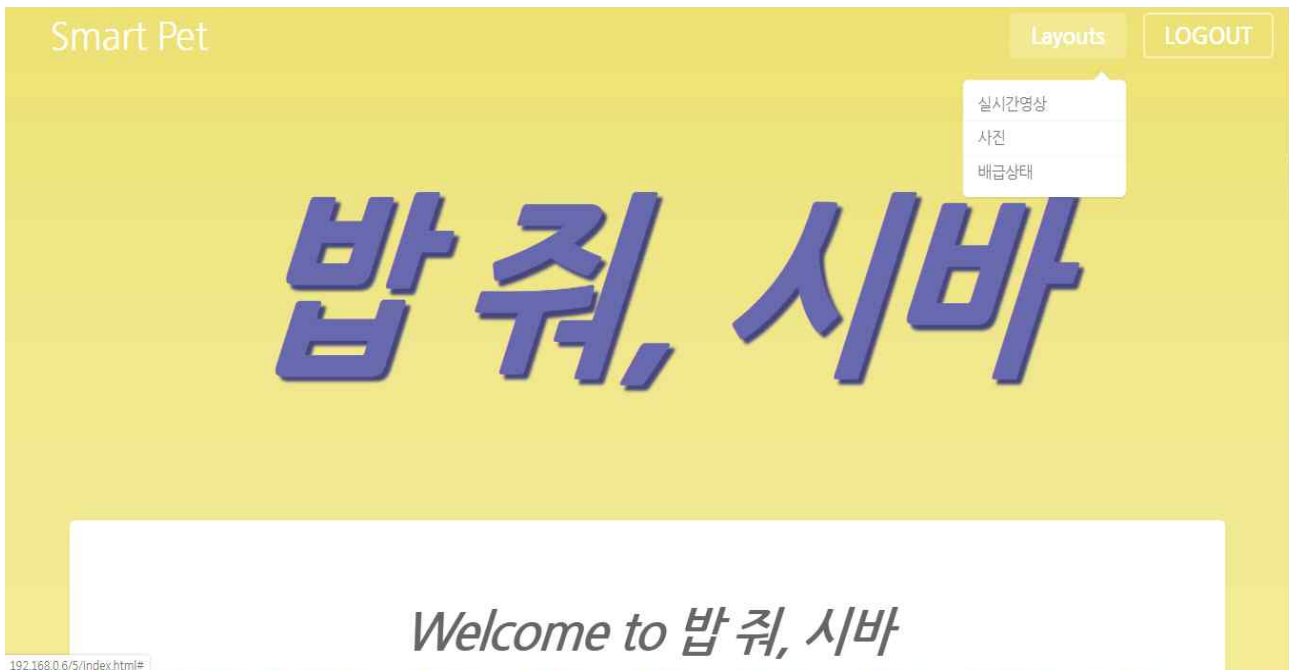


[그림 II-22] 협력 관계

5 웹 페이지



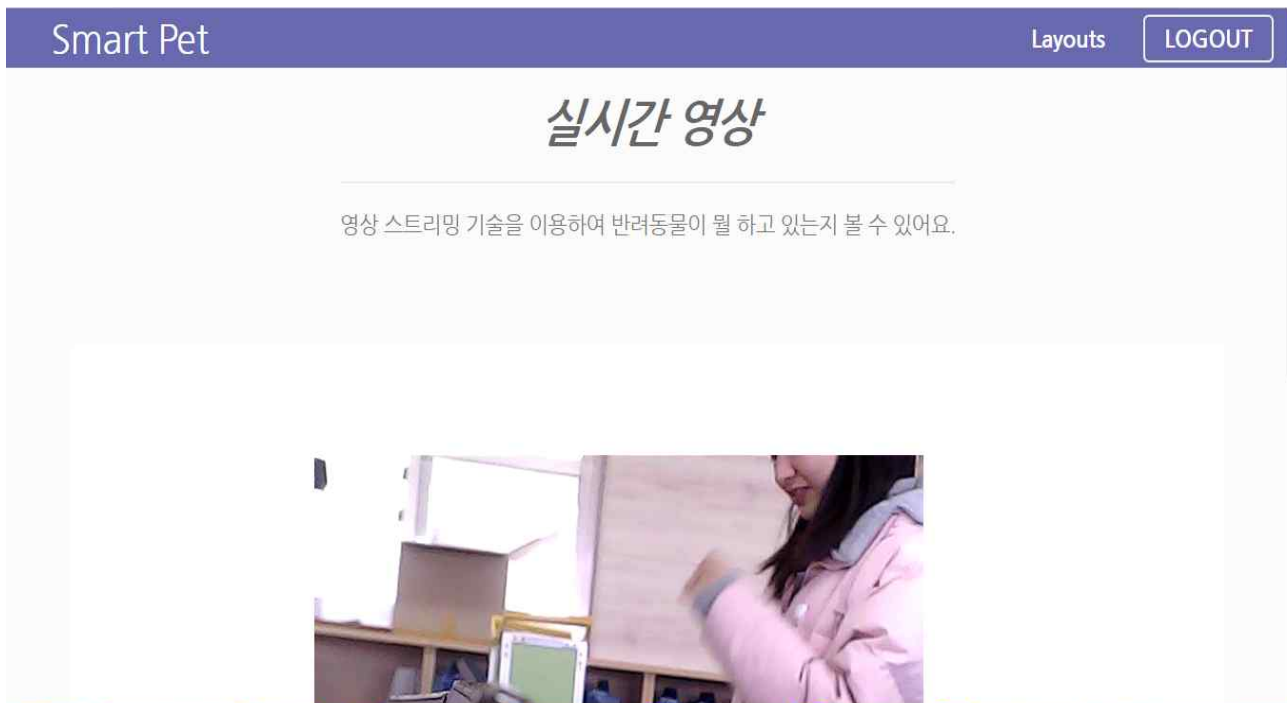
[그림 II-23] 웹 페이지



[그림 II-24] 웹 페이지 메인 화면



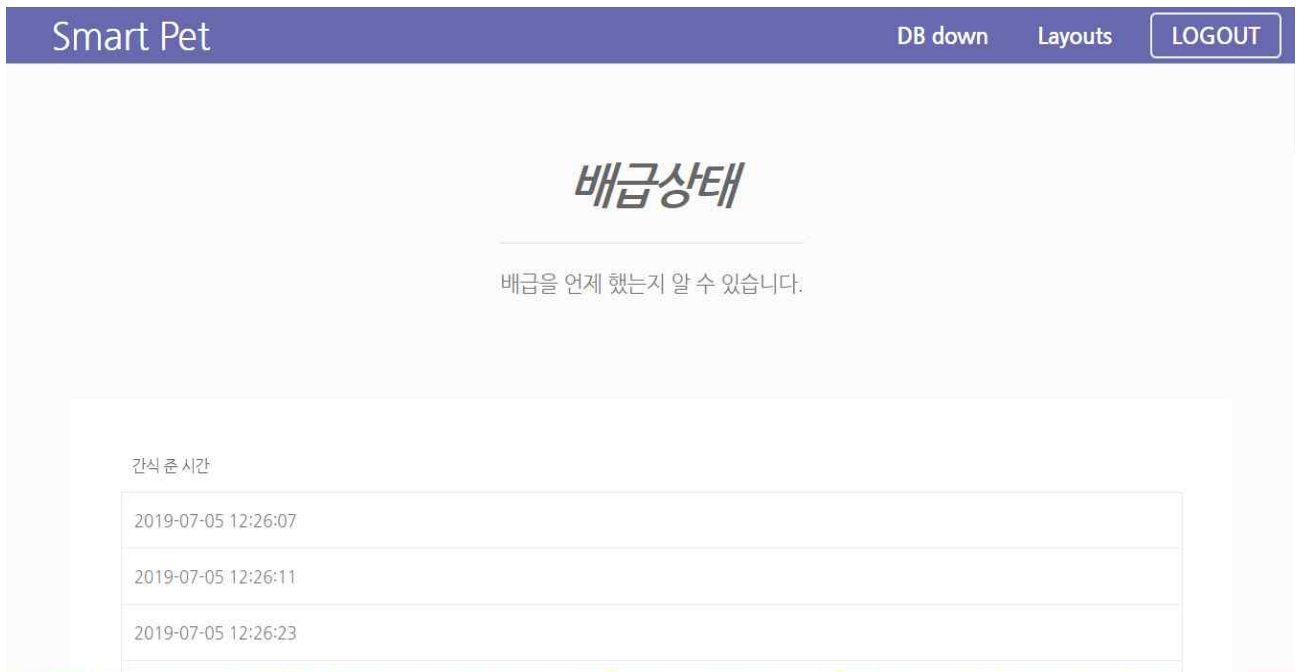
[그림 II-25] 웹 페이지 솔루션 화면



[그림 II-26] 웹 페이지 실시간영상 화면



[그림 II-27] 웹 페이지 사진 화면



[그림 II-28] 웹 페이지 배급상태 화면

III. 결론

1 결과

1. 실무결과

목표 계획 능력

프로젝트를 진행하면서 가장 크게 얻은 능력은 목표 계획 능력이라고 생각한다. 작품을 만들기 위해서는 해야 할 목표들을 하나하나 나열하고, 그것들을 이룰 수 있도록 계획을 해야한다.

자료 조사 능력

프로젝트는 우리가 여태 익숙하게 해왔던 강의 수업 방식이 아니라, 우리가 스스로 직접 자료를 찾아서 만들어나가는 수업이었다. 우리가 이미 아는 정보 내에서 할 수 있는 것도 있었지만, 할 수 없는 것이 더 많았다. Github, StackOverflow를 사용해보고, 더 나아가 논문을 읽으며 자료 조사 능력과 습득 능력이 생겼고, 전공 관련 영어 독해 능력까지 이전보다 월등하게 높아진 것 같다.

조원 협업 능력

프로젝트를 하는 첫 시간에 각자 오늘의 계획이 무엇인지 정하고, 오늘 그것만큼은 꼭 할 수 있도록 다같이 응원과 당부의 말을 남겼다. 올해 3학년 프로젝트 우리 조 슬로건은 “너도 알고 나도 알자”였다. 대체로 프로젝트를 진행하면, 역할을 나눠 각자 자기 일을 하기 바쁘는데, 그러면 일이 빨리 진행되는 것 같아도 결함을 할 때에는 많은 시간과 감정이 소모될 가능성이 높다. 우리는 매번 누군가 무엇을 성취할 때마다 간단하게 공유를 할 수 있도록 했다.

업무 처리 능력

Github, Notion, PowerPoint. 이 3가지가 업무 처리 능력을 대변해 준다. Github를 4명이서 같이 접근하여 소스를 공유하고, 행사같은 경우 Github의 캘린더 기능으로 기록한 다음 메모장으로 활용하였다.

Notion의 경우, 생산성을 엄청 높이는데 도움을 많이 받았다. 스마트폰으로, PC로 접근하여 언제 어디서든지 업무에 신경을 쓸 수 있고, 플래너는 물론, 북마크와 To-do 리스트 등 무궁무진한 기능을 사용하며 업무의 생산성을 높였다.

2. 기대효과

가. 스마트 팟 통합 플랫폼을 구현함으로써 반려견관 주인 모두에게 편리한 삶을 제공한다.

나. 반려견의 분리불안을 해소, 주인이 반려견을 향해 갖는 걱정을 통합플랫폼의 AI기능을 이용하여 편리하게 해소한다.

2 시사점

1. 강아지 소리 인식 구현에 실패한 점
2. 본체의 밸런스를 맞추지 못한 점 + 견고하게 만들지 못한 점
3. 전원선 꼬임의 문제로 본체 회전을 자유롭지 못하게 한 점
4. Github를 버전 관리에 사용하지 못한 점
5. Notion과 같은 업무 생산성 앱을 뒤늦게 사용한 점
6. PPT를 조장이 혼자 만들어 다른 조원들이 경험이 부족해진 점
7. 프로젝트 중간에 조장을 제외한 조원 세 명이 현장실습에 나가 조장이 모든 역할을 하게 된 점

3 느낀점

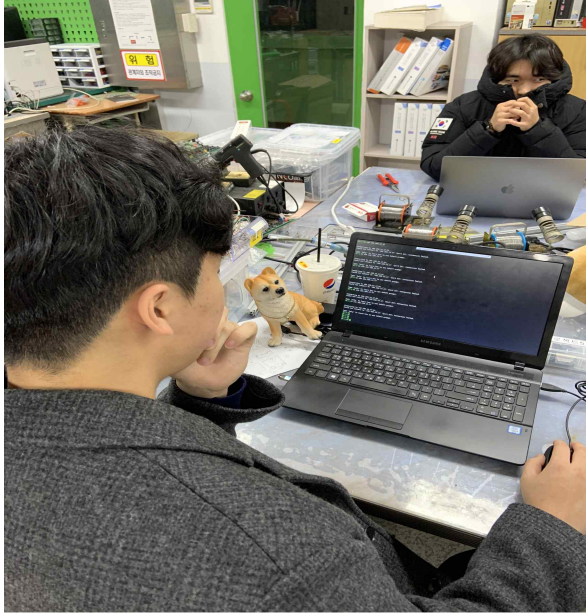
허성진(조장) : 나는 사실 프로젝트가 두려웠다. 하지만 마무리는 뿌듯함과 행복함으로 장식했다. 3년 동안 그랬듯이 나는 친구들의 추천으로 프로젝트의 조장을 맡게 되었다. 나는 선생님과 친구들 앞에서는 스트레스 안 받는 척, 쿨한 척 했지만 속은 아니었던 것 같다. 조원의 의견을 계속 수용해야 하는 건지, 아니면 내가 어느 정도 의견을 잘라가면서 단호하게 나서야 하는지 잘 알지 못했다. 사실 모든 것이 새롭고 스스로 배워야 하는 프로젝트 였기에, 나도 대부분의 분야에서는 지식이 전무했고, 그렇기에 의견을 단호하게 잘라내는 행위는 안 좋게 보여 위험하다고 생각했기 때문이다. 그런 고민은 짧은 시간에 해결되었다. 서로를 생각하며 의견을 내주고, 싸우는 것 같아도 프로젝트를 위해서 의견을 어필하는 조원들의 모습을 보며, 조장으로서의 부담을 날려 보낸 채 조원들과 함께 프로젝트를 할 수 있었다. 프로젝트는 모두가 처음인 것을 다음에는 처음이 아닐 수 있도록 미리 경험시켜주는 기회라고 생각한다. 배울 때에는 학생처럼, 작품을 제작할 때에는 현장의 전문가처럼. 이 모든 경험이 우리가 직장에 들어가 기술자가 되었을 때 큰 도움이 될 것 같다.

문은서(조원) : 1학년, 2학년을 지나 이제 3학년 마지막 프로젝트를 진행하게 되면서 3년간 다져온 실력을 보여드릴 수 있는 절호의 기회라고 생각하고 처음에는 너무 열정만 앞세웠었지만, 보다 노련해지고 성숙해진 저희였기 때문에 문제가 생겼을 때 빠르게 대처하고 이전보다 쉽게 대안을 떠올리고 실행할 수 있었습니다. 매년마다 각기 다른 프로젝트를 진행하게 되면서 3년이면 서당개도 풍월을 읊는다고 각자가 잘 하는 일을 스스로 찾아 누가 굳이 시키지 않아도 모두 알아서 척척 해내는 모습이 저희끼리도 대견스럽게 느껴졌습니다. 이전에는 어려웠던 작업들이 프로젝트를 거듭하며 노하우가 생기고 수월하게 진행 되는 것을 느끼며 할 수 있다라는 자신감과 긍지를 얻게 되었습니다. 이렇게 프로젝트를 진행하면서 얻은 경험과 지식은 실무에서 더없이 좋은 발 디딤 판이 될 것이라고 생각합니다. 마지막으로 이 프로젝트를 만드는데 많은 도움과 격려를 주신 우리 학교 선생님들과 함께 했던 조원들에게 감사드리며 느낀점 마무리 하겠습니다.

윤수화(조원) : 매년 프로젝트를 하면서 내가 발전되고 있다는 느낌이 든다. 선생님의 말씀대로만 행동해왔던 1학년 때의 나, 조원들과 상의해가면서 우리가 생각하는 대로 실행했지만 서툴렀던 2학년의 내가 생각한다. 하지만 이제는 기능에 대해 생각하면서 어떻게 구현해나갈 것인지, 구현 가능한 기능인지 머릿속으로 정리도 하고 부족한 부분은 인터넷, 선생님, 친구 등 여러 도움을 받으며 최대한 스스로 성공시키려 노력했다. 이번 프로젝트를 진행하면서 배운 점들이 많은 것 같아 뿌듯하다. 서버를 담당하면서 초반에는 무엇을 해야 하는지, 어디부터 시작해야 하는지 몰라서 잘 하는 친구의 소스를 받아 이해하면서 따라했지만 어느새 내가 다른 친구를 도와주고 있는 모습을 발견할 수 있었다. 전공뿐만 아니라 조원들과의 수많은 회의를 하면서 상대방의 의견을 존중하고 조율해나가는 방법 또한 배웠다고 할 수 있을 것 같다. 하드웨어에는 내 손이 닿은 곳을 찾기 힘들지만 서버와 안드로이드에서 많이 찾을 수 있기 때문에 아쉬운 마음은 없다. 중간엔 하루 종일 코딩만 하다보니까 계속 앉아 있어 몸이 답답한 감이 없지 않아 있어서 차라리 내가 하드웨어를 하고 싶다는 생각이 들기도 했다. 그때 포맥스를 자르러 갔는데 하나 자르는 데도 힘이 굉장히 많이 들고 쉽지 않았다. 역시 뭐든지 쉬워 보인다고 만만하게 보면 안 되는 것 같다. 우리 조원을 포함해서 다른 조원들에게 전부 수고했다고 한마디 해주고 싶다. 항상 프로젝트를 할 때는 원하는 대로 되지 않아 스트레스를 받기도 하고 2학기가 되면 대부분의 수업 시간에 프로젝트를 하다 보니 지쳐있을 때가 많지만 그만큼 보상받는 것이 많다고 생각하기 때문에 몇 년 후 지금을 되돌아볼 때면 좋은 경험으로 남아있을 것 같다.

정성운(조원) : 학교에서 하는 마지막 프로젝트라는 생각에 완벽하게 구현해서 후회하지 않고 3학년 프로젝트를 끝내야겠다고 마음을 잡았다. 프로젝트 초반에 조원들과 의견을 내고 회의도 하면서 의견대립과 갈등도 있었지만 서로서로 의견을 존중해주고 맞춰나가면서 하나 배워간다는 생각을 했다. 내 부분인 하드웨어를 하면서는 적외선 센서를 브레드보드에서 해보는데 정확하게 연결을 했다고 생각하고 작동시켜봤지만 될 듯 말 듯 불이 켜졌다가 꺼졌다가 해서 수정을 하려고 보니 수발광 센서가 정확히 일직선으로 만나야 정확하게 작동이 된다는 점을 발견했다. 스텝모터를 작동시키는데 a4988 모터 드라이버를 사용하기 위해 거의 한 달 동안 공부하면서 연결도 하고 수정도 해보았지만 끝내 작동이 안 되었다. 서보모터를 작동시킬 때는 사료를 날리는데 힘이 적게 들어 멀리 날아가지 못하는 상황이 발생을 하여 수정을 해봤지만 생각했던 거리보다 적게 날아가고 끝내었다. 이렇게 적외선 센서, 스텝모터, 서보모터를 작동해보면서 정확하게 했다고 생각을 해도 수정할 부분이 있다는 것을 느꼈고 무조건 한 번 이상은 검토를 해야겠다고도 생각을 했다. 마지막으로 1, 2학년때는 취업을 생각하지 않고 프로젝트에만 집중을 하고 조원들 끼리 끝까지 프로젝트를 끝내 많은 성취감을 얻을 수 있었지만 3학년에는 프로젝트도 있고 취업도 해야된다는 생각에 프로젝트에만 집중을 할 수 없었다. 취업이 확정이 된 다음에는 프로젝트에 집중을 잠시나마 했지만 2학기가 되고 바로 현장실습을 나가 나머지 조원들이 내 부분까지 해야 된다는 생각에 미안함 마음과 끝까지 다 끝내지 못해서 큰 성취감은 느끼지 못하는 것 같다. 그래도 고등학교 마지막 3학년 프로젝트 “밥 쥐, 시바”를 한 것에 대해서는 후회하지는 않고 기억에 많이 남을듯하다.

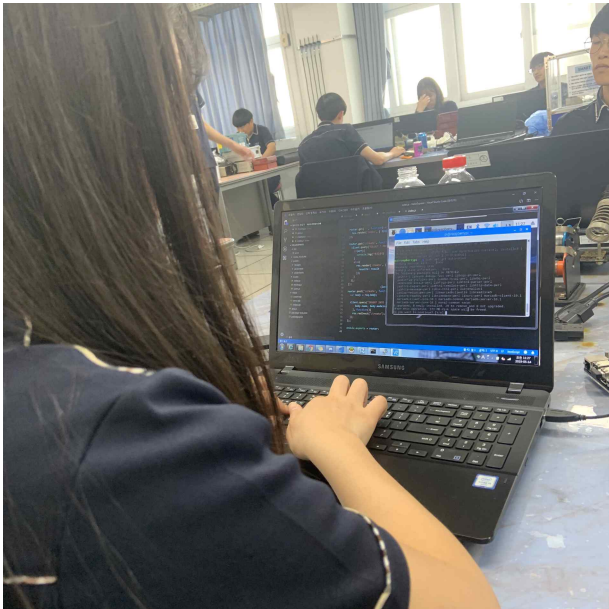
IV. 활동사진



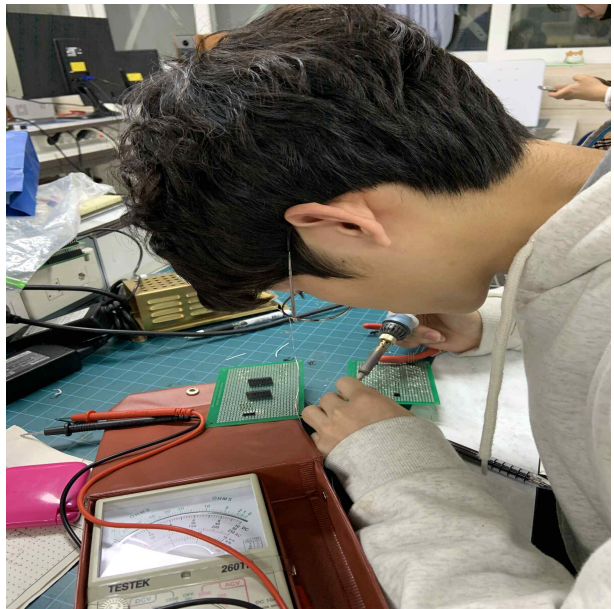
허성진



문은서



윤수화



정성운

[그림IV-1] 조원 활동사진

참고문헌

1. 심해펍킨, 심해펍킨의 심해도서관, “node.js 소켓통신”,
<https://m.blog.naver.com/PostView.nhn?blogId=platinasnow&logNo=220024486295&proxyReferer=https%3A%2F%2Fwww.google.com%2F>, 2019-04
2. WikiPedia, “PyTorch”, <https://ko.wikipedia.org/wiki/PyTorch>, 2019-04
3. WikiPedia, “FFmpeg”, <https://ko.wikipedia.org/wiki/FFmpeg>, 2019-04
4. zueseyera, “Yolo”,
https://github.com/zeuseyera/darknet-kr/blob/master/2_YOLO/yolo.md, 2019-05
5. Coding Run, “Servo Motor Control”, <https://codingrun.com/110>, 2019-05
6. 가치, 네이버 블로그, “DRV8825 Step Motor Driver”,
<http://blog.naver.com/PostView.nhn?blogId=ubicomputing&logNo=150173831399>,
2019-05
7. 비타민씩, 비타민씩의 포만감성라이프, “안드로이드 스튜디오 특징”
<http://blog.naver.com/PostView.nhn?blogId=vitaminsense&logNo=220680127586>,
2019-10
8. 두산백과, 네이버 지식백과, “오토캐드란”,
<https://terms.naver.com/entry.nhn?docId=3346150&cid=40942&categoryId=32836>,
2019-11
9. 컴퓨터인터넷IT용어대사전, 네이버 지식백과, “OrCAD란”,
<https://terms.naver.com/entry.nhn?docId=831635&cid=50376&categoryId=50376>,
2019-11
10. 두산백과, 네이버 지식백과, “OrCAD란”,
<https://terms.naver.com/entry.nhn?docId=3353659&cid=40942&categoryId=32836>,
2019-11
11. 망중립성, 네이버 지식백과, “TCP/IP”,
<https://terms.naver.com/entry.nhn?docId=2274898&cid=42171&categoryId=51118>,
2019-11