

DU-도전학기 결과보고서

성명		학번	
단과대학		학과(전공)	
도전학기 과제명	(한글) 원격 소형 금속탐지기 (영문) Remote Small Metal Detector		
지도교수 의견	<p>달성하고자 하는 목표를 모두 달성한 것으로 판단됨.</p> <ul style="list-style-type: none"> - 회로 분석 및 PCB 설계 완료 - 외형 제작을 위한 3D 프린팅 활용 - 스스로 주제를 정하고, 문제 해결 및 실행력이 우수하다고 판단됨. 		

1. 도전 과제의 목표

- 기존의 금속탐지기의 원리를 탐구하고, 응용하여 현재까지 없는 금속종류판별을 추가하고, 금속탐지기의 사용성을 증가 및 서버를 구축하여 원격으로 제어하도록 한다.

2. 도전 과제 내용

- 붙임1

3. 도전 과제의 성과

4. 자기평가

- 중간평가 이후에 LC발진회로 중 Colpitts Oscillator을 사용하였습니다. 그리고 Eagle 프로그램을 사용하여 PCB 기판을 스스로 제작하였고, Rhino을 이용하여 3D프린팅을 직접 설계하고, 제작하였습니다. 또한 라즈베리파이를 사용하여 인공지능 스피커를 제작하고, 이용하였습니다. 여러 부분에서 많이 부족한 실력이었지만, 다양한 경험을 할 수 있어서 좋았습니다.

5. 최종 결과물(반드시 사진 첨부)



1. Atmega128

㉔ Atmega128

> Atmega128이란 쉽게 말해서 기본적으로 CPU라고 생각하면 된다. CPU와 여러가지 I/O포트들을 합쳐 놓은 것이다.

메모리	128KB Flash, 4KB SRAM, 4KB EEPROM
입출력(I/O)	(A~F port) + G port = (8*6) + 5 = 53 I/O pin
디버깅	JTAG 인터페이스 지원
타이머/카운터	8비트 타이머/카운터 2개, 16비트 타이머/카운터 2개, 총4개
ADC	8채널 10비트 ADC 내장
통신	USART(시리얼통신) 2채널, TWI(I2C) 1채널, SPI 1채널

- 동작 전압

- 2.7V~5.5V : Atmega128L
- 4.5V~5.5V : Atmega128
- 2.7V~5.5V : Atmega128A

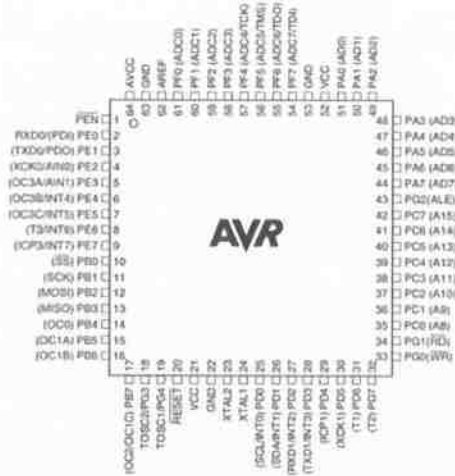
- 동작 속도

- 0~8MHz : Atmega128L
- 0~16MHz : Atmega128
- 0~16MHz : Atmega128A

> 하바드 구조와 반대되는 구조는 폰-노이만 구조이다 <

> 하버드 구조 = 메모리와 버스가 프로그램과 데이터로 분리, 하나의 명령이 처리되는 동안 다음 명령은 프로그램 메모리로부터 프리패치, 모든 클럭 사이클에 명령이 처리되는 것이 가능, 하바드 구조와 반대되는 구조는 폰-노이만 구조이다

CISC	RISC
하드웨어 ↑	소프트웨어 ↑
많고 복잡한 명령어 구조	적고 단순한 명령어 구조
Memory to Memory : 적은 레지스터	Register to Register : 많은 레지스터
느린 처리속도	빠른 처리속도
높은 호환성	낮은 호환성



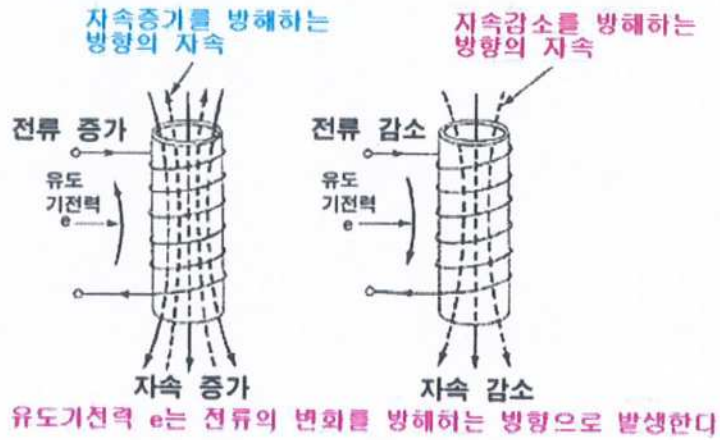
(Atmega128 내부 포트 및 구조)

포트	기능
포트A	<ul style="list-style-type: none"> > 외부 메모리 데이터 버스 / (Data Bus) > 외부 메모리 하위 어드레스 바이트 버스 / (Address Low Byte)
포트B	<ul style="list-style-type: none"> > 타이머/카운터 0(OC0) > 타이머/카운터 1(OC1A,OC1B,OC2/OC1C) > SPI(MISO,MOSI,SCK,SS)
포트C	<ul style="list-style-type: none"> > 외부 메모리 상위 어드레스 바이트 버스 / (Address High Byte)
포트D	<ul style="list-style-type: none"> > 타이머/카운터 1(T1,ICP1) > 타이머/카운터 2(T2) > 외부 인터럽트(INT0,INT1,INT2,INT3) > USART1(XCK1, TXD1, RXD1) > TWI 직렬통신(SDA,SCL)
포트E	<ul style="list-style-type: none"> > 타이머/카운터3(T3,ICP3,OC3A,OC3B,OC3C) > 외부 인터럽트(INT4, INT5, INT6, INT7) > USART0(XCK0, TXD0, RXD0) > 아날로그 비교기(AIN0[p],AIN1[n]) > SPI(PD0,PDI)-ISP
포트F	<ul style="list-style-type: none"> > ADC 입력 채널(ADC0~ADC7) > JTAG(TDI, TDO, TMS,TCK)
포트G	<ul style="list-style-type: none"> > 타이머/카운터 0(TOSC1, TOSC2)-RTC > 외부메모리 인터페이스(ALE,/RD,/WR)

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

㉔ 사용기능 이용하여 금속종류 탐지 기능

㉔ inductance[감응계수(H)]



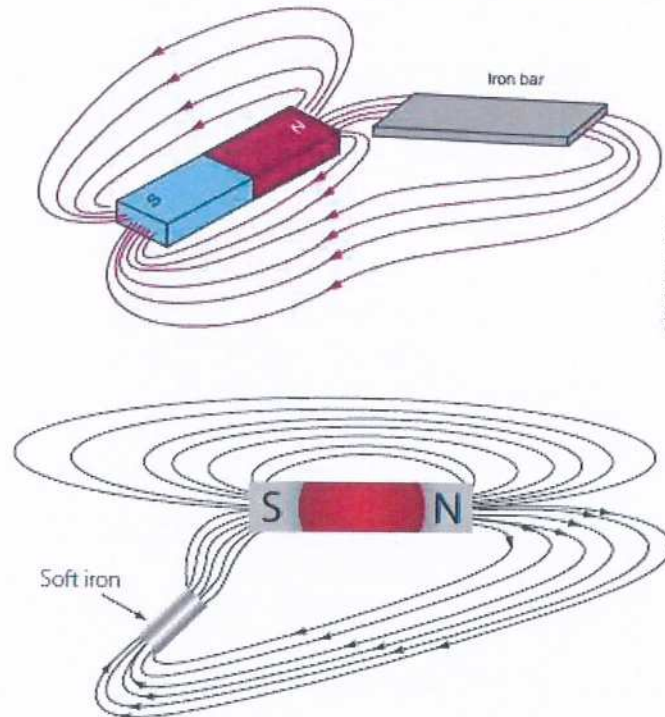
자체유도와 유도기전력

어떤 회로를 뚫고 나가는 자기력선속과 그 자기력선속을 생기게 하는 전류와의 비, 전자기 유도의 크기를 나타내는 자기력선속이 그 회로 자체의 전류에 의하여 생기는 경우를 자기 인덕턴스, 다른 회로의 전류에 의하여 생기는 경우를 상호 인덕턴스라고 하는데, 대개는 전자의 경우에 씀. 또는, 인덕턴스를 가진 회로 소자를 가리키는 경우도 있음. MKSA 단위는 헨리(henry).

> 송신, 수신 코일의 인덕턴스 변화를 Atmega128 Analog Digital Converter을 이용하여 판단한다.

■ FIGURE 18-12

Placing an iron bar in a magnetic field extends the magnetic field and magnetizes the iron bar.



```

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL
#include <stdio.h>
static int putchar0(char c, FILE *stream);
static FILE mystdout = FDEV_SETUP_STREAM(putchar0, NULL, _FDEV_SETUP_WRITE);
int putchar0(char c, FILE *stream)
{
    if (c == '\n')
        putchar0('\r', stream);
    while(!(UCSR0A & 0x20));
    UDR0 = c;
    return 0;
}
char getchar0()
{
    while (!(UCSR0A & 0x80));
    return(UDR0);
}
void init_uart()
{
    UBRROH = 0;
    UBRROL = 8;

    UCSROB = 0x18;
    UCSROC = 0x06;
}
int main()
{
    init_uart();
    stdout = &mystdout;
    ADMUX=0x80; ADCSRA=0xA7;
    while(1)
    {
        ADCSRA |=0x40;
        int a=ADCL+(ADCH<<8);
        printf("%d\n",a);
    }
}

```

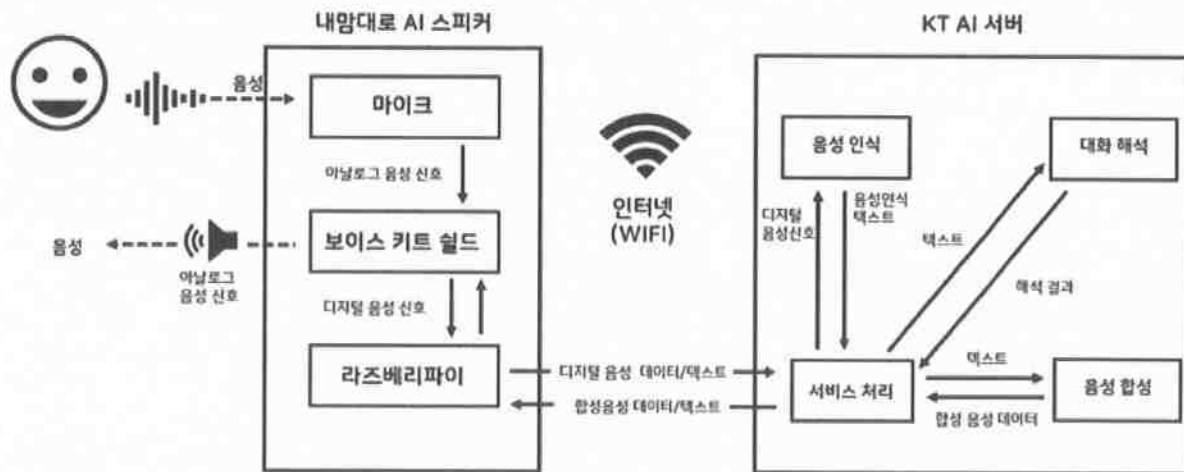
< ATMEGA 128 소스 >

2. 라즈베리파이의 인공지능 스피커 적용

㉠ 라즈베리파이

> 라즈베리파이는 컴퓨터와 마찬가지로 RAM, CPU가 장착되어있고, USB포트와 HDMI입력을 하여 사용할 수 있습니다. 또한 일반적으로 라즈베리파이는 리눅스 기반의 라즈비안 OS를 SD카드에 저장하여 운영체제를 구동할 수 있습니다. 한국에서는 운영체제로 주로 Windows를 많이 사용합니다.

㉡ 인공지능 서비스 이용



< 인공지능 서비스 절차 >

> 라즈베리파이를 사용하여 인공지능 서비스를 이용한다. Wifi를 통하여 사용자가 요구하는 정보를 KT AI서버를 통하여 알려주면서 라즈베리파이의 스피커를 통하여 확인 할 수 있도록 하였다.

㉔ JAVA 해석

```

const record=require('node-record-lpcm16');
const aikit=require('./aimakerskitutil'); // GRPC호출 기능을 제공하는 모듈(서버와 라즈베리파이
                                        사이의 통신에 사용되는 프로토콜.)

const client_id='';
const client_key='';
const client_secret='';
const json_path='/home/pi/Downloads/clientKey.json';
const cert_path='../data/ca-bundle.pem';
const proto_path='../data/gigagenieRPC.proto';
function initMic(){ // 마이크 캡처를 위한 코드
    return record.start({
        sampleRateHertz: 16000,
        threshold: 0,
        verbose: false,
        recordProgram: 'arecord',
        silence: '10.0',
    })
};
let writeFlag=0;
//aikit.initialize(client_id,client_key,client_secret,cert_path,proto_path); 초기화를 진행한다.
aikit.initializeJson(json_path,cert_path,proto_path);
const ktstt=aikit.getVoice2Text(); // 음성 데이터 해석을 위해 라이브러리를 호출한다.
ktstt.on('error',(error)=>{
    console.log('Error:'+error);
});
ktstt.on('data',(data)=>{ // Data수신, 음성으로 인식된 결과를 전달한다.
    console.log('data:'+JSON.stringify(data));
});
ktstt.on('end',()=>{ // 프로그램의 무한 루프를 탈출하기 위해 종료구문을 추가한다.
    console.log('pcm end');
    record.stop();
    writeFlag=0;
    ktstt.end();
});
ktstt.write({reqOptions:{mode:0,lang:0}}); // 음성인식 설정값을 KT AI서버로 전달한다.
writeFlag=1;
const mic=initMic();
mic.on('data',(data)=>{ // 서버의 결과를 마이크로 출력한다.
    if(writeFlag===1) ktstt.write({audioContent:data});
});
console.log('say something');

```



```

const aikit=require('./aimakerskitutil');
const client_id='';
const client_key='';
const client_secret='';
const json_path='/home/pi/Downloads/clientKey.json';
const cert_path='../data/ca-bundle.pem';
const proto_path='../data/gigagenieRPC.proto';
//aikit.initialize(client_id,client_key,client_secret,cert_path,proto_path);
aikit.initializeJson(json_path,cert_path,proto_path);
aikit.getText2VoiceUrl({lang:0,text:'안녕하세요. 만나서 반갑습니다.'}),(err,msg)=>{
    console.log('err:'+JSON.stringify(err)+' msg:'+JSON.stringify(msg));
}}

```

```

const Speaker=require('speaker'); // 합성된 음성을 출력하기 위해서 스피커를 호출한다.
const pcmplay=new Speaker({
    channels:1,
    bitDepth:16,
    sampleRate:16000
});
const aikit=require('./aimakerskitutil');
const client_id='';
const client_key='';
const client_secret='';
const json_path='/home/pi/Downloads/clientKey.json';
const cert_path='../data/ca-bundle.pem';
const proto_path='../data/gigagenieRPC.proto';
//aikit.initialize(client_id,client_key,client_secret,cert_path,proto_path);
aikit.initializeJson(json_path,cert_path,proto_path);
ktts=aikit.getText2VoiceStream({text:'안녕하세요. 반갑습니다.',lang:0,mode:0});
ktts.on('error',(error)=>{
    console.log('Error:'+error);
});
ktts.on('data',(data)=>{ // 합성된 음성이 전송 중에 출력할 문구를 준비한다.
    if(data.streamingResponse==='resOptions' && data.resOptions.resultCd===200)
console.log('Stream send. format:'+data.resOptions.format);
    if(data.streamingResponse==='audioContent') {
        pcmplay.write(data.audioContent);
    } else console.log('msg received:'+JSON.stringify(data));
});
ktts.on('end',()=>{ // 합성된 음성의 전송이 완료되면 출력한다.
    console.log('pcm end');
});
function finish(){
    console.log('tts played');
};
setTimeout(finish,5000);

```

```

const record=require('node-record-lpcm16');
const aikit=require('./aimakerskitutil');
//node version check
const nodeVersion=process.version.split('.')[0];
let ktkws=null;
if(nodeVersion==='v6') ktkws=require('./ktkws');
else if(nodeVersion==='v8') ktkws=require('./ktkws_v8');
//for playing pcm sound
const Speaker=require('speaker');
const fs=require('fs');
const soundBuffer=fs.readFileSync('../data/sample_sound.wav');
const pcmplay=new Speaker({
  channels:1,
  bitDepth:16,
  sampleRate:16000
});
const client_id='';
const client_key='';
const client_secret='';
const json_path='/home/pi/Downloads/clientKey.json';
const cert_path='../data/ca-bundle.pem';
const proto_path='../data/gigagenieRPC.proto';
const kwstext=['기가지니','지니야','친구야','자기야'];
const kwsflag=parseInt(process.argv[2]);
function initMic(){
  return record.start({
    sampleRateHertz: 16000,
    threshold: 0,
    verbose: false,
    recordProgram: 'arecord',
  })
};
ktkws.initialize('../data/kwsmodel.pack');
ktkws.startKws(kwsflag);
let mic=initMic();
//aikit.initialize(client_id,client_key,client_secret,cert_path,proto_path);
aikit.initializeJson(json_path,cert_path,proto_path);
let mode=0;//0:kws, 1:stt // Mode의 값을 설정하여 음성인식 데이터의 전달 방법을 결정한다.
// (Mode = 0이면 키워드 인식 모듈, Mode = 1이면 음성인식 객체)
let ktstt=null;
mic.on('data',(data)=>{
  if(mode===0){ // 마이크로 들어온 음성 데이터를 키워드 인식 모듈로 전달한다.
    result=ktkws.pushBuffer(data);
    if(result===1) { } // 인식인지 알림을 설정한다.
    console.log("KWS Detected");
    pcmplay.write(soundBuffer);
    setTimeout(startStt,1000); // 1초 Delay를 설정한다.
  }
  } else {
    ktstt.write({audioContent:data});
  }
});
console.log('say :'+kwstext[kwsflag]);
function startStt(){
  ktstt=aikit.getVoice2Text();
  ktstt.on('error',(error)=>{
    console.log('Error:'+error);
  });
  ktstt.on('data',(data)=>{
    console.log('stt result:'+JSON.stringify(data));
    if(data.resultCd!==200) mode=0;
  });
  ktstt.on('end',()=>{
    console.log('stt text stream end');
    mode=0; // 음성인식이 종료되면 Mode를 0으로 변경하여 종료한다.
  });
  ktstt.write({reqOptions:{mode:0,lang:0}});
  mode=1; // 음성인식의 시작되면 Mode를 1로 변경하여, 마이크 입력을 시작한다.
};

```

© 프로그램 실행

```

파일(F) 편집(E) 탭(T) 도움말(H)
pi@raspberrypi ~ $ cd ai-makers-kit/nodejs
pi@raspberrypi ~/ai-makers-kit/nodejs $ node ex8_kwsstdss js 0
say 기가지니
KWS Detected
Msg {"resultCd": 200, "uword": "금속이 뭐야", "sysAct": "11", "nAct": "1", "sPattern": "Describe", "action": [{"msg": "[CDATA[금속은 열이나 전기를 잘 전도하고, 펴지고 늘어나는 성질이 풍부하며, 특수한 광택을 가진 물질을 통틀어 일컫는 말입니다]]"}, {"aFeel": "happiness", "submsg": "N/A", "actGroup": "0", "actType": "99", "srchWord": "금속", "serviceId": "N/A"}]}
actType 99 msg: <![CDATA[금속은 열이나 전기를 잘 전도하고, 펴지고 늘어나는 성질이 풍부하며, 특수한 광택을 가진 물질을 통틀어 일컫는 말입니다]]>
Stream send format wav
msg received {"streamingResponse": "resOptions", "resOptions": {"resultCd": 200, "format": "wav"}, "audioContent": {"type": "Buffer", "data": []}}
pcm end
KWS Detected
Msg {"resultCd": 200, "uword": "지금 몇시야", "sysAct": "12", "nAct": "1", "sPattern": "Describe", "action": [{"msg": "지금 오후 10시 9분입니다", "aFeel": "Neutral", "submsg": "N/A", "actGroup": "0", "actType": "631", "srchWord": "오후 10시 9분", "serviceId": "N/A"}]}
actType 631 msg: 지금 오후 10시 9분입니다
Stream send format wav
msg received {"streamingResponse": "resOptions", "resOptions": {"resultCd": 200, "format": "wav"}, "audioContent": {"type": "Buffer", "data": []}}
pcm end
KWS Detected
Msg {"resultCd": 200, "uword": "음식 추천해줘", "sysAct": "11", "nAct": "1", "sPattern": "Yn_question", "action": [{"msg": "[CDATA[양념치킨은 어떠세요?]]"}, {"aFeel": "Neutral", "submsg": "N/A", "actGroup": "0", "actType": "99", "srchWord": "", "serviceId": "N/A"}]}
actType 99 msg: <![CDATA[양념치킨은 어떠세요?]]>
Stream send format wav
msg received {"streamingResponse": "resOptions", "resOptions": {"resultCd": 200, "format": "wav"}, "audioContent": {"type": "Buffer", "data": []}}
pcm end

```

> 질문 : 금속이 뭐야?
 > 답변 : 금속은 열이나 전기를 잘 전도하고, 펴지고 늘어나는 성질이 풍부하며, 특수한 광택을 가진 물질을 통틀어 일컫는 말입니다.

> 질문 : 지금 몇시야?
 > 답변 : 지금 오후 10시 9분입니다.

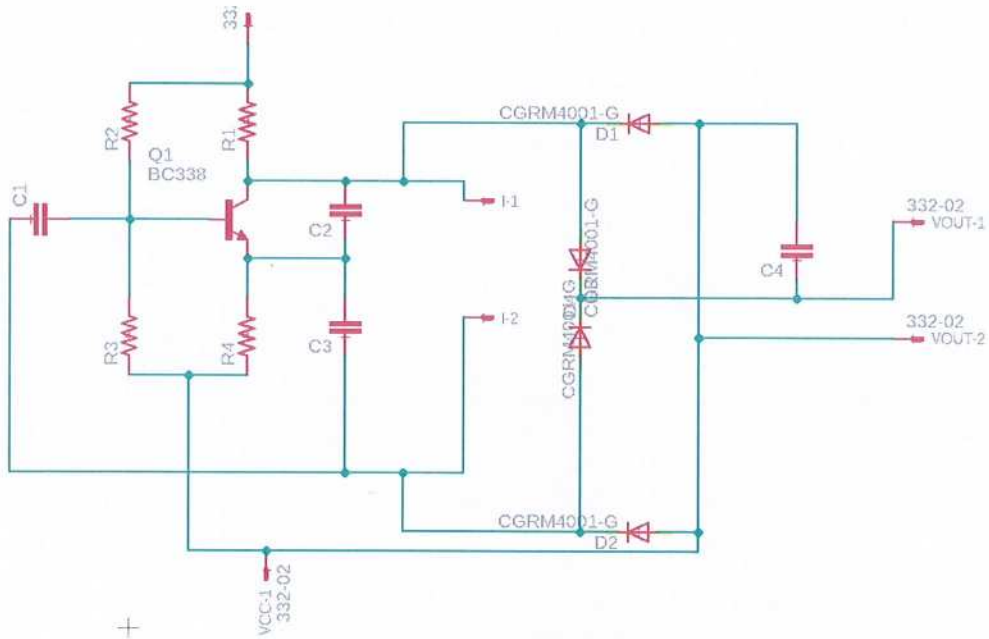
> 질문 : 음식 추천해줘.
 > 답변 : 양념치킨은 어떠세요?

3. PCB 제작

㉞ PCB

> PCB란 인쇄회로기판 (Printed Circuit Boards의 약자)이다. 에폭시 수지와 같은 절연판의 표면에 구리로 코팅한 후, 필요한 부분을 남기고 나머지 부분을 부식시켜 회로를 구성한다. 이러한 과정은 사람이 수작업으로 하는 것이 아닌 다음과 같은 공정을 통해서 이루어지기 때문에, 우리가 할 일은 부품의 배치 및 핀들간의 연결, 그리고 외관 디자인을 작업하게 된다. 디자인을 이글캐드를 통해서 작업하여, 거버파일을 생성하여 출력한다.

㉟ 회로구성



Properties

Part

Name: R2

Position: 0, 2,6

Gate: G\$1 (R-US)

Angle: 90

Mirror

Device: R-US_R3216 (R-US_) (Version 17)

Footprint: R3216 (Version 1)

3D Package: R3216 (Version 2)

Library: rcl (Version 5)

Value:

Attributes: Name^ Value
SPICEPREFIX R

Properties

Part

Name: C3

Position: 1,1, 1,5

Gate: G\$1 (C-EU-1)

Angle: 180

Mirror

Device: C-EUC1608 (C-EU) (Version 2)

Footprint: C1608 (Version 1)

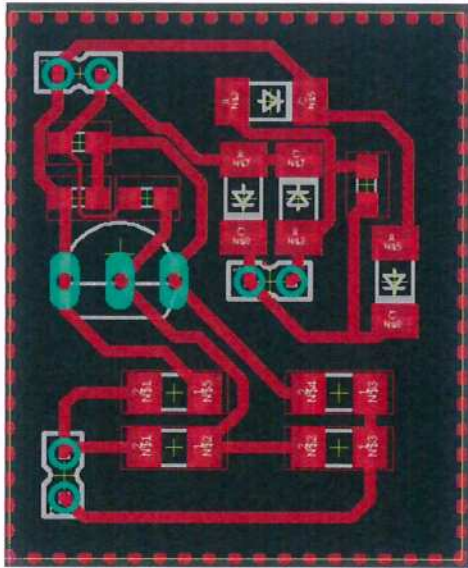
3D Package: C1608 (Version 2)

Library: resistor (Version 2)

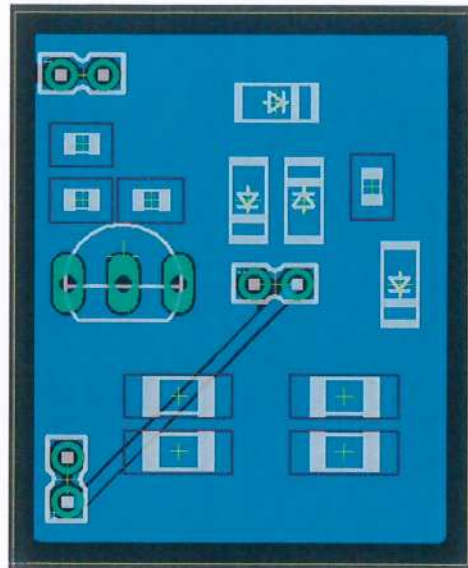
Value:

Smashed
CAPACITOR, European symbol

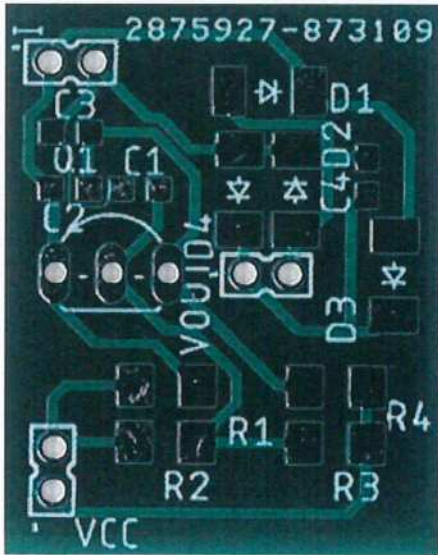
㉠ 보드설계



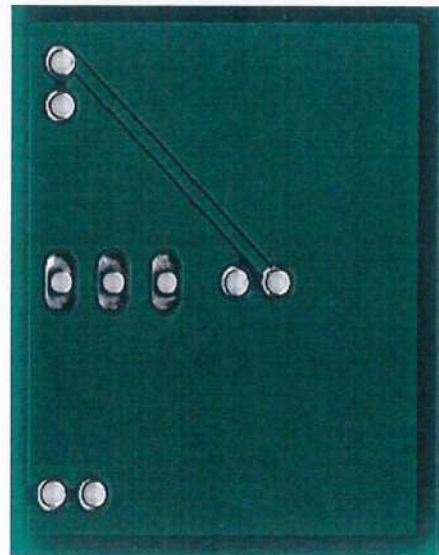
< 전면부 >



< 후면부 >

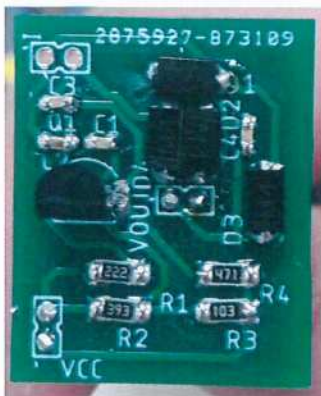


< 전면부 >



< 후면부 >

㉡ 납땜



4. 3D 프린팅 설계 및 제작

㉞ 3D 프린팅 설계

> 모델링 :

모델링은 직접 사람이 만들거나, 3D스캐너 등을 활용하여 역설계 방식으로 만들 수 있다. 3D모델링을 할 수 있는 프로그램은 정말 많다. 123D 디자인, Fusion 360 부터 많은 사람들이 들어보았을 오토 캐드, 라이노, 카티아, 3D MAX, sculptris, SketchUp 등 모든 모델링 프로그램을 사용할 수 있다. 그리고 사실 모든 모델링 프로그램들이 각자 특화되어 있는 부분이 다르기 때문에 만약에 모델링을 전문적으로 하는 사람이라면 다양한 프로그램을 모두 다룰 줄 아는 것이 좋을 것 같다.

3D모델링 작업이 끝났으면 STL파일로 변환해 주어야하는데, 여기서 STL파일은 3D프린터 출력 전에 거치게 되는 슬라이싱(slicing, 절편화)이라는 작업을 위한 파일이다. 슬라이싱도 나중에 프로그램이 하게 되는데, 모델링 프로그램들이 너무 많아 이 모든 파일들을 해석할 수 없는 문제로 STL이라는 파일 형식을 제정하였다.

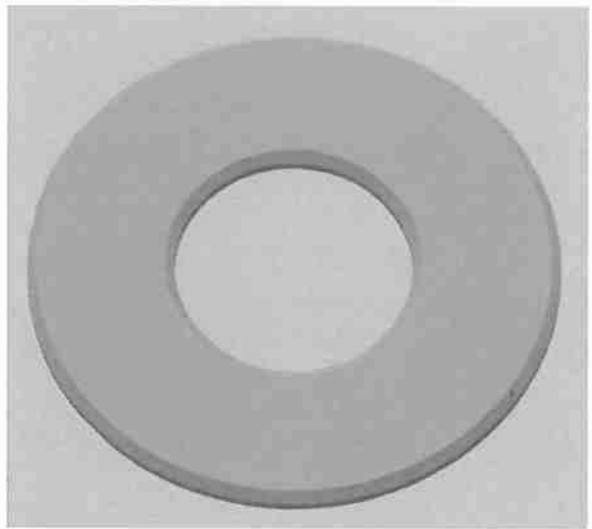
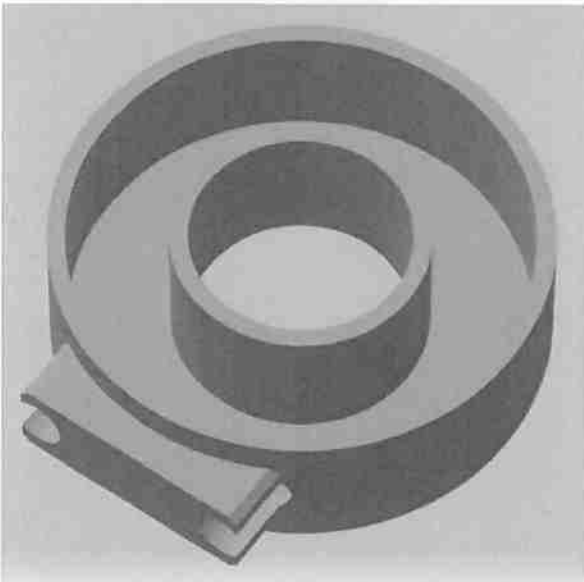
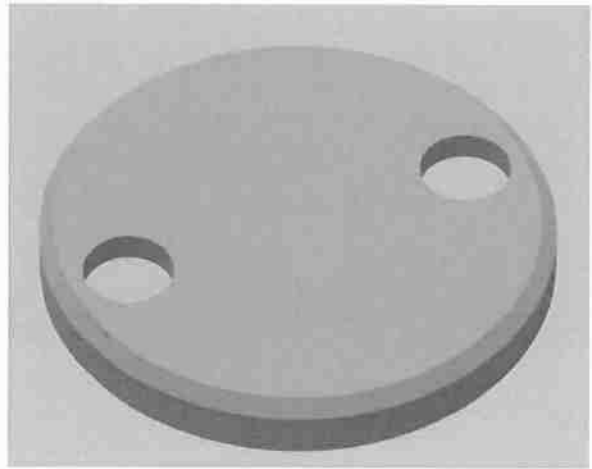
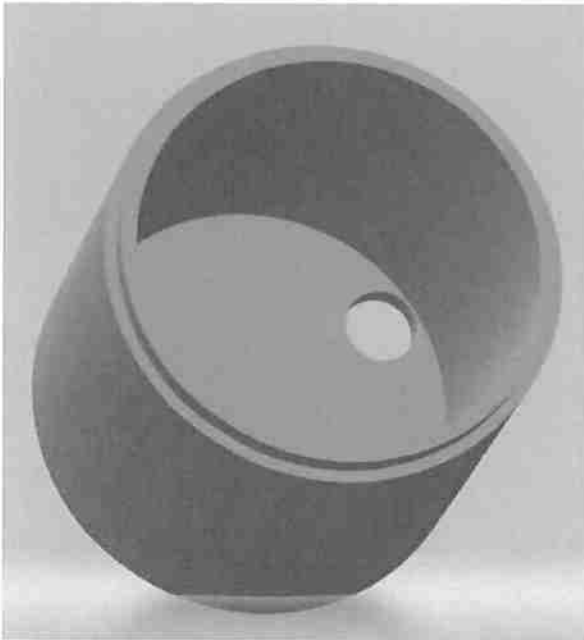
> 프린팅 :

프린팅 과정은 다시 2단계로 나뉜다. 첫번째는 슬라이싱 과정으로 이전의 모델링 과정에서 만든 STL파일을 불러서 3D프린터가 한 층씩 쌓을 수 있도록 한 층씩 잘라주는 작업이다. 이때 생성되는 것이 G 코드인데, 3D 프린터는 이 G 코드로 프린팅을 하게 된다. 두 번째는 실제로 프린팅하는 과정으로 한 층씩 쌓아서 만들며, 이때는 물리적인 여러 변수들(냉각, 프린팅 속도 등)이 간섭하기 때문에 의외로 프린트 하는데 시간이 약간 걸린다.

> 후처리 :

후처리는 프린팅 후 표면 가공, 도색 등을 포함하여 실제로 사용하기 전에 사용에 적합하게 하는 모든 작업을 칭한다. 분야에 따라서 그 방식들이 달라지기 때문에 이 부분에서 많은 전문적인 지식을 요구한다. 즉, 진행 과정상 어려움이 있는 부분은 세 부분으로, 모델링에서 실제 3차원 모델을 만드는 과정(그러나 3D 프린팅은 오픈소스를 지향하기 때문에 상당히 많은 모형이 인터넷에 공유되고 있다), 프린팅에서 여러 환경변수를 제어하는 과정, 적절한 후처리 과정 등이 있다.

㉠ Rhino을 이용한 설계



㉠ 3D 프린팅 제작

