

- 공학설계 공모전 입상 보고서

최종 결과 보고서

소 속	숭실대학교 IT대학 전자정보공학부 (오늘수업여기까지)조	작 성 자	20170633 조성준
실 험 일 시	2017년 12월 9일 공학설계입문경진대회 출품 및 최종 결과 발표	장 소	교육관 301호
실 험 주 제	아두이노와 라즈베리 파이를 이용한 유비쿼터스 스마트 홈 서비스		
실 험 준비물	나무 합판, 모니터, 유리, 반투명 필름, 아두이노 키트(나노,우노), 라즈베리 파이, 랜선, 케이블, 커튼재료(미니 수납함, 하드보드지빨대, 미니김밥말이), 공기청정기 재료(미니 선풍기), 스위치 재료(미니 스위치 전구), 보조배터리, 테이프, 본드, 칼, 톱, 멀티 탭		
실 험 구 성 및 과 정	<p><u>아래 밑줄 친 부분이 팀 프로젝트에서 제가 담당한 역할입니다.</u></p> <p>이제는 로봇이 인간의 일을 덜어주는 시대가 점점 다가오고 있다. 사람들은 집 안팎 어디서든 사람이 로봇에게 명령하기만 하면 집안일을 미리 대신해서 해주기를 원한다. 예를 들어 집안에 들어가기 전에 미리 전등불 켜놓기, 에어컨, 난방을 켜놓기를 원한다. 이렇게 어디서든 손쉽게 집안일을 리모컨으로 조절하는 것을 유비쿼터스 스마트 홈 서비스라고 한다. 우리 조는 이번 실험에서 이런 서비스를 간단하게 시연하고자 이 프로젝트를 시도하게 되었다.</p> <p>실험은 크게 아두이노 부분과 라즈베리 파이와 Node.js를 연계한 부분으로 나눌 수 있다. 공기청정기, 커튼, 스위치는 모두 아두이노를 이용하여 만들었다. 그리고 모니터와 음성인식을 연계시켜서 필요한 정보를 요청하는 것들은 모두 라즈베리 파이와 Node.js를 이용하여 만들었다. 그리고 아두이노와 음성 인식 프로그램 중심에는 라즈베리 파이가 있다고 할 수 있다.</p> <p><u>커튼은 스텝모터를 이용하여 만들었다. 스텝 모터가 360도 회전하는 특징을 이용하였다. 우선 수납함 양 옆에 구멍을 뚫어 빨대를 끼웠다. 그리고 스텝 모터의 축을 빨대에 접착제로 고정시켰다. 빨대에는 김밥 김말이를 매달아서 전체적인 커튼을 완성시켰다. 코드는 리모컨과 서보모터 라이브러리를 기본적으로 이용하였다. 스텝모터 신호핀을 11, 9, 10, 8번에 연결하고 리모컨 수신부를 나노의 12번 핀에 연결하는 것으로 코드를 설정하였다. 또한 리모컨의 각 버튼 코드(HEX 값을 따서 case문에 입력)를 따서 if문, case문을 새로 만들고 스텝 모터 코드를 그에 맞게 새로 짜서 리모컨이 명령하는 대로 스텝 모터가 구동되도록 하였다. 스텝 모터의 스피드는 30으로 맞췄다. 30 이상</u></p>		

으로 해보았는데 진동만 울릴 뿐 움직이지를 않아서 다른 방법도 알아보았다. 하지만 30이 최선의 속도여서 그냥 30으로 고정시키고 코드를 완성시켰다. 리모컨 버튼 왼쪽을 누르면 시계방향으로 버튼 오른쪽을 누르면 반시계 방향으로 회전되게 하였다. 아두이노 키트 조립을 할 때는 우선 우노 대신에 나노를 이용하였다. 아두이노를 처음 다운받을 때에는 우노 드라이버는 있었지만 나노 드라이버는 없어서 새로 다운받아야 했다. 나노가 우노에 비해 크기가 많이 작았지만 나노도 있을 건 다 있었다. 둘에 차이가 있다면 스텝모터 도선을 우노에서 할 때처럼 모듈 보드로 갔다가 빵판에다가 꽂았다가 우노의 GND와 VCC로 가는 것이 아니라 모듈 보드로 갔다가 직접 나노에다가 VCC, GND를 모두 꽂는 것이었다. 그리고 모듈 보드에서 나온 신호 도선 4개를 나노에 직접 꽂았다.

스위치는 서보모터를 이용하여 만들었다. 어떤 벽에 있는 스위치든 쉽게 원격으로 조절할 수 있다는 것을 보여주기 위해서 서보모터가 직접 스위치 끄는 것을 시연하였다. 커튼과 마찬가지로 리모컨의 각 버튼 코드(HEX값을 case문에 입력)를 따서 if문, case문을 새로 짜고 서보모터 코드와 연결시켰다. 그래서 버튼 1을 누르면 각도가 70도에 가도록, 버튼 2를 누르면 145도에 가도록 하여서 서보모터의 날개가 움직일 때 전구의 스위치 on, off에 들어맞도록 하였다. 서보모터 코드는 라이브러리를 이용했다. 또한 신호핀을 나노의 9번핀에 연결하고 리모컨 수신부를 11번핀에 연결하도록 코드에 입력했다. 아두이노 키트 조립을 할 때에는 서보 모터의 세 개의 선(GND, VCC, 신호9번핀)을 직접 나노에다가 꽂았다. 우노처럼 빵판을 거치지 않고 애초에 나노가 워낙 작다보니까 전체 회로의 크기가 작고 간단해졌다. 오히려 보조배터리와 나노를 연결하는 케이블이 크기비중을 많이 차지했다.

공기청정기는 미니 선풍기의 전선 피복을 벗겨서 아두이노 도선과 연결시켰다. 그리고 코드로 명령을 주면 선풍기가 작동되도록 하였다. 또한 미세 먼지 감지 센서를 연결시켜서 일정 양 이상의 먼지가 감지되면 미니 선풍기가 작동되도록 코드를 짰다.

라즈베리 파이는 모니터와 직접 연결한다. 라즈베리 파이라는 것은 모니터, 키보드, 마우스와 연결만 하면 구동이 되는 초소형 컴퓨터이다. 스마트 미러라는 개념을 구현시키기 위해서 정말 작은 컴퓨터라고 할 수 있는 라즈베리 파이를 이용하게 되었다. 그래서 나무 합판으로 만든 직육면체 안에 모니터를 넣고 라즈베리 파이를 연결시켜서 작은 상자 안에서 컴퓨터가 구동되게 한 것이다. 모니터에 보이는 프로그램은 Node.js를 이용하여 짰다. Node.js는 자바스크립트를 기반으로 한 것으로 아직 전자정보공학부 1학년 학생들은 배우지 않은 프로그래밍 언어이다. 그리고 송신부 센서의 코드는 아두이노를 기반으로 짰으며 리모컨의 각 값을 따와 case문과 연결하였다. 송신부 센서는 음성 명령 신호들을 총괄하여 적절한 곳으로 송신한다고 볼 수 있으며 간단히 말하면 음성 명령을 모두 모아서 송신부에서 수신부로 신호를 보낸다고 볼 수 있다. 우리 프로젝트의 최종 목적은 음성 인식으로 명령을 내려서 웹 브라우저를 이용해 원하는 정보를 가져오는 것이다. 그래서 지하철 도착 정보, 유튜브 동영상, 지도 등을 웹 브라우저에서 가져와 모니터에 띄우는 것이다. 또한 음성으로 명령(커튼, 스위치, 공기청정기 조작 명령)을 내리면 아두이노 리모컨 송신부로 신호를 보내서 리모컨의 역할을 송신부 센서가 대신 담당하는 것이다. 그러면 수신부 센서가 신호를 받아 일을 수행하는 것이다. 이렇게 사람은 목소리로 명령만 내리면 자동으로 컴퓨터(로봇)가 일을 수행하는 것이 우리 프로젝트의 최종 목적이다.

(음성명령-라즈베리파이-송신부 센서-수신부 센서-아두이노 요구사항 수행)

(음성명령-라즈베리파이-웹 브라우저-라즈베리파이-모니터에 요구사항 수행)

나의 역할을 요약하자면 위의 글에서도 대략적으로 알 수 있듯이 아두이노를 이용해 커튼과 스위치 만드는 것을 주로 담당하고 모니터와 나무 합판을 합치는 것을 보조 담당했다. 구체적으로는 커튼과 스위치 전체 구성과 코드 짜기, 도선 정리를 하였다. 처음에는 이 정도는 혼자 할 수 있을 것이라 생각하였지만 하다보니까 혼자 했다면 절대 해내지 못했을 것이다. 아두이노 나노라는 유용한 제품을 친구를 통해서 알았으며 함께 코드를 다 짜고 하드웨어를 구현시킬 때에도 여러 부분에서 협동을 해서 해낼 수 있었다. 커튼을 만들 때에도 머리를 맞대니 좀 더 간결하게 쉽게 만들 수 있었고 스위치로 껐다 켜다 하는 전구의 스위치 부분에 서보모터 날개를 붙일 때에도 정말 애먹었는데 결국에는 해낼 수 있었다. 진짜 사소한 것(예를 들어 접착제를 어디에 붙일까?)까지 함께 의논하고 하니 쉽고 빠르게 일을 진행할 수 있었다.

```
<송신부 센서 코드 1>
#include <IRremote.h>  // 적외선 송수신을 하기 위해서
IRRemot 라이브러리를 사용한다.
IRsend irsend;
int i = 0;
long inputInt;
void setup(){
  Serial.begin(9600);
}

void loop(){

  inputInt = Serial.parseInt();
  switch(inputInt){
    case 0:
      Serial.print(" ");
      break;
    case 1: //커튼올려
      for(i = 0; i < 3; i++){
        irsend.sendSony(0xFF5AA5,32);
        delay(40);
        Serial.print("1");
      }
      break;
    case 2: //커튼내려
      for(i = 0; i < 3; i++){
        irsend.sendSony(0xFF10EF,32);
        delay(40);
        Serial.print("2");
      }
      break;
```

```
<송신부 센서 코드 2>
case 3: //불켜
  for(i = 0; i < 3; i++){
    irsend.sendSony(0xFFA25D,32);
    delay(40);
  }
  Serial.print("불켜");
  break;
case 4: //불
  for(i = 0; i < 3; i++){
    irsend.sendSony(0xFF629D,32);
    delay(40);
  }
  Serial.print("불꺼");
  break;
case 5: //공기청정기를 켜
  for(i = 0; i < 5; i++){
    irsend.sendSony(0xFFE21D,32);
    delay(40);
  }
  break;
case 6: //공기청정기를 꺼
  for(i = 0; i < 5; i++){
    irsend.sendSony(0xFF22DD,32);
    delay(40);
  }
  break;
  }
  inputInt = 0;
  delay(200);
}
```

스텝모터 + 리모컨 코드

```
#include <Stepper.h>
#include <IRremote.h>

const int stepsPerRevolution = 1024;
Stepper myStepper(stepsPerRevolution,11,9,10,8);

int RECV_PIN=12;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup() {
  myStepper.setSpeed(30);
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop(){
  if(irrecv.decode(&results)){
    Serial.println(results.value, HEX);

    irrecv.resume();
    switch(results.value){
      case 0xFF10EF:
myStepper.step(stepsPerRevolution); break;

    } }

    if(irrecv.decode(&results)){
      Serial.println(results.value, HEX);

      irrecv.resume();
      switch(results.value){

        case 0xFF5AA5:
myStepper.step(-stepsPerRevolution); break;
      }
    } }
```

서보모터 + 리모컨 코드

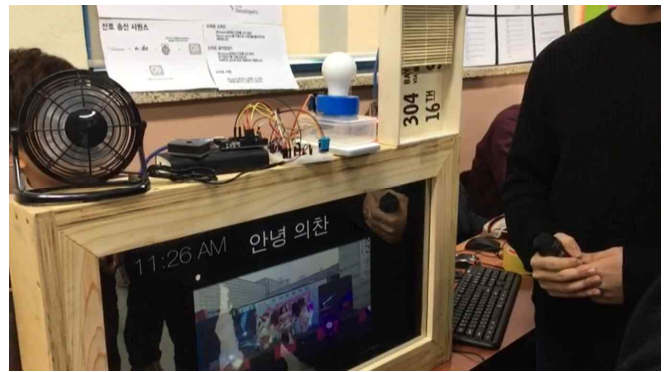
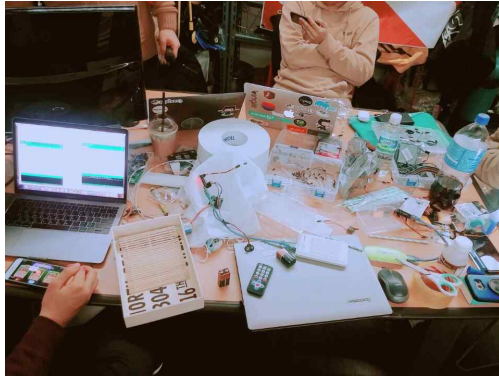
```
#include <IRremote.h>
#include <Servo.h>
Servo servo;

int RECV_PIN=11; //11번 핀에 적외선 리모콘 수신
부 연결

IRrecv irrecv(RECV_PIN);

decode_results results;
//
void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();
  servo.attach(9); // 서보모터 9번에 연결
}
//
void loop(){
  if(irrecv.decode(&results)){
    Serial.println(results.value, HEX);
    irrecv.resume();
    switch(results.value){
      case 0xFFA25D: servo.write(70); break;
// 버튼 1
      case 0xFF629D: servo.write(145); break;
// 버튼 2 //80, 135

    }
  }
}
```



실험 결과 및 분석

준비하면서 어려운 점이 정말 많았다. 사소하게 나무 합판 치수 재는 것부터 프로그램 코딩하는 것까지 종류도 다양했다. 가장 난해했던 점은 나무 합판을 모두 적절한 치수에 맞게 준비해왔는데 모니터의 받침대 높이 약 7cm 때문에 직사각형 안에 모니터가 들어가지 않는 것이다. 그래서 별의 별 방법을 모두 생각해보다가 결국 모니터 받침대 부분을 자르게 되었다. 아무리 대회를 위해서라지만 좀 아깝기도 하였고 왜 모니터 받침대 높이를 처음부터 생각하지 않았을까 하고 회의감이 들기도 하였다.

모니터 자르냐 마냐 뿐만 아니라 준비 과정의 95%가 총체적 난국이었다. 그저 사소하게 커튼 만드는 것, 스위치 만드는 것, 코드 한 줄 한 줄 짜는 것, 스위치에 서보모터를 붙이는 것 등 모두 머릿속으로 구상하고 있어도 막상 재료를 다이소에 가서 사려고 하고 코드를 손으로 쳐보고 실제로 서보모터를 붙여보니 현실적인 문제에 부딪히게 되며 걸림돌이 되는 것이 이만저만이 아니었다. 그럴 때에는 인터넷을 이용해보거나 최대한 비슷한 다른 방법을 동원해 해결하고자 정말 많이 노력하였다.

또한 이번에 프로젝트 협동을 하면서 서로에 대해서 좀 더 자세하게 알게 되었다. 또한 한 가지

	<p>주제를 가지고 각자 의견을 내면서 수렴하는 법도 제대로 배우는 계기가 되었고 아무리 어렵고 힘든 문제라도 계속 하다보면 언젠가는 해결될 수 있다는 것을 정말 처음 깨달았다. 원래의 나는 어려운 것은 뛰어넘고 할 수 있는 것부터 하자라고 보통 생각해온 편이다. 하지만 이번 프로젝트에서 안 되는 것을 붙들고 꼭 해내야 된다고 생각하며 계속 하다보니까 언젠가 결국 풀리는 것을 볼 수 있었다. 그리고 꼭 어려운 문제를 뛰어넘지 않아도 일이 결국에는 풀릴 수 있구나라는 것을 제대로 느낄 수 있었다.</p>		
토 의	<p>컴퓨터에 대해서 올해 처음 배워보는 나는 공학설계입문 시간 초반에 교수님이 수업하시는 내용들이 전혀 이해가 안가서 매우 심란했다. 고민하다가 결국 인터넷에서 개인적으로 아두이노를 구매가지고 혼자 집에서 해보았다. 그렇게 집에서 혼자 아두이노를 만져보고 코딩도 해보니 흥미도 늘어났다. 덕분에 단기간 동안 쌓은 실력과 몇몇 친구들의 유능한 코딩 실력이 함께 합쳐져서 이번 대회에서 좋은 성적을 얻을 수 있었지 않았나 생각한다. 결론은 단기간에 무에서 유를 창조한 것 같아서 진심으로 내 자신이 뿌듯하고 신기하다.</p>		
참 석 자	조성준, 전민규, 주현우, 이은찬, 전의찬	확 인	조성준 (인)