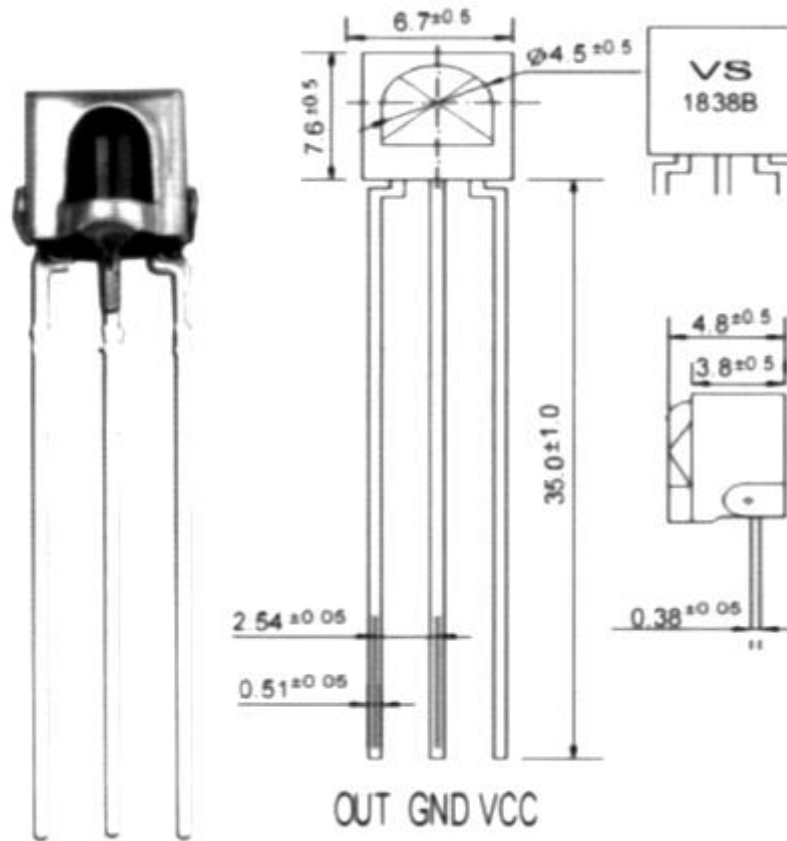


---

# 리모컨

# 리모컨



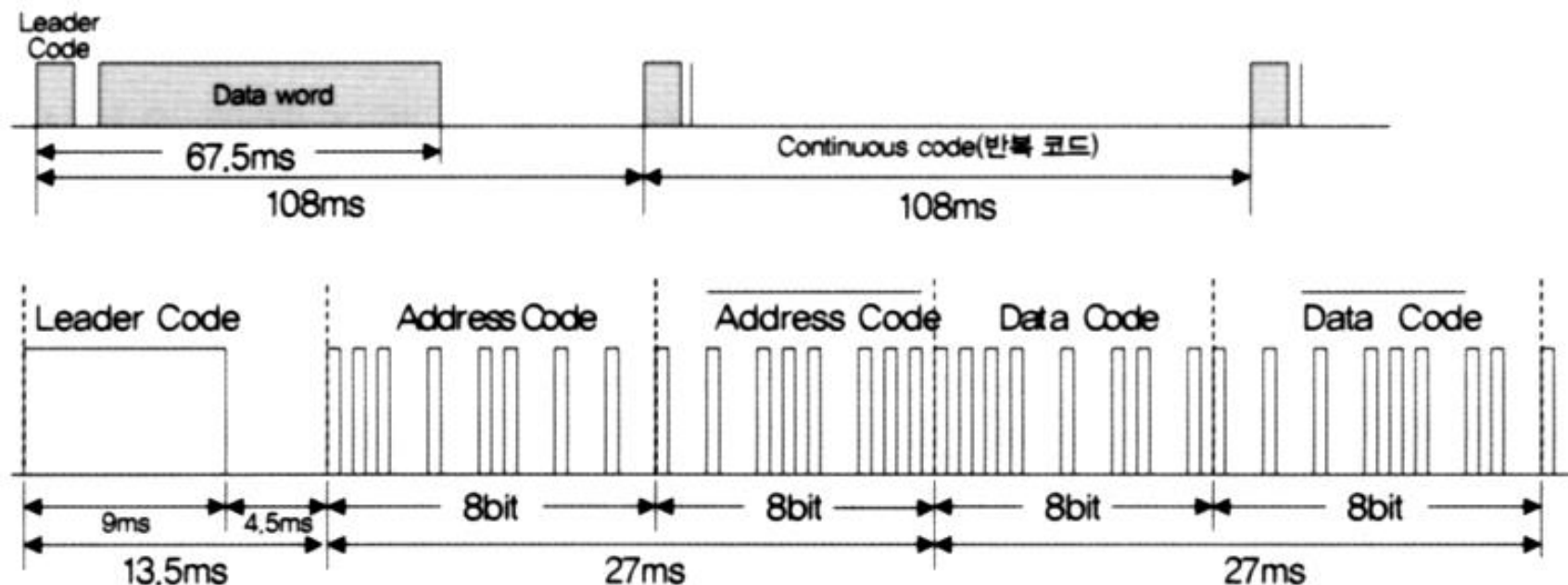
(a) IR Receiver



(b) 리모콘

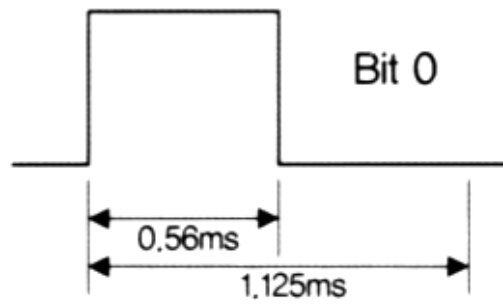
# 리모컨

Address Code(8비트) + Address Code에 대한 반전값(1의 보수, 8비트)  
+ Data Code(8비트) + Data Code에 대한 반전값(1의 보수, 8비트)

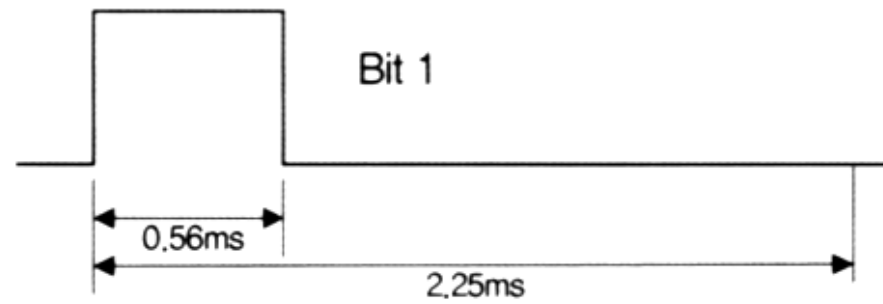


NEC 송신 코드

# 리모컨



(a) 비트 0



(b) 비트 1

NEC 방식에서의 비트 0과 1

# 리모컨

## ❖ 리모컨 라이브러리

o <https://github.com/z3t0/Arduino-IRremote>

변수	설명	비고
<code>decode_type_t decode_type;</code>	리모콘 신호 형식	UNKNOWN, NEC, SONY, RC5, ..
<code>unsigned long value;</code>	디코딩 데이터	클래스 변수
<code>int bits;</code>	디코드된 비트 수	클래스 변수

멤버 함수명	설명	사용 예
<code>IRrecv 객체명(int recvpin)</code>	적외선 수신센서 연결핀 지정	예) <code>IRrecv ir(2);</code>
<code>int decode(decode_results *rfts);</code>	적외선 수신기로부터 데이터 수신하여 rft에 저장	<code>decode_results rft;</code> <code>ir.decode(&amp;rft);</code>
<code>void enableIRin( );</code>	리모컨의 입력을 시작하며 setup() 함수에서 사용.	<code>ir.enableIRin();</code>
<code>void resume( );</code>	디코딩 종료된 후 다음 값 수신하고, 디코드 데이터 사용 후 실행	<code>ir.decode();</code> ..... <code>ir.resume();</code>

# 리모컨

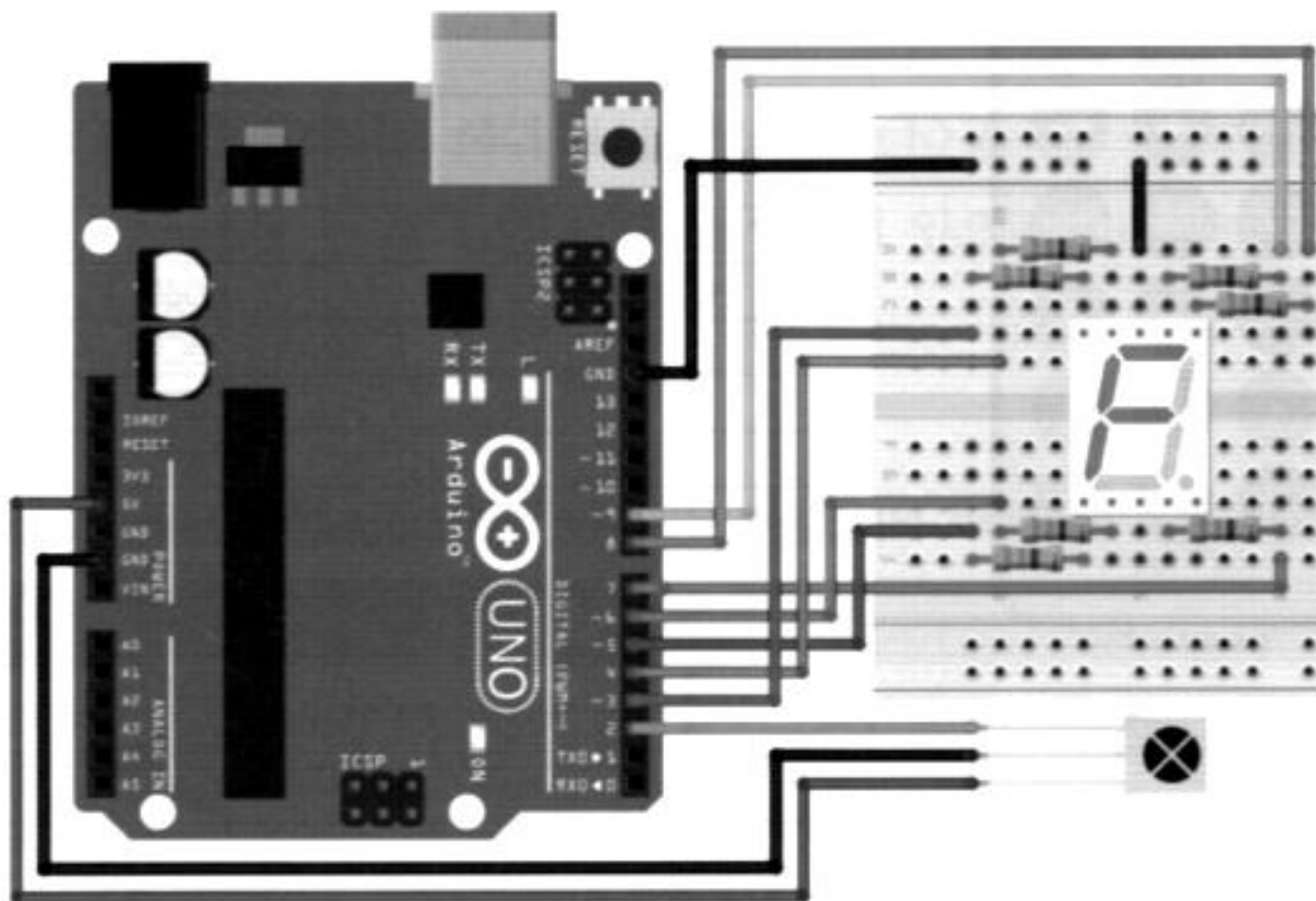
## ❖ 리모컨 키의 전송 데이터 코드



리모컨	키	전송 데이터 코드(16진)
	CH-	A25D
	CH	629D
	CH+	E21D
	:<<	22DD
	>>:	02FD
	>::	C23D
	-	E01F
	+	A857
	EQ	906F
	0	6897
	100+	9867
	200+	B04F
	1	30CF
	2	18E7
	3	7A85
	4	10EF
	5	38C7
	6	5AA5
	7	42BD
	8	4AB5
	9	52AD

# 리모컨

## ❖ 회로도



아두이노 연결핀	7-세그먼트 연결핀
9	7(a)
8	6(b)
7	4(c)
6	2(d)
5	1(e)
4	9(f)
3	10(g)
	수광기 연결핀
2	수광기 OUT

# 리모컨

## ❖ 리모컨 키 기능

키	7-세그먼트 표시값 (disp_val : 0~9)	키	7-세그먼트 표시값 (disp_val : 0~9)
CH-	-4	200+	+2
CH	0	1	1
CH+	+4	2	2
:<<	-3	3	3
>>:	+3	4	4
>::	9	5	5
-	-1	6	6
+	+1	7	7
EQ	5	8	8
0	0	9	9
100+	-2		



# 리모컨

---

## ❖ remocon1.ino

```
// 리모컨 눌려진 키 시리얼 모니터 표시하기
#include <IRremote.h>
const int REC_PIN = 2;

IRrecv myIR(REC_PIN);    // 리모컨 객체 myIR생성
decode_results rlt;      // 리모컨 수신 데이터 구조체 선언

void setup()
{
  Serial.begin(9600);    // 시리얼 통신속도 9600bps
  myIR.enableIRIn();     // 리모컨 입력 시작
}
```

# 리모컨

## ❖ remocon1.ino

```
void loop()
{
    byte key;

    if(!myIR.decode(&rlt)) return;    // 수신된 데이터가 없으면 리턴

    if(rlt.decode_type == NEC){        // NEC 방식인 경우에만 처리
        key = (unsigned char)rlt.value; // 반전 데이터 코드만 추출(하위 8비트)

        if(key == 0x5D) Serial.println("Press  CH-");    // CH- key
        else if(key == 0x9D) Serial.println("Press  CH");    // CH key
        else if(key == 0x1D) Serial.println("Press  CH+");    // CH+ key
        else if(key == 0xDD) Serial.println("Press  :<<");    // :<< key
        else if(key == 0xFD) Serial.println("Press  >>:");    // >>: key
        else if(key == 0x3D) Serial.println("Press  >::");    // >:: key
        else if(key == 0x1F) Serial.println("Press  -");    // - key
        else if(key == 0x57) Serial.println("Press  +");    // + key
        else if(key == 0x6F) Serial.println("Press  EQ");    // EQ key
        else if(key == 0x97) Serial.println("Press  0");    // 0 key
        else if(key == 0x67) Serial.println("Press  200-");    // 200- key
        else if(key == 0x4F) Serial.println("Press  200+");    // 200+ key
```

# 리모컨

## ❖ remocon1.ino

```
    else if(key == 0xCF) Serial.println("Press 1"); // 1 key
    else if(key == 0xE7) Serial.println("Press 2"); // 2 key
    else if(key == 0x85) Serial.println("Press 3"); // 3 key
    else if(key == 0xEF) Serial.println("Press 4"); // 4 key
    else if(key == 0xC7) Serial.println("Press 5"); // 5 key
    else if(key == 0xA5) Serial.println("Press 6"); // 6 key
    else if(key == 0xBD) Serial.println("Press 7"); // 7 key
    else if(key == 0xB5) Serial.println("Press 8"); // 8 key
    else if(key == 0xAD) Serial.println("Press 9"); // 9 key
  }
  myIR.resume();    // 새로운 데이터 수신
}
```

# 리모컨

## ❖ remocon2.ino

```
// 리모컨 키에 따라 7-세그먼트 표시 값 설정
#include <IRremote.h>
const int REC_PIN = 2;
const int segment_pin[8] = {9, 8, 7, 6, 5, 4, 3}; // A,B,C,D,E,F,G 연결 핀

// 7세그먼트 0~F 표시 패턴 {a, b, c, d, e, f, g}
const byte segment_pat[10][7] = {
    {1, 1, 1, 1, 1, 1, 0}, // 0
    {0, 1, 1, 0, 0, 0, 0}, // 1
    {1, 1, 0, 1, 1, 0, 1}, // 2
    {1, 1, 1, 1, 0, 0, 1}, // 3
    {0, 1, 1, 0, 0, 1, 1}, // 4
    {1, 0, 1, 1, 0, 1, 1}, // 5
    {1, 0, 1, 1, 1, 1, 1}, // 6
    {1, 1, 1, 0, 0, 0, 0}, // 7
    {1, 1, 1, 1, 1, 1, 1}, // 8
    {1, 1, 1, 0, 0, 1, 1}}; // 9

byte disp_val = 0; // 7-세그먼트 표시값
void segment_dsp(void); // 7-세그먼트 표시
IRrecv myIR(REC_PIN); // 리모컨 객체 myIR생성
decode_results rlt; // 리모컨 수신 데이터 구조체 선언
```

# 리모컨

---

## ❖ remocon2.ino

```
void setup()
{
  byte  n;

  myIR.enableIRIn();    // 리모컨 입력 시작

  for(n = 0;n < 7;n++)
    pinMode(segment_pin[n], OUTPUT);    // Segment 연결 핀 출력 설정

  segment_dsp();        // 초기상태 '0' 표시
}
```

# 리모컨

## ❖ remocon2.ino

```
void loop()
{
    byte key, disp;

    if(!myIR.decode(&rslt)) return;    // 수신된 데이터가 없으면 리턴

    if(rslt.decode_type == NEC){        // NEC 방식인 경우에만 처리
        key = (unsigned char)rslt.value; // 반전 데이터 코드만 추출(하위 8비트)

        if(key == 0x5D) disp_val = (disp_val + 6) % 10;    // CH- key(-4)
        else if(key == 0x9D) disp_val = 0;                // CH key(0)
        else if(key == 0x1D) disp_val = (disp_val + 4) % 10; // CH+ key(+4)
        else if(key == 0xDD) disp_val = (disp_val + 7) % 10; // :<< key(-3)
        else if(key == 0xFD) disp_val = (disp_val + 3) % 10; // >>: key(+3)
        else if(key == 0x3D) disp_val = 9;                 // >:: key(9)
        else if(key == 0x1F) disp_val = (disp_val + 9) % 10; // - key(-1)
        else if(key == 0x57) disp_val = (disp_val + 1) % 10; // + key(+1)
        else if(key == 0x6F) disp_val = 5;                 // EQ key(5)
        else if(key == 0x67) disp_val = (disp_val + 8) % 10; // 100+ key(-2)
        else if(key == 0x4F) disp_val = (disp_val + 2) % 10; // 200+ key(+2)
```

# 리모컨

## ❖ remocon2.ino

```
    else if(key == 0x97) disp_val = 0;           // 0 key
    else if(key == 0xCF) disp_val = 1;           // 1 key
    else if(key == 0xE7) disp_val = 2;           // 2 key
    else if(key == 0x85) disp_val = 3;           // 3 key
    else if(key == 0xEF) disp_val = 4;           // 4 key
    else if(key == 0xC7) disp_val = 5;           // 5 key
    else if(key == 0xA5) disp_val = 6;           // 6 key
    else if(key == 0xBD) disp_val = 7;           // 7 key
    else if(key == 0xB5) disp_val = 8;           // 8 key
    else if(key == 0xAD) disp_val = 9;           // 9 key
    segment_dsp();
}
myIR.resume();    // 새로운 데이터 수신
}

// 눌려진 키 16진 7-세그먼트 표시
void segment_dsp() {
    byte  n;

    for(n = 0;n < 7;n++)    // disp_val 패턴 출력
        digitalWrite(segment_pin[n], segment_pat[disp_val][n]);
}
```