

IOT 프로그래밍

# Multi Processing을 이용한 센서 제어 프로그램

---

오승은 201758074

# 시나리오

## 실험 환경 조정 프로그램

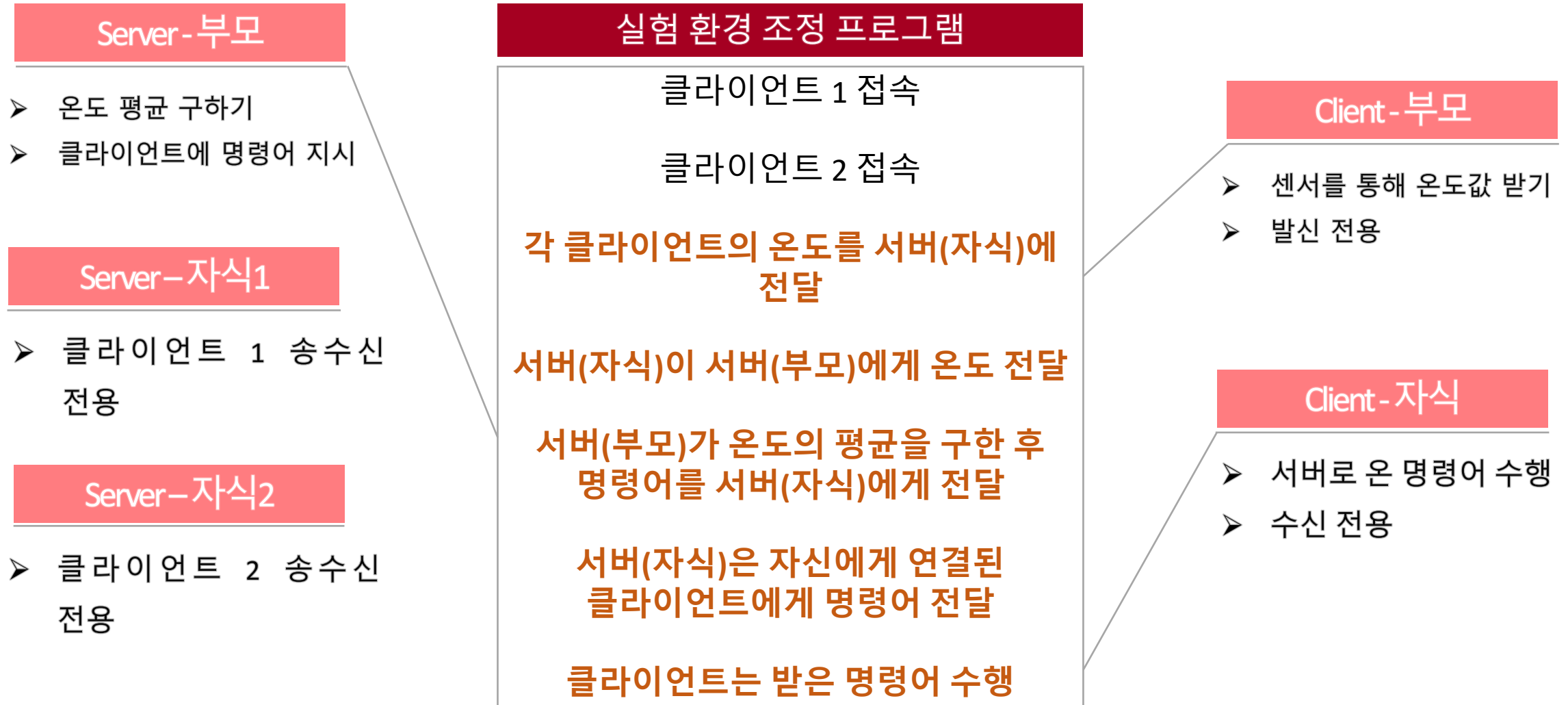


<https://blog.naver.com/mariakim98/191848061> [출처]

실험에서 대조군이 온도인 경우 실험이 이뤄지고 있는 각 공간의 온도가 서로 동일해야 한다.  
예로 좋은 말을 듣는 식물과 나쁜 말을 듣는 식물의 성장 차이를 확인하는 실험을 생각해 보자.  
실험군으로는 좋은 말과 나쁜 말이 있고, 대조군 중에는 온도가 있다.  
좋은 말을 듣는 식물이 자라는 공간과 나쁜 말을 듣는 식물이 자라는 공간의 온도는 대조군이기 때문에 온도가 서로 같지 않으면 실험이 실패하게 된다.  
그러므로 각 식물이 자라는 공간의 온도가 같도록 프로그램을 만들어 보았다.

프로그램 결론 : 각 공간의 온도를 동일하게 만들기

# 시스템 구성도



# Server - 부모

```
if (fork_num == 2)
{
    while (1)
    {
        read(fd1[0], buffer, sizeof(buffer));
        temp[0] = atof(buffer);
        read(fd2[0], buffer, sizeof(buffer));
        temp[1] = atof(buffer);
        getAve(temp);
        order(temp, buffer, 0);
        write(fd1[1], buffer, sizeof(buffer));
        order(temp, buffer, 1);
        write(fd2[1], buffer, sizeof(buffer));
        strcpy(buffer, "#0");
        sleep(5);
    }
}
```

## getAve() 함수

```
void getAve(double temp[])
{
    double tmp = (temp[0] + temp[1]) / 2.;
    temp[2] = tmp;
}
```

## order(double temp[],char \*buffer,int i) 함수

```
void order(double temp[], char *buffer, int i)
{
    strcpy(buffer, "#0");
    if (temp[i] > temp[2])
        strcpy(buffer, "hot");
    else if (temp[i] < temp[2])
        strcpy(buffer, "cold");
    else if (temp[i] == temp[2])
        strcpy(buffer, "average");
}
```

# Server - 자식

```
if (pid == 0) //자식
{
    close(x);
    while (1)
    {
        str_len = read(y, buffer, bufsiz);
        printf("receive message: %s\n", buffer);
        if (fork_num == 1)
        {
            write(fd1[1], buffer, sizeof(buffer));
            usleep(10);
            strcpy(buffer, "W0");
            read(fd1[0], buffer, sizeof(buffer));
            write(y, buffer, sizeof(buffer));
            puts(buffer);
            strcpy(buffer, "W0");
        }
        else
        {
            write(fd2[1], buffer, sizeof(buffer));
            usleep(10);
            strcpy(buffer, "W0");
            read(fd2[0], buffer, sizeof(buffer));
            write(y, buffer, sizeof(buffer));
            puts(buffer);
            strcpy(buffer, "W0");
        }
        sleep(5);
    }
}
```

## 자식 1일 때 (빨강 박스)

- fd1 파이프를 사용하여 부모와 통신
- 클라이언트와의 송수신 전용

## 자식 2일 때 (파랑 박스)

- fd2 파이프를 사용하여 부모와 통신
- 클라이언트와의 송수신 전용

자식 1과 자식 2가 같은 파이프를 쓰게 되면 메시지가 겹쳐 오류가 날 수 있기 때문에 자식별로 다른 파이프를 사용하게 함.

## Client - 부모

```
else
{
    if (wiringPiSetup() == -1)
    {
        return 1;
    }

    while (1)
    {
        temp = dht11_read_val();

        sprintf(message, "%f", temp);

        if (write(x, message, strlen(message) + 1) == -1)
        {
            perror("client: write");
            break;
        }
        strcpy(message, "#0");
        sleep(10);
    }
}
```

### float dht11\_read\_val() 함수

- 센서를 통해 온도를 측정하는 함수
- 온도를 float로 반환한다

### 온도를 받은 후

- float형을 문자열 배열로 변환한다
- 서버(자식)에 문자열 배열을 전송한다

## Client - 자식

```
else if (pid == 0)
{
    while (1)
    {
        int str_len = read(x, message, sizeof(message));
        if (str_len == -1)
        {
            perror("parent client: read");
            break;
        }
        if (str_len == 0)
            break;
        puts(message);
        moter(message);
        strcpy(message, "#0");
    }
    close(x);
    return 0;
}
```

### moter(char \*message) 함수

```
if (strcmp(mes, "") == 0 || strcmp(mes, "average") == 0)
```

DC 모터와 STEP 모터의 작동을 중지시킴

```
else if (strcmp(mes, "hot") == 0)
```

DC 모터를 일정 시간동안 작동시킴

```
else if (strcmp(mes, "cold") == 0)
```

STEP 모터를 일정 시간동안 작동시킴