

Micro-controller

미니프로젝트 #3 (감시 시스템)

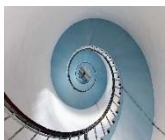
미니프로젝트발표: **11월25일(수)** 오후 **2시 ~ 4시**

금오공과대학교 기계시스템공학과

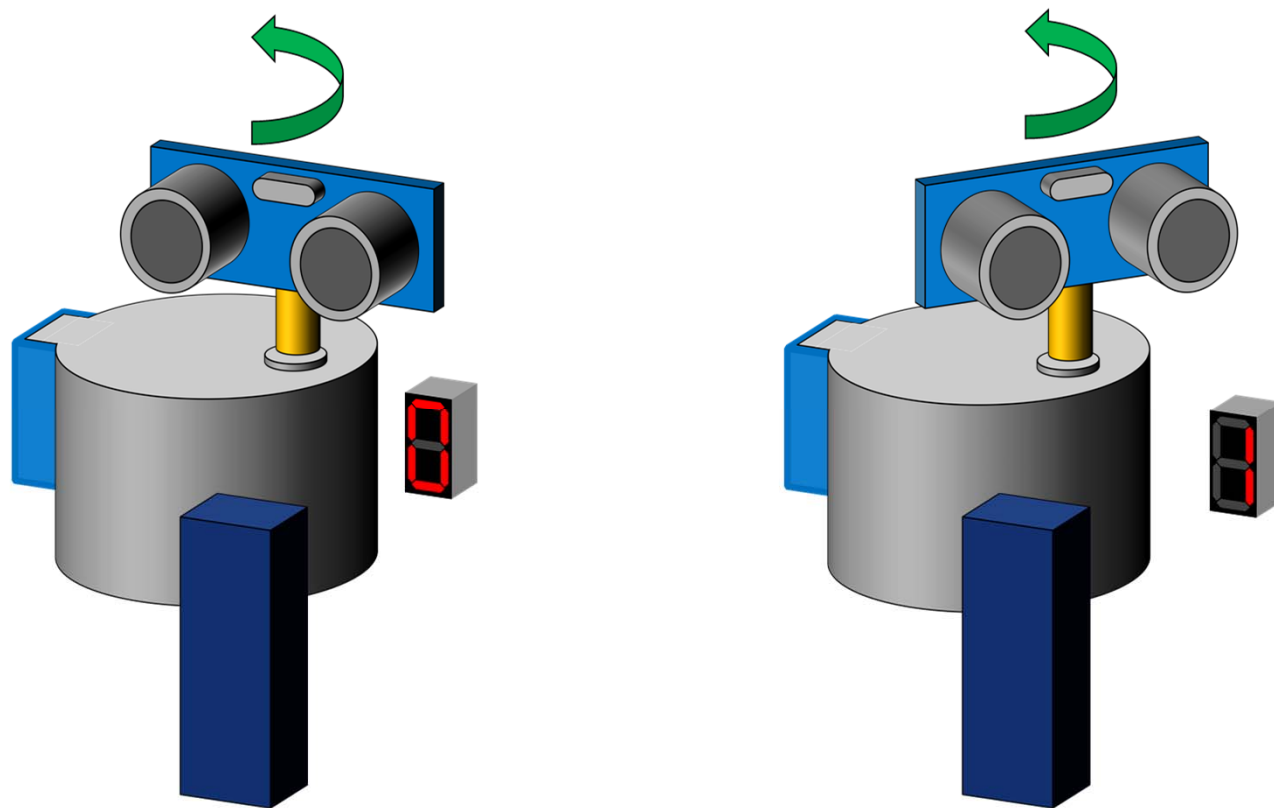


미니프로젝트 3 내용

1. 미니프로젝트 2에서 **DC**모터를 추가.
2. 주요 구성요소: **DC**모터/초음파거리센서/**7**-세그먼트/부저
3. 시스템 구성 방법:
 - ① **DC**모터에 초음파거리 센서를 회전이 가능하도록 설치
 - ② 브레드 보드에 **7**-세그먼트와 부저 설치
4. 작동 프로그램 구성
 - ① **DC**모터를 시계방향/반시계방향으로 지속적으로 회전시킴
 - ② 매 **20ms** 회전 시마다 초음파거리센서를 이용하여 전방의 물체와의 거리를 측정, 일정 값 이하가 될 때 **7**-세그먼트의 값을 **1**씩 증가시킴.
 - ③ **7**-세그먼트는 최초에는 '**0**'을 나타내도록 하고 물체가 감지될 때 마다 값을 증가시키며 **9**가 넘어가면 부저를 **1**회 울리고 다시 **0**부터 시작함.



미니프로젝트 3 시스템 예시



System 예시



미니 프로젝트 3 데모

11월25일 (수) 14시 ~ 17시

1. Webex 실시간 화상회의 시스템 활용
2. 2개의 그룹으로 나누어 발표 진행
1그룹: 학번 20140548 ~ 201600651 (14명) : 14시~15시
2그룹: 학번 20160653 ~ 20160768 (15명) : 15시~16시
3. 개인별로 준비된 시스템을 화상으로 데모함.

주의사항:

- 미팅방 링크 <https://kumoh.webex.com/join/swhong>
- 발표시작 5분전까지 입장하기 바랍니다.



미니프로젝트 3 코딩

```
#define BUZZER 11 // 부저를 11번 핀
#define TRIGPin 12 // 초음파센서 트리거 12
#define ECHOPin 13 //초음파 센서 에코 13
int ena = 10;          //모터 enable
int in1 = 14;          //모터 in 1 A0
int in2 = 15;          //모터 in 2 A1
int count=0;          //초음파센서 카운트
int LEDpins[] = {2, 3, 4, 5, 6, 7, 8, 9};
int NumLEDs = 8; // LED 개수
// 도, 레, 미, 파, 솔, 라, 시, 도
int tone_f[] = {261, 294, 330, 349, 392, 440, 494, 523};
int digitmap[10][8] = { //숫자에 대한 맵 (0 ~ 9)
    {0, 0, 0, 0, 0, 0, 1, 1}, //0
    {1, 0, 0, 1, 1, 1, 1, 1}, //1
    {0, 0, 1, 0, 0, 1, 0, 1}, //2
    {0, 0, 0, 0, 1, 1, 0, 1}, //3
    {1, 0, 0, 1, 1, 0, 0, 1}, //4
    {0, 1, 0, 0, 1, 0, 0, 1}, //5
    {0, 1, 0, 0, 0, 0, 0, 1}, //6
    {0, 0, 0, 1, 1, 1, 1, 1}, //7
    {0, 0, 0, 0, 0, 0, 0, 1}, //8
    {0, 0, 0, 0, 1, 0, 0, 1} //9
};
```



미니프로젝트 3 코딩

```
void setup ()
{
  for (int i = 0 ; i < NumLEDs; i++) {
    pinMode(LEDpins[i], OUTPUT);    //2~9 7세그먼트
  }
  pinMode(TRIGPin, OUTPUT);    //초음파센서
  pinMode(ECHOPin, INPUT);    //초음파센서
  pinMode(BUZZER, OUTPUT);    //부저
  pinMode(in1, OUTPUT);    //모터
  pinMode(in2, OUTPUT);    //모터
  pinMode(ena, OUTPUT);    //모터

  sevSEG(0);}

```



미니프로젝트 3 코딩

1. 정방향 회전(80으로 20ms) 동작 후 정지하여 거리 측정을 반복함. (20~40 회, for 문 이용하면 편리). 거리를 측정하여 정해진 거리(30cm) 이하이면 count 증가 시킴. count가 9가 넘으면 부저 작동 및 count를 0으로 치환. 7 세그먼트에 count를 출력.
2. 정방향 작업이 종료되면 역방향 회전(80으로 20ms) 동작 후 정지하여 거리 측정하는 작업을 반복함 (정방향에서와 마찬가지로 20~40회 반복). 거리를 측정하여 정해진 거리(30cm) 이하이면 count 증가시킴. count가 9가 넘으면 부저 작동 및 count를 0으로 치환. 7 세그먼트에 count를 출력.

```
//정방향 회전, 80으로 20ms 동작 후 정지
```

```
digitalWrite(in1, HIGH); digitalWrite(in2, LOW);  
analogWrite(ena, 80); delay(20); analogWrite(ena, 0);
```

```
//역방향 회전, 80을 20ms 동작 후 정지
```

```
digitalWrite(in1, LOW); digitalWrite(in2, HIGH);  
analogWrite(ena, 80); delay(20); analogWrite(ena, 0);
```



미니프로젝트 3 코딩 - 자체 정의 함수

```
void sevSEG(int i) {  
  for (j = 0 ; j < NumLEDs ; j++) { //Turn on or off the LEDs  
    digitalWrite(LEDpins[j], digitmap[i][j]);  
  }  
  delay(10);  
}
```



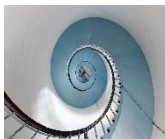

미니프로젝트 3 코딩 - 자체 정의 함수

```
int distance()
{
    int duration, cm;
    digitalWrite(TRIGPin, LOW);           //making a pulse
    delayMicroseconds(2);                  //
    digitalWrite(TRIGPin, HIGH);           //
    delayMicroseconds(10);                 //
    digitalWrite(TRIGPin, LOW);            //
    duration = pulseIn(ECHOPin, HIGH);     //receiving the echo
    // convert the time into a distance
    cm = duration/29/2;
    return(cm);
}
```



미니프로젝트 3 코딩 - 자체 정의 함수

```
void play() {  
  for (int i = 0; i < 10; i++)  
  {  
    tone(BUZZER, tone_f[0]);    //play DO-Do  
    delay(500);  
    tone(BUZZER, tone_f[7]);  
    delay(500);  
  }  
  noTone(BUZZER); // 소리 중단  
}
```



미니프로젝트 3 실행 동영상

