

# 웹 기반 원격 액츄에이터 제어 시스템 구현

배병채, 정동민, 이영동

창신대학교 소프트웨어공학과

## Implementation of Actuator Control System based on Web

Byeong Chae Bae, Dong Min Jeong, Young-Dong Lee

Department of Software Engineering, Changshin University

E-mail : ydlee@cs.ac.kr

### 요 약

최근 사물인터넷에 대한 관심이 높아지고 있으며, 여러 분야에서 사물인터넷을 접목한 응용 사례들이 늘어나고 있다. 본 논문에서는 웹을 통해 다양한 사물들의 제어를 위해 서버는 파이썬을 기반으로 한 SimpleHTTPServer를 활용하여 구현하였으며, 클라이언트에서는 웹페이지 메시지 전송 방식중 하나인 POST 방식을 이용하여 웹 서버를 경유하여 원격 지 액츄에이터 제어 시스템을 구현하였다.

### ABSTRACT

Recent people start to attention about IoT(Internet of Things) and applying IoT on many fields is rising. In this paper we have implemented to control things via web so we used SimpleHTTPServer with Python. Client was used POST, one of web request method which passes through web server and developed a remote control system.

### 키워드

사물인터넷, SimpleHTTPServer, 액츄에이터 제어, POST

## I. 서 론

사물인터넷에 대한 관심과 기술의 발달은 다양한 분야로 응용되어 우리의 삶에 크고 작은 영향을 주고 있다. 그 중에서도 원격으로 제어하는 기기들은 간단한 조작만으로 쉽게 사물들을 관리할 수 있게 되었다. 사물인터넷을 제어하기 위해 조종기나 스마트폰의 앱이 필요한데 기존 시스템은 여러 가지 제약점들을 안고 있다.

본 논문에서는 웹을 통해 다양한 사물들의 제어를 위해 파이썬(Python) [1]과 HTML을 기반으로 한 웹 기반 원격 액츄에이터 제어 시스템을 구현하고자 한다. 라즈베리파이를 이용하여 서버모터 1개와 DC모터 1개를 이용한 액츄에이터 제어 시스템으로서 RC카를 제작하였으며, 구동방식은 웹페이지를 이용하기 위해 SimpleHTTPServer [2]의 파이썬 코드로 서버를 제작하고 HTML코드로 조작장치를 구현하였다.

## II. 시스템 구성

본 논문에서 제안한 웹 기반 원격 액츄에이터 제어 시스템은 라즈베리파이, DC모터, 서버모터, 모터구동을 위한 모터드라이버, 외부전원으로 구성된다.

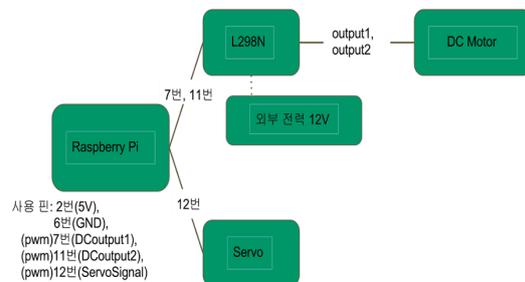


그림 1. 전체 시스템 구성도

모터드라이브는 L298N을 사용하였으며, DC모터의 속도 조절은 ChangeDutyCycle를 통해 조절하게 하여 DC모터에 들어가는 2개의 선에 PWM 신호를 두어 제어 가능하도록 하였다.

### III. 시스템 구현 결과

그림 2는 라즈베리에서 DC 모터 제어를 위한 global 변수 선언부를 나타낸다. 함수 안에 있는 변수를 전역 변수로 만들어 주기위해 변수 앞에 'global'명령어를 넣어줌으로 해당된 변수를 전역변수로 선언하였다. ChangeDutyCycle의 바뀔 수 있는 비는 0~100까지이므로 음수는 존재할 수 없으며, 소스코드는 그림 3과 같이 구현하였다.

```

pi@raspberrypi: ~/smartphoneControl2
File Edit Tabs Help
GNU nano 2.2.6 File: controlServer2.py

self.send_response(100)
self.send_header('Content-type', 'text/html')

return
return self.do_GET()

def setMotor(cmd):
    global currentGearValue1
    global currentGearValue2
    global curSpeedVal1
    global curSpeedVal2
    global SERVO_STEP_LIMIT
    global SERVO_MAX_STEP

    if cmd == 'GO':
        if currentGearValue1 <= MOTOR_MIN_STEP:
            currentGearValue1 = MOTOR_MIN_STEP
        else:

```

그림 2. global 변수 선언

```

pi@raspberrypi: ~/smartphoneControl2
File Edit Tabs Help
GNU nano 2.2.6 File: controlServer2.py

if (curSpeedVal2 < SERVO_STEP_LIMIT+1):
    SERVO_MAX_STEP = 1
else:
    SERVO_MAX_STEP = 2

if currentGearValue1 > 0 and currentGearValue1 <= MOTOR_FORWARD_MAX_STEP:
    p.ChangeDutyCycle(curSpeedVal1)
elif currentGearValue2 < 0 and currentGearValue2 >= MOTOR_BACKWARD_MAX_STEP:
    q.ChangeDutyCycle(curSpeedVal2*1)
elif currentGearValue1 == 0:
    p.ChangeDutyCycle(0)
    q.ChangeDutyCycle(0)
elif currentGearValue2 == 0:
    p.ChangeDutyCycle(0)
    q.ChangeDutyCycle(0)

def setSteering(cmd):
    global currentSteeringValue

```

그림 3. 모터 출력값 조정

```

pi@raspberrypi: ~/smartphoneControl2
File Edit Tabs Help
GNU nano 2.2.6 File: controlServer2.py

import cgi
import RPi.GPIO as gpio
from BaseHTTPServer import HTTPServer
from SimpleHTTPServer import SimpleHTTPRequestHandler
import math

#Value
SERVO_RANGE = 4
SERVO_STEP_LIMIT = 60

MOTOR_FORWARD_MAX_VAL = 90
MOTOR_BACKWARD_MAX_VAL = -75
MOTOR_RANGE_VAL = 15
MOTOR_DEFAULT_VAL = 30
MOTOR_MIN_STEP = math.ceil(MOTOR_DEFAULT_VAL / MOTOR_RANGE_VAL)
MOTOR_FORWARD_MAX_STEP = math.ceil(MOTOR_FORWARD_MAX_VAL / MOTOR_RANGE_VAL)
MOTOR_BACKWARD_MAX_STEP = math.floor(MOTOR_BACKWARD_MAX_VAL / MOTOR_RANGE_VAL)
MOTOR_BREAK_STEP_RANGE = 0.25

```

그림 4. DC모터를 사용한 모터 제어부 소스코드

RC카의 전진과 후진을 구분 하기 위해 후진의 값을 그림 4와 같이 음수로 취해 썼으며, 마지막에 출력될 때에는 '-1'을 곱함으로 문제없이 만들었다.

그림 5는 서버모터 제어를 나타내며, 함수안의 변수 'currentsteeringvalue'를 'global'명령어를 사용함으로 전역변수로 주었고, 좌우로 변환하는 버튼은 설정해놓은 크기만큼 변환 할 수 있다.

```

pi@raspberrypi: ~/smartphoneControl2
File Edit Tabs Help
GNU nano 2.2.6 File: controlServer2.py

p.ChangeDutyCycle(0)
q.ChangeDutyCycle(0)
if currentGearValue2 == 0:
    p.ChangeDutyCycle(0)
    q.ChangeDutyCycle(0)

def setSteering(cmd):
    global currentSteeringValue

    if cmd == 'Left':
        if currentSteeringValue < 7 + (SERVO_RANGE/2):
            currentSteeringValue = currentSteeringValue + (SERVO_RANGE/2/SERVO_MAX_STEP).ChangeDutyCycle(currentSteeringValue)

    elif cmd == 'Right':
        if currentSteeringValue > 7 - (SERVO_RANGE/2):
            currentSteeringValue = currentSteeringValue - (SERVO_RANGE/2/SERVO_MAX_STEP).ChangeDutyCycle(currentSteeringValue)

    elif cmd == 'C':
        currentSteeringValue = 7
        SERVO.ChangeDutyCycle(currentSteeringValue)

```

그림 5. 서버모터 제어

본 논문에서 제안한 웹 기반 원격 액츄에이터 제어 시스템은 RC카를 제작하여 동작 테스트하였으며, 웹 기반 동작 테스트는 그림 6과 같이 SimpleHTTPServer를 HTML를 사용하여 구현하였으며, 사용자는 웹 브라우저를 통해 원격지에 있는 RC카 제어가 가능하도록 하였다. 원격지에서 RC카 제어를 위해 그림 7과 같이 총 9개의 버튼을 사용하여 제어가 가능하도록 하였으며, 각각의 버튼에 전진, 후진, 좌우, 가속, 급가속, 정위치의 8가지의 기능을 두었다. 전진, 후진 버튼은 누르면 설정한 최소 속도로 움직이며, 'E'는 가속 버튼이며 누를 때 마다 전 후진의 속도가 점점 빨리진다. 'B'는 급가속 버튼이며 전진을 했을 때만 설정한 최고 속도로 움직인다. 'C'는 RC카의 방향을 정위치로 바꾸는 버튼이고, 좌우 버튼은 서버모터가 왼쪽, 오른쪽으로 회전해서 설정한 각도만큼 버튼을 눌러 최고 각도 까지 움직이게 하였다. 정지 버튼은 한번에 멈추는 것이 아닌 누를 때마다 감속하게 된다.

```

<html>
<head>
<meta charset="UTF-8">
<title>RC-CAR Controller</title>
<style>
div{
display:inline;
float:left;
}
</style>
<meta name="viewport" content="width=200, initial-scale=1, maximum-scale=1">
</head>

<body>
<div style="margin: 0 0 10px 10px">
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="E" onClick="cmd('E')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="left" onClick="cmd('left')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="C" onClick="cmd('C')">
</ul>
</div>

<div style="margin: 0 0 10px 10px">
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="Go" onClick="cmd('GO')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="Stop" onClick="cmd('Stop')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="Back" onClick="cmd('BACK')">
</ul>
</div>

<div style="margin: 0 0 10px 10px">
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="B" onClick="cmd('B')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="Right" onClick="cmd('Right')">
</ul>
<ul>
<input type="button" style=font-size:20pt;width:70;height:60;
value="" $
</ul>
</div>
    
```

그림 6. 웹 기반 소스코드 구현

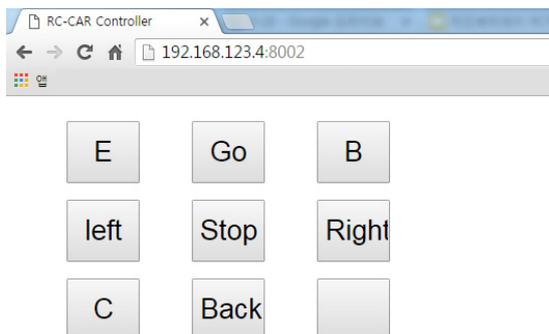


그림 7. 웹 브라우저를 통한 원격 액츄에이터 제어 화면

#### IV. 결 론

사물인터넷의 발달로 사람들의 생활에 변화가 오고, 그에 따라 사물인터넷은 여러 분야에 응용된 사례들이 나오고 있다. 웹을 통해 다양한 사물을 제어 할 수 있도록 파이썬과 HTML을 이용하여 서버와 조작기능을 가진 웹페이지를 구현하였다. 라즈베리파이를 이용한 RC카를 만들어 웹기반의 원격 액츄에이터를 제작해 만들어진 웹페이지를 이용하여 조작이 가능하게 제작하였다. 통신 인프라가 갖춰진 곳이라면 언제, 어디서나 웹을 이용하여 정보를 받거나 조작이 가능할 것으로 판단된다.

#### 참고문헌

- [1] Python, <https://docs.python.org/3/>
- [2] SimpleHTTPServer, <https://docs.python.org/2/library/simplehttpserver.html>