

IoT 시스템 응용 팀 프로젝트

프로젝트명 : 강아지 자동 사료 급식기

12조 : 김주혁, 김기현, 최규동

목차

1. 개요	3
2. 개발 과정	4
3. 수행 내역(주차별)	5
4. 시스템 구성도	6
5. 코드	8
6. 테스트 시연	18
7. 추가 및 변경사항	23
8. 재료	25
9. Q&A	26

개요

- 자동 사료 급식기

- 사용자가 웹 페이지에 사료 급여 횟수와 각 횟수별 배급 시간들을 입력, 배급 시간이 되면 자동으로 사료를 배급
- 매일 정해진 급여 시간으로 강아지의 체중과 단식을 관리
- 외출 또는 여행 시 강아지 사료 자동 급여 가능
- 시중의 자동 급식기를 구매 시 드는 높은 비용을 피해서 직접 제작

개발 과정

- **계획 단계**

- 프로젝트 계획서 작성

- **설계 단계**

- 서보모터, LCD Display 작동 모듈 및 웹페이지 설계

- **구현 단계**

- 서보모터, LCD Display 작동 모듈 및 웹페이지 구현
- 사료통 제작

- **테스트 단계**

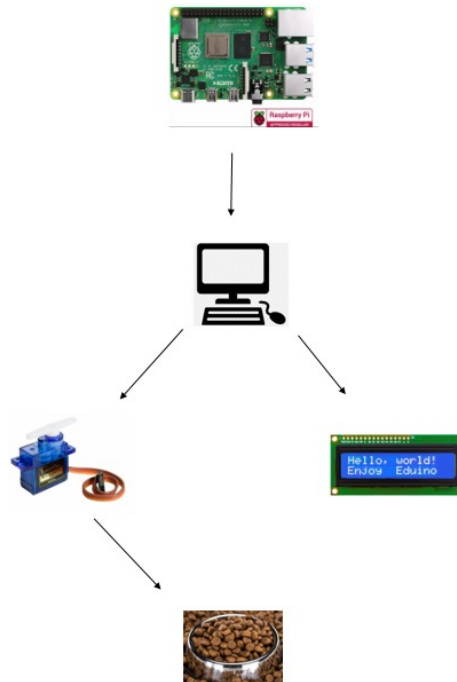
- 단위 테스트 수행
- 통합 테스트 수행

수행 내역(주차별)

- 9주차: 프로젝트 계획서 작성 및 조원간 역할 분담
- 10주차: Flask를 활용하여 웹 서버 구축한 후 웹 페이지 구현
- 11주차: 서보모터, LCD Display 코드 구현
- 12주차: 단위/통합 테스트 진행, 코드 수정 및 최종 확인
- 13주차: 사료 급식기 제작 및 기기 장착
- 14주차: 프로젝트 발표 및 시연

시스템 구성도 - HW

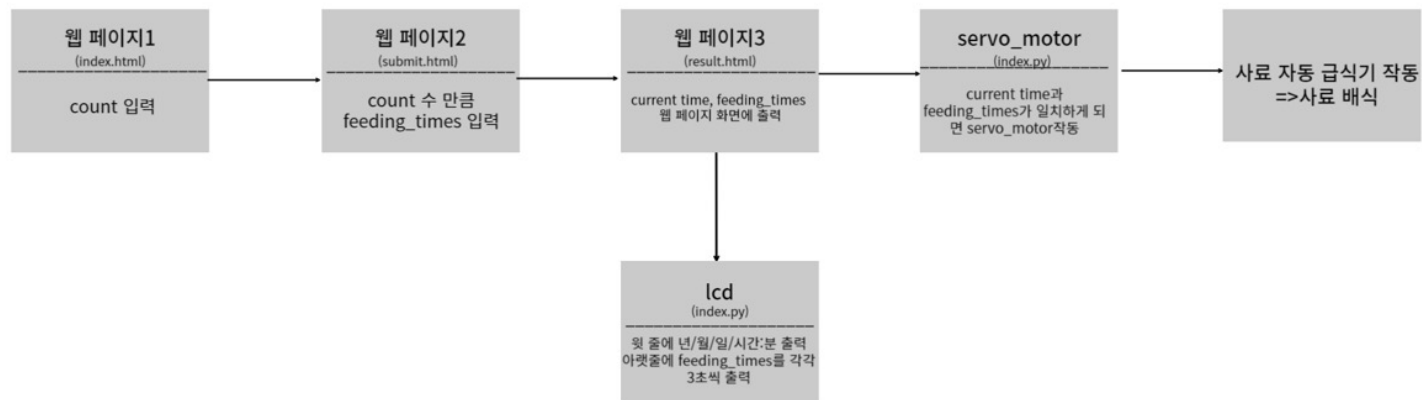
- 하드웨어 구성도
- 전체 구성도



시스템 구성도- SW

- 소프트웨어 구성도

- 전체 구성도



코드

1. index.py

- Flask 웹 어플리케이션을 구성하고 실행하는 파이썬 파일
- 서버모터, LCD Display 실행 함수

2. index.html

- 일일 사료 횟수를 입력받는 Page1

3. submit.html

- index.html에서 입력받은 count만큼 time을 입력받는 Page2

4. result.html

- 현재 시간과 입력 받은 feeding times를 화면으로 보여주는 Page3

코드 1 – index.py

- 크게 5부분으로 구성
 - (1) 선언부
 - (2) GPIO pin 설정과 초기화
 - (3) 서보모터 함수
 - (4) LCD Display 함수
 - (5) Flask application에서 들어오는 HTTP 요청 처리

코드 1 – index.py

(1) 선언부 : 라이브러리와 모듈 선언

```
from flask import Flask, request, render_template, jsonify
import datetime
import threading
import time
from RPLCD.i2c import CharLCD
import board
import busio
import RPi.GPIO as GPIO
```

코드 1 – index.py

(2) GPIO pin 설정과 초기화 : LCD, 서보모터

```
# LCD 설정
lcd_columns = 16
lcd_rows = 2
i2c = busio.I2C(board.SCL, board.SDA)
lcd = CharLCD(i2c_expander='PCF8574', address=0x27, port=1, cols=lcd_columns, rows=lcd_rows)

# 서보모터 설정
SERVO_PIN = 18
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SERVO_PIN, GPIO.OUT)
servo = GPIO.PWM(SERVO_PIN, 50)
servo.start(0)
```

코드 1 – index.py

(3) 서보모터 함수: servo_motor_runnin0 | web page3로부터 feeding_times를 받으면서 True가 되고 while문 실행, 현재 시간(00초)과 feeding_times를 비교하여 같으면 서보모터 동작

```
def run_servo_motor():
    global servo_motor_running, feeding_times

    while servo_motor_running:
        current_time = datetime.datetime.now().strftime("%H:%M:%S")
        if current_time.endswith(":00"):
            if current_time[:-3] in feeding_times:
                servo.ChangeDutyCycle(7.5)
                time.sleep(1)
                servo.ChangeDutyCycle(2.5)
                time.sleep(1)

            time.sleep(1) # 1초마다 확인

    print("Servo motor stopped.")
    servo.stop()
    GPIO.cleanup()
```

코드 1 – index.py

(4) LCD Display 함수 : LCD 윗 줄에 현재 시간, 아래줄에 feeding_times 출력

```
def display_lcd():
    global feeding_times

    while True:
        current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M")
        lcd.clear()
        lcd.write_string(current_time)
        lcd.cursor_pos = (1, 0)
        if feeding_times:
            for time_val in feeding_times:
                lcd.write_string(time_val)
                time.sleep(3) # 각 feeding time을 3초간 표시
            lcd.clear()
            lcd.write_string(current_time)
            lcd.cursor_pos = (1, 0)
        else:
            lcd.write_string("No feeding times")
            time.sleep(5) # 5초 동안 메시지 표시
            lcd.clear()
```

코드 1 – index.py

(5) Flask application에서 들어오는 HTTP 요청 처리

```
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/submit', methods=['POST'])
def submit():
    count = int(request.form['count'])
    return render_template('submit.html', count=count)

@app.route('/result', methods=['POST'])
def result():
    global feeding_times, servo_motor_running

    current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    count = int(request.form['count'])

    new_feeding_times = []
    for i in range(1, count + 1):
        time_val = request.form.get(f'time{i}', '')
        new_feeding_times.append(time_val)

    if not servo_motor_running:
        feeding_times = new_feeding_times
        servo_motor_running = True
        servo_thread = threading.Thread(target=run_servo_motor)
        lcd_thread = threading.Thread(target=display_lcd)
        servo_thread.start()
        lcd_thread.start()

    return render_template('result.html', current_time=current_time, feeding_times=feeding_times)

@app.route('/get_current_time')
def get_current_time():
    current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    return jsonify(current_time=current_time)

@app.route('/get_feeding_times')
def get_feeding_times():
    global feeding_times
    return jsonify(feeding_times=feeding_times)

if __name__ == '__main__':
    app.run(debug=True)
```

코드 2 – index.html

- Page1: 사료 급식 횟수를 입력하는 폼 생성, Feed count는 POST 방식으로 “/submit”에 전송

```
<html>
<head>
  <title>Enter feed count</title>
  <style>
    /* css code */
  </style>
</head>
<body>
  <h2>Enter feed count</h2>
  <form action="/submit" method="post">
    <label type="count">Feed count:</label>
    <input type="number" id="count" name="count" min="1" required>
    <input type="submit" value="Enter">
  </form>
</body>
</html>
```

코드 3 – submit.html

- Page2: count만큼 time을 입력받는 폼 생성, times은 POST 방식으로 “/result”에 전송

```
<html>
<head>
  <title>Enter feed time</title>
</head>
<body>
  <h2>Enter feed time</h2>
  <p>Example: 9:30, 12:00, 21:30</p>
  <form action="/result" method="post">
    <input type="hidden" name="count" value="{{ count }}">
    {% for i in range(1, count +1) %}
      <label for="time{{ i }}">Count {{ i }} time (h m):</label>
      <input type="text" name="time{{ i }}" id="time{{ i }}"><br>
    {% endfor %}
    <input type="submit" value="Enter">
  </form>
</body>
</html>
```


코드 4 – result.html

- Page3: 현재 시간(“년 월 일 시간:분:초”), feeding_times 보여줌

```
<html>
<head>
  <title>feeding time results</title>
</head>
<body>
  <h2>feeding time results</h2>
  <p>current time: <span id="current_time"></span></p>
  <p>feeding times:</p>
  <ul id="feeding_times">
    <li>{% for time in feeding_times %}
      <li>{{ time }}</li>
    </li>{% endfor %}</ul>

  <script>
    function updateCurrentTime() {
      var currentTimeSpan = document.getElementById("current_time");

      function displayTime() {
        fetch('/get_current_time')
          .then(response => response.json())
          .then(data => currentTimeSpan.textContent = data.current_time);
      }

      displayTime();
      setInterval(displayTime, 1000); // 1초마다 현재 시간 업데이트
    }

    function displayFeedingTimes() {
      fetch('/get_feeding_times')
        .then(response => response.json())
        .then(data => {
          const feedingTimes = data.feeding_times || [];
          const feedingTimesList = document.getElementById("feeding_times");

          // 기존 리스트 초기화
          feedingTimesList.innerHTML = '';

          // 새로운 시간들로 리스트 업데이트
          feedingTimes.forEach(time => {
            const listItem = document.createElement("li");
            listItem.textContent = time;
            feedingTimesList.appendChild(listItem);
          });
        });
    }

    setInterval(displayFeedingTimes, 1000); // 1초마다 먹이 시간 업데이트
    displayFeedingTimes(); // 페이지 로드 시 먹이 시간 업데이트
    updateCurrentTime(); // 페이지 로드 시 현재 시간 업데이트
  </script>
</body>
</html>
```

테스트 시연

1. 데이터 입력 (웹 페이지)

- count = 3

- feeding times : 9:00, 12:00, 18:00

2. LCD Display에 윗 줄에 현재시간, 아래줄에 feeding times를 각각 3초동안 출력

3. 현재 시간이 사용자가 입력한 feeding times가 되면 서보모터 동작

4. 사료 급여

테스트 시연 1 - 데이터 입력 (웹 페이지)

①

Enter feed count

Feed count:

②

Enter feed time

Example: 9:30, 12:00, 21:30

Count 1 time (h m):

Count 2 time (h m):

Count 3 time (h m):

③

Feeding time results

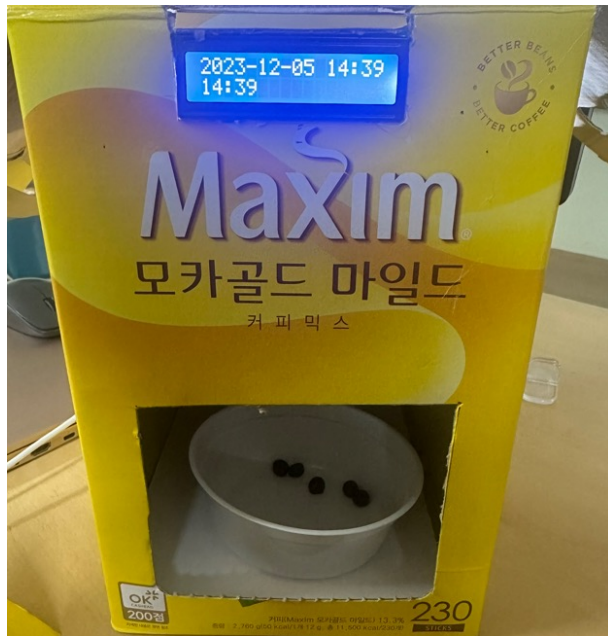
Current time: 2023-12-05 15:35:51

Feeding times:

- 9:00
- 12:00
- 18:00

테스트 시연 2 – LCD Display

(* 시연한 LCD 사진을 찾지 못해 다른 테스트 사진으로 대체함)



테스트 시연 3 – 서보모터 동작

- 급식기의 내부 모습으로 서보모터가 화살표 방향으로 동작하여 사료 급여



테스트 시연 4 – 사료 급여

(* 시연한 사진을 찾지 못해 다른 테스트 사진으로 대체함)



추가 및 변경사항

1. Web Page2 입력 예시 추가 (11/15)

- Web Page2에서 feeding times 입력 예시로 "Example: 9:30, 12:00, 21:30"을 추가

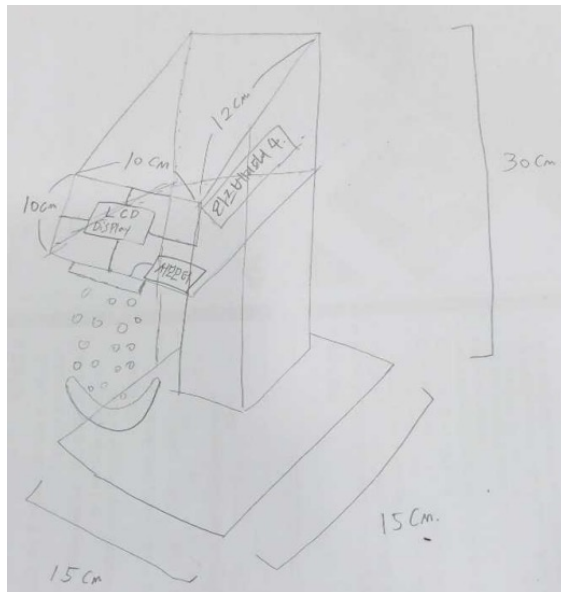
2. LCD Display 화면 출력 방법 변경 (11/22)

- 기존 : LCD Display의 feeding times 출력이 한 칸씩 왼쪽으로 이동하면서 보여줌
=> feeding times를 한 개씩 1개씩 보여줌
=> (11/28 변경) 1초->3초

추가 및 변경사항

3. 사료 급식기 변경

- 기존: 사료 급식기 전체를 직접 제작



- 변경: 박스를 이용



재료

- 라즈베리파이
- 암암/암수/수수 점퍼선
- 1602 LCD display
- 서보 모터
- 우드락
- 맥심 모카골드 마일드 커피믹스 230T 박스
- 글루건

