

02. 버튼

학습내용

1. 버튼 한 개를 눌렀다 뗐다 해보기(시리얼 모니터 출력)

소스	01._button_one_status.ino
개념	<div>   </div> <p>버튼은 실생활에서 자주 쓰이는 모듈이다. 예) 엘리베이터의 버튼, 스마트폰의 음량 조절 버튼 등</p>
준비물	<div>    </div> <div> <p><컴퓨터></p> <p><USB Micro 5Pin 케이블></p> <p><ET보드></p> </div>

회로 구성	 <p> <컴퓨터> ← <USB Micro 5Pin케이블> <ET보드> </p>
회로 구성	<div data-bbox="301 786 796 1202">  <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="301 1238 796 1655">  <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소스 코드	<pre> int button_red = D6; // 빨강 버튼 void setup(){ pinMode(button_red, INPUT); // 핀 모드 설정 Serial.begin(115200); } </pre>

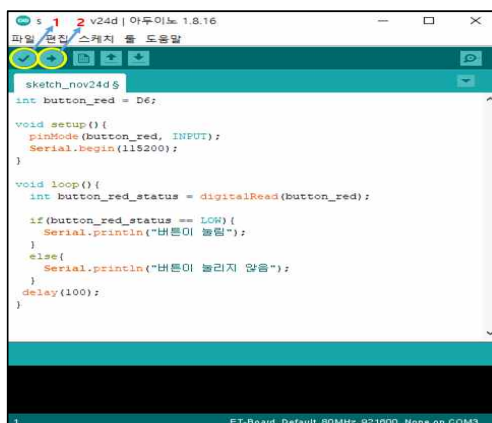
```

void loop(){
    // 빨강 버튼의 값을 저장
    int button_red_status = digitalRead(button_red);
    // 버튼이 눌렸는지 체크 버튼이 눌리면 LOW 눌리지 않으면 HIGH
    if(button_red_status == LOW){
        Serial.println("버튼이 눌림");
    }
    else{
        Serial.println("버튼이 눌리지 않음");
    }
    delay(100);
}

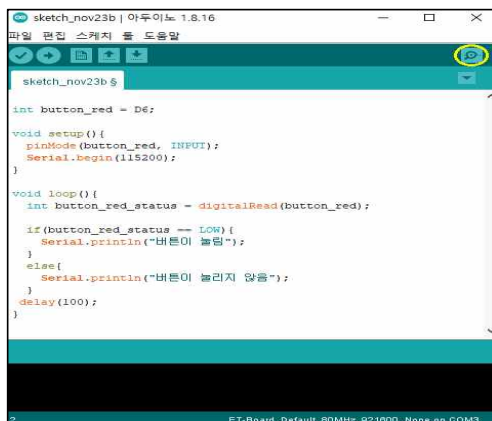
```

// 0.1초 대기

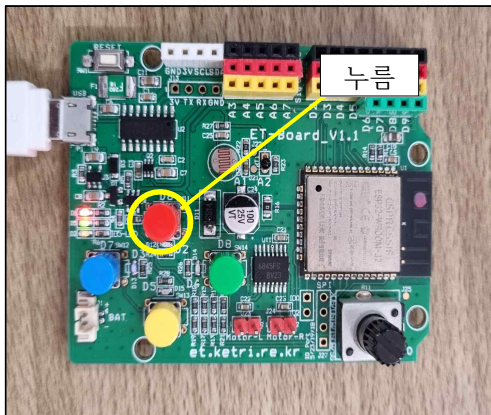
동작
과정



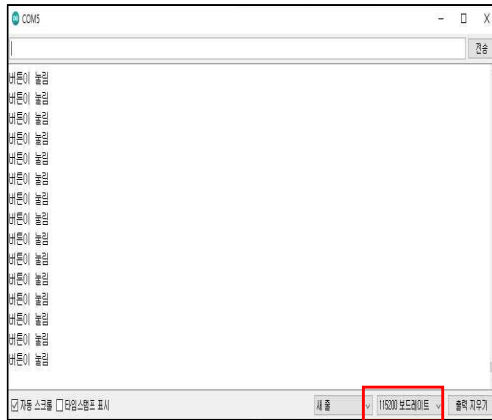
① 아두이노 코드입력 후 상단의 1[확인]아이콘을 클릭하고 완료되면 2[업로드] 아이콘도 클릭해 펌웨어를 ET보드에 업로드합니다.



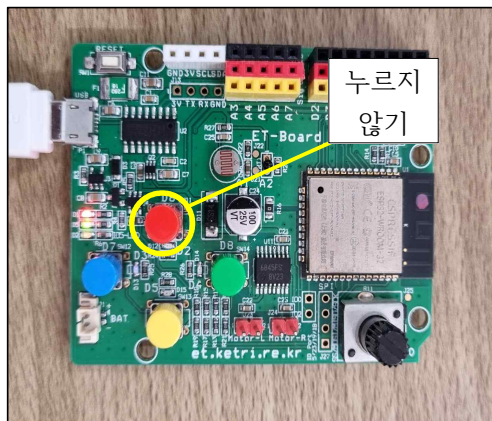
② 업로드까지 마치면 우측 상단 돋보기 모양의 "시리얼모니터"를 클릭해줍니다.



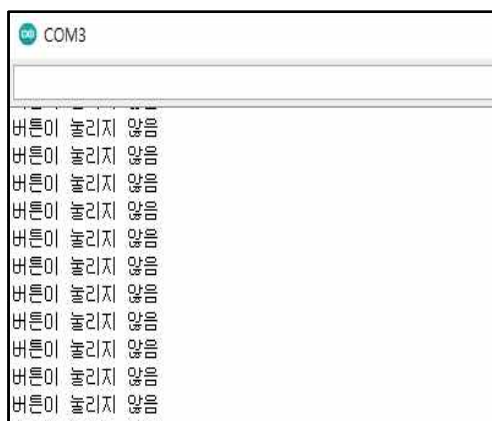
③ 빨간색 버튼을 눌러줍니다.



④ 시리얼모니터에 “버튼이 눌림” 이라 출력됩니다.
(하단의 보드레이트 값을 115200 로 조정합니다.)



⑤ 빨간색 버튼을 누르지 않습니다.



⑥ 시리얼 모니터에 “버튼이 눌리지 않음” 이라 출력됩니다.

참
고
사
항

① 회로

○ button에 대한 참고 링크 :

<https://ko.wikipedia.org/wiki/%EB%B2%84%ED%8A%BC>

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gu04005&logNo=221235520661>

② 소스코드

○ pinMode에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/digital-io/pinmode/>

<http://www.iamamaker.kr/ko/tutorials/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB>

[%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4-pinmode-%ED%95%A8%EC%88%98/](#)

○ Serial.begin에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-begin-%ed%95%a8%ec%88%98/>

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-println-%ed%95%a8%ec%88%98/>

○ digitalRead에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-digitalread-%ed%95%a8%ec%88%98/>

○ delay에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/time/delay/>

02. 버튼

학습내용

2. 버튼 두 개를 눌렀다 뗐다 해보기

소스	02._button_two_status.ino
개념	<div data-bbox="300 882 758 1223" data-label="Image"> </div> <div data-bbox="813 882 1272 1223" data-label="Image"> </div> <p>버튼은 실생활에서 자주 쓰이는 모듈이다. 예) 엘리베이터의 버튼, 스마트폰의 음량 조절 버튼 등</p>
준비물	<div data-bbox="309 1572 598 1713" data-label="Image"> </div> <div data-bbox="397 1771 507 1805" data-label="Caption"> <p><컴퓨터></p> </div> <div data-bbox="668 1536 876 1742" data-label="Image"> </div> <div data-bbox="620 1771 927 1805" data-label="Caption"> <p><USB Micro 5Pin 케이블></p> </div> <div data-bbox="973 1559 1197 1729" data-label="Image"> </div> <div data-bbox="1034 1771 1152 1805" data-label="Caption"> <p><ET보드></p> </div>

회 로 구 성	
회 로 구 성	<div data-bbox="301 786 796 1202">  </div> <div data-bbox="863 954 1353 1032"> <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="301 1238 796 1655">  </div> <div data-bbox="863 1429 1434 1464"> <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소 스 코 드	<pre> int button_red = D6; // 빨강 버튼 int button_blue = D7; // 파랑 버튼 void setup(){ pinMode(button_red, INPUT); // 핀 모드 설정 pinMode(button_blue, INPUT); // 핀 모드 설정 Serial.begin(115200); </pre>


```

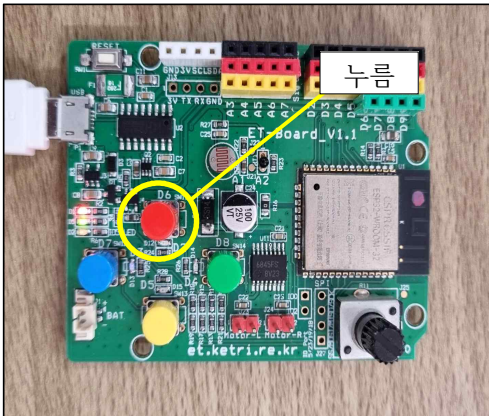
}

void loop(){
  // 빨강 버튼의 값을 저장
  int button_red_status = digitalRead(button_red);
  // 파랑 버튼의 값을 저장
  int button_blue_status = digitalRead(button_blue);

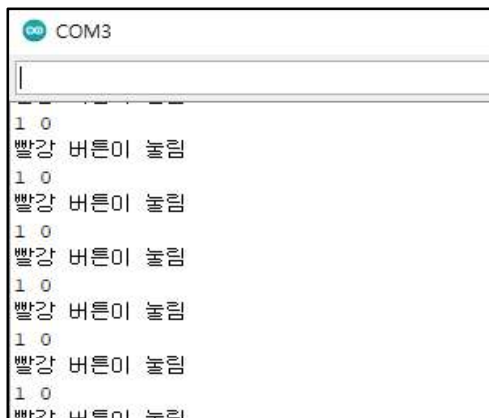
  // 버튼이 눌렸는지 체크 버튼이 눌리면 LOW 눌리지 않으면 HIGH
  if(button_red_status == LOW){
    Serial.println("빨강 버튼이 눌림");
  }
  if(button_blue_status == LOW){
    Serial.println("파랑 버튼이 눌림");
  }
  Serial.print(button_red_status);
  Serial.print(" ");
  Serial.print(button_blue_status);
  delay(100);          // 0.1초 대기
}

```

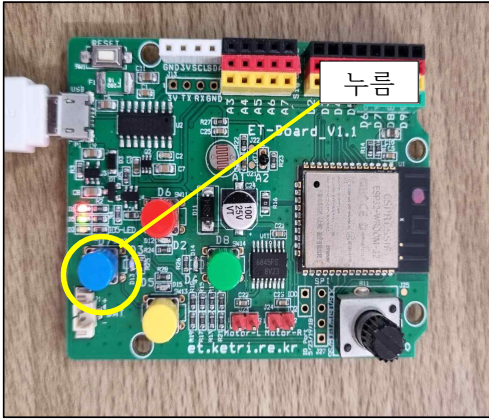
동작
과정



① 빨간색 버튼을 누릅니다.



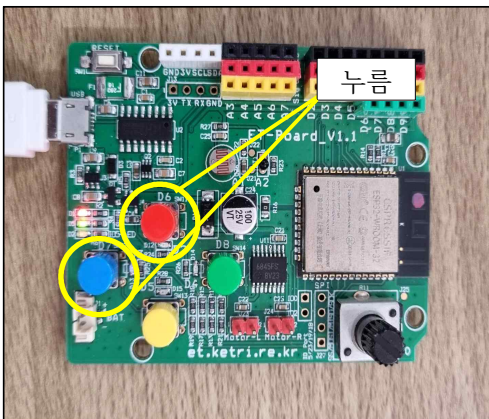
② 시리얼 모니터에 “빨강 버튼이 눌림”이라 출력됩니다.



③ 파랑 버튼을 누릅니다.



④ 시리얼 모니터에 “파랑 버튼이 눌림”이라 출력됩니다.



⑤ 빨강 / 파란 버튼을 동시에 누릅니다.



⑥ 시리얼 모니터에 “빨강 버튼이 눌림”, “파랑 버튼이 눌림” 이라 출력됩니다.

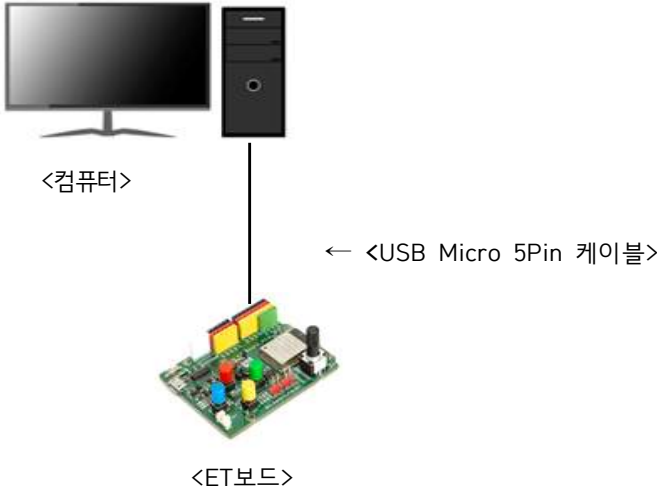

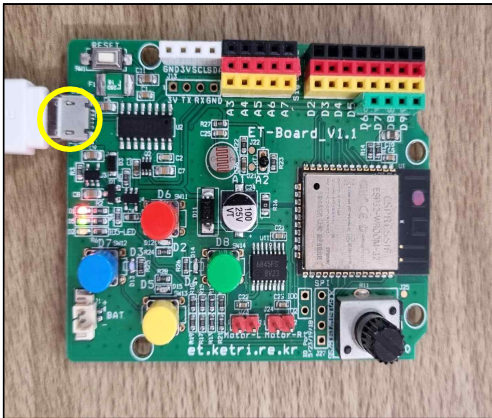
참 고 사 항	<p>① 회로</p> <ul style="list-style-type: none"> ○ button에 대한 참고 링크 : https://ko.wikipedia.org/wiki/%EB%B2%84%ED%8A%BC https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gu04005&logNo=221235520661 <p>② 소스코드</p> <ul style="list-style-type: none"> ○ pinMode에 대한 참고 링크 : https://www.arduino.cc/reference/ko/language/functions/digital-io/pinmode/ http://www.iamamaker.kr/ko/tutorials/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4-pinmode-%ED%95%A8%EC%88%98/ ○ Serial.begin에 대한 참고 링크 : http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-begin-%ed%95%a8%ec%88%98/ http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-println-%ed%95%a8%ec%88%98/ ○ digitalRead에 대한 참고 링크 : http://www.iamamaker.kr/ko/tutorials/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-digitalread-%ed%95%a8%ec%88%98/ ○ delay에 대한 참고 링크 : https://www.arduino.cc/reference/ko/language/functions/time/delay/
---------	--

02. 버튼

학습내용

3. 모든 버튼을 눌렀다 뗐다 해보기(시리얼 모니터 출력)

소스	03._button_all_status.ino
개념	<div data-bbox="296 862 754 1205" data-label="Image"> </div> <div data-bbox="815 862 1273 1205" data-label="Image"> </div> <p>버튼은 실생활에서 자주 쓰이는 모듈이다. 예) 엘리베이터의 버튼, 스마트폰의 음량 조절 버튼 등</p>
준비물	<div data-bbox="309 1538 598 1680" data-label="Image"> </div> <div data-bbox="397 1736 507 1769" data-label="Caption"> <p><컴퓨터></p> </div> <div data-bbox="668 1503 876 1709" data-label="Image"> </div> <div data-bbox="620 1736 927 1769" data-label="Caption"> <p><USB Micro 5Pin 케이블></p> </div> <div data-bbox="973 1525 1193 1697" data-label="Image"> </div> <div data-bbox="1034 1736 1150 1769" data-label="Caption"> <p><ET보드></p> </div>

회로 구성	
회로 구성	<div data-bbox="293 833 788 1249">  </div> <div data-bbox="863 1003 1353 1081"> <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="300 1314 794 1731">  </div> <div data-bbox="863 1507 1434 1543"> <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소스 코드	<pre> int button_red = D6; // 빨강 버튼 int button_blue = D7; // 파랑 버튼 int button_green = D8; // 초록 버튼 int button_yellow = D9; // 노랑 버튼 </pre>

```

void setup(){
    pinMode(button_red, INPUT);           // 핀 모드 설정
    pinMode(button_blue, INPUT);          // 핀 모드 설정
    pinMode(button_green, INPUT);         // 핀 모드 설정
    pinMode(button_yellow, INPUT);        // 핀 모드 설정
    Serial.begin(115200);
}

void loop(){
    int button_red_status = digitalRead(button_red);    // 빨강 버튼의 값을 저장
    int button_blue_status = digitalRead(button_blue);  // 파랑 버튼의 값을 저장
    int button_green_status = digitalRead(button_green); // 초록 버튼의 값을 저장
    int button_yellow_status = digitalRead(button_yellow); // 노랑 버튼의 값을 저장

    // 빨강 버튼이 눌렸는지 체크
    if(button_red_status == LOW){
        Serial.println("빨간버튼이 눌림");
    }

    // 파랑 버튼이 눌렸는지 체크
    if(button_blue_status == LOW){
        Serial.println("파란버튼이 눌림");
    }

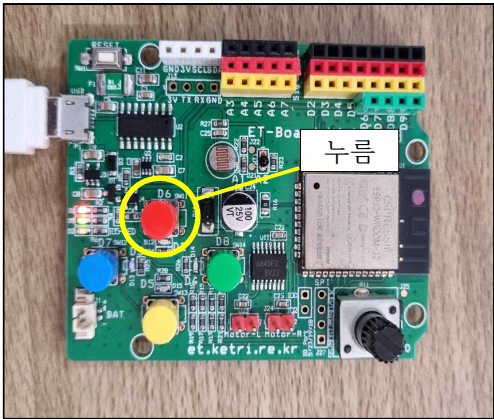
    // 초록 버튼이 눌렸는지 체크
    if(button_green_status == LOW){
        Serial.println("초록버튼이 눌림");
    }

    // 노랑 버튼이 눌렸는지 체크
    if(button_yellow_status == LOW){
        Serial.println("노란버튼이 눌림");
    }

    // 시리얼 플롯를 보기 위한 코드입니다.
    Serial.print(button_blue_status);           // 파랑 버튼 상태 체크
    Serial.print(" ");
    Serial.print(button_red_status);            // 빨강 버튼 상태 체크
    Serial.print(" ");
    Serial.print(button_green_status);          // 초록 버튼 상태 체크
    Serial.print(" ");
    Serial.println(button_yellow_status);        // 노랑 버튼 상태 체크
}

```

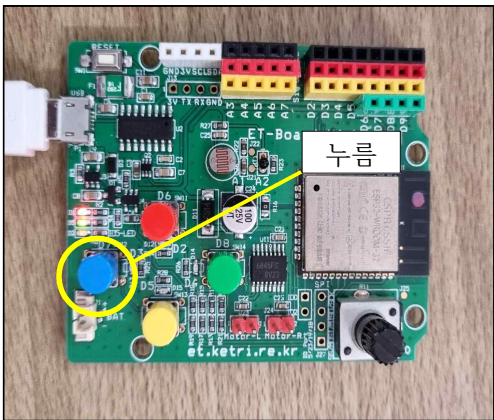
```
delay(100);  
}
```



① 빨강 버튼을 누릅니다.



② 시리얼 모니터에 “빨간버튼이 눌림”이라 출력됩니다.

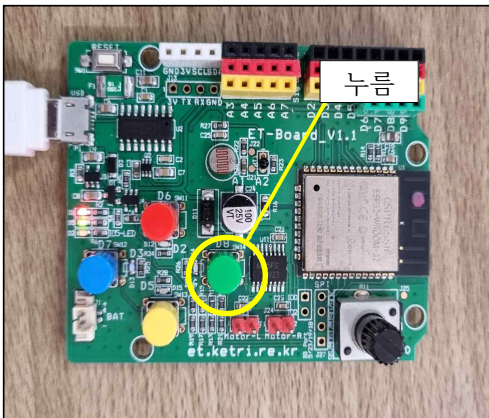


③ 파랑 버튼을 누릅니다.

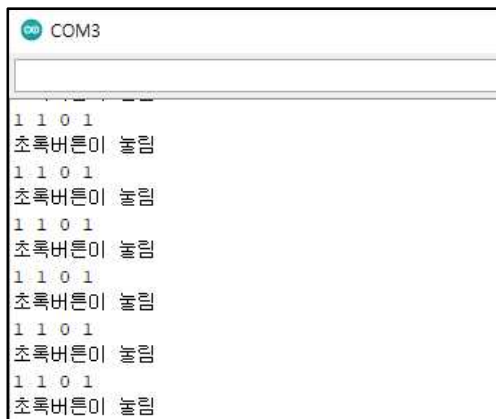
동 작
과 정



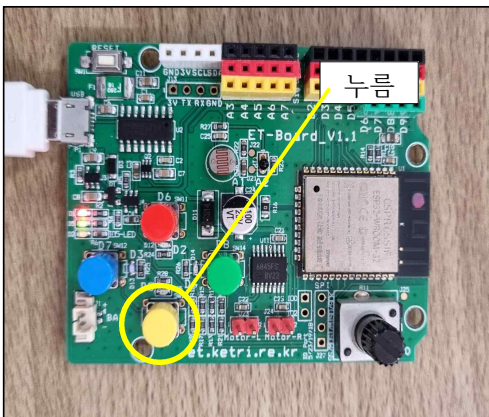
④ 시리얼 모니터에 “파란버튼이 눌림”이라 출력됩니다.



⑤ 초록 버튼을 누릅니다.



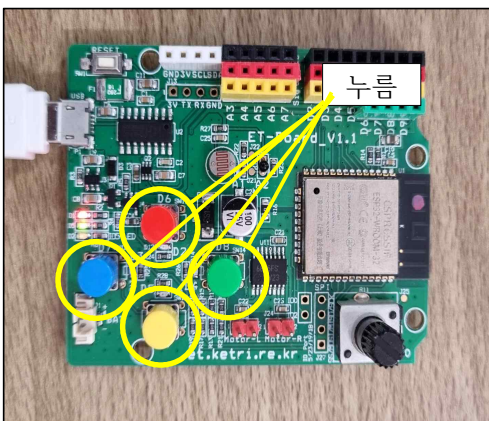
⑥ 시리얼 모니터에 “초록버튼이 눌림”이라 출력됩니다.



⑦ 노랑 버튼을 누릅니다.



⑧ 시리얼 모니터에 “노란버튼이 눌림”이라 출력됩니다.



⑨ 모든 버튼을 누릅니다.



⑩ 시리얼 모니터에 “빨간버튼이 눌림”, “파란버튼이 눌림”, “초록버튼이 눌림”, “노란버튼이 눌림”이라 출력 됩니다.

① 회로

○ button에 대한 참고 링크 :

참고
사항

<https://ko.wikipedia.org/wiki/%EB%B2%84%ED%8A%BC>

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gu04005&logNo=221235520661>

② 소스코드

○ pinMode에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/digital-io/pinmode/>

<http://www.iamamaker.kr/ko/tutorials/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4-pinmode-%ED%95%A8%EC%88%98/>

○ Serial.begin에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-begin-%ed%95%a8%ec%88%98/>

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-println-%ed%95%a8%ec%88%98/>

○ digitalRead에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-digitalread-%ed%95%a8%ec%88%98/>

○ delay에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/time/delay/>

02. 버튼

학습내용

4. 버튼과 같은 색의 LED 켜다 꺼보기

소스	04._button_led_control.ino
개념	<div data-bbox="300 875 756 1216" data-label="Image"> </div> <div data-bbox="818 875 1275 1216" data-label="Image"> </div> <p>버튼은 실생활에서 자주 쓰이는 모듈이다. 예) 엘리베이터의 버튼, 스마트폰의 음량 조절 버튼 등</p>
준비물	<div data-bbox="312 1547 600 1688" data-label="Image"> </div> <div data-bbox="399 1747 509 1780" data-label="Caption"> <p><컴퓨터></p> </div> <div data-bbox="671 1514 879 1720" data-label="Image"> </div> <div data-bbox="622 1747 928 1780" data-label="Caption"> <p><USB Micro 5Pin 케이블></p> </div> <div data-bbox="978 1536 1198 1709" data-label="Image"> </div> <div data-bbox="1037 1747 1153 1780" data-label="Caption"> <p><ET보드></p> </div>

회로 구성	
회로 구성	<div data-bbox="303 831 798 1247">  <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="303 1296 798 1713">  <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소스 코드	<pre> int led_red = D2; // 빨강 LED int led_blue = D3; // 파랑 LED int led_green = D4; // 초록 LED int led_yellow = D5; // 노랑 LED </pre>

```

int button_red    = D6;           // 빨강 버튼
int button_blue   = D7;           // 파랑 버튼
int button_green  = D8;           // 초록 버튼
int button_yellow = D9;           // 노랑 버튼

int button_red_value    = 0;       // 빨강 버튼의 상태
int button_red_old_value = 1;       // 빨강 버튼의 이전 상태
int led_red_status      = 0;       // 빨강 LED 상태

int button_blue_value    = 0;       // 파랑 버튼의 상태
int button_blue_old_value = 1;       // 파랑 버튼의 이전 상태
int led_blue_status      = 0;       // 파랑 LED 상태

int button_green_value    = 0;       // 초록 버튼의 상태
int button_green_old_value = 1;       // 초록 버튼의 이전 상태
int led_green_status      = 0;       // 초록 LED 상태

int button_yellow_value    = 0;       // 노랑 버튼의 상태
int button_yellow_old_value = 1;       // 노랑 버튼의 이전 상태
int led_yellow_status      = 0;       // 노랑 LED 상태

void setup(){
    pinMode(led_red, OUTPUT);         // 핀 모드 설정
    pinMode(led_blue, OUTPUT);        // 핀 모드 설정
    pinMode(led_green, OUTPUT);       // 핀 모드 설정
    pinMode(led_yellow, OUTPUT);      // 핀 모드 설정

    pinMode(button_red, INPUT);        // 핀 모드 설정
    pinMode(button_blue, INPUT);       // 핀 모드 설정
    pinMode(button_green, INPUT);      // 핀 모드 설정
    pinMode(button_yellow, INPUT);     // 핀 모드 설정
}

void loop(){
    // 빨강 버튼 으로 빨간 LED 제어
    button_red_value = digitalRead(button_red);

    if(button_red_value == 0 && button_red_old_value == 1){
        led_red_status = 1- led_red_status;
    }

```

```

button_red_old_value = button_red_value;
if(led_red_status == 1){
    digitalWrite(led_red, HIGH);
}else{
    digitalWrite(led_red, LOW);
}

// 파랑 버튼 으로 파랑 LED 제어
button_blue_value = digitalRead(button_blue);

if(button_blue_value == 0 && button_blue_old_value == 1){
    led_blue_status = 1- led_blue_status;
}
button_blue_old_value = button_blue_value;
if(led_blue_status == 1){
    digitalWrite(led_blue, HIGH);
}else{
    digitalWrite(led_blue, LOW);
}

// 초록 버튼 으로 초록 LED 제어
button_green_value = digitalRead(button_green);

if(button_green_value == 0 && button_green_old_value == 1){
    led_green_status = 1- led_green_status;
}
button_green_old_value = button_green_value;
if(led_green_status == 1){
    digitalWrite(led_green, HIGH);
}else{
    digitalWrite(led_green, LOW);
}

// 노랑 버튼 으로 노랑 LED 제어
button_yellow_value = digitalRead(button_yellow);

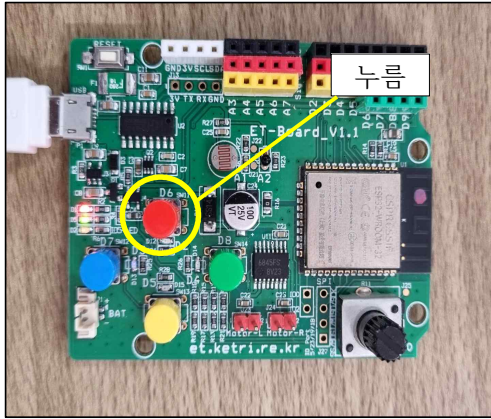
if(button_yellow_value == 0 && button_yellow_old_value == 1){
    led_yellow_status = 1- led_yellow_status;
}
button_yellow_old_value = button_yellow_value;
if(led_yellow_status == 1){
    digitalWrite(led_yellow, HIGH);
}

```

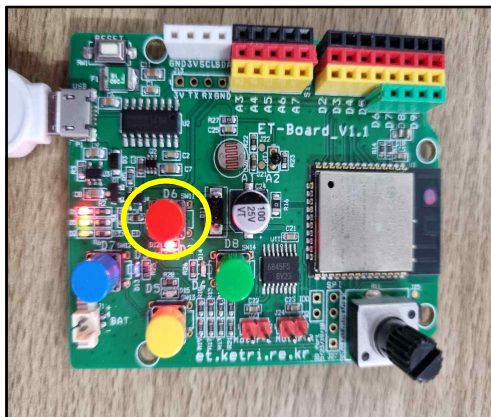
```

}else{
    digitalWrite(led_yellow, LOW);
}
}

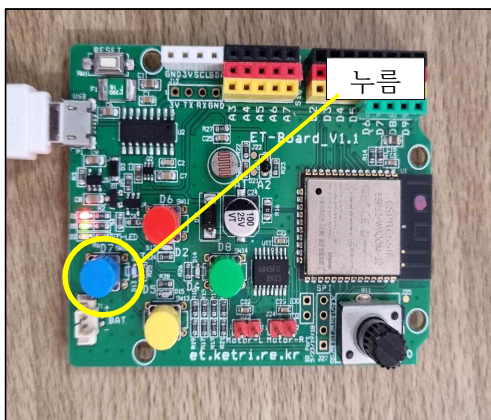
```



① 빨강 버튼을 누릅니다.



② 빨강 LED가 켜집니다.

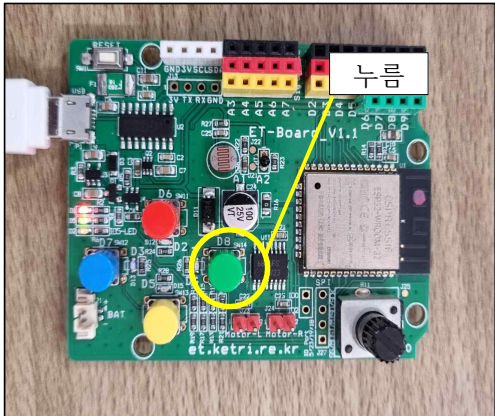


① 파랑 버튼을 누릅니다.

동 작
과 정



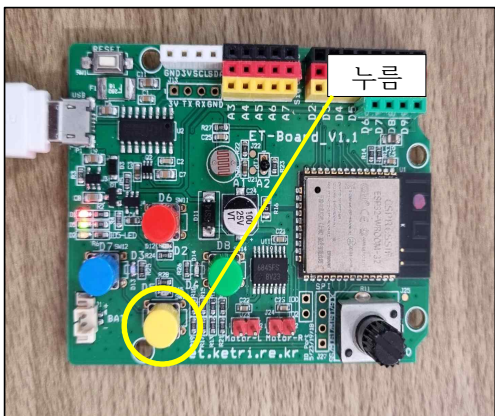
② 파랑 LED가 켜집니다.



① 초록 버튼을 누릅니다.



② 초록 LED가 켜집니다.



① 노랑 버튼을 누릅니다.



② 노랑 LED가 켜집니다.

① 회로

○ button에 대한 참고 링크 :

<https://ko.wikipedia.org/wiki/%EB%B2%84%ED%8A%BC>

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gu04005&logNo=221235520661>

② 소스코드

○ pinMode에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/digital-io/pinmode/>

<http://www.iamamaker.kr/ko/tutorials/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4-pinmode-%ED%95%A8%EC%88%98/>

○ Serial.begin에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-begin-%ed%95%a8%ec%88%98/>

<http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-println-%ed%95%a8%ec%88%98/>

○ digitalWrite에 대한 참고 링크 :

<http://www.iamamaker.kr/ko/tutorials/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-digitalread-%ed%95%a8%ec%88%98/>

○ delay에 대한 참고 링크 :

<https://www.arduino.cc/reference/ko/language/functions/time/delay/>

참 고
사 항

02. 버튼

학습내용

5. 빨강 버튼을 눌러 빨강 LED를 켜보기

소스	05._button_led_control_2.ino
개념	<div>   </div> <p>버튼은 실생활에서 자주 쓰이는 모듈이다. 예) 엘리베이터의 버튼, 스마트폰의 음량 조절 버튼 등</p>
준비물	<div>    </div> <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>

회 로 구 성	
회 로 구 성	<div data-bbox="301 790 796 1207">  </div> <div data-bbox="301 1245 796 1662">  </div> <div data-bbox="863 958 1434 1473"> <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소 스 코 드	<pre> int led_red = D2; // 빨강 LED int button_red = D6; // 빨강 버튼 int button_red_value = 0; // 빨강 버튼의 상태 int button_red_old_value = 1; // 빨강 버튼의 이전 상태 int led_red_status = 0; // 빨강 LED 상태 </pre>

```

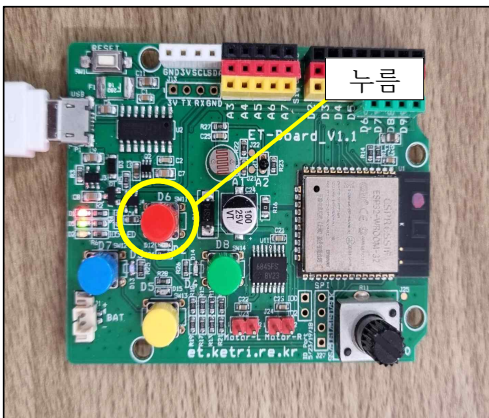
void setup(){
    pinMode(led_red, OUTPUT);           // 핀 모드 설정
    pinMode(button_red, INPUT);         // 핀 모드 설정
}

void loop(){
    // 빨강 버튼 으로 빨강 LED 제어
    button_red_value = digitalRead(button_red);

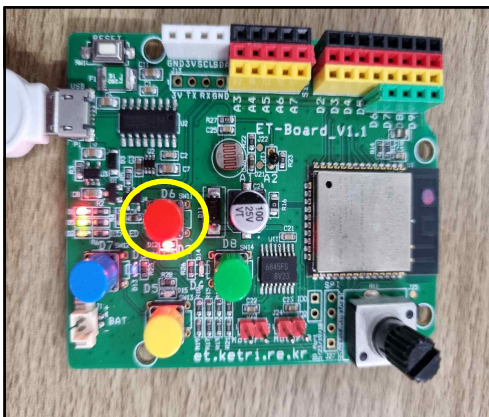
    if(button_red_value == LOW && button_red_old_value == HIGH){
        led_red_status = !led_red_status;
    }
    button_red_old_value = button_red_value;
    if(led_red_status == HIGH){
        digitalWrite(led_red, HIGH);
    }else{
        digitalWrite(led_red, LOW);
    }
}

```

동
작
과
정



① 빨강 버튼을 누릅니다.



② 빨강 LED가 켜집니다.

참 고 사 항	<p>① 회로</p> <ul style="list-style-type: none"> ○ button에 대한 참고 링크 : https://ko.wikipedia.org/wiki/%EB%B2%84%ED%8A%BC https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gu04005&logNo=221235520661 <p>② 소스코드</p> <ul style="list-style-type: none"> ○ pinMode에 대한 참고 링크 : https://www.arduino.cc/reference/ko/language/functions/digital-io/pinmode/ http://www.iamamaker.kr/ko/tutorials/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4-pinmode-%ED%95%A8%EC%88%98/ ○ Serial.begin에 대한 참고 링크 : http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-begin-%ed%95%a8%ec%88%98/ http://www.iamamaker.kr/ko/tutorials/arduino/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-serial-println-%ed%95%a8%ec%88%98/ ○ digitalRead에 대한 참고 링크 : http://www.iamamaker.kr/ko/tutorials/%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%a0%88%ed%8d%bc%eb%9f%b0%ec%8a%a4-digitalread-%ed%95%a8%ec%88%98/ ○ delay에 대한 참고 링크 : https://www.arduino.cc/reference/ko/language/functions/time/delay/
---------	--