

2017년 1학기

창의적설계

최종보고서

제목: 전자석을 이용한 타자 연습기

제출일: 2017년 6월 1일

지도교수명	김 동 한
조명	608호
조원명	오종렬, 윤상윤, 정준영
대표연락처	E-MAIL: jjy920517@khu.ac.kr 핸드폰: 010-4726-4009

1. 설계 작품 배경

A. 설계 작품의 중요성/필요성 (200자 이내로 요약)

키보드란 컴퓨터의 5대 요소 중 하나인 입력장치의 대표적인 기기로서, 반드시 필요한 장치이다. 이러한 키보드의 자판 배열은 여러 종류가 있으며, 영어 컴퓨터 자판으로는 “QWERTY 자판”이 표준이고, 한글 컴퓨터 자판은 “두벌식 자판”이 표준이다. 두 자판 모두 빠르고 정확한 속기를 위해 개발되고 사용 중이지만, 아직 컴퓨터에 익숙하지 못한 사용자들은 자판을 입력하는데 어려움을 느낀다. 이러한 사용자들은 장년층과 유아기가 있을 것이며 이들을 위해 효율적인 타자 연습을 위한 기기를 고안하였다. 이 기기는 전자석의 인력을 이용하여 각 문자마다 알맞은 손가락이 위치하도록 작동한다.

B. 설계 작품의 창의성 (200자 이내로 요약)

대부분의 사람들이 ‘한컴타자연습’을 통해 타자를 처음 접했을 것이다. 물론 화면을 보면서 자리를 하나하나씩 익히는 것도 좋다. 하지만 두발자전거를 처음 탈 때 뒤에서 누군가가 잡아주면 쉽게 배우는 것처럼, 처음 타자를 배울 때 손가락을 누군가가 옮겨주면 더 쉽게 배우지 않을까라고 생각하였다. 본인이 사용하던 키보드에 **키보드 패널ⁱ**을, 손에는 **핑커팁ⁱⁱ**을 장착하고 어플리케이션을 실행한다. 특정문자에 대해서만 키보드 패널과 핑커팁의 전자석에 전류가 흘러 인력이 발생한다.

2. 설계 작품에 대한 문제 정의

- (1) 키보드 제품마다 각기 다른 모양과 규격
- (2) Windows Application과 키보드 패널, 키보드 패널과 핑커팁의 통신 속도의 불확실성
- (3) 키보드 패널의 크기에 대한 불확실성(전자석의 크기에 대한 기전력의 세기를 실험하여 최적의 외형을 갖추도록 설계)

3. 설계 작품 내용

[a] 시작품에 대한 아이디어 창출

전체 프로젝트는 다음 세 부분으로 나눌 수 있다.

1) Windows Application

- 다른 기기들과 통신
- 한글 자모음의 타자연습이 가능한 노트 출력

2) 키보드 형태의 기기

- ‘키보드 패널’이라 정의
- Windows 컴퓨터와 통신
- 전자석 사용

3) 장갑 형태의 기기

- ‘핑커팁’이라 정의
- 전자석 사용

[b] 시작품에 대한 아이디어 다듬기

1) Windows Application

- 키보드 패널과 Serial 통신
- 영어 알파벳과 단어의 타자연습이 가능하도록 변경
- 직접 타자연습 노트를 제작하고 파일입출력을 통해 사용
- 키보드 인터럽트에 대한 기능 수행
- 키보드 인터럽트 발생 시, 키보드 패널의 상태 변화

2) 키보드 패널

- 기존 컴퓨터 키보드 각각의 키캡 위에 전자석을 부착하는
형태로 변경
- 컴퓨터 키보드 자판에 대응하는 전자석과 버튼이 존재
- 버튼을 누르면 컴퓨터 키보드의 자판이 따라 눌림
- Windows 컴퓨터와 Serial 통신
- 핑커팁과 Bluetooth 통신
- Windows Application의 신호 수신 시, 내부 회로의 전원공

- 급을 통해 특정 키캡에 전류 발생
- 캐캡에 전류 발생 시, 특정 전자석 활성화
- 전자석의 상태 변경 시, Windows Application에 상태 전달

3) 핑커틱

- 키보드 패널과 Bluetooth 통신
- 키보드 패널의 신호 수신 시, 내부 회로의 전원공급을 통해 특정 손가락에 전류 발생
- 특정 손가락에 전류 발생 시, 특정 전자석 활성화

[c] 시작품에 대한 아이디어 판정

- 1) 한글 단어의 키보드 입력은 자음과 모음, 받침을 순차적으로 입력해야 한다. 또한 된소리의 입력을 위해선 Shift 키를 누른 상태에서 해당 자음을 입력해야 한다. 이를 연습하기 위한 타자 연습기는 고려해야 할 사항의 가짓수가 많기 때문에 영어 알파벳과 영단어 입력을 위한 타자 연습기를 구상하였다.
- 2) Windows Application이 키보드 패널에 신호를 전달하고 키보드 패널은 핑커틱에게 신호를 전달함으로써 사용자가 순차적으로 타자를 입력할 환경을 제공하였다.
- 3) 키보드 패널을 키보드 형태가 아닌 키보드 위에 위치한 기기로 변경함으로써 기존 컴퓨터 키보드의 활용이 가능해졌다.

[d] 시작품 목표 사양 및 설계 방법

1) Windows Application

- Windows 7버전 이상에서 사용
- MFC 라이브러리 사용
- Serial 통신 사용

2) 키보드 패널

- AVR 보드 사용
- Serial 통신 사용
- Bluetooth 통신 사용

3) 핑커팁

- AVR 보드 사용
- Bluetooth 통신 사용

4. 설계 요구사항 정의

1) 성능적 제한 요구사항

- 타자연습을 위한 별도의 어플리케이션을 제작하고, 전자석을 이용한 타자연습 기능은 어플리케이션 실행시에만 작동
- 영어 키보드만 지원
- 직접 제작한 솔레노이드 전자석의 세기가 충분 여부

2) 비용적 제한 요구사항

- 특수문자와 기능 및 숫자 키는 제외(타자연습에 필요한 자음과 모음 키만 적용)

3) 환경(동작온도 등)적 제한 요구사항

- 전자유도장해 발생가능

4) 디자인적 제한 요구사항

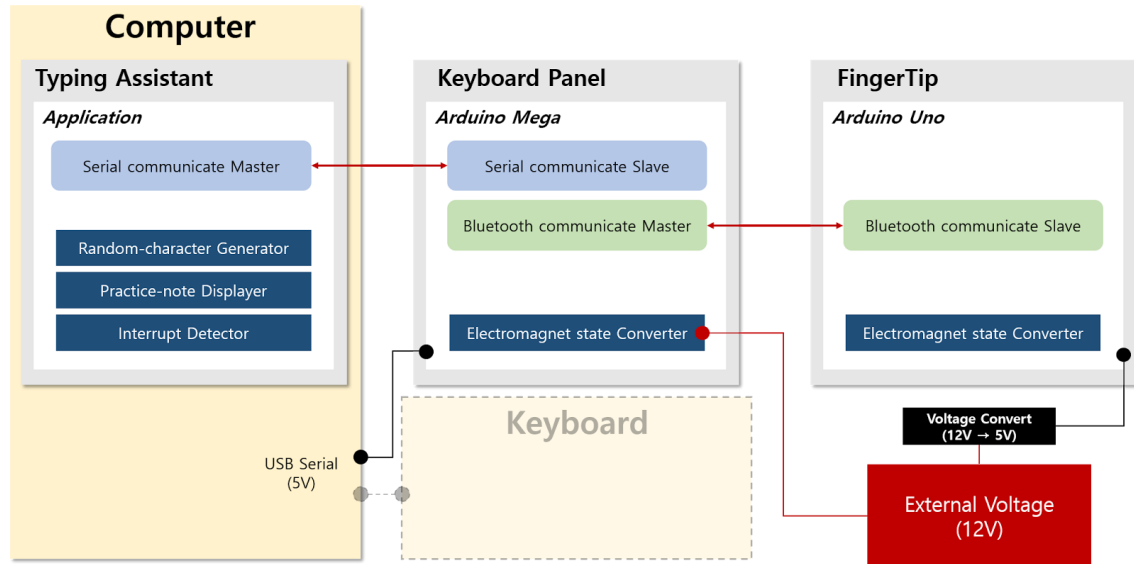
- 정해진 규격의 키보드에만 사용 가능(키캡 높이의 제한)
- 손가락에 장착할 컨트롤러와 키보드 위에 장착할 컨트롤러의 부피가 커질 우려가 있음

5. 설계 작품 요약

- 1) “전자석을 이용한 타자 연습기”는 전자석을 통한 자기력을 이용하여, 타자 연습의 효율을 높이는 타자 연습기다.
- 2) 컴퓨터 응용 프로그램인 “Typing Assistant”는 Node.js와 Chromium을 사용한 Electron을 통해 개발한다.
- 3) 키보드 위에 올려놓고 사용하는 “Keyboard Panel”은 아두이노 메가와 16채널 릴레이 모듈을 사용하여 전자석을 제어한다.
- 4) 장갑 형태의 “FingerTip”은 아두이노 우노와 4채널 릴레이 모듈을 사용하여 전자석을 제어한다.

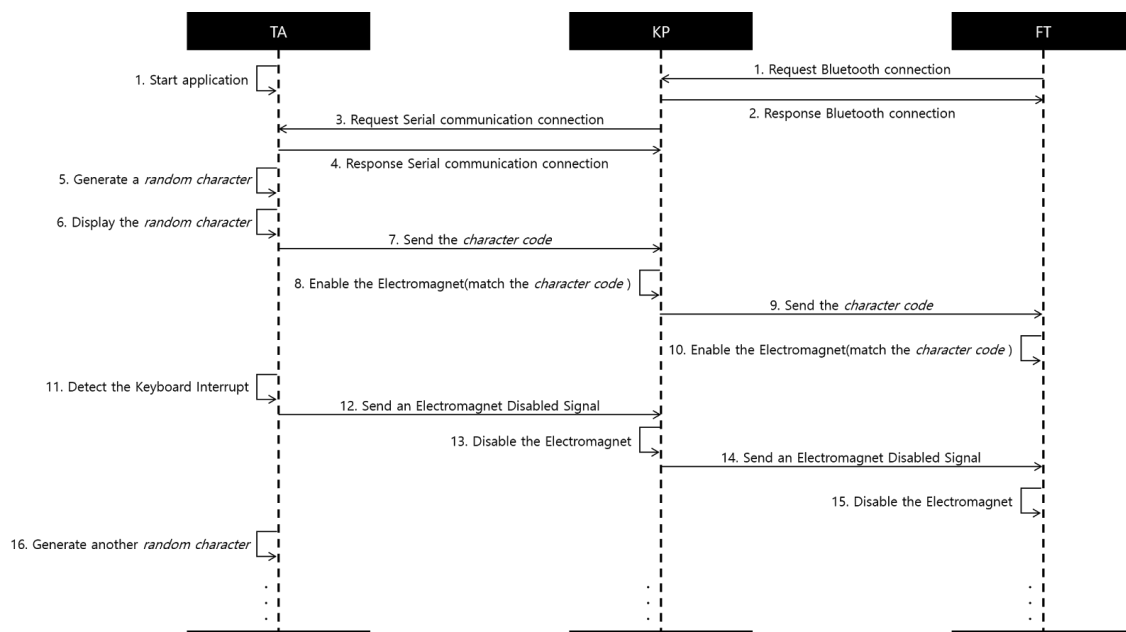
6. 시스템 설계 규격

1) System Architecture



[그림 1]과 같이 “전자석을 이용한 타자 연습기”는 컴퓨터 응용 프로그램 램인 “Typing Assistant”와 키보드 위에 올려놓고 사용하는 “Keyboard Panel”, 장갑 형태의 “FingerTip”으로 이루어져 있다.

2) Sequence Diagram



본 프로젝트의 동작 과정을 [그림 2]의 Sequence Diagram 으로 나타내었다.

“FingerTip”과 “Keyboard Panel”은 전원이 인가되면 서로 블루투스 연결 을 맺는다(과정 1, 2). 그리고 “Typing Assistant” 가 실행되면(과정 1) “Keyboard Panel”과 시리얼 연결을 맺는다(과정 3, 4). 그 후 “Typing Assistant”는 입력해야 할 문자를 무작위로 생성하며(과정 5) 컴퓨터 화면에 출력하고(과정 6), 사용자는 이 문자를 보고 타자 연습을 시작한다. “Typing Assistant”는 문자의 ASCII 코드를 “Keyboard Panel”에 전송하며 (과정 7), Keyboard Panel”은 이에 해당하는 전자석을 활성화 시킨다(과정 8). 그리고 “FingerTip”에도 똑같은 ASCII 코드를 전송하여(과정 9) “FingerTip”의 전자석도 활성화 시킨다(과정 10). 이때 “Keyboard Panel”과 “FingerTip”의 전자석은 서로 다른 극성을 가지기 때문에, 인력이 작동한다. 사용자가 이 인력을 인지하고 올바른 손가락으로 타이핑을 하면, 키보드 인 터럽트를 “Typing Assistant”가 감지하여(과정 11) “Keyboard Panel”에게 신호를 보낸다(과정 12). 이 신호로 인해 “Keyboard Panel”의 전자석은 비 활성화 되며(과정 13), “FingerTip”에도 같은 신호를 보내어(과정 14) 전자 석이 비활성화 되도록 한다(과정 15). 이 후 “Typing Assistant”는 새로운 문자를 생성하고(과정 16) 위의 과정을 반복한다.

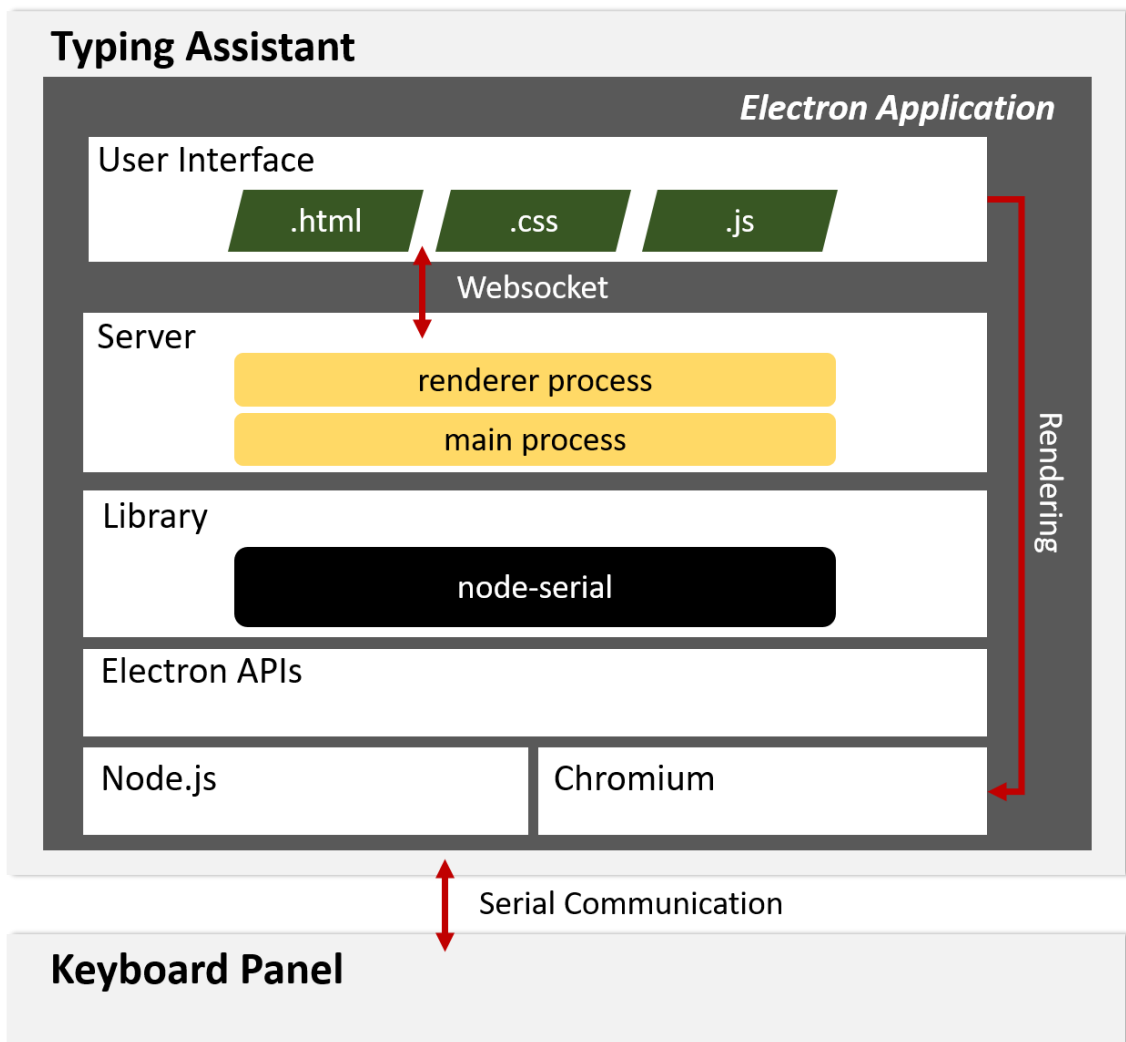
7. 모듈 설계 규격

1) Typing Assistant(TA)

- 개발환경(Development Environment)

“Typing Assistant”는 JetBrains사의 WebStorm Ver.2016.3.2를 통해 개발되었다. WebStorm은 웹의 Front-End 개발을 위해 사용되는 IDE다. 이를 통해 알 수 있듯이 “Typing Assistant”는 웹을 기초로 만들어졌다.

- 시스템 구성도(System Architecture)



“Typing Assistant”는 네 가지 기능을 가지고 있다. 첫째, 타자연습을 할 노트(입력해야 할 문자의 배열이 들어있는 텍스트 파일)를 읽는 기능. 둘째, 읽은

노트를 화면에 출력하는 기능. 셋째, 키보드인터럽트를 감지하여 다음 문자로 변경시키는 기능. 넷째, 시리얼 통신을 통해 “Keyboard Panel”과 통신하는 기능이다. 이 기능들을 사용하여 “Typing Assistant”는 입력해야 할 문자를 “Keyboard Panel”에게 알려주고, 사용자가 키보드 입력을 하면 다음 문자를 다시 “Keyboard Panel”에게 알려줄 수 있다.

[그림 3]와 같이 “Typing Assistant”의 개발에는 Node.js 와 Chromium 을 사용하여 데스크톱 어플리케이션 개발이 가능한 Electron 을 사용한다. 따라서 “Typing Assistant”는 Windows, Linux, MacOS 의 운영체제에서 실행할 수 있다. 그리고 어플리케이션과 컴퓨터 운영체제와의 통신을 위해 websocket 을 사용하며, “Keyboard Panel”과의 통신을 위해 시리얼 통신 을 이용한다.

- 인터페이스(Interface) : 시리얼 통신

“Typing Assistant”는 시리얼 통신을 통해 “Keyboard Panel”에게 Character 형의 코드를 전송한다.

시리얼 통신이란 PC 와 디바이스 간 직렬 방식으로 데이터를 주고받는 통신 프로토콜을 의미하며, 대표적으로 UART(Universal Asynchronous serial Receiver and Transmitter) 통신이 있다. 본 프로젝트에서는 UART 통신을 사용하지만, 아두이노 개발환경에서 UART 통신을 시리얼 통신으로 지칭하므로, 본 프로젝트에서도 시리얼 통신이라 하겠다.

PC 와 디바이스의 시리얼 통신을 사용하기 위한 조건은 두 기기의 보드 속도(Baud rate)를 일치시키는 것이다. 보드 속도가 일치되면 데이터를 전송하는 기기는 시작 비트(Start bit; 1bit), 데이터 비트(Data bit; 8bits), 패리티 비트(Parity bit; 1bit), 정지 비트(Stop bit; 1bit)로 구성된 데이터 형태(총 11bits)를 만든다. 이때 전송할 정보는 데이터 비트에 적재되어 전달되고 수신하는 기기에서는 적재된 정보를 데이터 비트에서 꺼내어 사용한다.

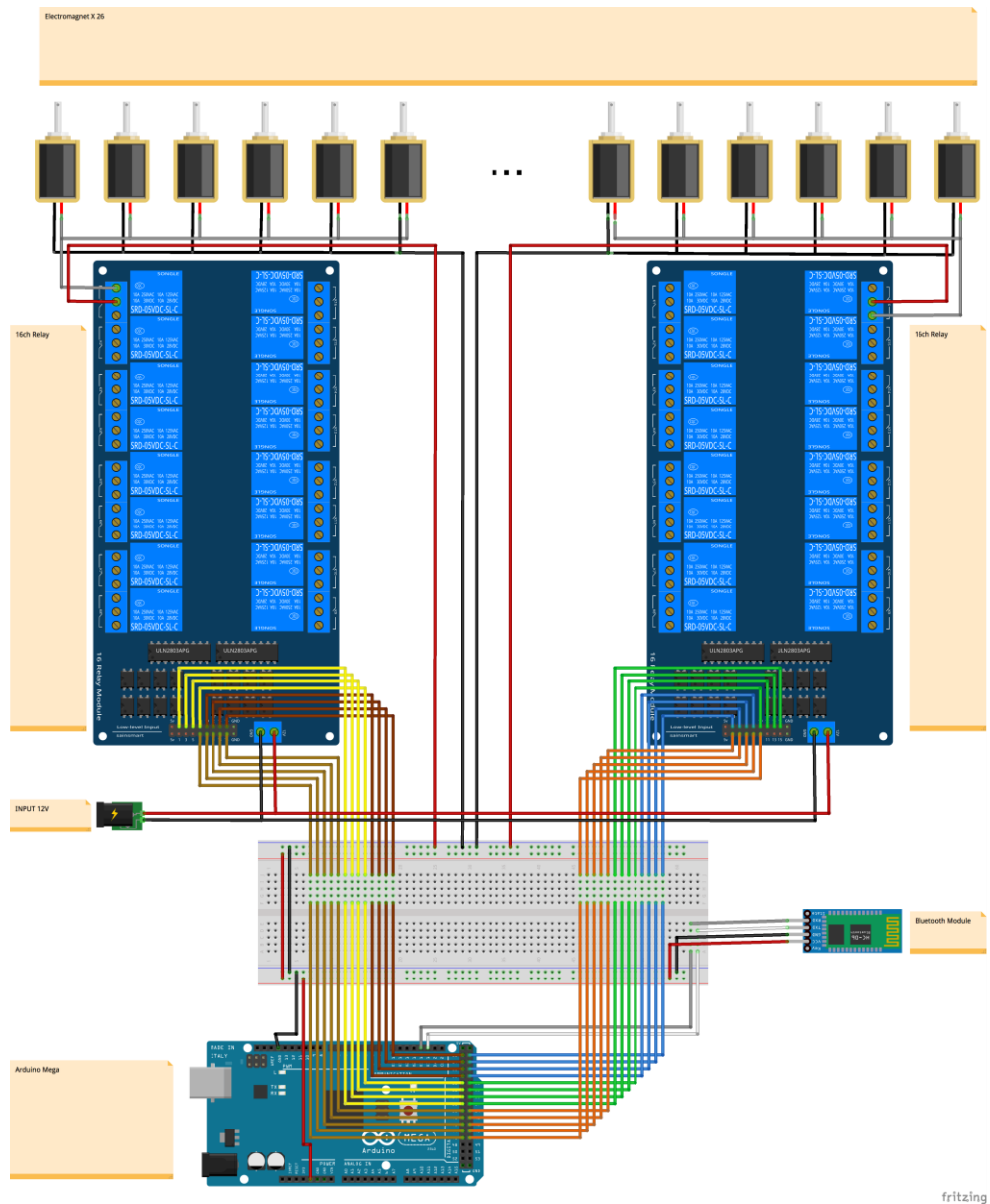
“Typing Assistant”가 수행되는 PC는 아두이노 메가로 만들어진 “Keyboard Panel”과 USB 케이블로 연결되어 있으며, “Typing Assistant”는 “Keyboard Panel”의 보드 속도를 알고 있다. 따라서 두 구성요소는 시리얼 통신을 할 수 있으며, “Typing Assistant”는 ASCII code(총 7bits)를 데이터 비트에 실고 송신한다.

2) Keyboard Panel(KP)

- 개발환경(Development Environment)

아두이노의 펌웨어는 개발자가 직접 수정이 불가능하므로, 호스트 PC를 통해 필요한 기능을 컴파일 후 아두이노 보드로 업로드해야 한다. “Keyboard Panel”은 IDE로 Arduino Sketch v1.8.2를 사용한다.

- 회로도(Circuit design)



[그림 4] Keyboard Panel 회로도

[그림 4]와 같이 “Keyboard Panel”은 아두이노 메가 보드와 16

채널 릴레이 2개, 전자석 26개를 사용한다. “Keyboard Panel”에는 특정 전자석을 활성화하는 기능이 있다. 이 기능은 “Typing Assistant”로부터 시리얼 통신으로 받은 값에 따라 동작하며, 전자석의 활성화 후 “FingerTip”에게 블루투스 통신으로 활성화되어야 할 전자석을 알려준다.

- 인터페이스(Interface) : 시리얼 통신

“Keyboard Panel”은 시리얼 통신을 통해 “Typing Assistant”에게 Character 형 코드를 전달 받는다.

시리얼 통신이란 PC와 디바이스 간 직렬 방식으로 데이터를 주고받는 통신 프로토콜을 의미하며, 대표적으로 UART(Universal Asynchronous serial Receiver and Transmitter) 통신이 있다. 본 프로젝트에서는 UART 통신을 사용하지만, 아두이노 개발환경에서 UART 통신을 시리얼 통신으로 지칭하므로, 본 프로젝트에서도 시리얼 통신이라 하겠다.

PC와 디바이스의 시리얼 통신을 사용하기 위한 조건은 두 기기의 보드 속도(Baud rate)를 일치시키는 것이다. 보드 속도가 일치되면 데이터를 전송하는 기기는 시작 비트(Start bit; 1bit), 데이터 비트(Data bit; 8bits), 패리티 비트(Parity bit; 1bit), 정지 비트(Stop bit; 1bit)로 구성된 데이터 형태(총 11bits)를 만든다. 이때 전송할 정보는 데이터 비트에 적재되어 전달되고 수신하는 기기에서는 적재된 정보를 데이터 비트에서 꺼내어 사용한다.

“Typing Assistant”가 수행되는 PC는 아두이노 메가로 만들어진 “Keyboard Panel”과 USB 케이블로 연결되어 있으며, “Typing Assistant”는 “Keyboard Panel”의 보드 속도를 알고 있다. 따라서 두 구성요소는 시리얼 통신을 할 수 있으며, “Keyboard Panel”은 데이터 비트에 저장되어 있는 ASCII code(총 7bits)를 수신한다.

- 인터페이스(Interface) : 블루투스 통신

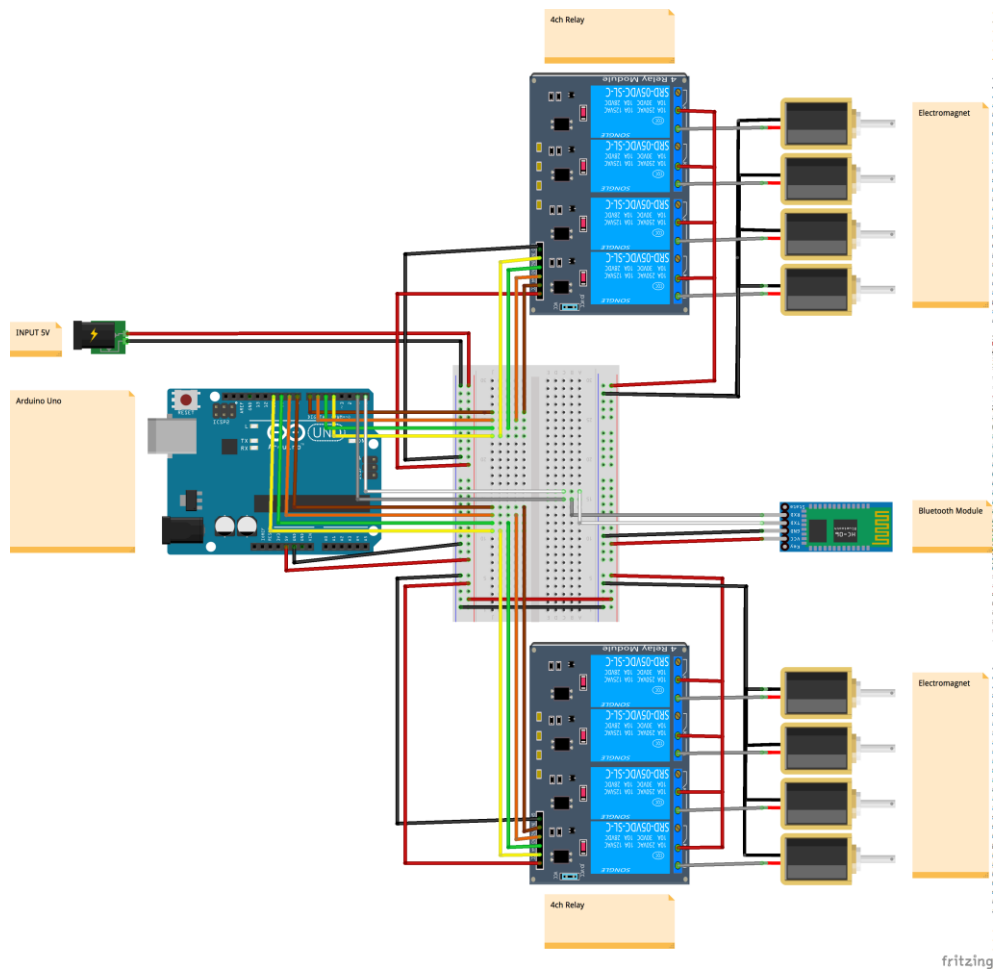
“Keyboard Panel”은 “FingerTip”과 블루투스 통신을 통해 데이터를 전송한다.

블루투스 통신이란 기기 간 저전력 데이터 통신을 하여 데이터를 주고받는 일대일 근거리 무선 통신을 의미한다. 블루투스 통신을 하기 위해서는 마스터(Master) 기기와 슬레이브(Slave) 기기로 구성하여야 한다. 본 프로젝트에서는 “Keyboard Panel”을 마스터, “FingerTip”을 슬레이브 기기로 정의한다. 블루투스 통신 과정은 다음과 같다.

- 1) 블루투스 모듈(HC-06)에 전원을 인가한다.
- 2) Master 기기인 “Keyboard Panel”과 연결된 블루투스 모듈이 Frequency hopping 을 통해 연결 가능한 기기를 탐색한다.
- 3) 기기의 이름, Mac address, Pair password 를 비교하여 일치하는 기기와 커넥션을 맺는다.
- 4) 연결된 기기와 데이터 통신을 진행한다.

3) FingerTip(FT)

- 개발환경(Development Environment)
아두이노의 펌웨어는 개발자가 직접 수정이 불가능하므로, 호스트 PC를 통해 필요한 기능을 컴파일 후 아두이노 보드로 업로드해야 한다. “Keyboard Panel”은 IDE로 Arduino Sketch v1.8.2를 사용한다.
- 회로도(Circuit design)



[그림 5] FingerTip 회로도

[그림 5]와 같이 “FingerTip”은 아두이노 우노 보드와 4채널 릴레이 2개, 전자석 8개를 사용한다. “FingerTip”에는 특정 전자석을 활성화하는 기능이 있다. 이 기능은 “Keyboard Panel”로부터 블루투스 통신으로 받은 값에 따라 동작하며, “Keyboard Panel”가 보내는 값에 따라 비활성화시킬 수 있다.

- 인터페이스(Interface) : 블루투스 통신

“Keyboard Panel”은 “FingerTip”과 블루투스 통신을 통해 데이터를 전송한다. 블루투스 통신이란 기기 간 저전력 데이터 통신을 하여 데이터를 주고 받는 일대일 근거리 무선 통신을 의미한다. 블루투스 통신을 하기 위해서는 마스터(Master) 기기와 슬레이브(Slave) 기기로 구성하여야 한다. 본 프로젝트에서는 “Keyboard Panel”을 마스터, “FingerTip”을 슬레이브 기기로 정의한다. 블루투스 통신 과정은 다음과 같다.

- 1) 블루투스 모듈(HC-06)에 전원을 인가한다.
- 2) Master 기기인 “Keyboard Panel”과 연결된 블루투스 모듈이 Frequency hopping 을 통해 연결 가능한 기기를 탐색한다.
- 3) 기기의 이름, Mac address, Pair password 를 비교하여 일치하는 기기와 커넥션을 맺는다.
- 4) 연결된 기기와 데이터 통신을 진행한다.

8. 설계작품의 시험 및 성능 결과

- 1) TA 실험
- 2) KP - FT 간 통신 및 릴레이 제어 시험
 - ① KP 와 FT 의 블루투스 통신
최초의 통신 데이터의 Type 을 Byte 형으로 송수신하였다.

```
#include <SoftwareSerial.h>
SoftwareSerial btSerial(12, 13);

void setup() {
    Serial.begin(9600);
    btSerial.begin(9600);
}

void loop() {
```

```

volatile byte data;
data = Serial.read();
if (Serial.available()) {
    delay(5);

}
if (btSerial.available()) {
    delay(5);
    while (btSerial.available()) {
        Serial.write(btSerial.read());
    }
}
}

```

블루투스의 데이터 비트의 크기가 8bit 이다. 그리고 KP 와 FT 사이 주고받는 데이터 Type 을 byte 로 설정하였을 때 송수신하는 데이터의 크기가 4 bit 였다. 그로 인해 송수신하는 데이터의 양과 속도가 증가할 때 데이터가 손실하는 경우가 발생하여 데이터의 type 을 byte -> char 형으로 수정하였다.

```

#include <SoftwareSerial.h>
SoftwareSerial btSerial(12, 13);

void setup() {
    Serial.begin(9600);
    btSerial.begin(9600);
}

void loop() {
    volatile char data;
    data = Serial.read();
    if (Serial.available()) {
        delay(5);

    }
}

```

```

        if (btSerial.available()) {
            delay(5);
            while (btSerial.available()) {
                Serial.write(btSerial.read());
            }
        }
    }
}

```

그 결과 블루투스 통신 중 데이터의 손실이 줄어드는 효과를 얻을 수 있었다.

② KP 와 FT 의 Relay 제어

KP 에서는 16 Channel Relay 2 개로 총 32 개의 channel 중 26 개의 channel 을 사용하였고 FT 에서는 4 Channel Relay 2 개로 총 8 개의 channel 모두를 사용하였다.

특정 Relay 를 제어하는 함수에서 처음에는 if ~ elseif 문을 사용하여 제어하였다.

<KP relay 제어 함수>

```

void turnOn(char data)
{
    if(data == 'q') I1(data);
    else if(data == 'a') I2(data);
    else if(data == 'z') I3(data);
    else if(data == 'w') I4(data);
    else if(data == 's') I5(data);
    else if(data == 'x') I6(data);
    else if(data == 'e') I7(data);
    else if(data == 'd') I8(data);
    else if(data == 'c') I9(data);
    else if(data == 'r') I10(data);
    else if(data == 'f') I11(data);
    else if(data == 'v') I12(data);
    else if(data == 't') I13(data);
}

```



```

else if(data == 'g') l14(data);
else if(data == 'b') l15(data);

else if(data == 'y') r1(data);
else if(data == 'h') r2(data);
else if(data == 'n') r3(data);
else if(data == 'u') r4(data);
else if(data == 'j') r5(data);
else if(data == 'm') r6(data);
else if(data == 'i') r7(data);
else if(data == 'k') r8(data);
else if(data == 'o') r9(data);
else if(data == 'l') r10(data);
else if(data == 'p') r11(data);
}

```

<FT relay 제어함수>

```

void TurnOn(char data)
{
    if(data=='1'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(l1, HIGH); // 왼손 새끼 손가락 On
    }

    elseif(data=='2'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(l2, HIGH); // 왼손 약지 On
    }

    elseif(data=='3'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(l3, HIGH); // 왼손 중지 On
    }

    elseif(data=='4'){

```

```
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(l4, HIGH); // 왼손 검지 On
    }

    elseif(data=='5'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(r1, HIGH); // 오른손 검지 On
    }

    elseif(data=='6'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(r2, HIGH); // 오른손 중지 On
    }

    elseif(data=='7'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(r3, HIGH); // 오른손 약지 On
    }

    elseif(data=='8'){
        turnOff(); // 모든 전자석을 Off 한다.
        digitalWrite(r4, HIGH); // 오른손 새끼 손가락 On
    }
}
```

8 개의 channel 만을 제어하는 FT 에서는 문제가 발생하지 않았지만 26 개의 channel 을 제어하는 KP 에서 입력이 빠르게 들어오는 경우 해당 channel 이 On 이 되지 않는 경우가 발생하여 해당 channel 을 찾는 데에 binary search 알고리즘을 적용하여 코드를 수정함으로써 depth 가 26 -> 5 개로 감소시켰다.

<KP relay 제어 함수>

```
void turnOn(char data)
{
    if (data < 'm')
    {
        if (data < 'f')
        {
            if (data < 'c')
            {
                if (data < 'b')
                {
                    l2(data); //a
                }
                else
                {
                    l15(data); //b
                }
            }
            else if (data > 'c')
            {
                if (data > 'd')
                {
                    l8(data); //e
                }
                else
                {
                    l7(data); //d
                }
            }
            else
            {
                l9(data); //c
            }
        }
    }
}
```

```
if (data > 'f')
{

    if (data < 'i')
    {
        if (data < 'h')
        {
            l14(data); //g
        }
        else
        {
            r2(data); //h
        }
    }
    else if (data > 'i')
    {

        if (data < 'k')
        {
            r5(data); // j
        }
        else if (data > 'k')
        {
            r10(data); //l
        }
        else
        {
            r8(data); //k
        }
    }
    else
    {
        r5(data); //i
    }
}
```

```

        }
        else
        {
            l11(data); //f
        }
    }
    //////////////////////////////////////
    if (data > 'm')
    {
        if (data < 't')
        {
            if (data < 'p')
            {
                if (data < 'o')
                {
                    r3(data); //n
                }
                else
                {
                    r9(data); //o
                }
            }
        }
        else if (data > 'p')
        {
            if (data < 'r')
            {
                l1(data); //q
            }
            else if (data > 'r')
            {
                l5(data); //s
            }
            else
            {

```

```

        l10(data); //r
    }
}
else
{
    r11(data); //p
}
}
if (data > 't')
{
    if (data < 'w')
    {
        if (data < 'v')
        {
            r4(data); //u
        }
        else
        {
            l12(data); //v
        }
    }
    else if (data > 'w')
    {
        if (data < 'y')
        {
            l6(data); //x
        }
        else if (data > 'y')
        {
            l3(data); //z
        }
        else
        {
            r1(data); //y
        }
    }
}

```

```

        }
    }
    else
    {
        l4(data); //w
    }
}
else
{
    l13(data); //t
}
}
else
{
    r6(data); //m
}
}

```

<FT relay 제어함수>

```

void TurnOn(char data)
{
    if(data < '4')
    {
        if(data<'2')
        {
            TurnOff();
            digitalWrite(relay_l1, LOW);    // set the LED on
            Serial.println("1");
        }
        else if(data>'2')
        {
            TurnOff();
            digitalWrite(relay_l3, LOW);    // set the LED off
            Serial.println("3");
        }
    }
}

```

```
}  
else  
{  
    TurnOff();  
    digitalWrite(relay_l2, LOW);    // set the LED off  
    Serial.println("2");  
}  
}  
else if(data > '4')  
{  
    if(data < '6')  
    {  
        TurnOff();  
        digitalWrite(relay_r1, LOW);    // set the LED on  
        Serial.println("5");  
    }  
    else if(data > '6')  
    {  
        if(data > '7')  
        {  
            TurnOff();  
            digitalWrite(relay_r4, LOW);    // set the LED off  
            Serial.println("8");  
        }  
        else  
        {  
            TurnOff();  
            digitalWrite(relay_r3, LOW);    // set the LED off  
            Serial.println("7");  
        }  
    }  
}  
else  
{  
    TurnOff();  
    digitalWrite(relay_r2, LOW);    // set the LED off
```



```

        Serial.println("6");
    }
}
else
{
    TurnOff();
    digitalWrite(relay_I4, LOW);    // set the LED off
    Serial.println("4");
}
}

```

depth 가 26 에서 5 로 감소시킨 수정 전 발생했던 해당 channel 이 on 되지 않았던 문제가 해결되었다.

3) 전자석의 동작환경 시험

① 전자석간 자기력 세기

전자석의 자기력이 활성화되면 손으로 강력한 힘을 주어야만 떼어낼 수 있었다. TA 에 알파벳이 출력되면 KP 와 FT 에서는 전자석을 활성화시킨다. 사용자가 키보드를 typing 하면 새로운 알파벳이 출력되고 KP 와 FT 가 전체 전자석을 비활성 후 새로운 전자석을 활성화 시키는 방법을 사용했다.이를 통해 필요할 때 전자석의 강력한 세기를 잘 활용하고, 다음 입력을 위해 큰 힘을 들이지 않고 손가락을 움직일 수 있게끔 구현했다.

② 발열온도

처음 전자석을 선정할 때 가격과 성능, 세기와 발열온도 사이에서 여러 제품을 두고 고민하였다. 합리적인 가격에서 결정하다 보니 1 분 이상 사용시, FT 에서 손가락에 부착된 형태로 사용하기 어려울 만큼 발열이 심하다는 단점이 있었다. 따라서 새로운 전체 전자석을 비활성화하는 방법으로, 전자석의 발열시간을 최소화하였다.

·
·

☎ 4항의 제한요구사항을 만족하도록 상세설계(8항)를 수행하고, 최종결과보고서에 제한요구사항을 만족함을 보이는 시험(실험), 성능 등의 결과를 보

일 것.

9. 8항의 시험 결과가 4항의 제한요구사항을 만족하는가?

평가 항목		목표	가중치 (%)	평가 기준안
성 능	[FT-KP] 블루투스 연결	프로그램 시작 시 FT→KP: 블루투스 접속 시도 KP→FT: 블루투스 접속 허용	3	39 해당 절차가 빠짐없이 제대로 수행되었는가?
	[KP-TA] 시리얼 통신	KP→TA: 시리얼 통신 접속 시도 TA→KP: 시리얼 통신 접속 허용	3	
	[TA] 알파벳 랜덤 출력	임의의 알파벳 출력	3	
	[TA-KP] ASCII코드 송수신	TA에 출력된 알파벳에 해당하는 ASCII코드 전송	3	
	[KP] 전자석 활성화	KP의 해당 릴레이 및 전자석 활성화	3	
	[KP-FT] ASCII코드 송수신	KP가 수신한 ASCII코드를 FT로 전송	3	
	[FT] 전자석 활성화	FT의 해당 릴레이 및 전자석 활성화	3	
	[TA] 키보드 인터럽트 감지	키보드 자판의 입력 감지	3	
	[TA-KP] 전자석 비활성화 신호 송수신	TA에서 KP로 비활성화 신호 전송	3	
	[KP] 전자석 비활성화	KP의 해당 릴레이 및 전자석 비활성화	3	
	[KP-FT] 전자석 비활성화 신호 송수신	KP에서 FT로 비활성화 신호 전송	3	
	[FT] 전자석 비활성화	FT의 해당 릴레이 및 전자석 비활성화	3	
	[TA] 새로운 알파벳 출력	새로운 알파벳을 랜덤으로 출력	3	

비용	200,000원대에서 제작	15	예산 범위 내에서 제작되었는가?
환경(동작온도)	전자석의 발열온도 및 발열시간 최소화, 전자유도장해 발생억제	30	구동 시 사고나 기기의 이상이 없도록 발열을 고려하였는가? 전자유도장해를 최소화하였는가?
제품 디자인	키캡 높이의 제한(최대 3cm) KP와 FT의 부피 최소화	16	사용자의 편리성과 심미성을 고려하였는가?

10. 업무 분장

임무 중요도 업무		1순위 개발	2순위 개발	3순위 개발
TA	Application 디자인	정준영	오종렬	윤상윤
	Application 개발	정준영	오종렬	윤상윤
	Serial 통신	정준영	오종렬	윤상윤
KP	HW 제작	윤상윤	정준영	오종렬
	Serial 통신	윤상윤	정준영	오종렬
	Bluetooth 통신	윤상윤	정준영	오종렬
	릴레이 제어	윤상윤	정준영	오종렬
FT	HW 제작	오종렬	윤상윤	정준영
	Bluetooth 통신	오종렬	윤상윤	정준영
	릴레이 제어	오종렬	윤상윤	정준영

11. 설계 진행일정

구 분 주	추진 일정									진도율 (%)
	2	4	6	8	10	12	14	16	포트폴리오	
설계 세부 업무										
자료수집 기술분석 설계 문제 정의	←-----→									100
설계계획서 작성 및 제출	←-----→									100
시스템설계규격 작성		←-----→								100
모듈설계규격 작성 스퀀스 다이어그램 작성 회로도 설계			←-----→							100
필요 물품 주문 및 준비				←-----→						100
H/W 및 S/W 상세설계				←-----→						100
중간보고서 제출 발표				←-----→						100
모듈 테스트 성능 테스트 유저 테스트 제한조건요구사항 확인시험					←-----→					100
최종 보고서 발표							←-----→			100
학습성과 포트폴리오 업로드 공학인증 사정보고서 제출					←-----→					100
총 진도율	30			40			30			100

12. 공학인증 사정보고서 (개인별)

공학인증 대상 아님

ⁱ 3. 설계내용 작품 참조

ⁱⁱ 3. 설계내용 작품 참조