

초보자도 간단하게 만들수 있는

앱 제작 프로그램 App Inventor





MIT App Inventor

App Inventor 응용

Section 1. 블루투스 응용

Section 2. 음성인식 응용

Section 3. 센서와 DB 응용

Section 4. MIDI 응용

App Inventor 응용

● 체크 포인트

1. Internet Explorer에서는 실행이 안된다.
2. Google 계정이 필요하다.
3. Wi-Fi를 이용할 경우 AP가 동일해야 한다.
4. MIT AI2 Companion의 어플을 이용하여 디버깅 기능

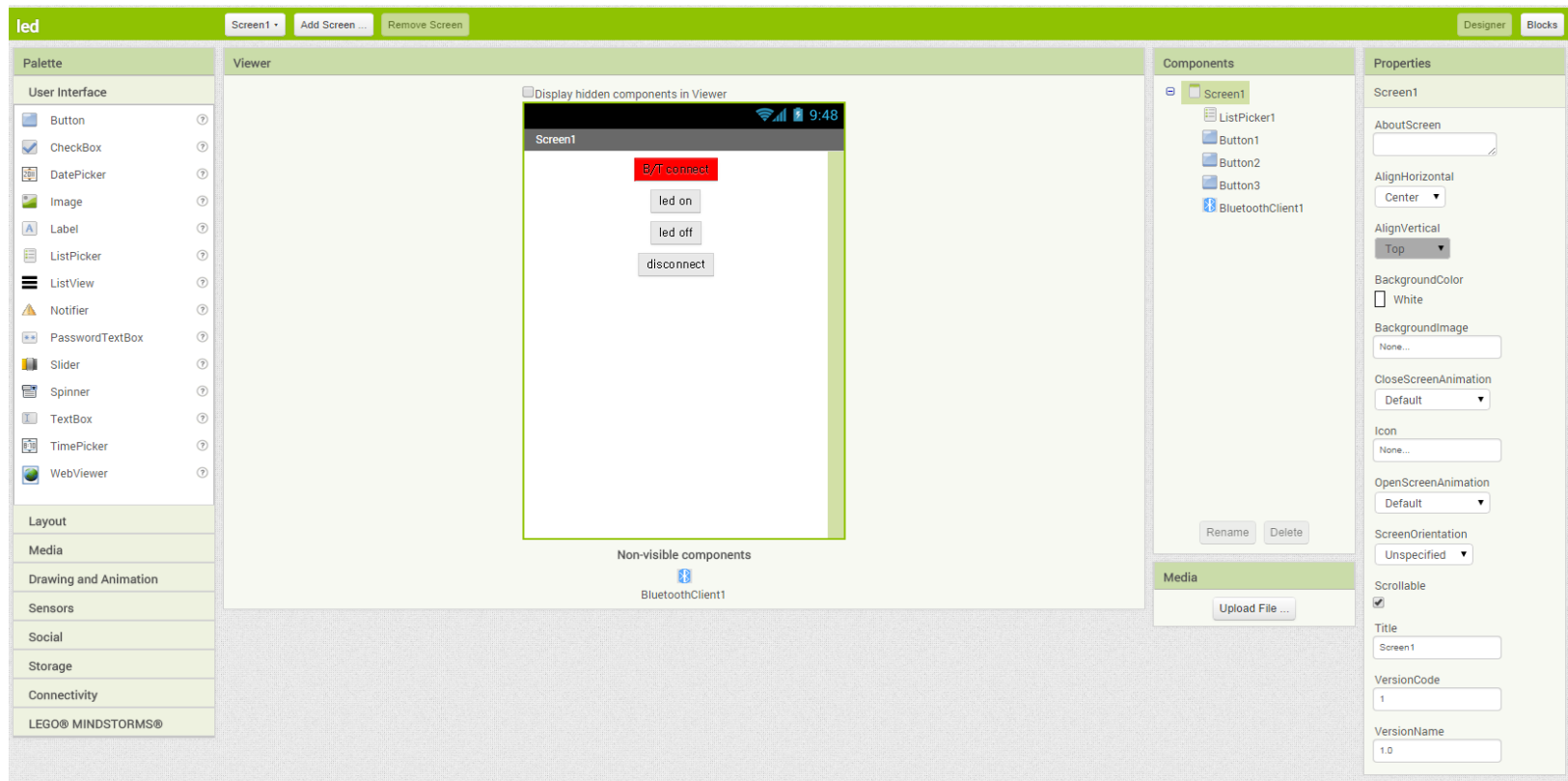
App Inventor 응용

Section 1. B/T 응용

- B/T는 App Inventor에서 Connectivity부분에서 가장 많이 활용하는 접근성을 제공한다.
- 기본적인 예제로 LED 제어
- IOT 단말 장치 중 아두이노 활용

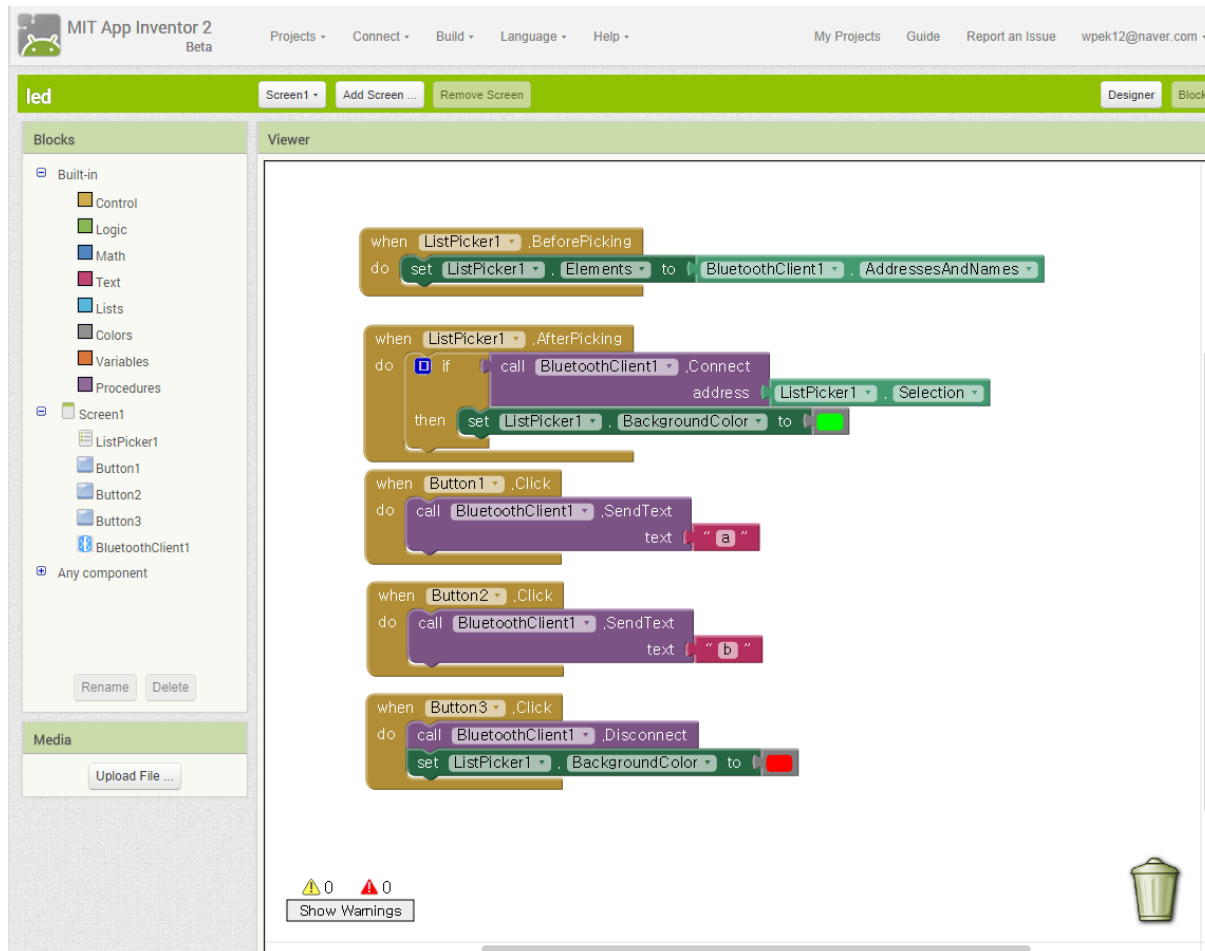
App Inventor 응용

- Start new project 실행
- Designer에서 UI 제작



App Inventor 응용


- Blocks으로 컴포넌트와 동작 제어를 생성
- 아래의 그림은 App Inventor의 전체 블록



App Inventor 응용

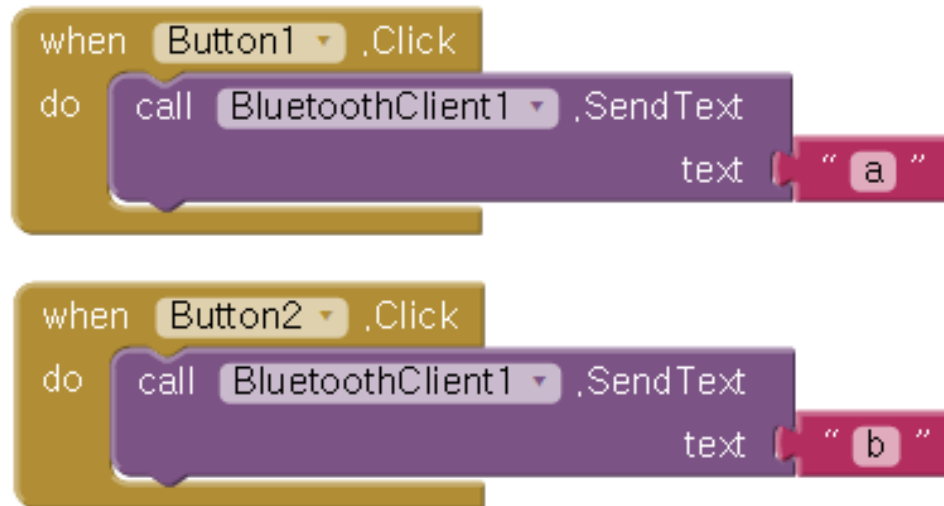
- B/T 페어링 방법
- 주의할 점은 App Inventor에서 검색을 해주는것이 아니라 이미 페어링을 해놓은 기기만을 검색을 해서 연결
- B/T와 연결이 되면 BackgroundColor를 이용하여 B/T의 연결상태의 여부를 원하는 색깔로 확인 가능하다.

```
when ListPicker1 .BeforePicking  
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```

```
when ListPicker1 .AfterPicking  
do if call BluetoothClient1 .Connect  
    address ListPicker1 . Selection  
    then set ListPicker1 . BackgroundColor to 
```

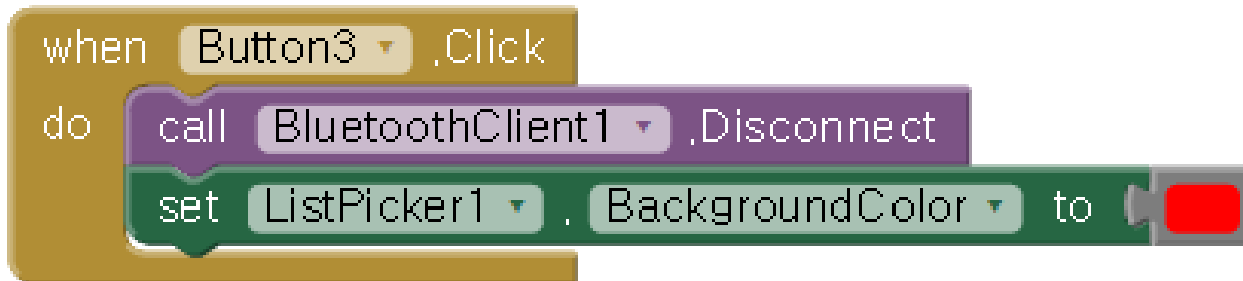
App Inventor 응용

- 각각 Button1(led on) Button2(led off)를 클릭을 하면 B/T를 통해서 각각 a,b라는 텍스트 전송한다.



App Inventor 응용

- Button3(disconnect)로 B/T 해제
- 색깔(빨강) 해제 상태로 표시한다.



App Inventor 응용

● IOT 디바이스, 아두이노에서 LED 제어 소스코드

```
test_LED
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(2,3);
byte a=0;
int LED=13;
void setup(){
    Serial.begin(9600);
    Serial.println("Hello!");
    BTSerial.begin(9600);
    pinMode(LED,OUTPUT);
}

void loop(){
    if(Serial.available()){
        a = Serial.read();
        Serial.print(a);
        if(a==97){
            Serial.println("led on");
            digitalWrite(LED,HIGH);
        }
        if(a==98){
            Serial.println("led off");
            digitalWrite(LED,LOW);
        }
    }
}
```

App Inventor 응용

- 소스코드 중 byte 주목
- App Inventor에서 B/T를 사용할 경우, ASCII코드 byte형식으로 전송한다.

```
#include <SoftwareSerial.h>

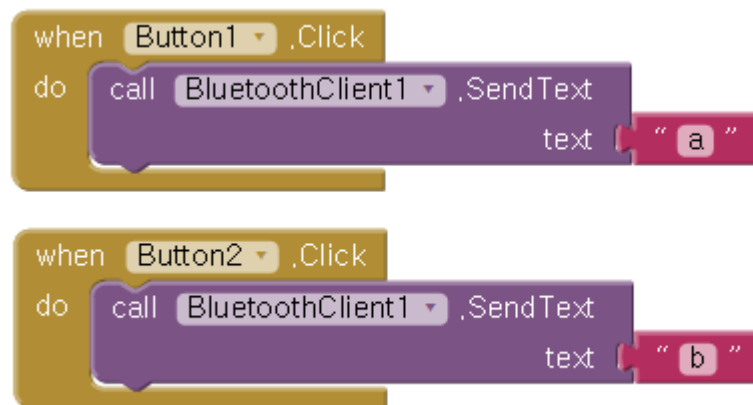
SoftwareSerial BTSerial(2,3);
byte a=0;
int LED=13;
void setup(){
  Serial.begin(9600);
  Serial.println("Hello!");
  BTSerial.begin(9600);
  pinMode(LED, OUTPUT);
}
```

App Inventor 응용

- ASCII 코드/App Inventor/아두이노 코드
- a=97, b=98로 전송 됨을 확인 가능하다.

```
void loop(){  
  if(Serial.available()){  
    a = Serial.read();  
    Serial.print(a);  
    if(a==97){  
      Serial.println("led on");  
      digitalWrite(LED,HIGH);  
    }  
    if(a==98){  
      Serial.println("led off");  
      digitalWrite(LED,LOW);  
    }  
  }  
}
```

DEC	HEX	OCT	Char
97	61	141	a
98	62	142	b



App Inventor 응용

Section 2. 음성인식 응용

- 음성 명령으로 LED제어를 하는 방법
- Section1에서 텍스트로 전송
- 여기서는 숫자로 전송한다.

App Inventor 응용

● 디자이너 영역

Viewer

☐ Display hidden components in Viewer

Screen1

not connected

B/T 연결

LED ON

LED OFF

Voice Order

B/T 연결해제

Non-visible components

SpeechRecognizer1

BluetoothClient1

Components

Screen1

Label1

ListPicker1

Button1

Button2

Button3

Button4

SpeechRecognizer1

BluetoothClient1

Rename

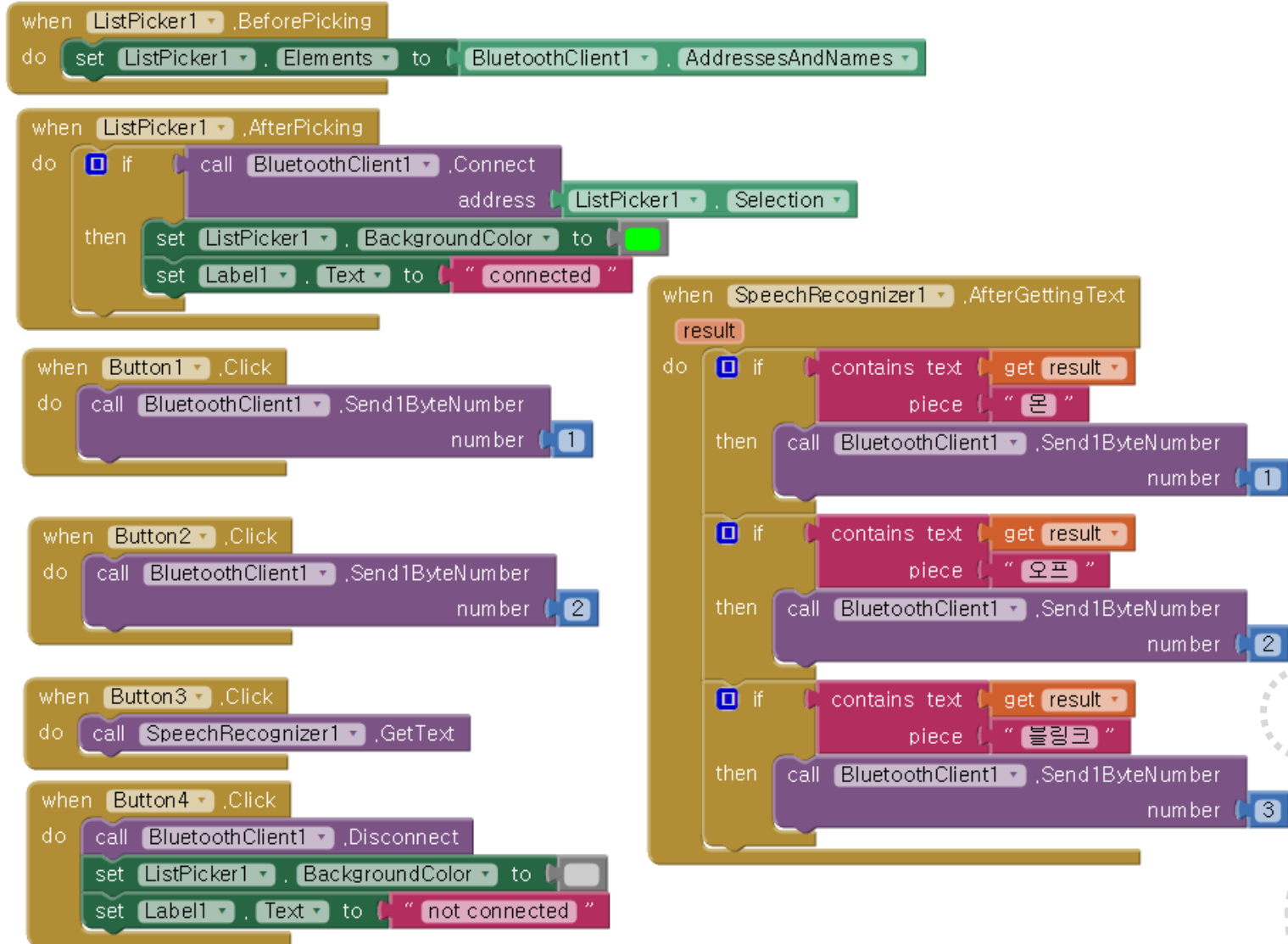
Delete

Media

Upload File ...

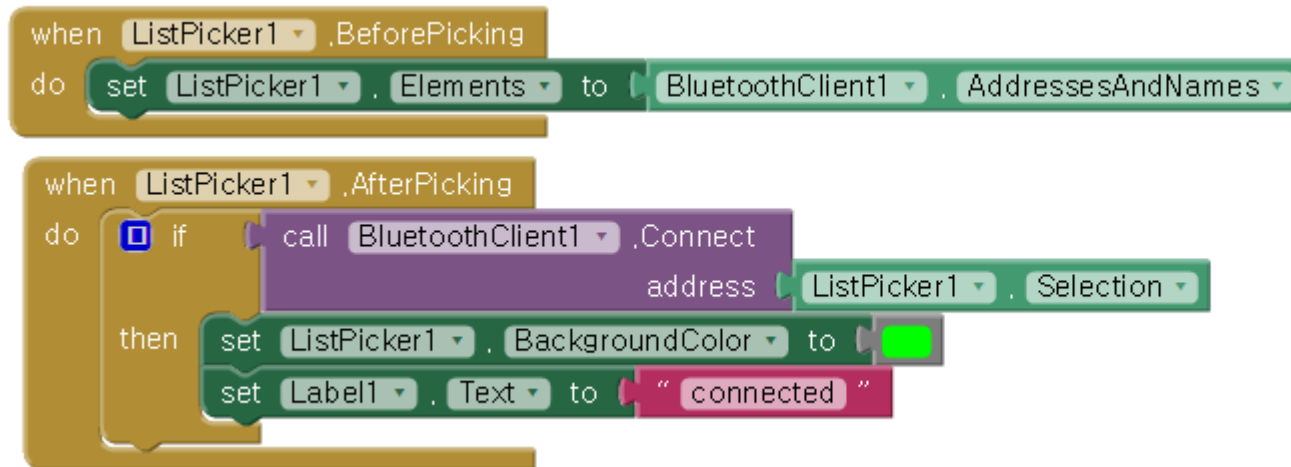
App Inventor 응용

● 음성인식의 전체 블록도



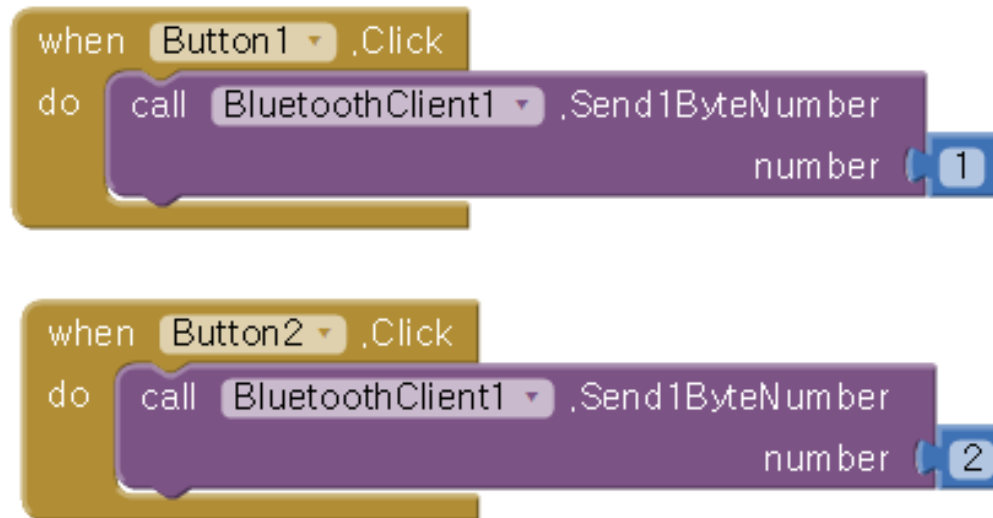
App Inventor 응용

- 아래의 코드는 위에 설명했듯이 연결이 되면 색으로 알려주는 코드입니다. 또 이번에는 문자도 변경되는 것을 추가한다.
- 이처럼 사용자가 생각하는데로 블록만 추가해서 사용할 수 있는것을 확인 할 수 있다.



App Inventor 응용

- 이 블록은 B/T로 1Byte의 숫자로 넘겨주는 블록을 사용해서 바로 인식한다.
- 위에서는 문자로 넘겨줘서 ASCII문자표와 대조를 하였지만 간단하게 B/T를 실행하고자 할 때는 이와 같이 숫자로 넘겨주면 매우 편하다.



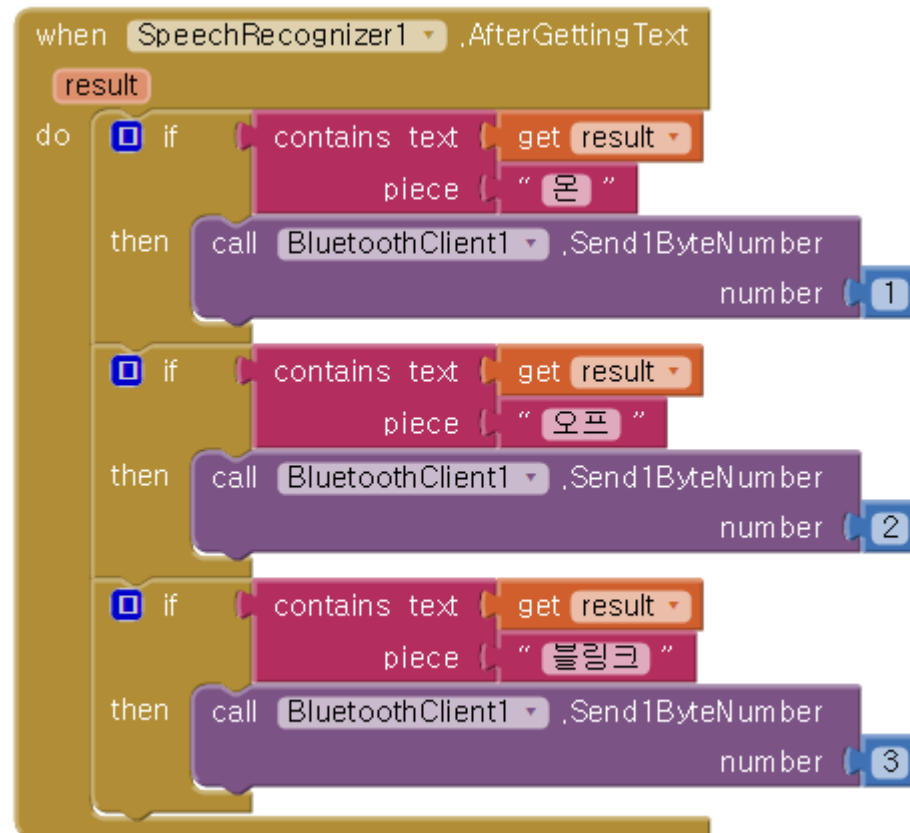
App Inventor 응용

- 음성인식의 시작단계
- SpeechRecognizer는 구글의 음성인식기능을 사용하며, App Inventor에서는 구글이 제공하는 음성인식 기능을 그대로 사용해서 Text로 읽는다.



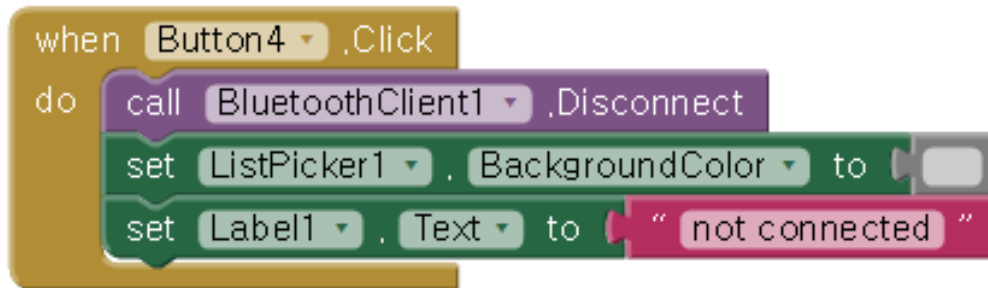
App Inventor 응용

- 음성인식 실행단계
- 음성인식을 통해서 나온 결과값을 result에 저장해서 piece에 지정한 텍스트와 동일하면 실행하는 구조



App Inventor 응용

- 이전과 동일하게 B/T를 해제하고, 버튼의 B/T의 연결상태 색상과 Label의 텍스트를 변경한다.



App Inventor 응용

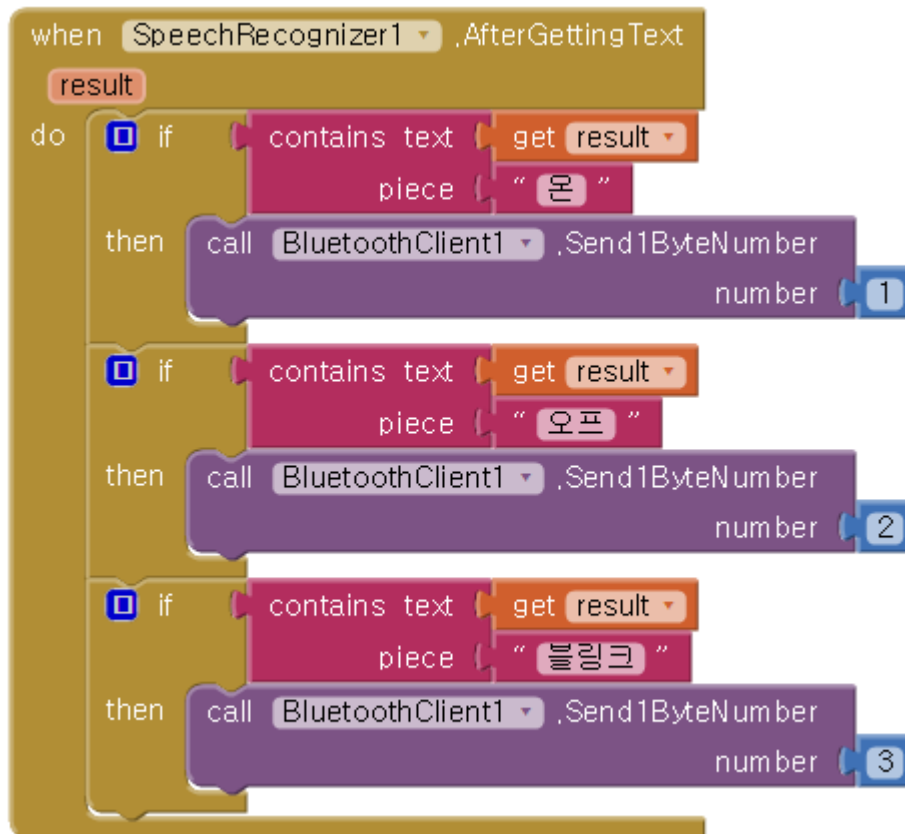
● 아두이노 소스코드

```
#include <SoftwareSerial.h>
|
SoftwareSerial BTSerial(2,3);
byte a=0;
int LED=13;
void setup(){
    Serial.begin(9600);
    Serial.println("Hello!");
    BTSerial.begin(9600);
    pinMode(LED,OUTPUT);
}
```

```
void loop(){
    if(Serial.available()){
        a = Serial.read();
        Serial.print(a);
    }
    switch(a)
    {
        case 1:digitalWrite(LED,HIGH);
        break;
        case 2:digitalWrite(LED,LOW);
        break;
        case 3:digitalWrite(LED,HIGH);
        delay(100);
        digitalWrite(LED,LOW);
        delay(100);
        default:
        break;
    }
}
```

App Inventor 응용

- 아래 2개가 각각 코드에 대응한다. “온”일 경우 case1이 실행이 되며, “오프”일 경우 case2가 실행이 되며, “블링크”일 경우 case3이 실행이 되는 구조이다.



```
switch(a)
{
    case 1: digitalWrite(LED, HIGH);
              break;
    case 2: digitalWrite(LED, LOW);
              break;
    case 3: digitalWrite(LED, HIGH);
              delay(100);
              digitalWrite(LED, LOW);
              delay(100);
    default:
              break;
}
```

Section 3. 센서와 DB응용

- DB는 2가지 종류가 있다.
- Web에 올라가는 TinyWebDB와 기기 자체내에 저장하는 TinyDB 2가지가 있다.
- 이중 TinyDB를 사용해서 B/T를 통해 수신된 값의 카운트에 따라서 DB에 저장한다.

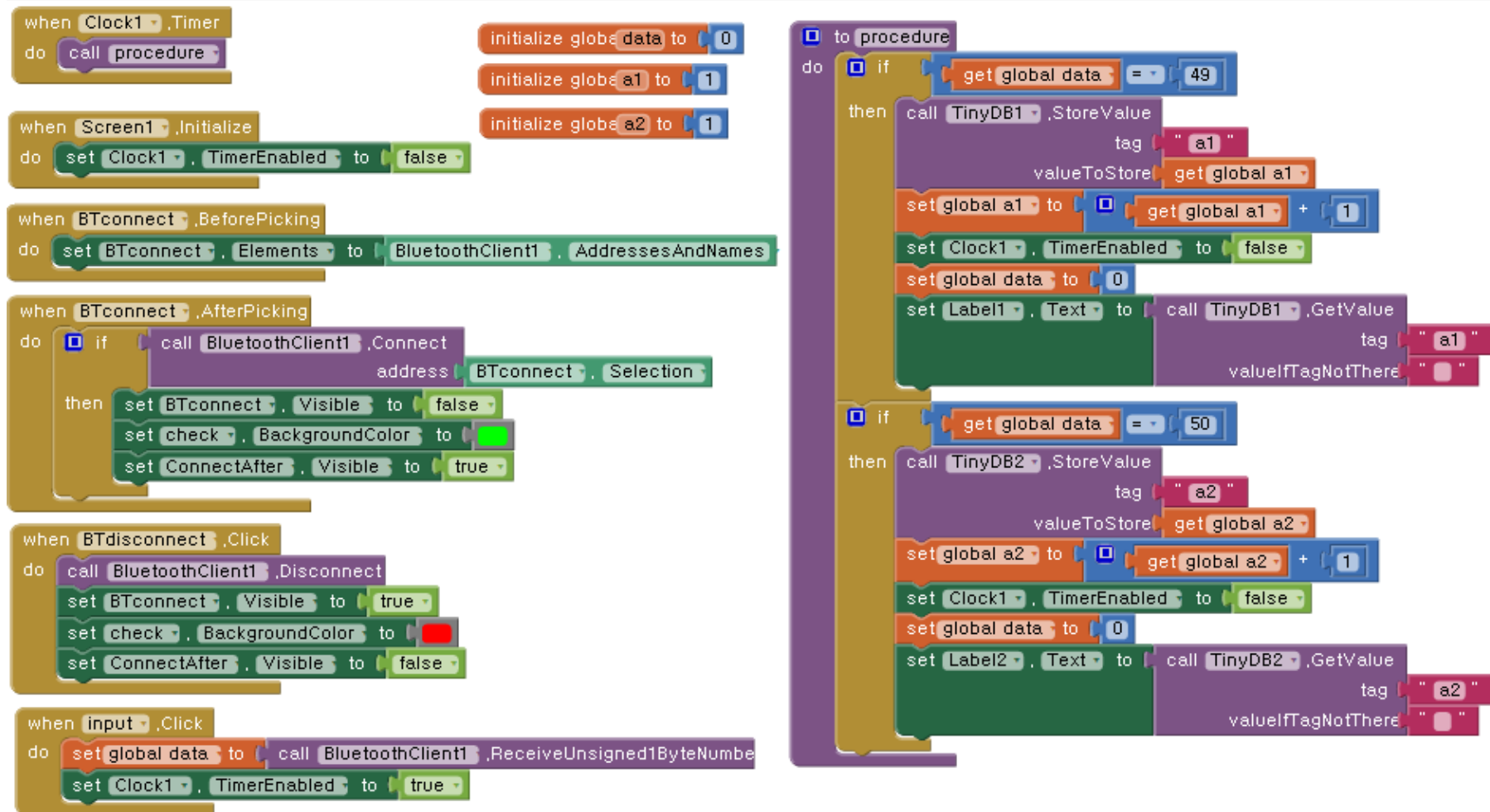
App Inventor 응용

● DB의 디자이너 영역



App Inventor 응용

● DB의 전체블록도



App Inventor 응용

- 데이터의 저장될 장소를 초기화 한다.
- Clock 초기화 한다.

initialize global **data** to **0**

initialize global **a1** to **1**

initialize global **a2** to **1**


when **Screen1** .Initialize

do **set Clock1** . **TimerEnabled** to **false**

App Inventor 응용

- B/T 연결
- 이번에는 조금 독특하게 디자이너 부분에서 미리 안보이게 했다가 연결이 되면 보이게 한다.

```
when BTconnect .BeforePicking  
do set BTconnect . Elements to BluetoothClient1 . AddressesAndNames
```

```
when BTconnect .AfterPicking  
do ☒ if call BluetoothClient1 .Connect  
                                address BTconnect . Selection  
    then set BTconnect . Visible to false  
          set check . BackgroundColor to   
          set ConnectAfter . Visible to true
```

App Inventor 응용

- 다음 그림이 처음 시작화면이고 연결시의 화면이다.
- 마지막 그림이 디자이너 영역 화면이다.
- 이와 같이 하나의 화면을 Visible을 이용하여 꾸밀수 있다.



블루투스 연결

연결상태



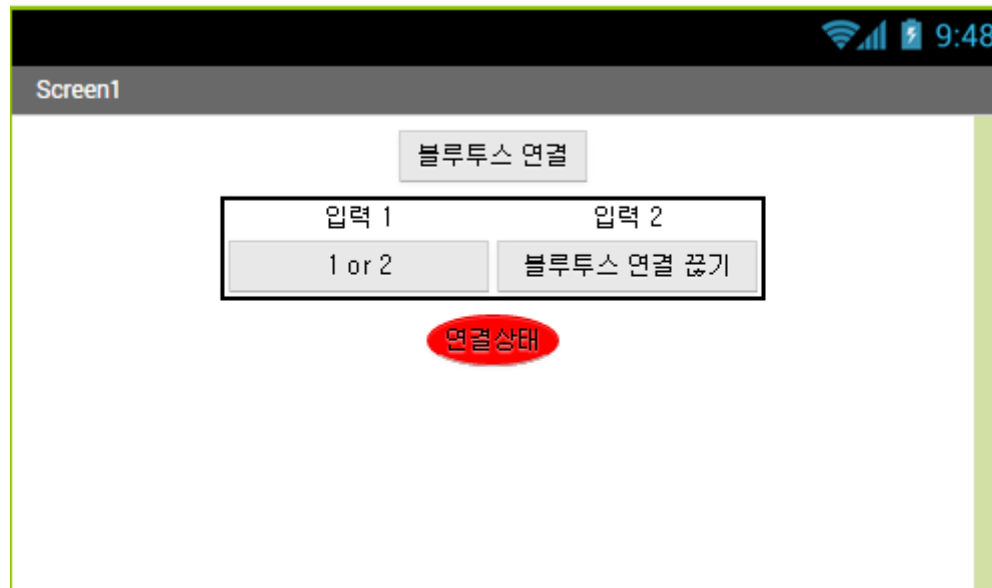
입력 1

입력 2

1 or 2

블루투스 연결 끊기

연결상태



App Inventor 응용

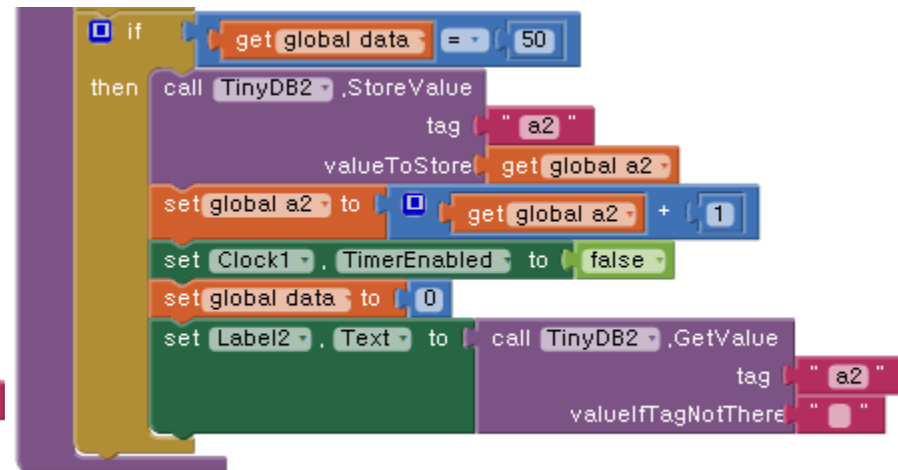
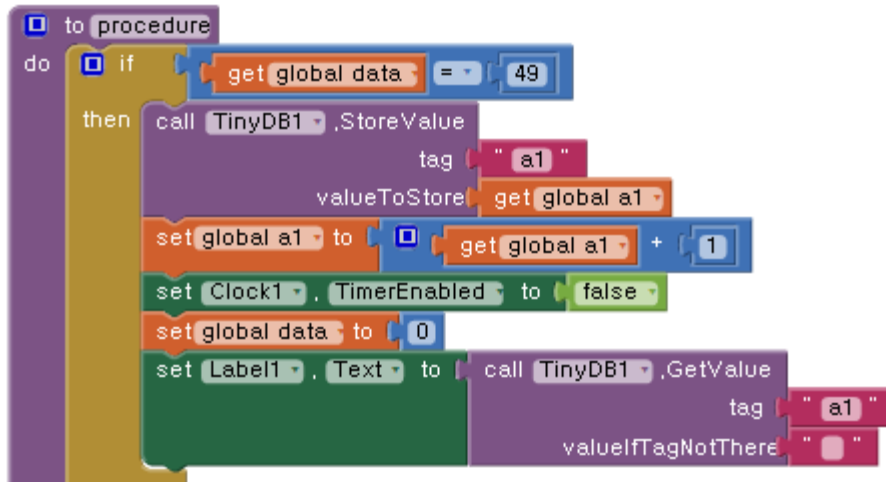
- 이 부분은 마땅한 인터럽트 기능이 없어서 타이머를 사용해 B/T를 통한 입력을 받는 블록이다.
- 본격적인 DB가 존재하는 타이머가 작동한다.

```
when input Click
do
  set global data to call BluetoothClient1 ReceiveUnsigned1ByteNumbe
  set Clock1 TimerEnabled to true
```

```
when Clock1 Timer
do
  call procedure
```

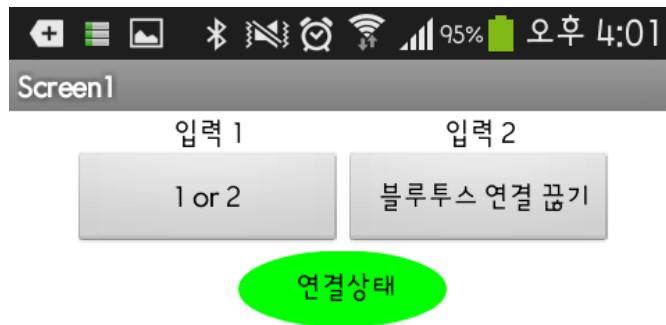
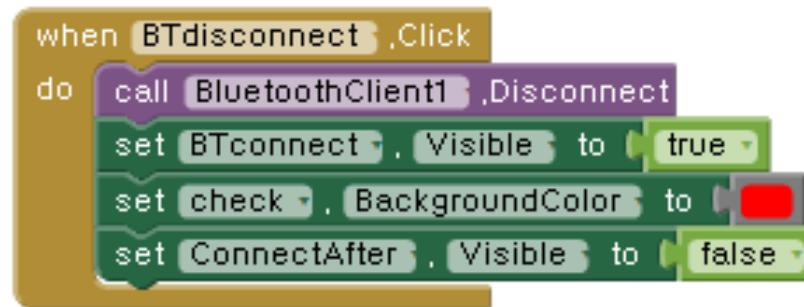
App Inventor 응용

- 이 블록이 타이머가 동작하는 영역이다.
- 타이머가 동작하면서 B/T를 통한 입력이 올 때까지 대기하고, ASCII코드의 byte값이 입력되면 여기서 49=1, 50=2가 입력이되면 각각 a1, a2에 +1을 하여서 DB에 저장하고 Label에 DB에 저장된 a1, a2의 값을 출력을 해주는 블록이다.



App Inventor 응용

- B/T 해제
- 위에서 보여준것 처럼 Visible을 이용해서 초기의 화면으로 복귀한다.



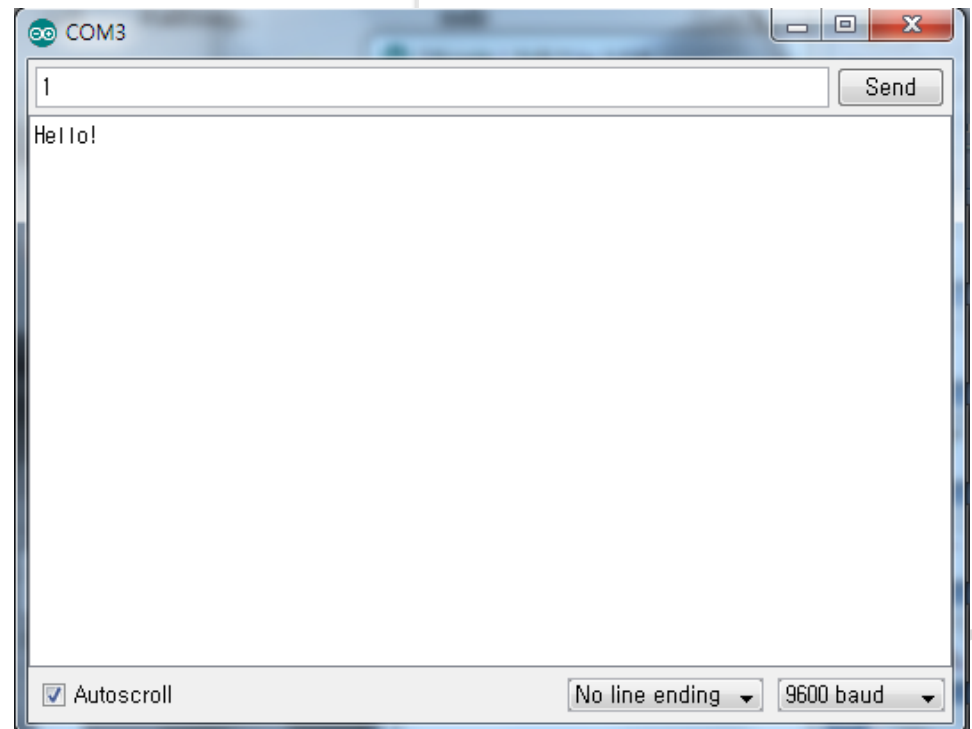
App Inventor 응용

- 아두이노 소스코드
- 앞에서는 앱에서 보낸값만 받았다면
- 이번엔 아두이노쪽에서 값을 보내는 경우이다.

```
#include <SoftwareSerial.h>

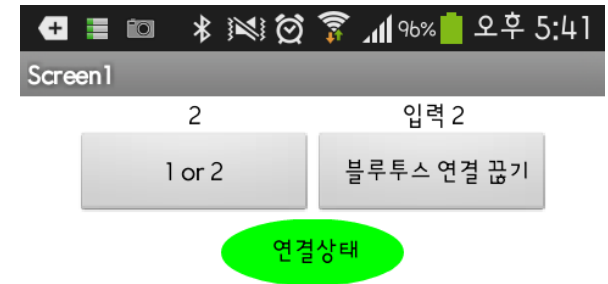
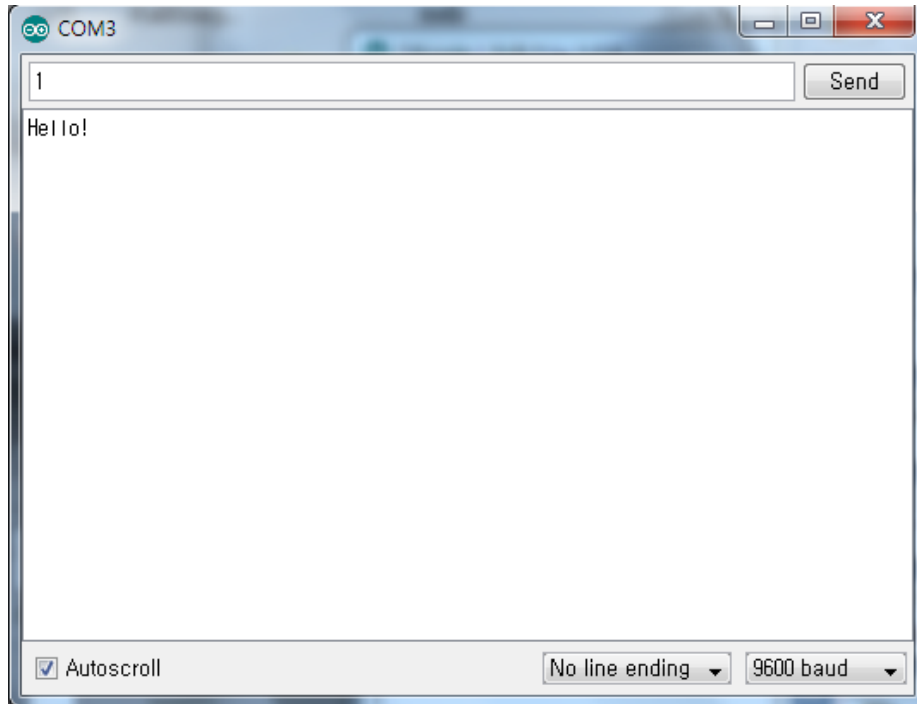
SoftwareSerial BTSerial(2,3);
byte a=0;
void setup(){
  Serial.begin(9600);
  Serial.println("Hello!");
  BTSerial.begin(9600);
}

void loop(){
  if(Serial.available()){
    BTSerial.write(a);
  }
}
```



App Inventor 응용

- 시리얼쪽에서 B/T를 통해 앱으로 신호를 보내면 카운트가 증가하는 것을 확인 할 수 있다.



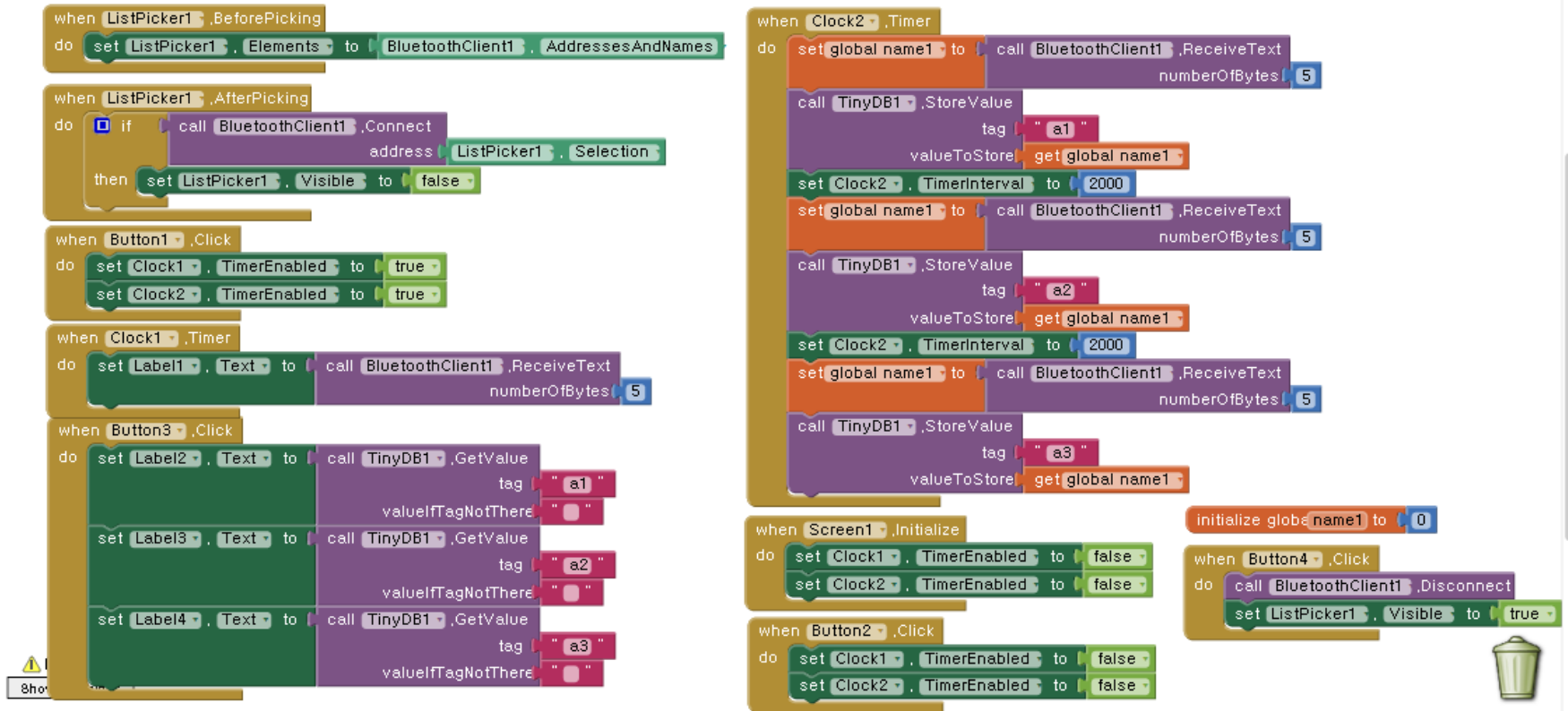
App Inventor 응용

- 빛 센서 데이터를 수집해서 DB에 저장하고 출력하기
- 디자이너 영역



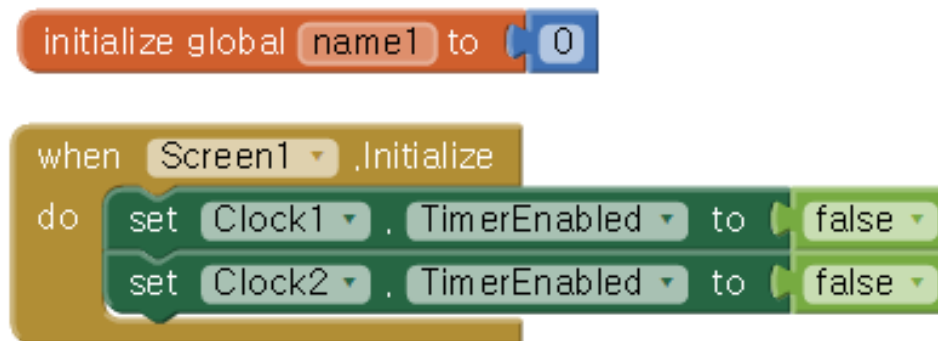
App Inventor 응용

● 전체 블록도



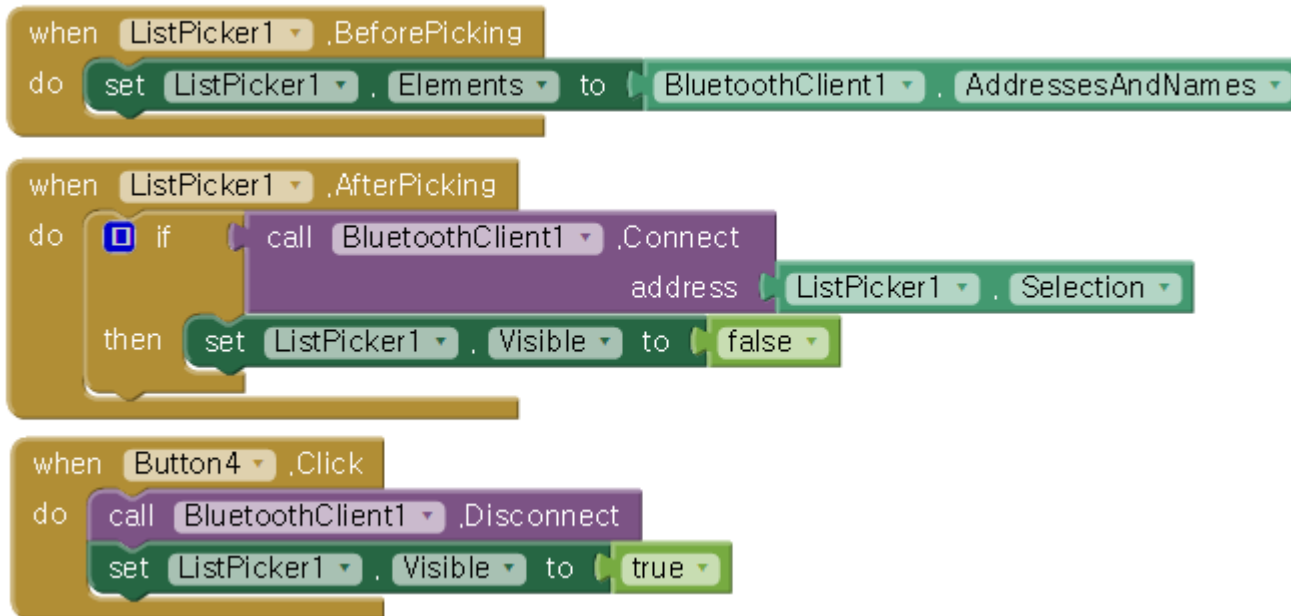
App Inventor 응용

- 블록 초기화
- DB와 타이머의 초기화



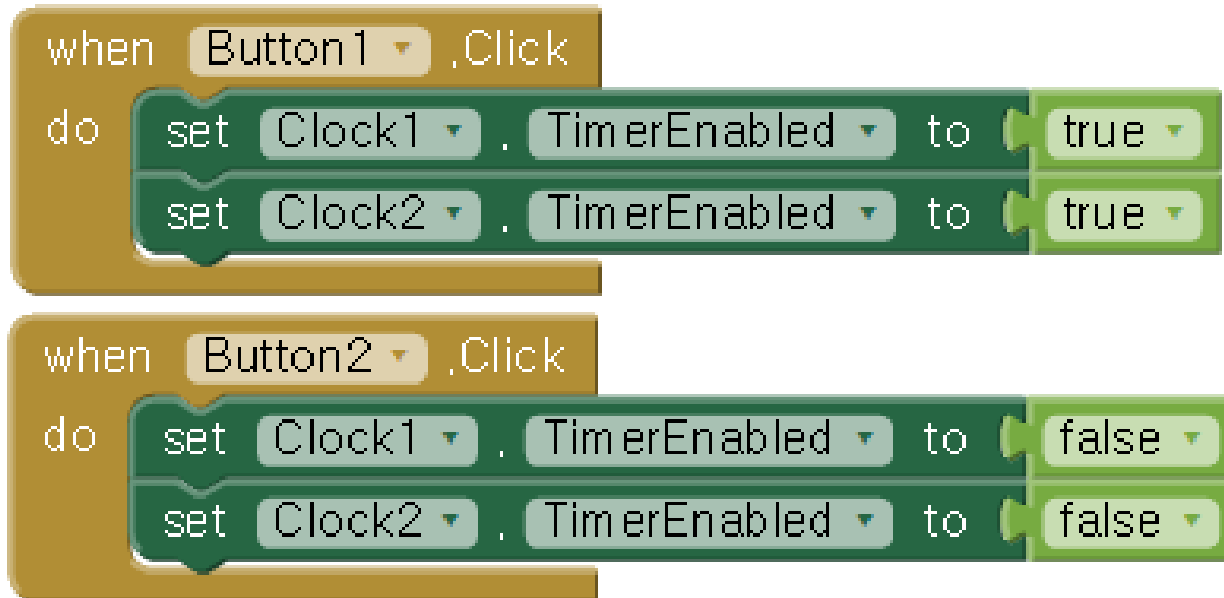
App Inventor 응용

- B/T 블록도
- B/T의 연결과 해제 블록도



App Inventor 응용

- 타이머를 제어하는 블록
- 타이머를 시작하고 종료하는 블록이다.



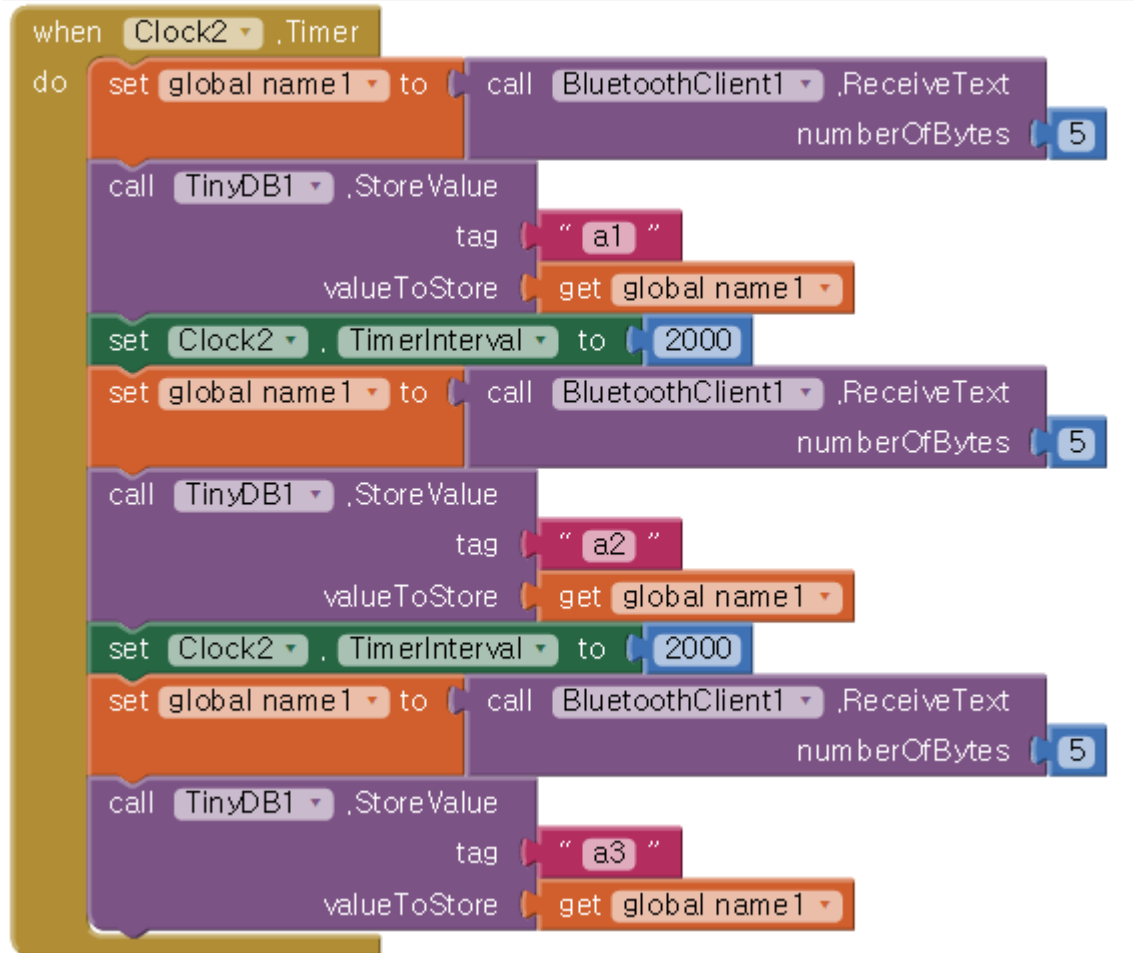
App Inventor 응용

- 아래 블록이 타이머1 현재 데이터를 보내는 값을 확인하기 위해서 따로 만들었다.



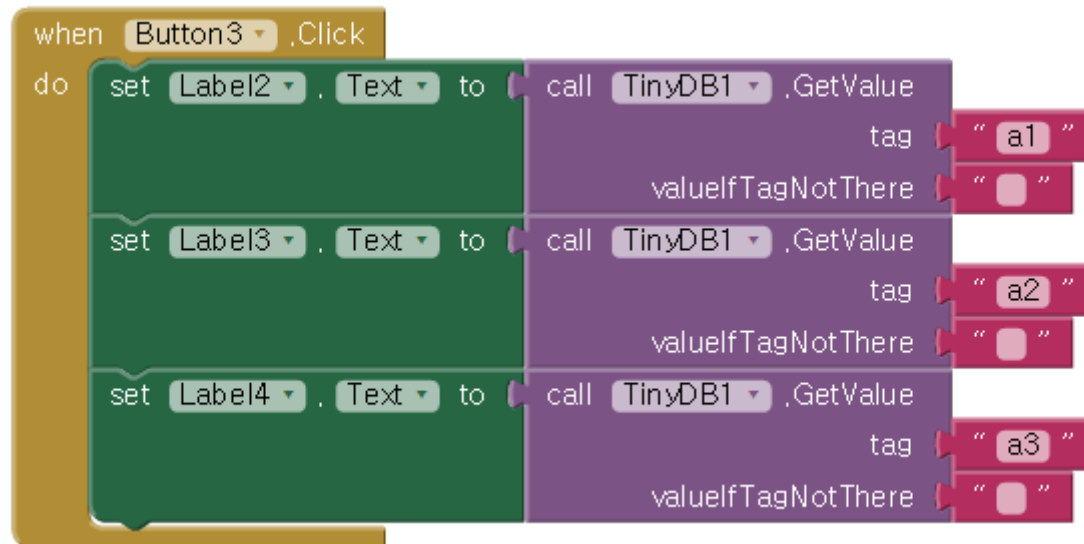
App Inventor 응용

- 타이머2에 의해서 DB에 2초 간격으로 저장이 된다.



App Inventor 응용

- DB에 저장된 값을 출력을 해주는 블록도



● 아두이노의 소스코드

cds

```
#include <SoftwareSerial.h>

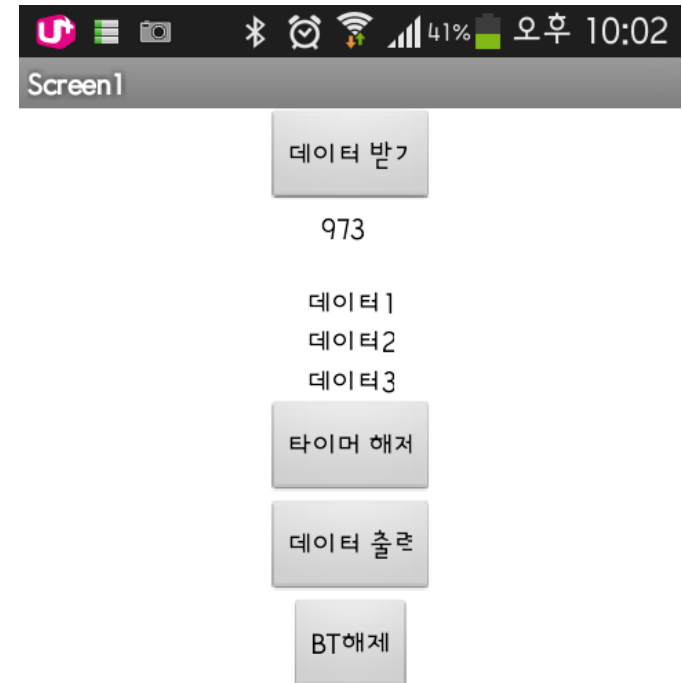
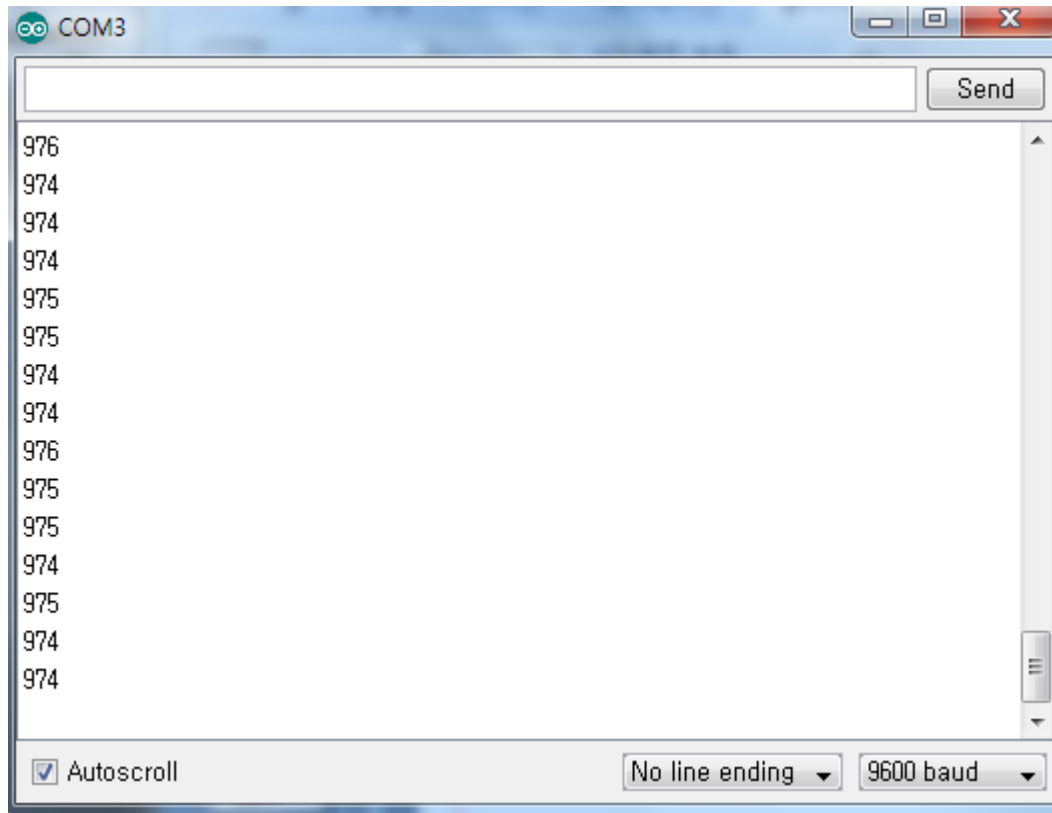
SoftwareSerial BTSerial(2,3);
int analogPin = A5; // 아날로그 핀 정의 : A5
int cds_value;      // CDS 센서 변수

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

void loop()
{
    cds_value=analogRead(analogPin);
    Serial.println(cds_value);
    delay(1000);
    if(Serial.available()>0){
        BTSerial.write(cds_value);
        delay(1000);
    }
}
```

App Inventor 응용

- 아두이노에서 빛 센서의 데이터를 Serial 모니터에 출력
- 출력이 되는 동안 타이머를 동작하면 데이터가 앱에서 표시가 되며, DB에 2초 간격으로 저장



App Inventor 응용

- 데이터 출력을 눌러주면, DB에 저장되어있던 데이터 확인



데이터 받기

973

데이터 1

데이터 2

데이터 3

타이머 해저

데이터 출력

BT해제



데이터 받기

973

97b

972

974

타이머 해저

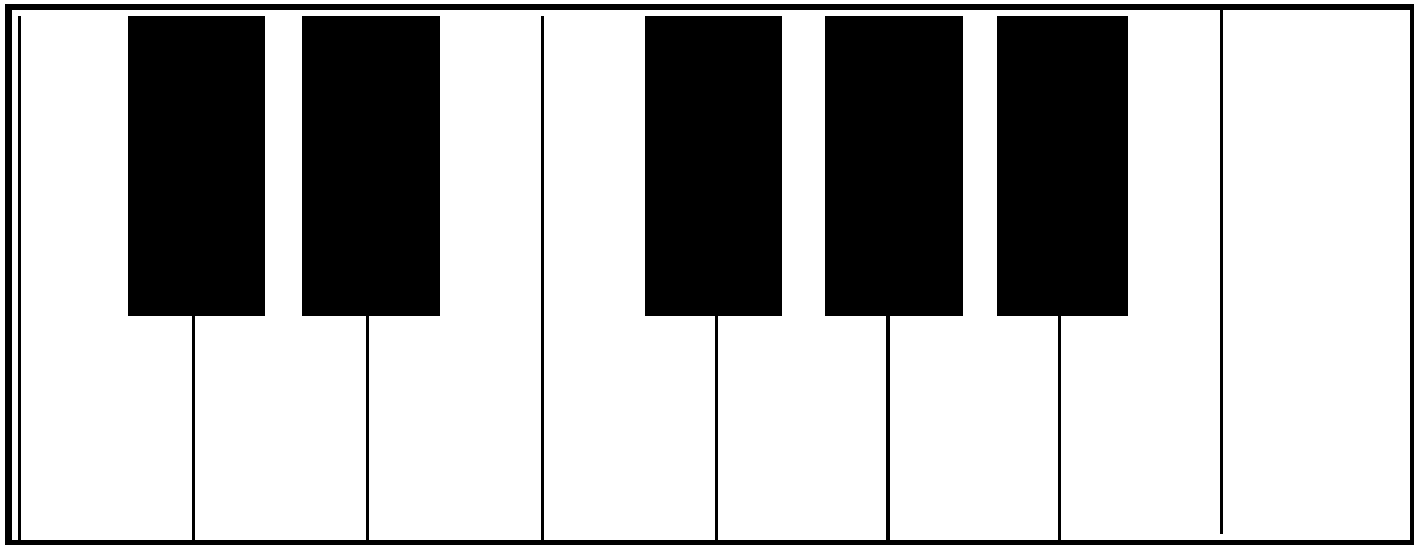
데이터 출력

BT해제

App Inventor 응용

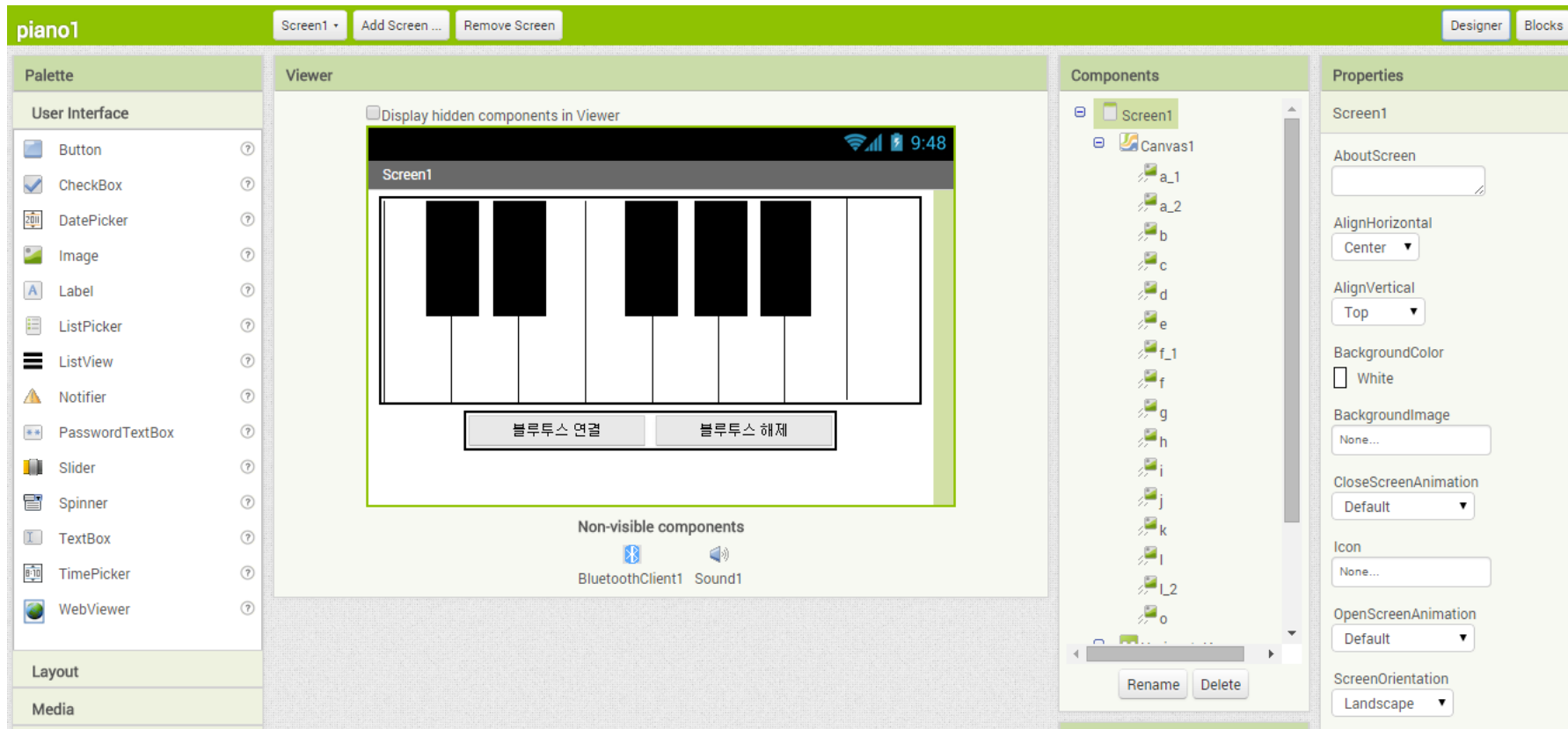
Section 4. MIDI 응용

- 피아노 앱 제작
- 아두이노와 연결된 피에조 센서 제어



App Inventor 응용

● 디자이너 영역



App Inventor 응용

- 전체블록도
- 블록도는 B/T 연결 및 해제
- 각 건반에 해당되는 신호를 B/T로 전송

```
when Button1.Click
do
  call BluetoothClient1.Disconnect
  set ListPicker1.BackgroundColor to [red]
```

```
when ListPicker1.BeforePicking
do
  set ListPicker1.Elements to BluetoothClient1.AddressesAndNames
```

```
when ListPicker1.AfterPicking
do
  if
    call BluetoothClient1.Connect
    address ListPicker1.Selection
  then
    set ListPicker1.BackgroundColor to [blue]
```

```
when a_2.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 1
```

```
when b.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 2
```

```
when o.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 13
```

```
when c.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 3
```

```
when d.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 4
```

```
when e.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 5
```

```
when f.TouchDown
do
  call BluetoothClient1.Send1ByteNumber
  number 6
```



App Inventor 응용

- 전체블록도
- 블록도는 B/T 연결 및 해제
- 각 건반에 해당되는 신호를 B/T로 전송

when **a_2** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **1**

when **b** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **2**

when **c** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **3**

when **d** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **4**

when **e** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **5**

when **f** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **6**

when **g** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **7**

when **h** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **8**

when **i** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **9**

when **j** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **10**

when **k** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **11**

when **l** .TouchDown
do
call **BluetoothClient1** .Send1ByteNumber
number **12**



App Inventor 응용

- 아두이노 소스코드
- B/T로 보내진 데이터를 case문을 통해 실행
- tone 함수를 통해 주파수로 피에조 센서 제어

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(2,3);
byte a=0;

int speakerPin = 12;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440, 466, 494, 523};
//          mid C  C#  D   D#  E   F   F#  G   G#  A   A#  B   C

void setup()
{
    Serial.begin(9600);
    Serial.println("Hello!");
    BTSerial.begin(9600);
}

void loop()
{
    if(BTSerial.available()){
        a = BTSerial.read();
        Serial.println(a);
    }
}
```

```
switch(a)
{
    case 1:
        tone(speakerPin, tones[0]);
        delay(100);
        noTone(speakerPin);
        break;
    case 2:
        tone(speakerPin, tones[1]);
        delay(100);
        noTone(speakerPin);
        break;
    case 3:
        tone(speakerPin, tones[2]);
        delay(100);
        noTone(speakerPin);
        break;
    case 4:
        tone(speakerPin, tones[3]);
        delay(100);
        noTone(speakerPin);
        break;
    case 5:
        tone(speakerPin, tones[4]);
}
```

App Inventor 응용

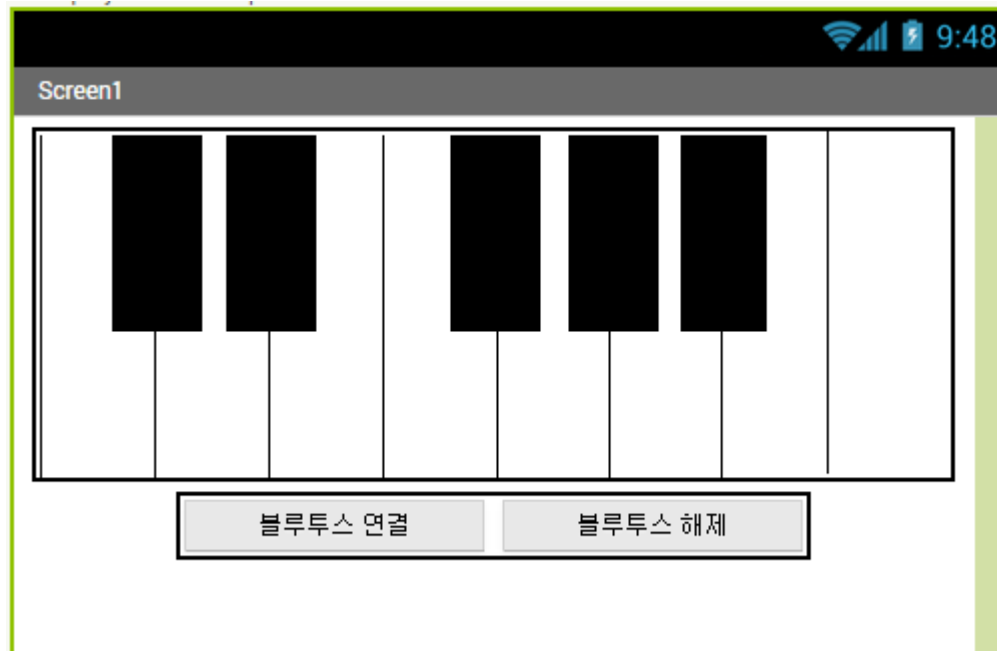
- 아두이노 소스코드
- B/T로 보내진 데이터를 case문을 통해 실행
- tone 함수를 통해 주파수로 피에조 센서 제어

```
case 5:
tone(speakerPin, tones[4]);
delay(100);
noTone(speakerPin);
break;
case 6:
tone(speakerPin, tones[5]);
delay(100);
noTone(speakerPin);
break;
case 7:
tone(speakerPin, tones[6]);
delay(100);
noTone(speakerPin);
break;
case 8:
tone(speakerPin, tones[7]);
```

```
case 11:
tone(speakerPin, tones[10]);
delay(100);
noTone(speakerPin);
break;
case 12:
tone(speakerPin, tones[11]);
delay(100);
noTone(speakerPin);
break;
case 13:
tone(speakerPin, tones[12]);
delay(100);
noTone(speakerPin);
break;
}
}
```

App Inventor 응용

- 앱을 실행하면, 터치를 할 때 마다 아두이노에 연결된 피에조 센서가 동작하는 것을 확인할 수 있다.



Q&A

