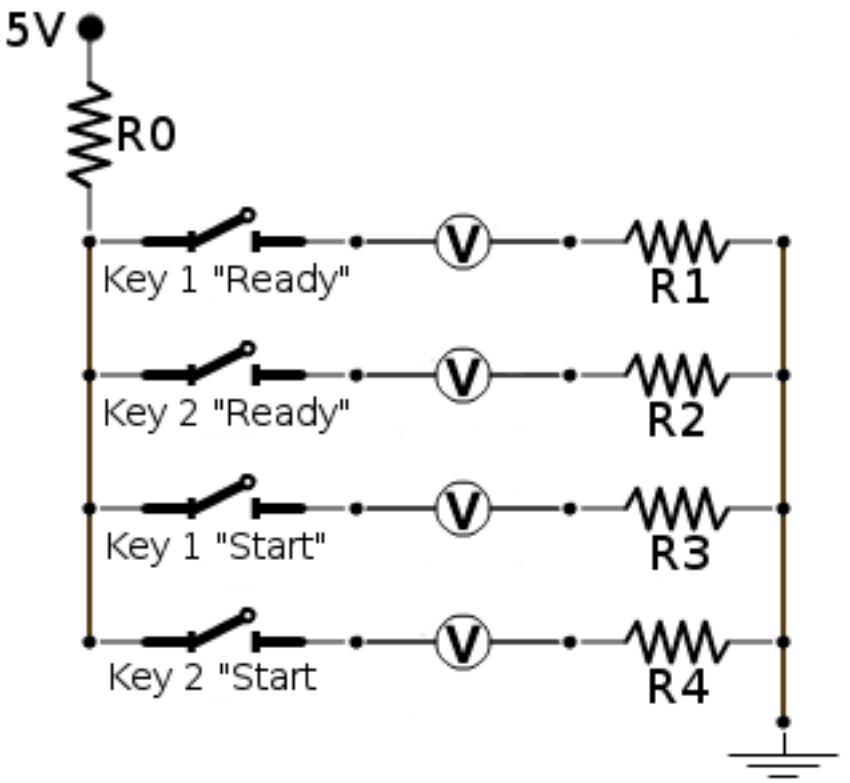
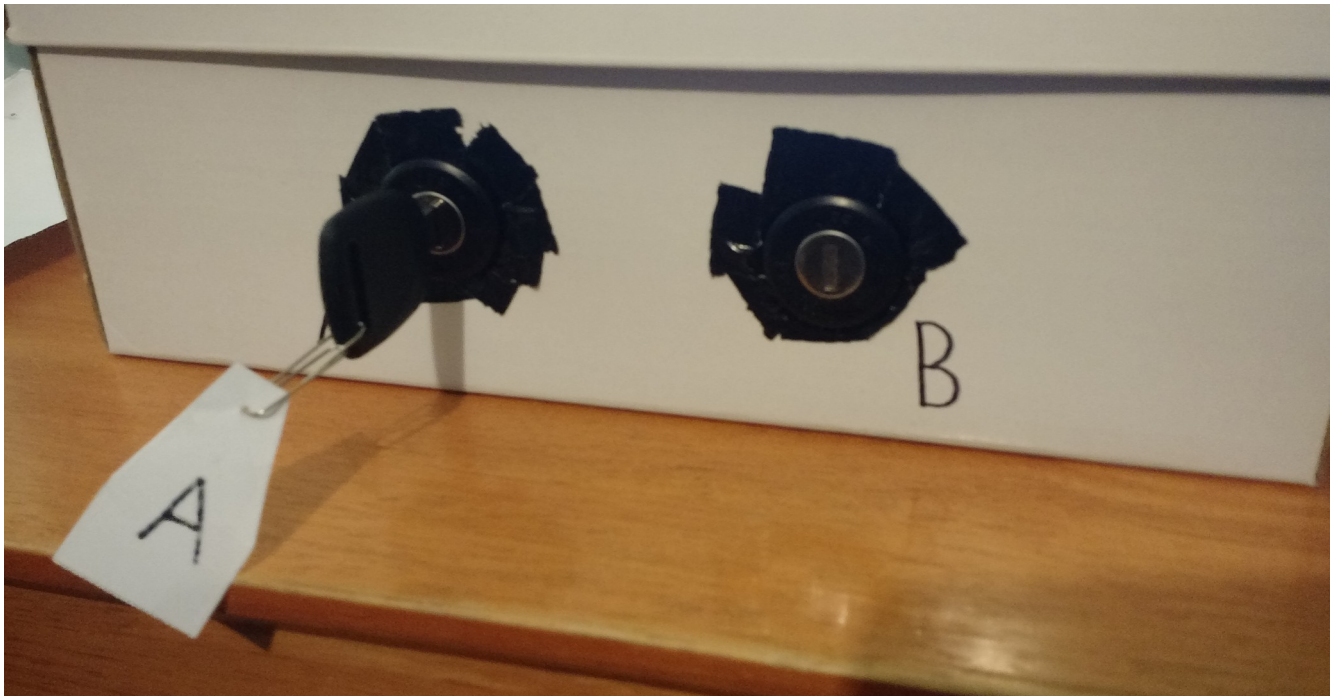
Two-Man-Rule Encryption

```
int len;
if (str.length() < EELEN) len = str.length();
else len = EELEN;
byte cypher[len];
char encrypted[str.length()+1];
for (int i = 0; i < len; i++) {
  cypher[i] = random(91);
}
for (int i = 0; i < str.length(); i++) {
  encrypted[i] = (str.charAt(i)-32 + cypher[i%len])%91 + 32;
}
encrypted[str.length()] = 0;
eWrite(cypher, len);
return (String) encrypted;
```



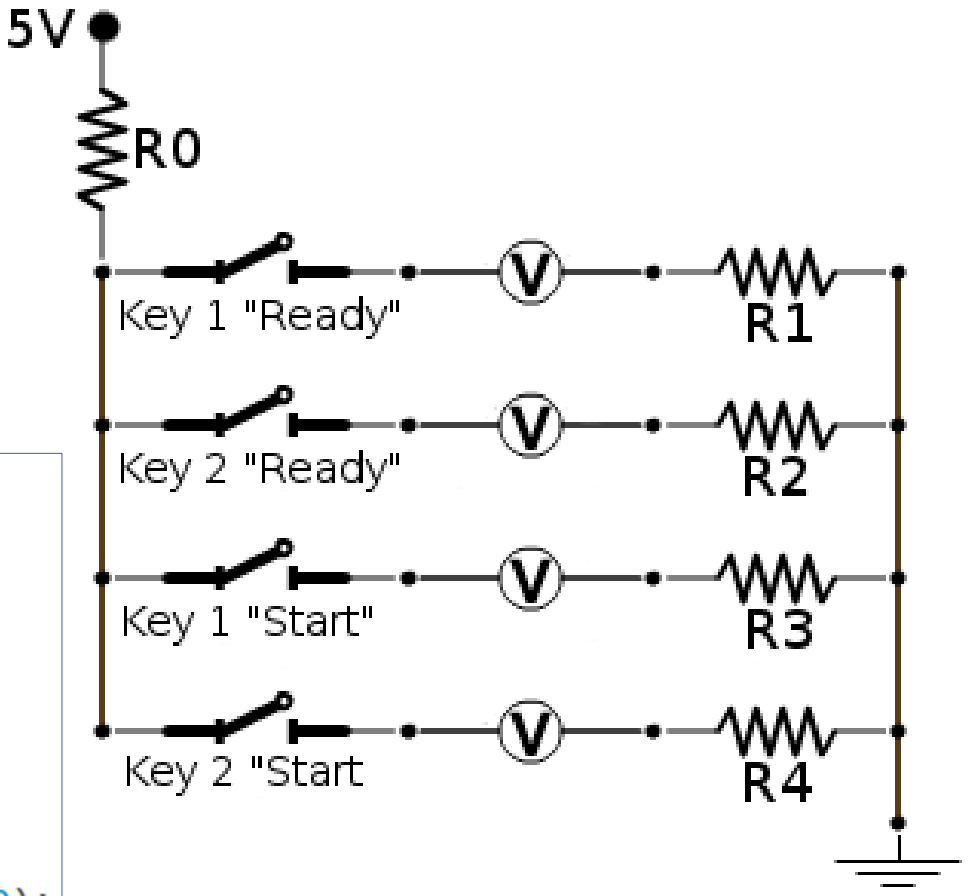
The Two Man Rule

- Two physical keys required for security
- Extra protection against accidental or unauthorized access
- Used in nuclear weapons facilities



Implementation

- 4 calls to `analogRead()`
- Easy detection of state of key switches

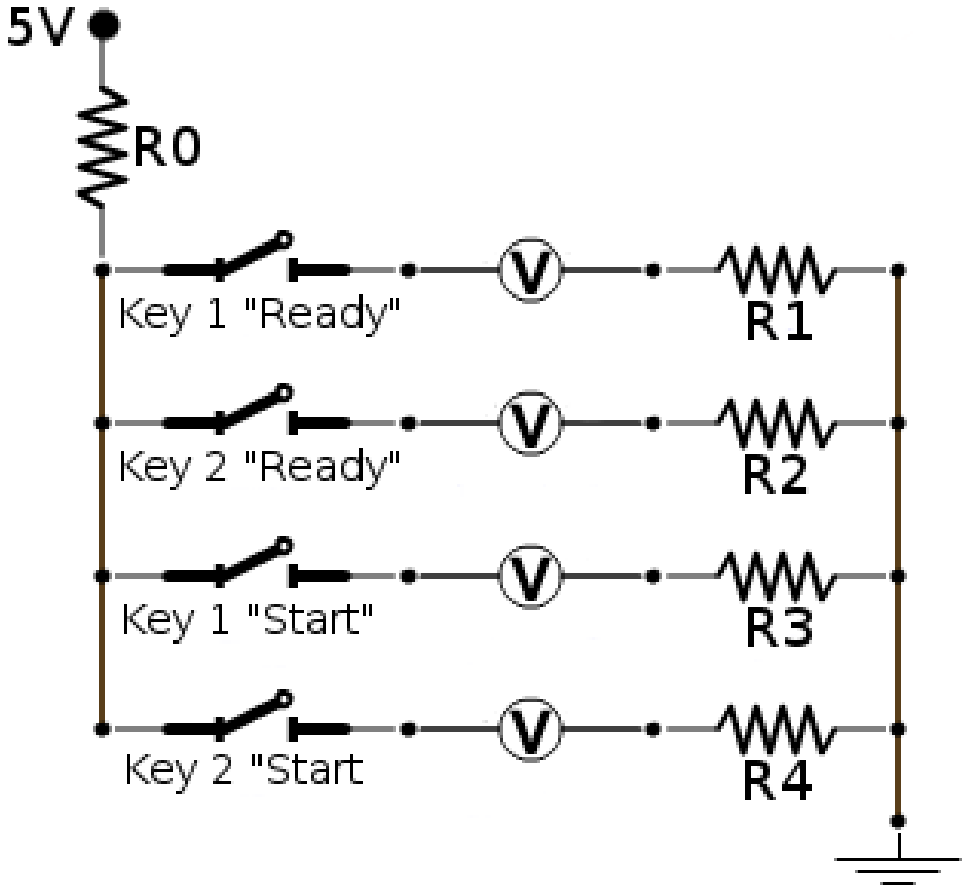



```
do {
  delay(100);
  if (checkKillSwitch()) return false;
  a5 = analogRead(A5);
  a4 = analogRead(A4);
  a3 = analogRead(A3);
  a2 = analogRead(A2);
} while (a5 > 10 || a4 > 10 || a3 > 10 || a2 > 10);
```

Security

```
while (!(a5 > 300 && a5 < 400)|| !(a4 > 300 && a4 < 400) || !(a3 > 300 && a3 < 400) || !(a2 > 300 && a2 < 400));
```

Key 1	Key 2	A5	A4	A3	A2
Off	Off	A4	0	0	0
Off	Ready	0	0	0	~850
Off	Start	0	~600	0	~600
Ready	Off	0	0	~800	0
Ready	Ready	0	0	~700	~700
Ready	Start	0	~500	~500	~500
Start	Off	~450	0	~450	0
Start	Ready	~400	0	~400	~400
Start	Start	~350	~350	~350	~350



Vigenère Cypher

- Every character has a number value
- Create a random key string
- Add the each character in the key to its corresponding character in the given string
- If they key is as long as the original text, it is impossible to break

Text:	`teststring`
Key:	`gpecheqlgr`
Result:	`ztwvzxhttx`

Key Storage – EEPROM

- EEPROM is 1024 byte flash storage on Arduino—stored when powered down
- Key is stored on EEPROM and never seen by computer

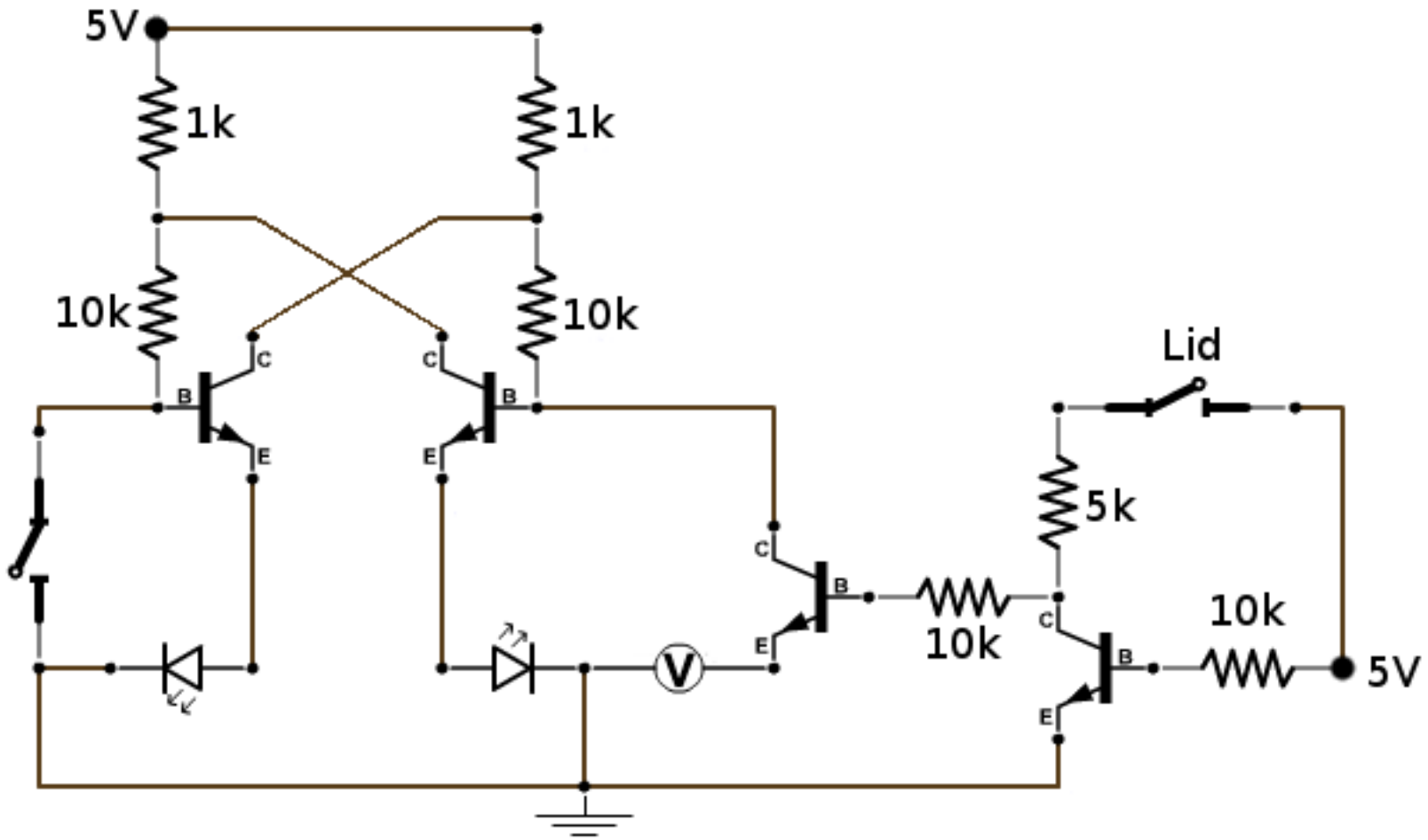
```
int len = str.length();
byte cypher[len];
char encrypted[str.length()+1];
for (int i = 0; i < len; i++) {
  cypher[i] = random(91);
}
for (int i = 0; i < str.length(); i++) {
  encrypted[i] = (str.charAt(i)-32 + cypher[i%len])%91 + 32;
}
encrypted[str.length()] = 0;
```

Tamper Detection

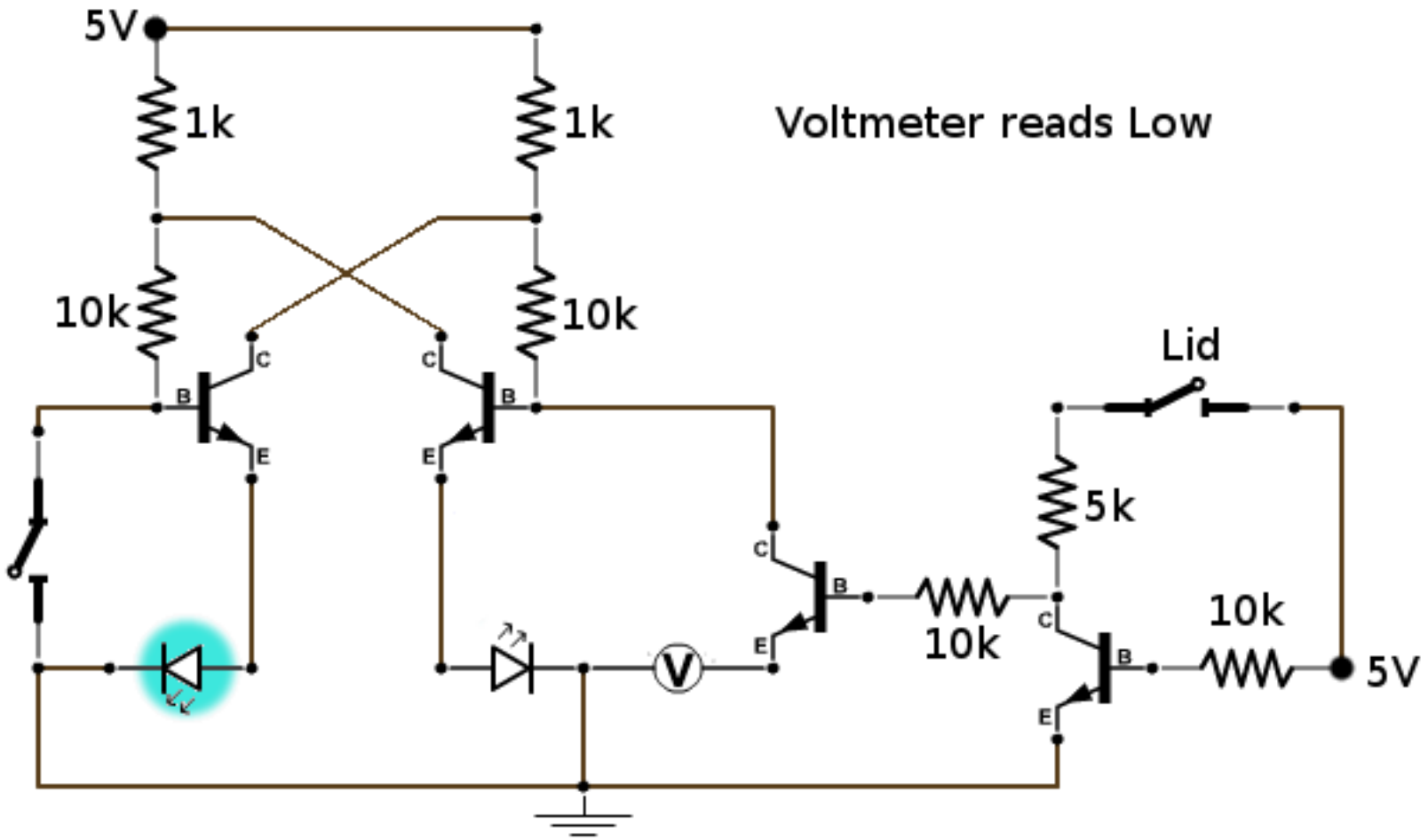
- EEPROM is cleared (key deleted) if lid is opened
- Using “flip-flop” circuit, device “remembers” lid being opened even after it is closed

```
bool checkKillSwitch() {
  //if box has been opened, erase EEPROM
  int voltage = analogRead(A0);
  if (voltage < 970 || voltage > 990) {
    eClear();
    return true;
  }
  return false;
}
```

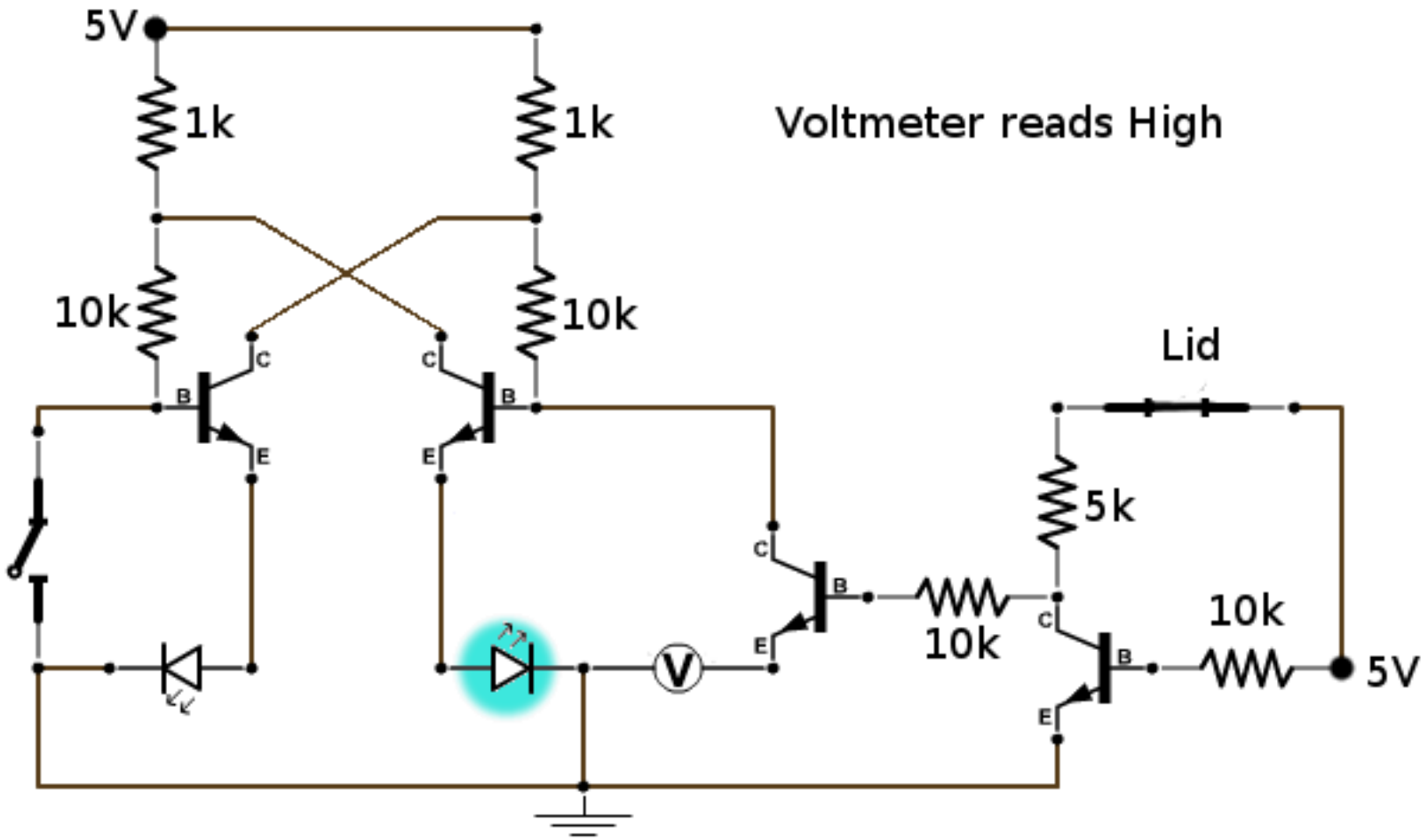
Circuit



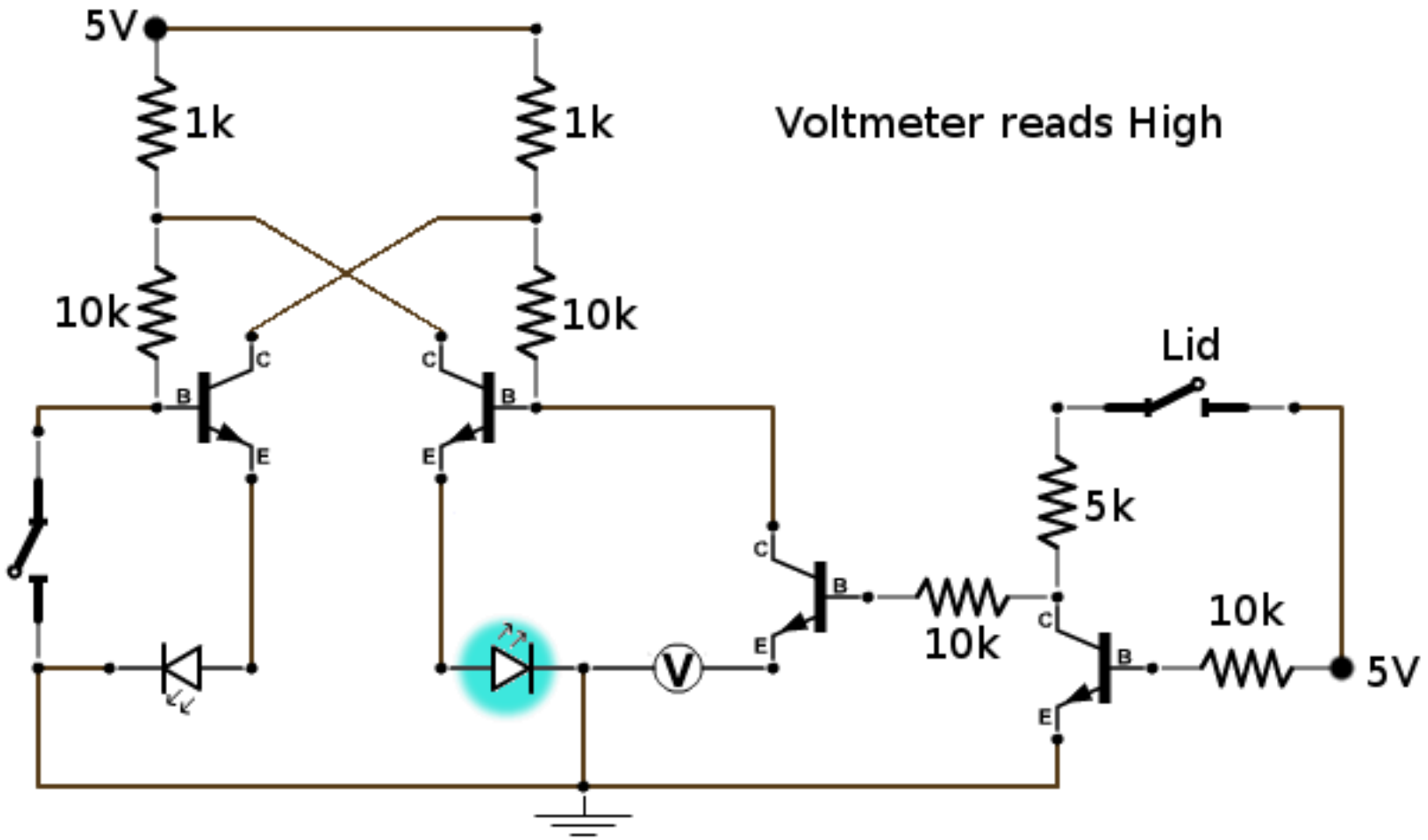
Initial State



Open Lid



Lid Closed



;