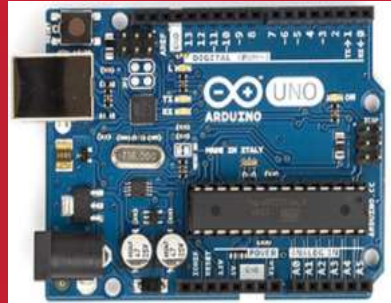
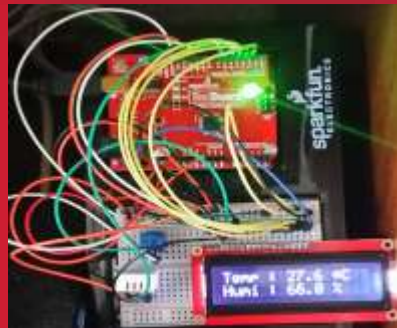


# Arduino-basic

[wk14]



## Various elements



Learn how to code Arduino from scratch

Comsi, INJE University

2<sup>nd</sup> semester, 2018

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)

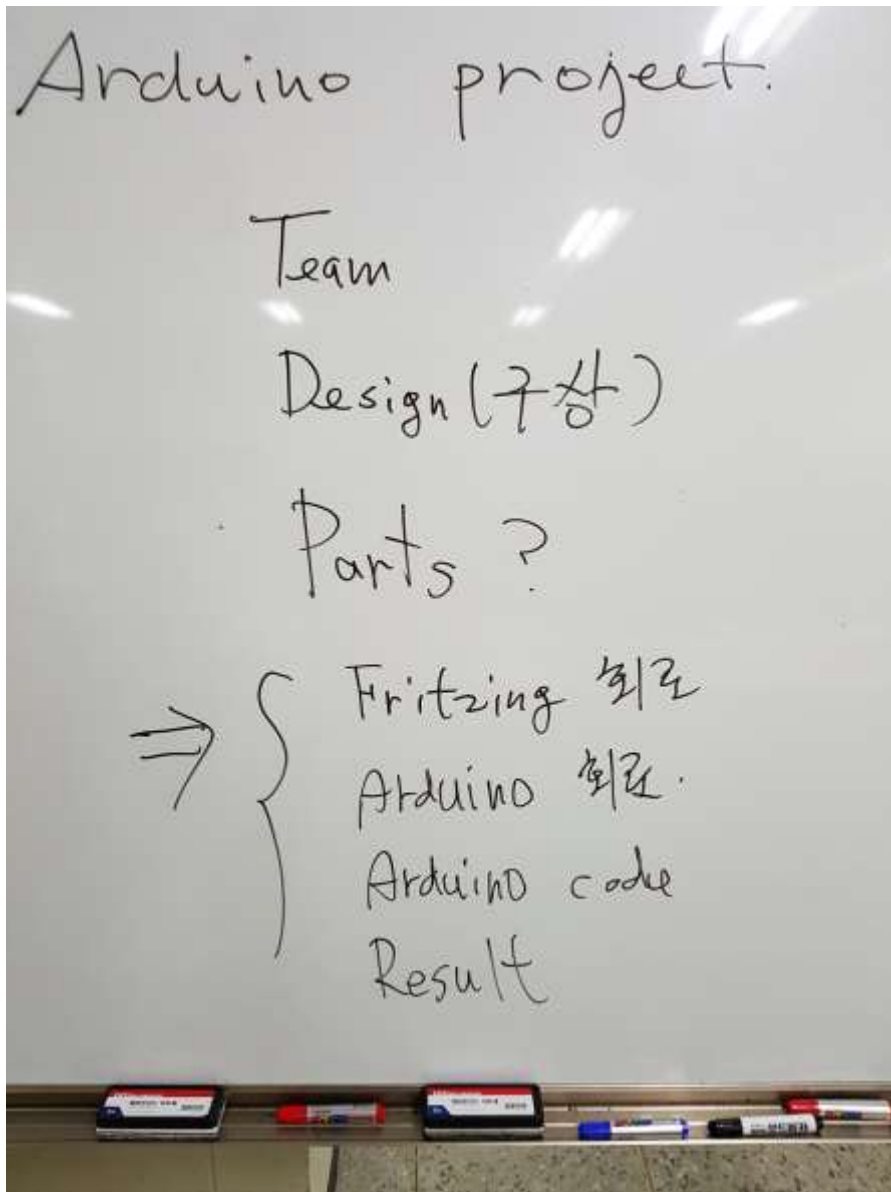


# My ID (ARnn)

성명	ID
백동진	AR01
김도훈	AR02
김희찬	AR03
류재현	AR04
문민규	AR05
박진석	AR06
이승현	AR07
이승협	AR08
이후정	AR09
최민구	AR10

김다영	AR11
공진영	AR12
김해인	AR13
류성현	AR14
류재환	AR15
박상현	AR16
박해주	AR17
백지혜	AR18
송원식	AR19
신송주	AR20
윤지훈	AR21
정은성	AR22

# Arduino team project



- 2명/팀
- 구상 소개 (11.22, 11.29), ppt 준비
- 부품은 수업 세트 기준 (추가신청은 22일까지)
- 팀당 발표 자료 준비
- 발표 : 12월6일
- 참고

# Arduino team

- ✓ • 류성현, 박진석 (약분류기)
- ✓ • 문민규, 이희정 (심박센서)
- ✓ • 백동진, 이승협 (코그)
- ✓ • 김도훈, 김희찬 (자동문)
- ✓ • 박상현, 정은성 (위험지역 탐색용 RC카)
- ✓ • 김해민, 신송주 (선풍기)
  - 최민지, 이승현 (신호등)
  - 윤지훈, 박해주 (눈발력 테스트)
  - 류재현, 공진영 (도어락)
  - 송원식, 류재환 (선풍기)
  - 백지혜, 김다영 (수면등)

# wk15 : 기말고사 안내

## [1] 실기 - 팀플

- 시간: 12월 6일 오후 2 시~
- 장소: E323 실습실
- 배점: 6점

**A(4팀), B(4팀), C(3팀)**

- 팀원의 기여도 반영 (6~0)
- 결과 제출
  - \* 발표 자료 (pdf)
  - \* 아두이노 스케치(ino file)
- 강의 **github**에 **project** 폴더로 업!

## [2] 필기

- 시간: 12월 13일 오후 2 시~3시
- 장소: E323 실습실
- 배점: 15 점 (보너스 1점 포함)



# [Review]

## ◆ [wk13]

- **Arduino : Infrared remote**
- **Complete your project**
- **Submit file : ARnn\_Rpt10.zip**

# wk13 : Practice-10 : ARnn\_Rpt10.zip

## ◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn\_Rpt10.zip**

- 압축할 파일들

① **ARnn\_servo\_motor.ino**

② **ARnn\_remote\_LCD.png**

③ **ARnn\_remote\_LED.png**

④ **ARnn\_remote\_LED.ino**

**Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)**

**[ 제목 : id, 이름 (수정) ]**

```
// 모터 각도 변수 설정
int motorAngle;
int motorAngleOld;
const int cwPin = 2;
const int ccwPin = 3;
boolean c = false;
boolean c_c = false;
void setup() {

    // 서보모터 설정. 0.6ms 부터 2.4ms 범위로 설정
    motor1.attach(servoMotorPin,600,2400);
    pinMode(ccwPin, INPUT_PULLUP);
    pinMode(cwPin, INPUT_PULLUP);
    // 시리얼 통신 설정
    Serial.begin(9600);
}

void loop(){
    // 포텐쇼미터 값을 읽어옴

    int cw = digitalRead(cwPin);
    int ccw = digitalRead(ccwPin);
    // 포텐쇼미터 값을 모터 각도로 변환한다
    if(cw == LOW){
        c = true;
        c_c = false;
    }
    else if(ccw == LOW){
        c = false;
        c_c = true;
    }
    if(c == true){
        motor1.write(0);
    }
    else if(c_c == true){
        motor1.write(270);
    }
    // 모터에 각도값을 전달한다
```

```
// 모터 각도 변수 설정
int motorAngle;
int motorAngleOld;
int sw1 = 4;
int sw2 = 5;
boolean a = false;
void setup() {

    // 서보모터 설정. 0.6ms 부터 2.4ms 범위로 설정
    motor1.attach(servoMotorPin,600,2400);
    pinMode(sw1, INPUT_PULLUP);
    pinMode(sw2, INPUT_PULLUP);
    // 시리얼 통신 설정
    Serial.begin(9600);
}

void loop(){
    // 포텐쇼미터 값을 읽어옴
    int potentiometer = analogRead(potentiometerPin);
    int sw1Input = digitalRead(sw1);
    int sw2Input = digitalRead(sw2);

    // 포텐쇼미터 값을 모터 각도로 변환한다
    motorAngle = map(potentiometer,0,1023,0,180);

    // 모터에 각도값을 전달한다
    if(!a)
    {
        if (sw1Input==LOW)
        {
            motor1.write(motorAngle);

            a=true;
        }
    }
    if(sw2Input==LOW)
    {
        motor1.write(motorAngle-motorAngleOld);
        a=false;
    }
}
```



```
// 모터 각도 변수 설정
const int right=4;
const int left=5;
int count=90;
int count1;
void setup() {

    // 서보모터 설정. 0.6ms 부터 2.4ms 범위로 설정
    motor1.attach(servoMotorPin,600,2400);

    // 시리얼 통신 설정
    Serial.begin(9600);
    pinMode(right,INPUT_PULLUP);
    pinMode(left,INPUT_PULLUP);
}

void loop(){
    // 모터에 각도값을 전달한다
    if(digitalRead(right)==LOW)
    {
        count=180;
    }
    if(digitalRead(left)==LOW)
    {
        count=0;
    }
    motor1.write(count);
```

```
void loop(){

    // 포텐쇼미터 값을 읽어옴
    int potentiometer = analogRead(potentiometerPin);
    int b= digitalRead(sw1);
    int c= digitalRead(sw2);
    motorAngle= map(potentiometer,0,1023,0,180);

    // 포텐쇼미터 값을 모터 각도로 변환한다
    if(!G)
    {
        if (b==LOW)
        {
            if (motorAngle<=180)
            {
                motor1.write (motorAngle);
                G=true;
            }
        }

        // 모터에 각도값을 전달한다
        if (c==LOW)
        {
            motor1.write (motorAngle-motorAngleOld);
            G=false;
        }
    }
```

```
void loop() {
  int leftInput = digitalRead(leftPin);
  int rightInput = digitalRead(rightPin);

  if (digitalRead(leftPin) == LOW) {
    cnt = 0;
  }
  if (digitalRead(rightPin) == LOW) {
    cnt = 180;
  }

  motor1.write(cnt);

  // 이전각도와 현재 각도가 같지 않으면 시리얼 모니터에
  // 각도를 출력한다.
  if(cnt != cntOld){
    Serial.print("Servo Motor Angle is: ");
    Serial.println(cnt);
  }

  // 현재의 모터 각도를 저장한다.
  cntOld = cnt;

  delay(20);
}
```

```
void loop() {
  // 포텐쇼미터 값을 읽어옴
  int potentiometer = analogRead(potentiometerPin);

  int swInput1 = digitalRead(inputPin1);
  int swInput2 = digitalRead(inputPin2);
  // 포텐쇼미터 값을 모터 각도로 변환한다
  motorAngle = map(potentiometer, 0, 1023, 0, 180);

  // 모터에 각도값을 전달한다
  motor1.write(motorAngle);

  // 이전각도와 현재 각도가 같지 않으면 시리얼 모니터에 각도를 출력한다.
  if(motorAngle != motorAngleOld){
    Serial.print("Servo Motor Angle is: ");
    Serial.println(motorAngle);
  }

  if(swInput1 == LOW){
    for(angle = 0; angle < 350; angle++){
      motor1.write(angle);
      delay(5);
    }
  }

  if(swInput2 == LOW){
    for(angle = 0; angle > 350; angle--){
      motor1.write(angle);
      delay(5);
    }
  }
}
```

```
// LED에 연결된 핀을 설정한다.
int led1 = 2;
int led2 = 3;
int led3 = 4;
int led4 = 5;

// LED 제어용 코드 (리모컨에 맞게 수정한다)
long on1  = 0xFFA25D;
long off1 = 0xFF629D;
long on2  = 0xFFE21D;
long off2 = 0xFF22DD;
long on3  = 0xFF02FD;
long off3 = 0xFFC23D;
long on4  = 0xFFE01F;
long off4 = 0xFFA857;

// 적외선 수신부가 연결된 핀을 리모컨 수신 핀으로 설정한다.
IRrecv irrecv(irPin);

// 수신된 신호의 결과를 results 변수로 설정한다.
decode_results results;

void setup()
{
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("LED");
  lcd.setCursor(0,1);
  lcd.print("");
  // 시리얼 통신을 설정한다.
  Serial.begin(9600);

  // 적외선 리모컨 수신을 시작한다.
  irrecv.enableIRIn();

  // 13번 핀에 연결된 LED를 리모컨 수신시 점멸시킨다.
  irrecv.blink13(true);

  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
}
```

```
void loop()
{
  // 수신된 코드가 있을 때 실행한다.
  if (irrecv.decode(&results)) {
    // 0xFFFFFFFF 값을 제외하고 출력한다.
    if(results.value != 0xFFFFFFFF){
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Remote");
      lcd.setCursor(0,1);
      // 수신된 코드가 on1과 같을 때
      if(results.value == on1){
        digitalWrite(led1, HIGH);
        Serial.println("LED1 is ON");
        lcd.print("LED1 is ON");
      }
      // 수신된 코드가 off1과 같을 때
      if(results.value == off1){
        digitalWrite(led1, LOW);
        Serial.println("LED1 is OFF");
        lcd.print("LED1 is OFF");
      }
      // 수신된 코드가 on2와 같을 때
      if(results.value == on2){
        digitalWrite(led2, HIGH);
        Serial.println("LED2 is ON");
        lcd.print("LED2 is ON");
      }
      // 수신된 코드가 off2와 같을 때
      if(results.value == off2){
        digitalWrite(led2, LOW);
        Serial.println("LED2 is OFF");
        lcd.print("LED1 is OFF");
      }
    }
  }
}
```

```
void loop()
{
    // 수신된 코드가 있을 때 실행한다.
    if (irrecv.decode(&results)) {
        // 0xFFFFFFFF 값을 제외하고 출력한다.
        if(results.value != 0xFFFFFFFF){
            // 수신된 코드가 on1과 같을 때
            if(results.value == on1){
                digitalWrite(led1, HIGH);
                Serial.println("LED1 is ON");
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("LED1 is ON");
                delay(100);
                lcd.setCursor(0,1);
                lcd.print(results.value,HEX);
            }
            // 수신된 코드가 off1과 같을 때
            if(results.value == off1){
                digitalWrite(led1, LOW);
                Serial.println("LED1 is OFF");
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("LED1 is OFF");
                delay(100);
                lcd.setCursor(0,1);
                lcd.print(results.value,HEX);
            }
        }
    }
}
```



# 9. Various elements

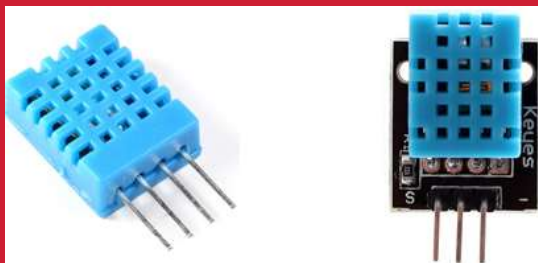
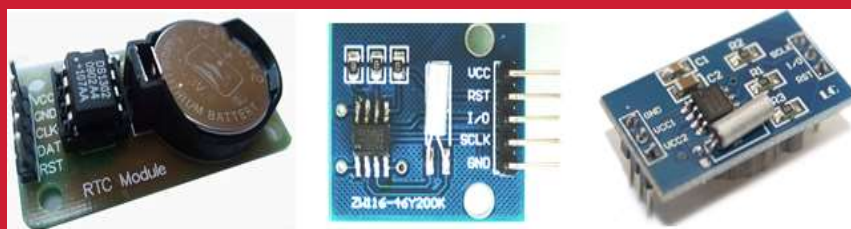


그림 9. 2 DHT11과 DHT11 모듈



## 9. 여러가지 부품들

9.1 버저

9.2 온습도 센서

9.3 실시간 클럭모듈

9.4 RFID

9.5 초음파 거리센서



# 9.1 버저



마그네틱 버저,

피에조 버저,

기계식 버저



## 버저(Buzzer)



그림 9.1 마그네틱 버저, 피에조 버저, 기계식 버저

- ✓ 전기적 신호로 진동판을 진동시켜 소리를 출력하는 부품
- ✓ 마그네틱 버저, 피에조 버저, 기계식 버저 등이 있음
- ✓ 피에조 버저는 일정 주파수를 입력시켜 다양한 음을 낼 수 있음

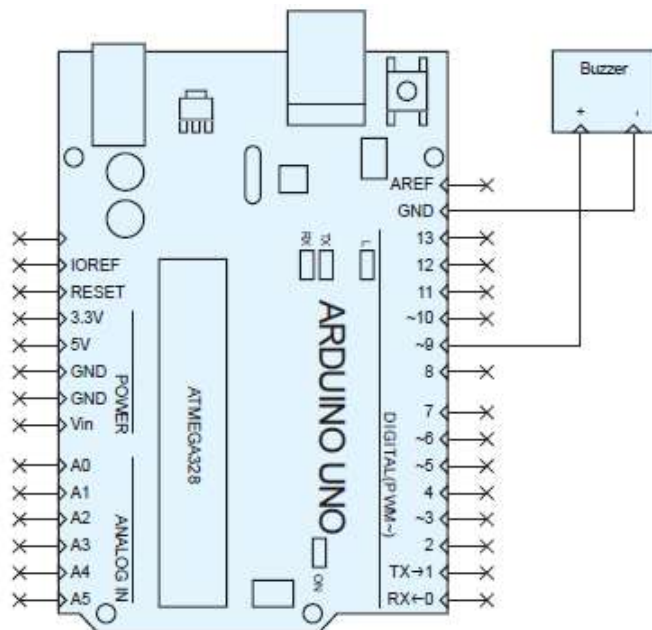


# 9.1.1 피에조 버저

## EX 9.1 피에조 버저를 이용한 소리 출력 (1/3)

**실습목표** 피에조 버저를 이용하여 다양한 소리를 출력한다.

- Hardware**
1. 버저의 (+)핀을 Arduino의 9번핀에 연결한다.
  2. 버저의 (-)핀을 Arduino의 GND에 연결한다.



## EX 9.1

## 피에조 버저를 이용한 소리 출력 (2/3)

**Commands**    • `pinMode(핀번호, 설정)`

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT\_PULLUP'을 적는다.

• `for(변수=시작 값 ; 조건 ; 변수의 증분){ }`

변수의 시작 값부터 조건이 만족하는 경우 '{ }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

• `tone(핀 번호, 주파수, 시간)`

해당 주파수와 50%의 듀티비의 사각파를 핀에 출력한다. 시간은 밀리초 단위로 설정할 수 있다.

• `for(변수=시작 값 ; 조건 ; 변수의 증분){ }`

변수의 시작 값부터 조건이 만족하는 경우 '{ }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

## 9.1.3 피에조 버저

### EX 9.1

### 피에조 버저를 이용한 소리 출력 (3/3)

- Sketch 구성**
1. 버저를 디지털 입출력핀 9번으로 설정한다.
  2. '도레미파솔라시도' 음에 대하여 피에조 버저의 진동 주파수를 설정한다.
  3. 도레미송 악보를 데이터화하여 시간에 맞춰 해당 주파수로 피에조 버저를 진동시킨다.

**실습 결과**    도레미송이 반복하여 연주된다.

# 9.1.4 피에조 버저: code-1

ex\_9\_1\_start

```

1  /*
2  예제 9.1
3  피에조 버저를 이용한 소리 출력
4  */
5
6  int buzzerPin = 9;
7  int songLength = 16;
8
9  // 노래 데이터, 공백은 쉬는 구간을 나타낸다
10 char notes[] = "cee egg dff abb ";
11 // 음의 길이, 노래 데이터와 맞춰 음의 길이를 설정한다.
12 int beats[] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
13
14 // 노래의 빠르기를 설정한다.
15 int tempo = 200;
16
17 void setup()
18 {
19   // 부저핀을 출력으로 설정한다
20   pinMode(buzzerPin, OUTPUT);
21 }

```

```

24 void loop()
25 {
26   // 부저 출력 시간에 사용할 변수 설정
27   int duration;
28
29   // 노래 길이 데이터 갯수만큼 반복한다
30   for (int i = 0; i < songLength; i++){
31     // 한 음의 시간을 계산한다.
32     duration = beats[i] * tempo;
33
34     if (notes[i] == ' '){      // 공란일 경우 음을 출력하지 않는다
35       delay(duration);
36     }
37     else{
38       // tone 명령어를 통하여 부저 핀으로 사각파를 출력한다
39       tone(buzzerPin, frequency(notes[i]), duration);
40       delay(duration);
41     }
42     // 음이 바뀔 때 잠시 쉬어준다
43     delay(tempo / 10);
44   }
45 }

```

## 9.1.4 피에조 버저: code-2

```

47 int frequency(char note){
48     // 노래 데이터를 주파수 값으로 변경하기 위한 함수
49
50     int i;
51     // 음계의 갯수 설정
52     int notes = 8;
53
54     char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
55     int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523};
56
57     // 노래 데이터를 주파수 값으로 변경하기 위해 반복하여 비교한다
58     for (i = 0; i < notes; i++){
59         if (names[i] == note){
60             // 맞는 값을 찾았을 경우 이 값을 회신한다
61             return(frequencies[i]);
62         };
63     };
64     // 앞의 for문에서 맞는 값을 못찾았을 경우 0을 회신한다
65     return(0);
66 }

```

## 9.1.5 피에조 버저: DIY

DIY

응용 문제

세 개의 스위치 입력을 받아 각 스위치가 ‘도’, ‘레’, ‘미’ 음을 내어 연주할 수 있는 스케치를 만들어 보자.

아두이노 스케치 코드를

**ARnn\_doremi.ino** 로 저장...



# 9.4

# RFID



## RFID (Radio-Frequency Identification)

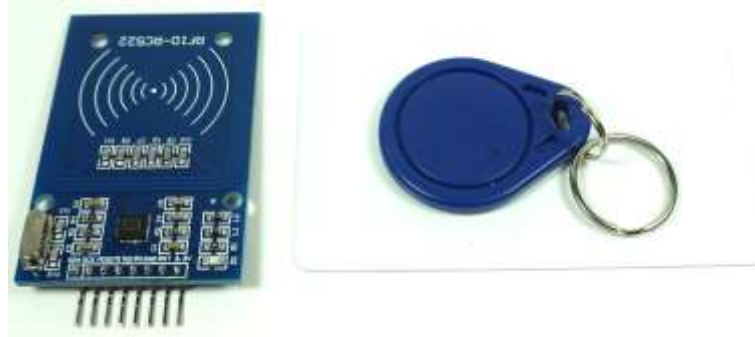


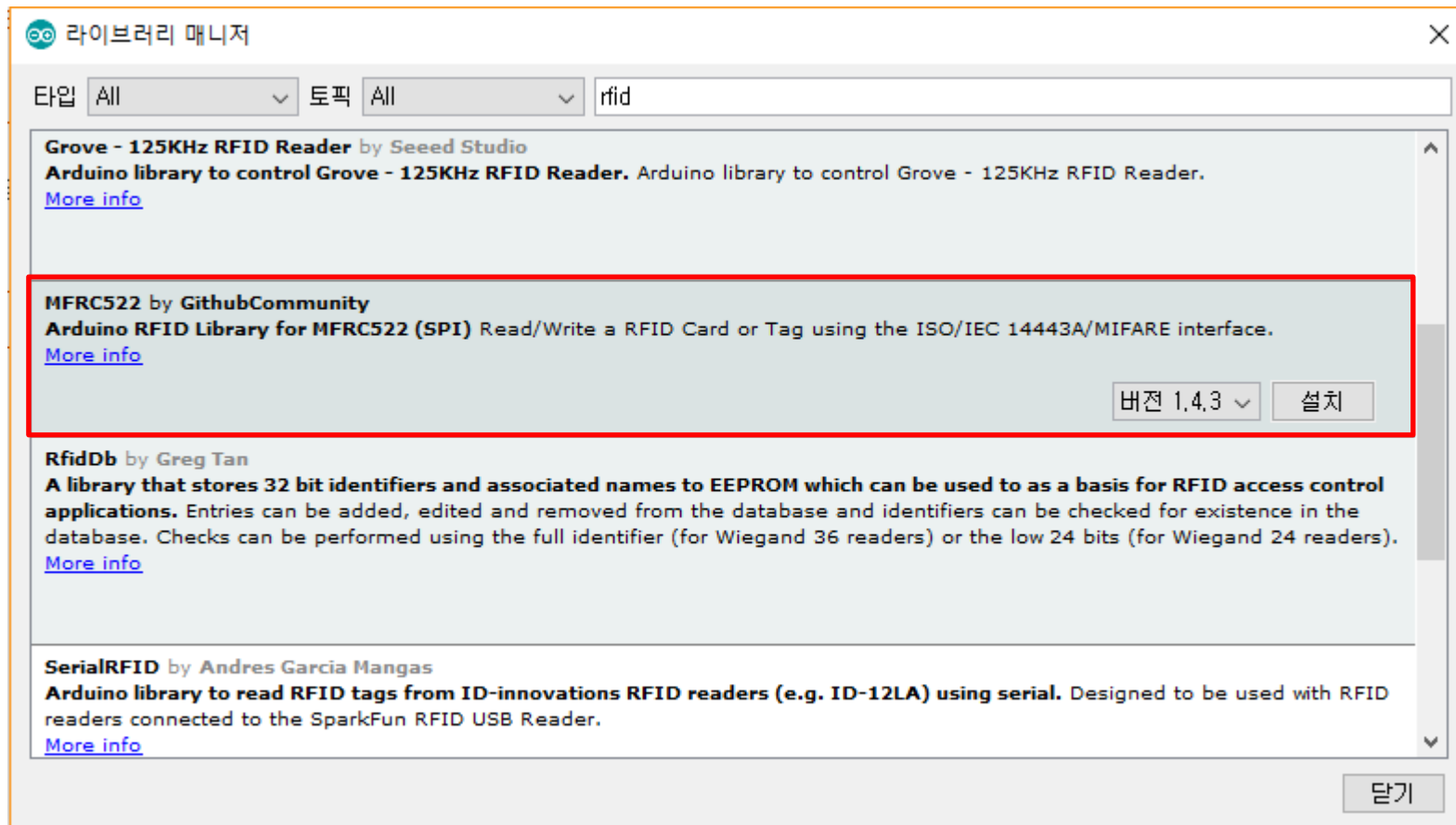
그림 9. 5 실험에 사용할 RC522 RFID 모듈과 태그

- ✓ 전파를 이용하여 원거리의 정보를 인식하는 기술
- ✓ RFID 태그(RFID tag)와 RFID 판독기(RFID reader)로 구성
- ✓ RFID 태그에는 안테나와 직접회로가 내장되어 있어 RFID 판독기에 접근하였을 때 무선통신으로 데이터 송수신
- ✓ RFID 태그의 전원 유무에 따라 수동형, 반수동형, 능동형으로 구분
- ✓ SPI 통신을 통해 MRFC522 IC를 이용한 RFID 판독기 모듈과 통신

※ 하기의 주소에서 라이브러리를 다운받아 설치 할 것  
<https://github.com/miguelbalboa/rfid>



✓ 라이브러리 관리' 메뉴에서 라이브러리 매니저를 실행 시켜 RFID 라이브러리를 설치하자.



## EX 9.4 RFID (1/3)

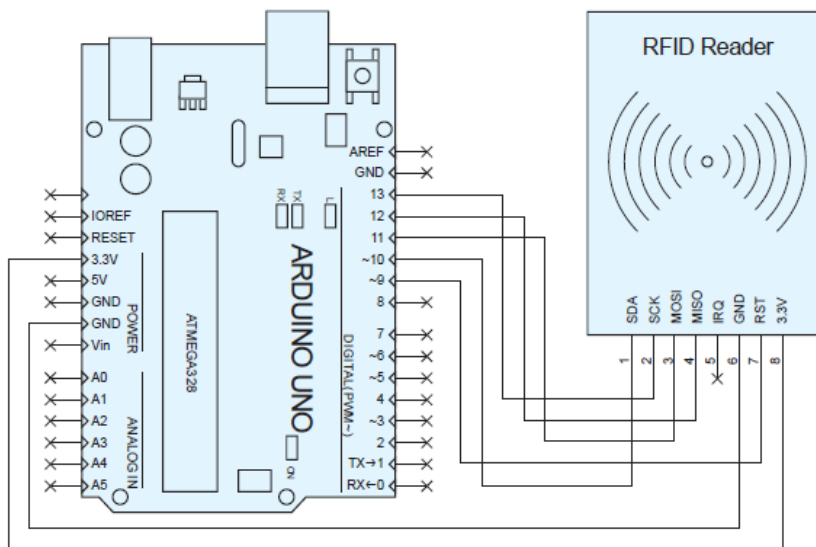
**실습목표** 1. RFID 태그를 읽는다.

2. RFID 태그의 UID와 PICC type을 판독하여 시리얼 통신으로 출력한다.

**Hardware** 1. RFID 리더와 Arduino는 SPI 통신으로 데이터를 주고 받는다.

2. RFID 리더의 3.3V, GND 핀을 Arduino의 3.3V, GND에 연결한다.

3. RFID 리더의 SDA, SCK, MOSI, MISO, RST 핀을 Arduino의 10, 13, 11, 12, 9번 핀에 연결한다.



## EX 9.4 RFID (2/3)

### Commands

- MFRC522 mfrc522(SS\_PIN, RST\_PIN)  
'mfrc522'란 이름으로 RFID 리더를 설정한다.
- mfrc522.PICC\_IsNewCardPresent  
'mfrc522'란 이름의 RFID 리더에 새로운 카드가 입력되었을 때 'TRUE' 값을 출력한다.
- mfrc522.PICC\_ReadCardSerial()  
'mfrc522'란 이름의 RFID 리더에서 카드의 내용을 읽는다.
- mfrc522.uid.uidByte[ ]  
'mfrc522'란 이름의 RFID 리더에서 읽어낸 데이터 중 uid 값에 데이터
- mfrc522.PICC\_GetType(mfrc522.uid.sak)  
'mfrc522'란 이름의 RFID 리더에서 읽어낸 데이터 중 PICC 데이터를 읽는다.
- mfrc522.PICC\_GetTypeName(picctype)  
'mfrc522'란 이름의 RFID 리더에서 읽어낸 PICC 이름 데이터.
- mfrc522.PICC\_HaltA()  
'mfrc522'란 이름의 RFID 리더를 중단한다.
- mfrc522.PCD\_StopCrypto1()  
'mfrc522'란 이름의 RFID 리더를 초기화 한다.

## EX 9.4

## RFID (3/3)

- Sketch 구성**
1. 'mfrc522'란 이름으로 RFID 리더를 설정한다.
  2. 새로운 RFID 입력이 있을 경우 데이터를 읽는다.
  3. UID와 PICC를 읽는다.
  4. UID와 PICC를 시리얼 통신으로 출력한다.
  5. RFID 리더를 초기화 하고 새로운 카드 수신을 대기한다.

**실습 결과** RFID 태그를 판독기에 접근시킬 때 UID와 PICC type이 출력된다.

```
COM11 (Arduino/Genuino Uno)

Please touch your card

Card UID: 86 17 C5 1F   PICC type: MIFARE 1KB

Card UID: D9 F6 B6 C3   PICC type: MIFARE 1KB
```

ex\_9\_4

```

1 /*
2 예제 9.4
3 RFID
4 */
5
6 // SPI 라이브러리를 불러온다.
7 #include <SPI.h>
8 // MFRC522 라이브러리를 불러온다
9 #include <MFRC522.h>
10
11 // SS 핀을 10번 핀으로 설정한다.
12 #define SS_PIN 10 //Arduino Uno, Data
13 // Reset 핀을 9번 핀으로 설정한다.
14 #define RST_PIN 9
15
16 // mfrc522란 이름의 RFID 판독기를 설정한다.
17 MFRC522 mfrc522(SS_PIN, RST_PIN);
18
19 void setup() {
20 // 시리얼 통신을 설정한다.
21 Serial.begin(9600);
22 // SPI 통신을 시작한다.
23 SPI.begin();
24 // 앞서 설정한 mfrc522란 이름의 RFID 판독기를 시작한다.
25 mfrc522.PCD_Init(); // Init MFRC522 card
26 Serial.println("Please touch your card");
27 Serial.println(" ");
28 }

```

```

30 void loop() {
31 // 새로운 카드를 기다린다.
32 if (!mfrc522.PICC_IsNewCardPresent()) return;
33 // 카드를 읽는다.
34 if (!mfrc522.PICC_ReadCardSerial()) return;
35 // 'Card UID: '메세지를 출력한다.
36 Serial.print("Card UID:");
37
38 // 판독기에 입력된 UID의 바이트수 만큼 읽어낸다.
39 // UID는 카드의 종류에 따라 최대 8바이트를 갖는다.
40 for (byte i = 0; i < mfrc522.uid.size; i++) {
41 // UID를 시리얼 통신으로 출력한다.
42 Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
43 Serial.print(mfrc522.uid.uidByte[i], HEX);
44 }
45
46 // 시리얼 통신으로 'PICC type: '메세지를 출력한다.
47 Serial.print(" PICC type: ");
48 // picType 변수에 picc type을 저장한다.
49 byte piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
50 // 시리얼 통신으로 picc type을 출력한다.
51 Serial.println(mfrc522.PICC_GetTypeName(piccType));
52 // 줄바꿈
53 Serial.println(" ");
54
55 // mfrc522 판독기를 초기화 한다
56 mfrc522.PICC_HaltA(); // Halt PICC
57 mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD
58 }

```

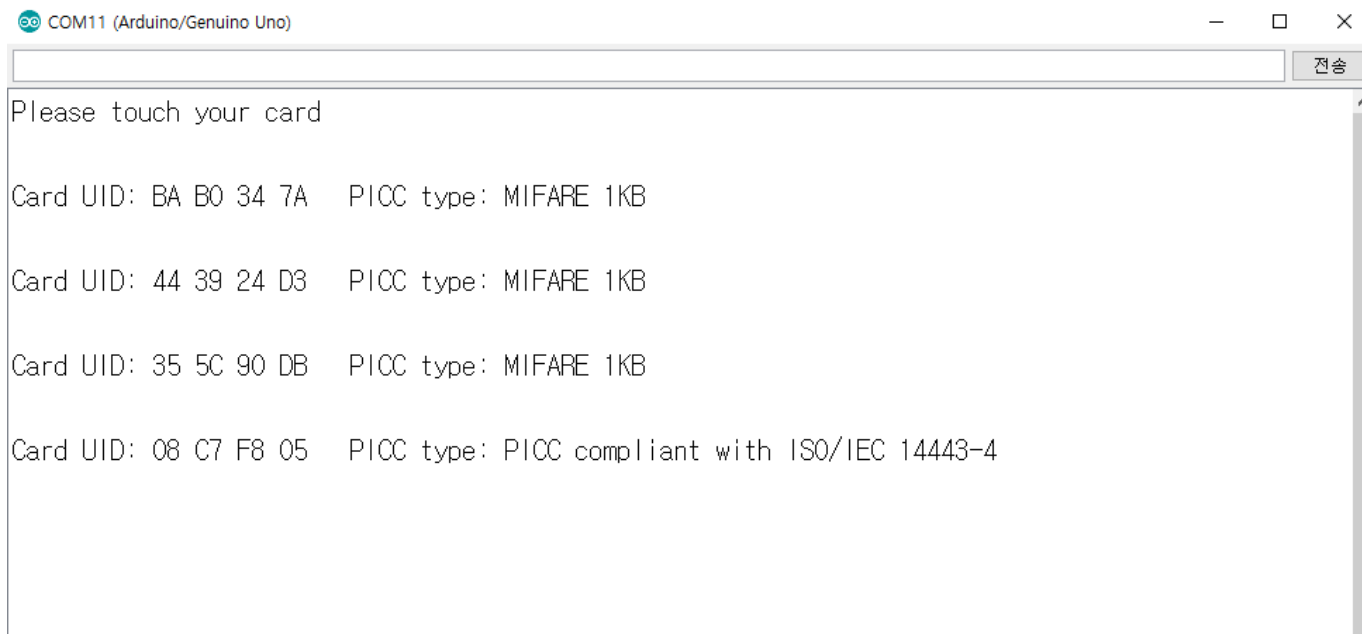
DIY

1. 갖고 있는 교통카드를 판독기에 접근시켜 읽어보자.

응용 문제

2. 폰의 NFC를 활성화 시킨 후 uid와 PICC type을 출력해보자.

직렬모니터 출력 화면을  
**ARnn\_rfid.png** 로 저장...



COM11 (Arduino/Genuino Uno)

Please touch your card

Card UID: BA B0 34 7A   PICC type: MIFARE 1KB

Card UID: 44 39 24 D3   PICC type: MIFARE 1KB

Card UID: 35 5C 90 DB   PICC type: MIFARE 1KB

Card UID: 08 C7 F8 05   PICC type: PICC compliant with ISO/IEC 14443-4



# 9.5

## 초음파 거리센서



# 9.5 초음파 거리센서 (HC-SR04)

## 초음파 거리센서 (HC-SR04)



그림 9.6 실험에 사용할 HC-SR04 초음파센서

- ✓ 약 40 kHz의 주파수의 초음파를 발사하여 물체에 반사되어 돌아오는 시간을 측정
- ✓ 외부 환경에 강한 특징을 갖고 있고, 물체의 색깔에 상관없이 사용할 수 있으며, 투명한 물체도 감지 가능하며 물이나 먼지 등이 있더라도 감지할 수 있는 장점이 있음
- ✓ 외부에 초음파 발신부가 노출되어야 함

$$L = \frac{\text{에코 펄스 폭} * 340[m/s]}{2}, L : \text{물체와의 거리}$$

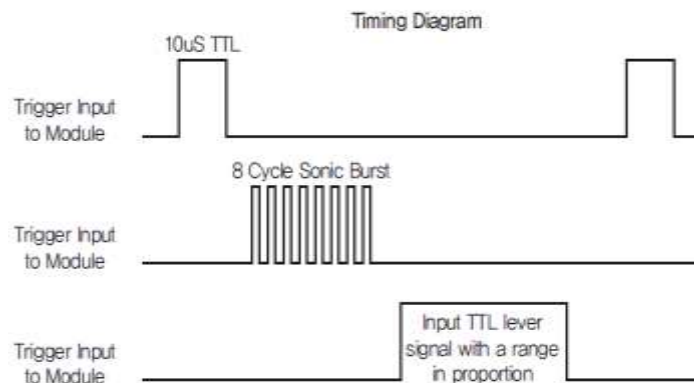


그림 9.7 HC-SR04의 타이밍 다이어그램

표 9.1 HC-SR04 사양표

동작 전압	DC 5V
소비 전류	13mA
동작 주파수	40kHz
최대 감지거리	4m
최소 감지거리	2cm
측정 각도	15°
프러거 임계 신호	10us TTL 펄스
에코 출력 신호	거리에 따른 TTL 레벨의 신호
사이즈	45*20*15mm



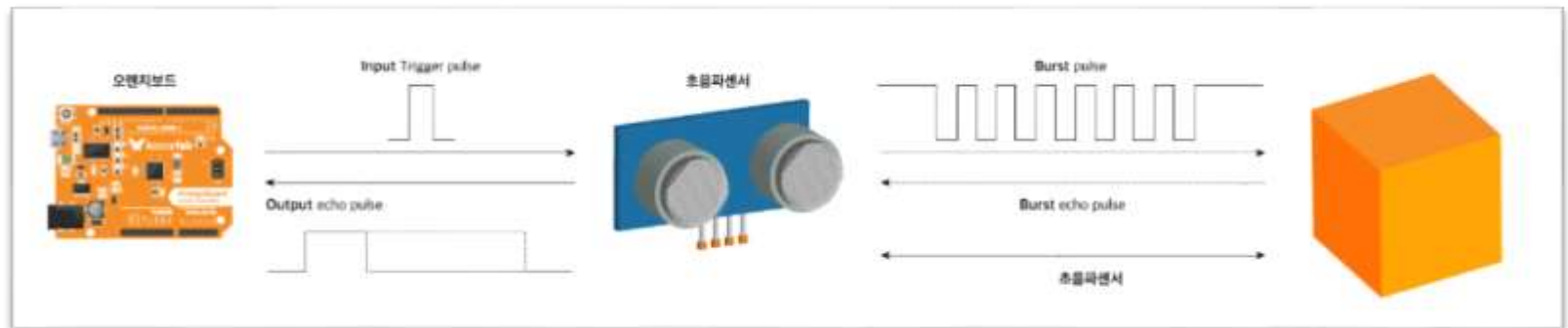
# 9.5 초음파 거리센서 (HC-SR04)

$$t = \frac{2 \times L(\text{물체와의 거리m})}{V_s(\text{음속m/s})}$$

t: 신호가 되돌아 올때까지 걸리는 시간(s)

**$t = 2 * 0.01 / 340 = 58.824\mu\text{s}$**  로, 초음파가 **1cm**를 이동하는데 걸리는 시간은 약  **$29\mu\text{s}$** 가 걸리며, 초음파가 반사된 물체와의 거리는 다음과 같이 구할 수 있습니다.

**측정 거리 (cm) = duration (왕복에 걸린 시간) / 29 / 2 (왕복)**  
**= duration (왕복에 걸린 시간) / 58**



<https://kocoafab.cc/tutorial/view/357>

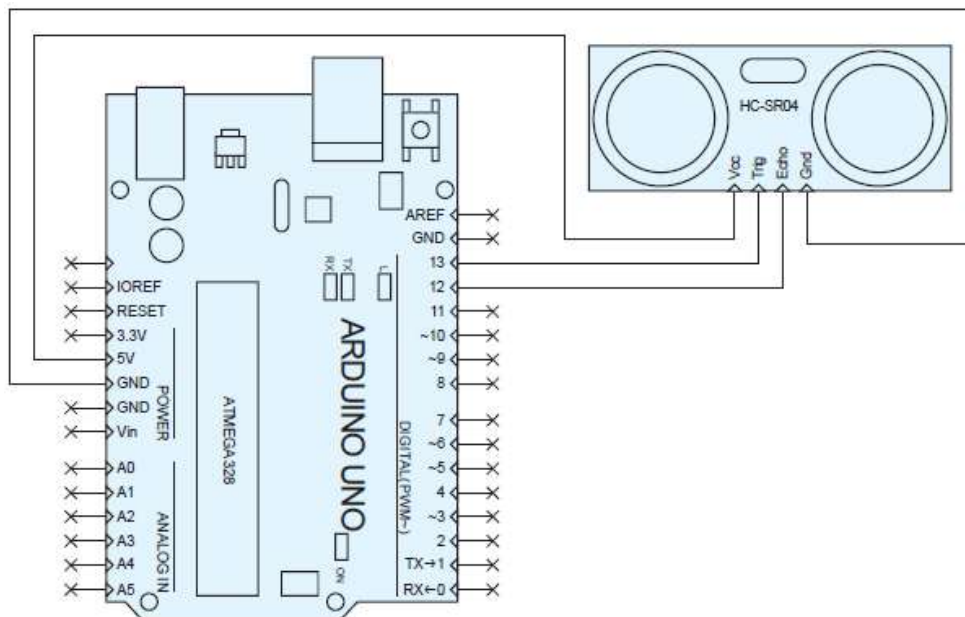
# 9.5.1 초음파 거리센서 (HC-SR04)

EX 9.5

## 초음파 거리센서를 이용한 거리 측정 (1/3)

- 실습목표**
1. 초음파 모듈 센서를 이용하여 거리를 측정한다.
  2. 측정한 거리의 변화가 있을 때 시리얼 통신을 이용하여 모니터에 출력한다.

- Hardware**
1. HC-SR04 모듈의 Vcc와 GND를 Arduino의 5V와 GND에 연결한다.
  2. HC-SR04 모듈의 Echo핀과 Trig핀을 Arduino의 12, 13번핀에 연결한다.



## 9.5.2 초음파 거리센서 (HC-SR04)

EX 9.5

### 초음파 거리센서를 이용한 거리 측정 (2/3)

#### Commands

- `pulseIn`(핀번호, 값, 초과시간)

핀번호로 입력된 펄스에 대하여 펄스의 폭을 마이크로초( $\mu\text{s}$ ) 단위로 측정한다. '핀번호'엔 펄스를 입력받을 핀의 번호, '값'엔 HIGH 펄스폭을 측정할 때는 HIGH, LOW 펄스폭을 측정할 때는 LOW를 적는다. 초과시간은 최대 측정 시간으로서 초기값은 1이다.

- `delayMicroseconds`(지연시간)

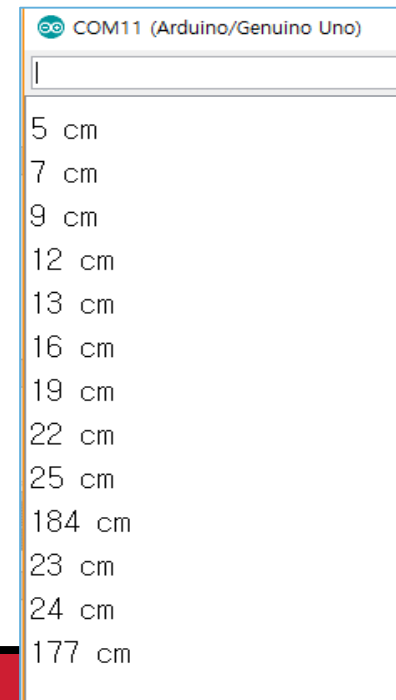
지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 마이크로초 단위로 넣는다.

## EX 9.5

### 초음파 거리센서를 이용한 거리 측정 (3/3)

- Sketch 구성**
1. 디지털 입출력핀 12와 13을 각각 에코핀과 트리거핀으로 설정한다.
  2. 폭 10  $\mu$ s의 펄스를 트리거 핀으로 출력한다.
  3. 'pulseIn' 명령어를 이용하여 에코핀으로 입력되는 HIGH 펄스의 폭을 측정한다.
  4. 펄스폭에 대하여 초음파의 속도를 물체와의 거리를 측정한다.
  5. 현재 측정한 거리와 이전에 측정한 거리가 상이할 때 그 값을 시리얼 통신으로 출력한다.

**실습 결과** 초음파 센서 앞의 물체와의 거리에 따라 측정값이 시리얼 모니터에 출력된다.  
거리의 변화가 없을 때는 출력하지 않는다.



ex\_9\_5\_start

```

1 /*
2  예제 6.7
3  초음파 거리센서를 이용한 거리 측정
4 */
5
6 // 트리거 핀과 에코 핀 번호를 설정한다.
7 const char trigPin = 13;
8 const char echoPin = 12;
9
10 // 펄스 폭과 거리 변수 설정
11 int pulseWidth;
12 int distance;
13 int distanceOld;
14
15 void setup() {
16   // 시리얼 통신 설정
17   Serial.begin(9600);
18   // 트리거 핀은 출력으로, 에코핀은 입력으로 설정
19   pinMode(trigPin, OUTPUT);
20   pinMode(echoPin, INPUT);
21   // 트리거 핀의 초기값을 LOW로 한다
22   digitalWrite(trigPin, LOW);
23 }

```

```

25 void loop() {
26   // 10us의 트리거 신호를 HC-SR04로 내보낸다.
27   digitalWrite(trigPin, HIGH);
28   delayMicroseconds(10);
29   digitalWrite(trigPin, LOW);
30
31   // Echo 펄스 폭을 측정하여 pulseWidth 변수에 저장한다.
32   pulseWidth = pulseIn(echoPin, HIGH);
33   // 거리를 계산한다.
34   distance = pulseWidth / 58;
35
36   // 감지거리인 2~200cm 범위의 거리값만 사용한다.
37   if(distance <= 200 || distance >= 2){
38     // 이전의 거리값과 비교하여 변화가 있을 경우에만
39     // 시리얼 통신으로 전송한다.
40     if(distance != distanceOld){
41       Serial.print(distance);
42       Serial.println(" cm");
43     };
44   };
45   distanceOld = distance;
46   delay(100);
47 }

```

DIY

1. 황색, 청색, 적색 LED를 Arduino에 연결하자.

응용 문제

2. 물체와의 거리가 2~30cm에서는 황색 LED, 31~60cm에서는 청색 LED, 그 이상의 거리에서는 적색 LED가 점등 되게 스케치를 작성하여라.

동작 중 사진을

**ARnn\_ultrasonic.png** 로 저장하고 제출,

어두이노 스케치 코드를

**ARnn\_ultrasonic.ino** 로 제출



# [Practice]

## ◆ [wk14]

- **Arduino : various elements**
- **Complete your project**
- **Submit file : ARnn\_Rpt11.zip**

# wk14 : Practice-11 : ARnn\_Rpt11.zip

## ◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn\_Rpt11.zip**

- 압축할 파일들

- ① **ARnn\_doremi.ino**
- ② **ARnn\_rfid.png**
- ③ **ARnn\_ultrasonic.png**
- ④ **ARnn\_ultrasonic.ino**

**Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)**

**[ 제목 : id, 이름 (수정) ]**



## ● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling
- ✓ <https://www.youtube.com> Youtube



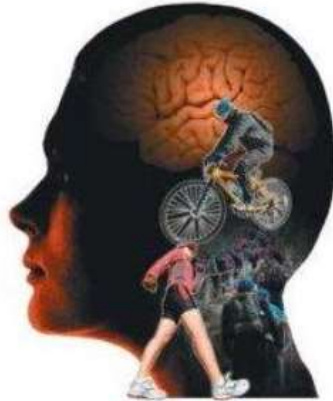
Search or jump to...

Pull requests

Issues

Marketplace

Explore



**Redwoods Yi**

Redwoods

Add a bio

GimHae, Republic of Korea

chaos21c@gmail.com

Overview

Repositories 7

Stars 2

Followers 1

Following 0

## Pinned repositories

Customize your pinned repositories

Py

Lectures on coding python from scratch to the advanced level.

Jupyter Notebook

Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.

Lec

All lectures by Redwoods in Inje University

Jupyter Notebook

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)


Arduino

171 contributions in the last year




Contribution settings

Redwoods/Arduino: Lect

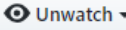
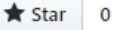

GitHub, Inc. [US] | https://github.com/Redwoods/Arduino


 Search or jump to...

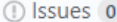
[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

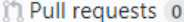
  


Redwoods / Arduino


 1  0  0

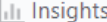
 Code

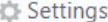
 Issues 0

 Pull requests 0

 Projects 0

 Wiki

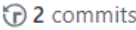
 Insights


 Settings


Lectures on learning Arduino from scratch to the advanced level in iot environment.

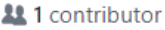
Edit

Add topics

 2 commits

 1 branch

 0 releases

 1 contributor

Branch: master


New pull request

Create new file


Upload files

Find file

Clone or download


 Redwoods 2018 start

Latest commit 38ca9e0 28 minutes ago

 ar-basic


2018 start

28 minutes ago

 ar-iot


2018 start

28 minutes ago

 README.md

Initial commit

43 minutes ago

 README.md

Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.





아두이노

상급 키트

High-Level Kit for Arduino

LEVEL 03



당신의 상상을  
실현해줄  
아두이노스토리

[http://arduinostory.com/goods/goods\\_view.php?goodsNo=1000000306](http://arduinostory.com/goods/goods_view.php?goodsNo=1000000306)



## 상급키트 구성품

<b>1</b> 1EA  <b>아두이노 우노 R3 DIP</b> 아두이노 우노 R3 (DIP) 호환보드 기본 메인보드입니다.	<b>2</b> 1EA  <b>9V 배터리 홀더</b> 9V 배터리를 연결하여 아두이노에 외부전원을 공급할 수 있습니다.	<b>3</b> 1EA  <b>7세그먼트 4채널</b> 7세그먼트가 4개 연결된 형태의 부품입니다. 총 12개의 핀을 사용합니다.	<b>4</b> 1EA  <b>7세그먼트 1채널</b> 공통 음극 7세그먼트 시계나 점수 등의 숫자를 표현 할 때 많이 사용됩니다.
<b>5</b> 1EA  <b>74HC595N</b> 기본 메인보드입니다. 74HC595N LED, 드로메트릭스, NFD 제어 IC 입니다.	<b>6</b> 1EA  <b>65핀 점퍼 와이어</b> 브레드보드에 연결할 때 사용하는 65핀 점퍼와이어 입니다.	<b>7</b> 1EA  <b>무지개 점퍼선 F-M 20cm</b> M타입과 F타입이 양쪽으로 달린 무지개 점퍼선입니다.	<b>8</b> 1EA  <b>투명 부품 케이스 대,소</b> 키트 구성품을 담을 수 있는 투명 부품 케이스입니다.
<b>9</b> 1EA  <b>가변저항10K</b> 물리변 저항값이 바뀝니다. (0~10KΩ)	<b>10</b> 1EA  <b>1602 I2C LCD</b> 아두이노 16x2 I2C LCD 모듈입니다. LCD입니다.	<b>11</b> 1EA  <b>저항</b> 100, 220, 330, 1K, 2K, 4.7K, 10K, 47K, 100K	<b>12</b> 1EA  <b>브레드 보드 830홀</b> 브레드 보드 830홀(봉무형) 센서 테스트나, 회로 프로토타입을 작성할 때 사용됩니다.

<b>13</b> 1EA  <b>수동부저</b> 아두이노의 tone함수를 통해 소리를 내는 부저입니다.	<b>14</b> 6EA  <b>택트스위치 (12x12x7)</b> 스위치를 누르고 있을 경우만 ON됩니다.	<b>15</b> 3EA  <b>택트스위치 캡 (피랑,노랑,초록,빨강,하양)</b> 택트스위치를 사용할 때 스위치간의 구분을 할 수 있습니다.	<b>16</b> 3EA  <b>조도센서</b> 빛을 감지하거나 빛의 밝기를 아날로그로 출력해주는 CDS 센서입니다.
<b>17</b> 5EA  <b>LED 5mm (빨강,노랑,초록,하양,파랑)</b> 기본으로 사용되는 LED입니다. 동작전압 : 2.2~2.4V 사용전류 : 20mA 미만	<b>18</b> 1EA  <b>헤더핀 1x40/2.54mm</b> 핀 간격은 2.54mm이며 헤더핀의 길이는 약 1.15cm입니다.	<b>19</b> 1EA  <b>USB케이블 50cm</b> PC와 아두이노 우노 보드를 연결하여 프로그램을 다운로드 할 때 사용합니다.	<b>20</b> 1EA  <b>저항값 카드</b> 저항값을 쉽게 확인 할 수 있는 카드입니다. 사이즈 : 60mm x 50mm
<b>21</b> 1EA  <b>능동부저</b> Signal 단자가 HIGH 일 때 약 2.5kHz의 음이 발생합니다.	<b>22</b> 1EA  <b>5V 1채널 릴레이 모듈</b> 아두이노의 디지털 핀과 모듈 하단의 IN 핀들을 연결해 릴레이를 제어할 수 있는 모듈입니다.	<b>23</b> 1EA  <b>8x8 도트 매트릭스 모듈</b> LED로 다양한 연출을 할 수 있습니다.	<b>24</b> 1EA  <b>4x4 16 키패드 모듈</b> 16개의 버튼을 사용할 수 있습니다.

# 아두이노 키트(Kit) : Part-2

<div>25</div> <div>1EA</div> <div></div> <div>무선 리모콘 키트</div> <div>핵파선을 사용해서 리모콘 기능을 구현할 수 있습니다.</div>	<div>26</div> <div>2EA</div> <div></div> <div>가열기 센서 스위치</div> <div>센서의 가열기에 따라 스위치 역할을 합니다.</div>	<div>27</div> <div>1EA</div> <div></div> <div>or</div> <div>사운드 센서 모듈</div> <div>아두이노와 호환되는 사운드센서 모듈입니다.</div>	<div>28</div> <div>1EA</div> <div></div> <div>불꽃 센서</div> <div>근거리 화재, 불꽃을 감지하는 센서입니다.</div>	<div>37</div> <div>1EA</div> <div></div> <div>DC 5V 스텝 모터</div> <div>28BYJ 28BYJ48 스텝 모터 중 저렴한 편에 속하는 모델입니다. 5개의 핀을 사용합니다.</div>	<div>38</div> <div>1EA</div> <div></div> <div>DS1302 RTC 모듈</div> <div>아두이노 등 마이크로컨트롤러에서 사용이 가능합니다.</div>	<div>39</div> <div>1EA</div> <div></div> <div>아두이노 우노 프로토 쉼드</div> <div>UNO 보드에서 회로를 간단히 짜기 위해 보드 위에 얹어 사용하는 쉼드입니다.</div>	<div>40</div> <div>1EA</div> <div></div> <div>3축 가속도 센서 모듈</div> <div>가속도를 측정할수 있는 센서입니다.</div>
<div>29</div> <div>1EA</div> <div></div> <div>모터 드라이버 모듈</div> <div>ULN2003 스텝 모터 드라이버 모듈 5V ~ 12V를 사용할가능합니다.</div>	<div>30</div> <div>1EA</div> <div></div> <div>LM35 온도 센서</div> <div>온도를 아날로그 값으로 출력합니다.</div>	<div>31</div> <div>1EA</div> <div></div> <div>수위 센서 모듈</div> <div>센서가 액체에 잠긴 정도를 아날로그 값으로 출력합니다.</div>	<div>32</div> <div>1EA</div> <div></div> <div>SG90 서보모터</div> <div>Vcc, GND, 신호선, 총 3개의 핀이 있습니다. 로봇팔이나 자동차, 비행기 조종에 사용됩니다.</div>	<div>41</div> <div>1EA</div> <div></div> <div>5V DC모터</div> <div>5V DC모터</div>	<div>42</div> <div>1EA</div> <div></div> <div>인체 감지 센서 모듈</div> <div>핵파선을 이용해 움직임 감지하는 센서입니다. 오선이 감지되면 HIGH 신호를 출력합니다.</div>	<div>43</div> <div>5EA</div> <div></div> <div>다이오드 1N4001</div> <div>다이오드 1N4001</div>	<div>44</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (22pF)</div> <div>세라믹 캐패시터 (22pF)</div>
<div>33</div> <div>1EA</div> <div></div> <div>초음파 거리 센서 모듈</div> <div>5V를 사용하여 만작 거리는 2cm에서 500cm입니다.</div>	<div>34</div> <div>1EA</div> <div></div> <div>조이스틱 모듈</div> <div>기본적으로 조이스틱 모듈은 두개의 가변저항이 서로 수직으로 회전하는 형태로 되어있습니다.</div>	<div>35</div> <div>1EA</div> <div></div> <div>온습도 센서 모듈</div> <div>아두이노 온습도 센서중 가장 대중적으로 사용되는 DHT11 디지털 센서입니다.</div>	<div>36</div> <div>1EA</div> <div></div> <div>RGB LED 모듈</div> <div>RGB LED 모듈로 RGB LED 세개를 하나로 모은 상품입니다.</div>	<div>45</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (1uF)</div> <div>세라믹 캐패시터 (1uF)</div>	<div>46</div> <div>5EA</div> <div></div> <div>트랜지스터 2N2222</div> <div>트랜지스터 2N2222</div>	<div>47</div> <div>5EA</div> <div></div> <div>트랜지스터 BC547</div> <div>트랜지스터 BC547</div>	<div>48</div> <div>5EA</div> <div></div> <div>트랜지스터 BC557</div> <div>트랜지스터 BC557</div>
<div>49</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 10uF)</div> <div>전해 캐패시터 (50V 10uF)</div>	<div>50</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 100uF)</div> <div>전해 캐패시터 (50V 100uF)</div>	<div></div>					

# [참고 : 저항 값 읽기]



Color	First	Second	Third	Multiplier	Tolerance
Black	0	0	0	x1	
Brown	1	1	1	x10	1%
Red	2	2	2	x100	2%
Orange	3	3	3	x1000	
Yellow	4	4	4	x10 000	
Green	5	5	5	x100 000	0,50%
Blue	6	6	6	x1 000 000	0,25%
Violette	7	7	7	x10 000 000	0,10%
Gray	8	8	8		
White	9	9	9		
Silver				x0,01	10%
Gold				x0,1	5%

