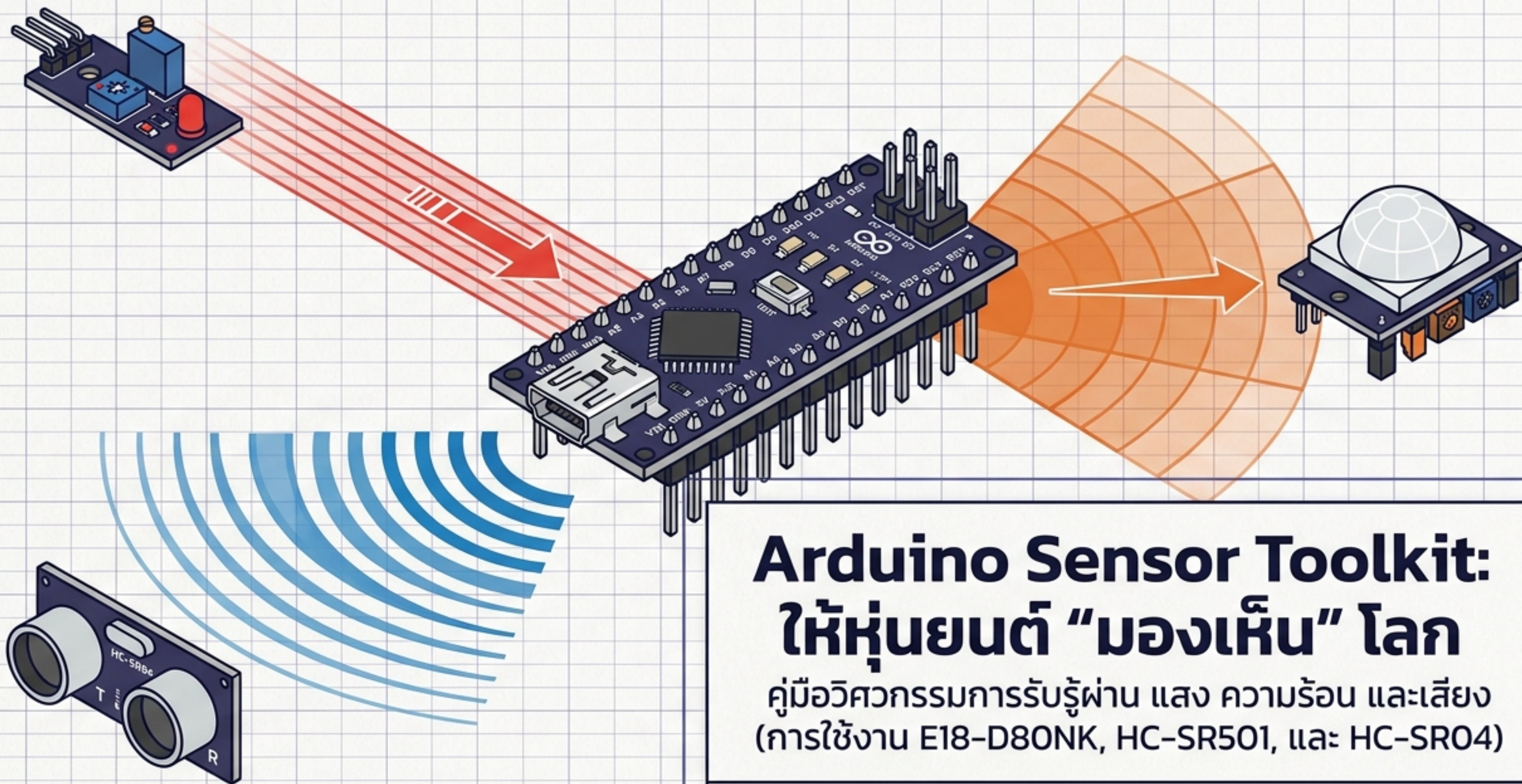




Arduino Sensor Toolkit: ให้หุ่นยนต์ “มองเห็น” โลก

คู่มือวิศวกรรมการเรียนรู้ผ่าน แสง ความร้อน และเสียง
(การใช้งาน E18-D80NK, HC-SR501, และ HC-SR04)

หลักสูตรไมโครคอนโทรลเลอร์ (20105-2105)



แสงอินฟราเรด (Light)

E18-D80NK

(IR Proximity Sensor)

บทบาท: การสัมผัสระยะประชิด
(3-80 cm)



รังสีความร้อน (Heat)

HC-SR501

(PIR Motion Sensor)

บทบาท: การจับความเคลื่อนไหว
สิ่งมีชีวิต (3-7 m)



คลื่นเสียงความถี่สูง (Sound)

HC-SR04

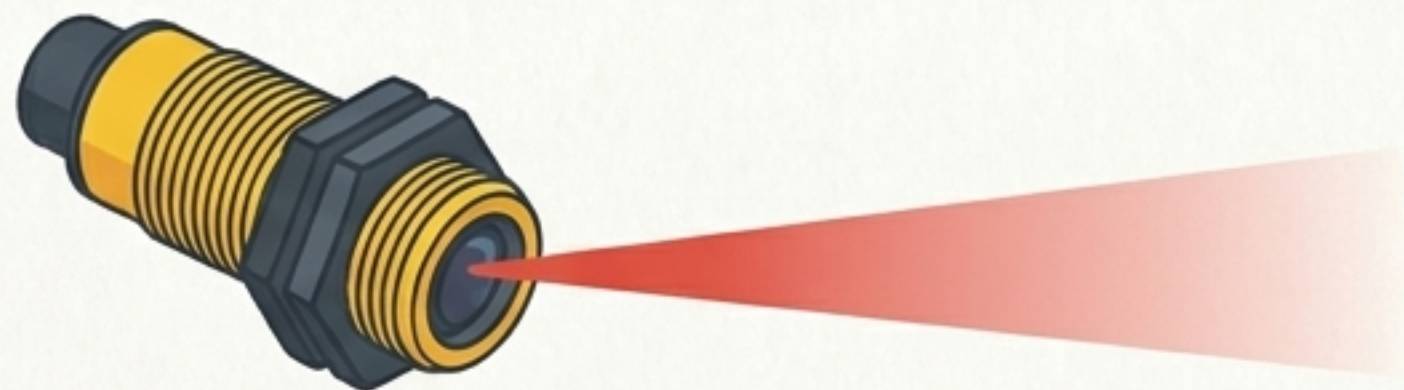
(Ultrasonic Sensor)

บทบาท: การวัดระยะทางที่แม่นยำ
(2-400 cm)

เซนเซอร์คือ 'ประสาทสัมผัส' ที่แปลงสภาพแวดล้อมทางกายภาพให้เป็นสัญญาณดิจิทัล (0 และ 1) สำหรับบอร์ด Arduino

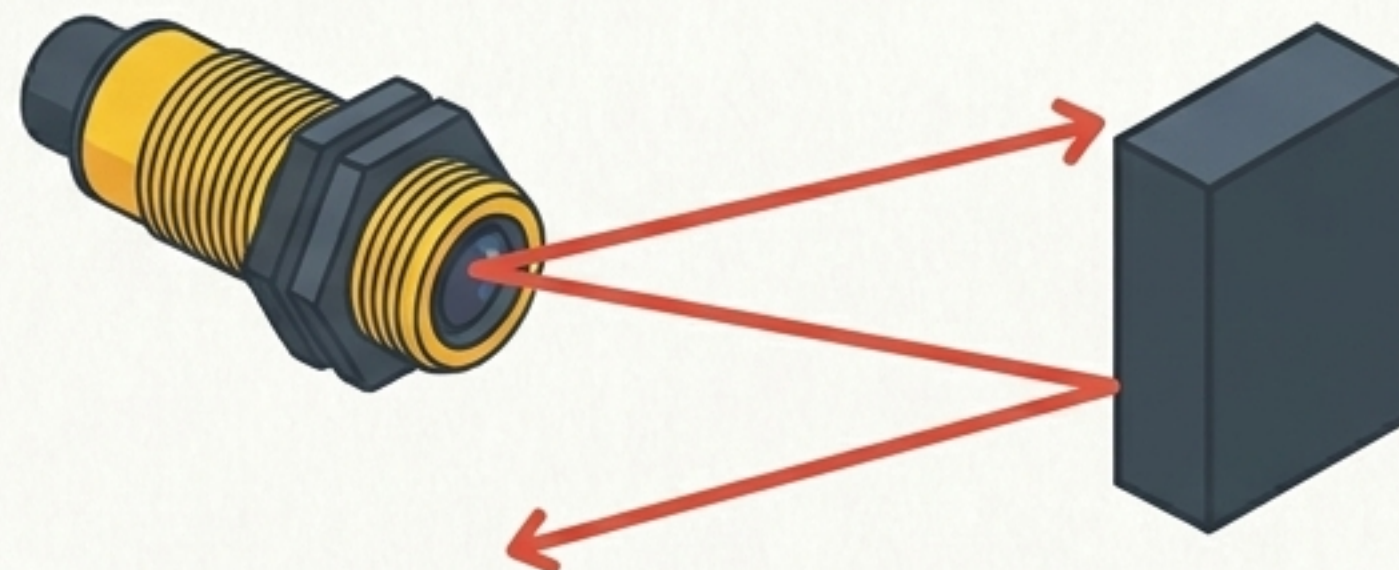
เครื่องมือที่ 1: เซนเซอร์อินฟราเรด E18-D80NK (ตรวจจับวัตถุ)

ไม่มีวัตถุ (No Object)



LED ดับ | Logic: 1 (HIGH)

พบวัตถุ (Object Detected)



LED สว่าง | Logic: 0 (LOW)

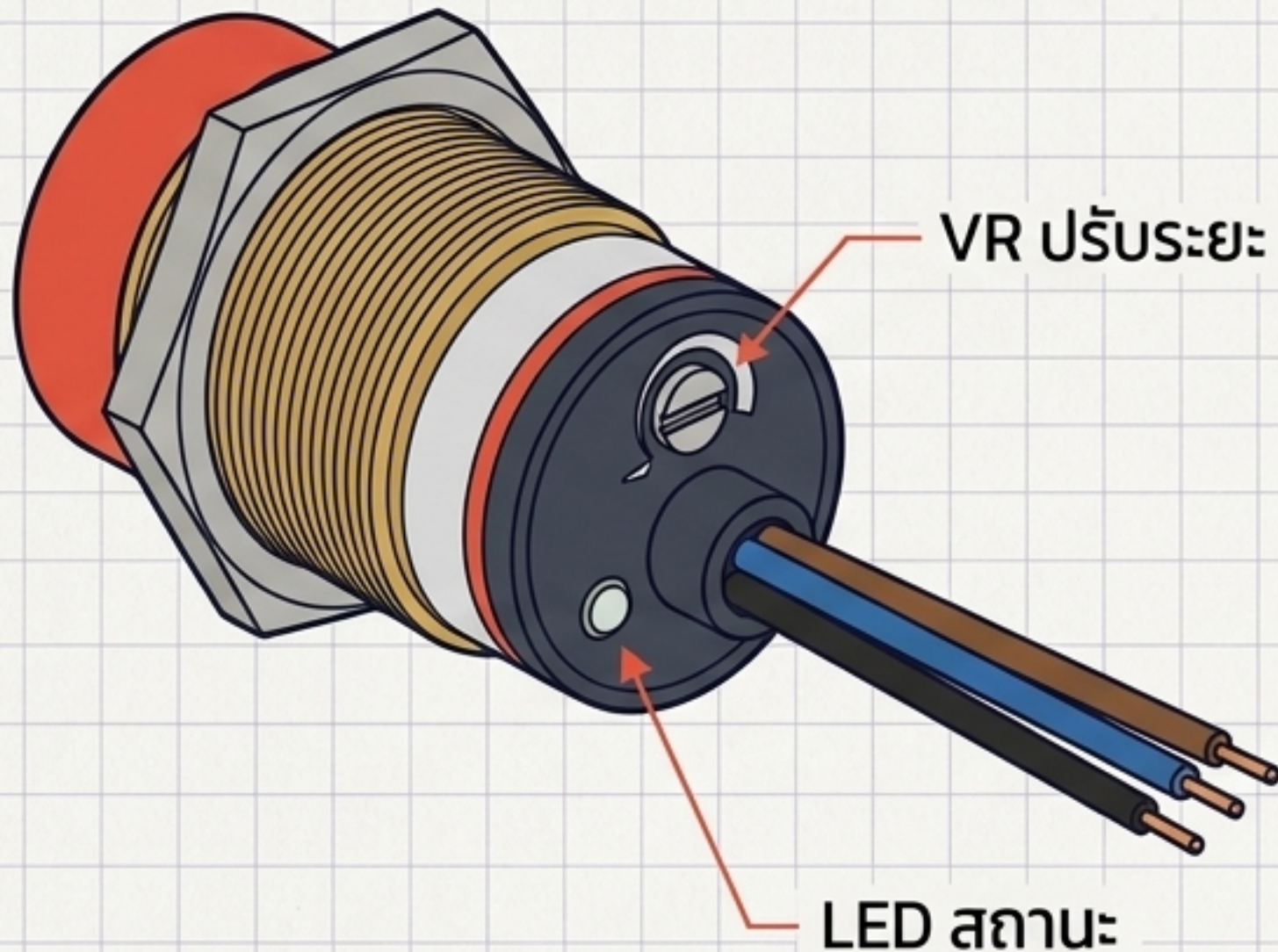
Key Concept

หลักการสะท้อน (Reflection)

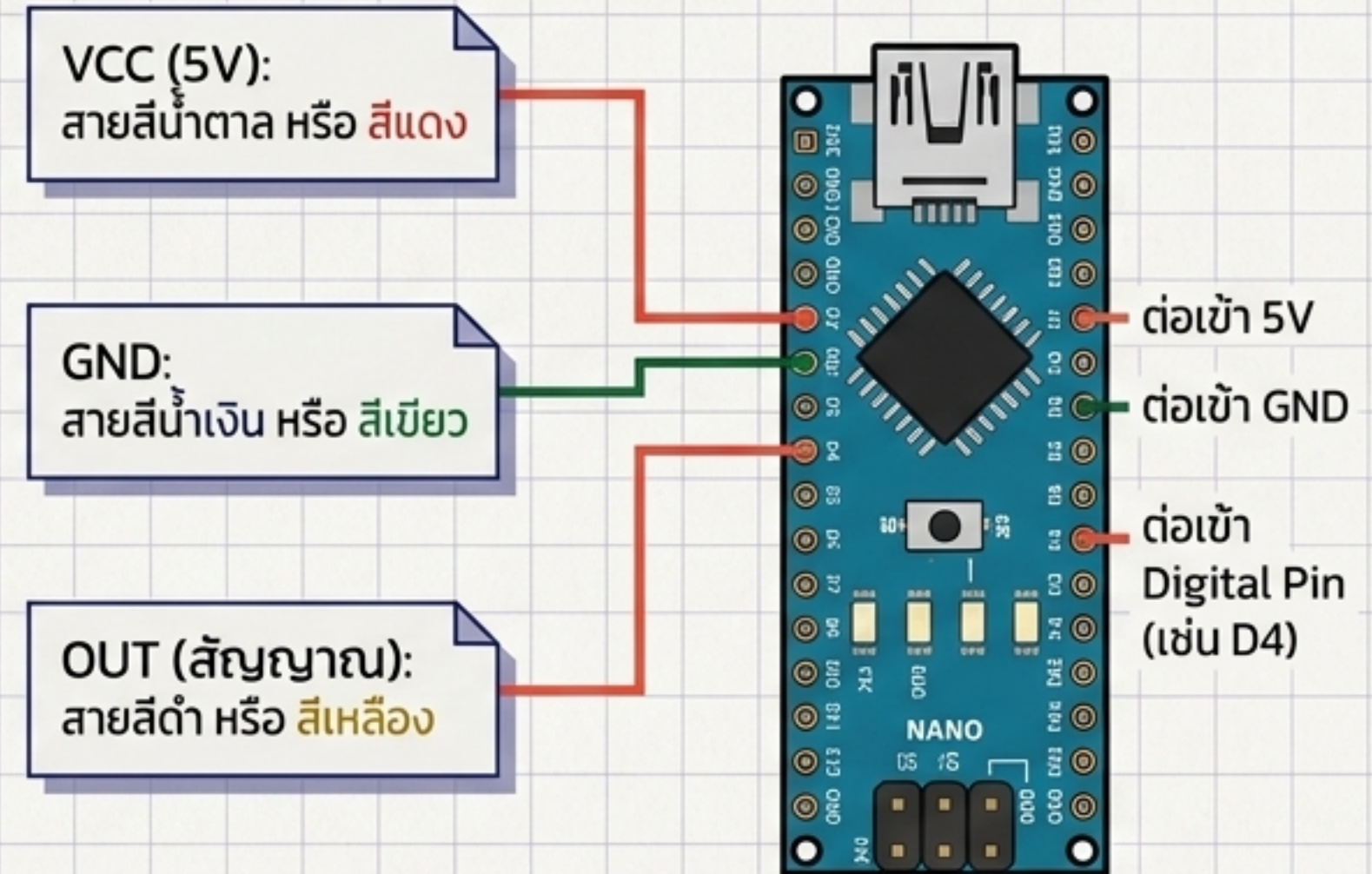
อาศัยการยิงแสงอินฟราเรด หากมีวัตถุทึบแสงขวางกั้น แสงจะสะท้อนกลับไปที่ตัวรับสัญญาณ ให้ผลลัพธ์เป็น Active LOW (0)
หมายเหตุ: ปรับระยะตรวจจับได้ 3-80 ซม. ด้วย VR (ตัวต้านทานปรับค่าได้) ด้านท้ายเซนเซอร์

กายวิภาคและการต่อสายไฟ (IR Sensor Wiring)

E18-D80NK Sensor



Connection Map



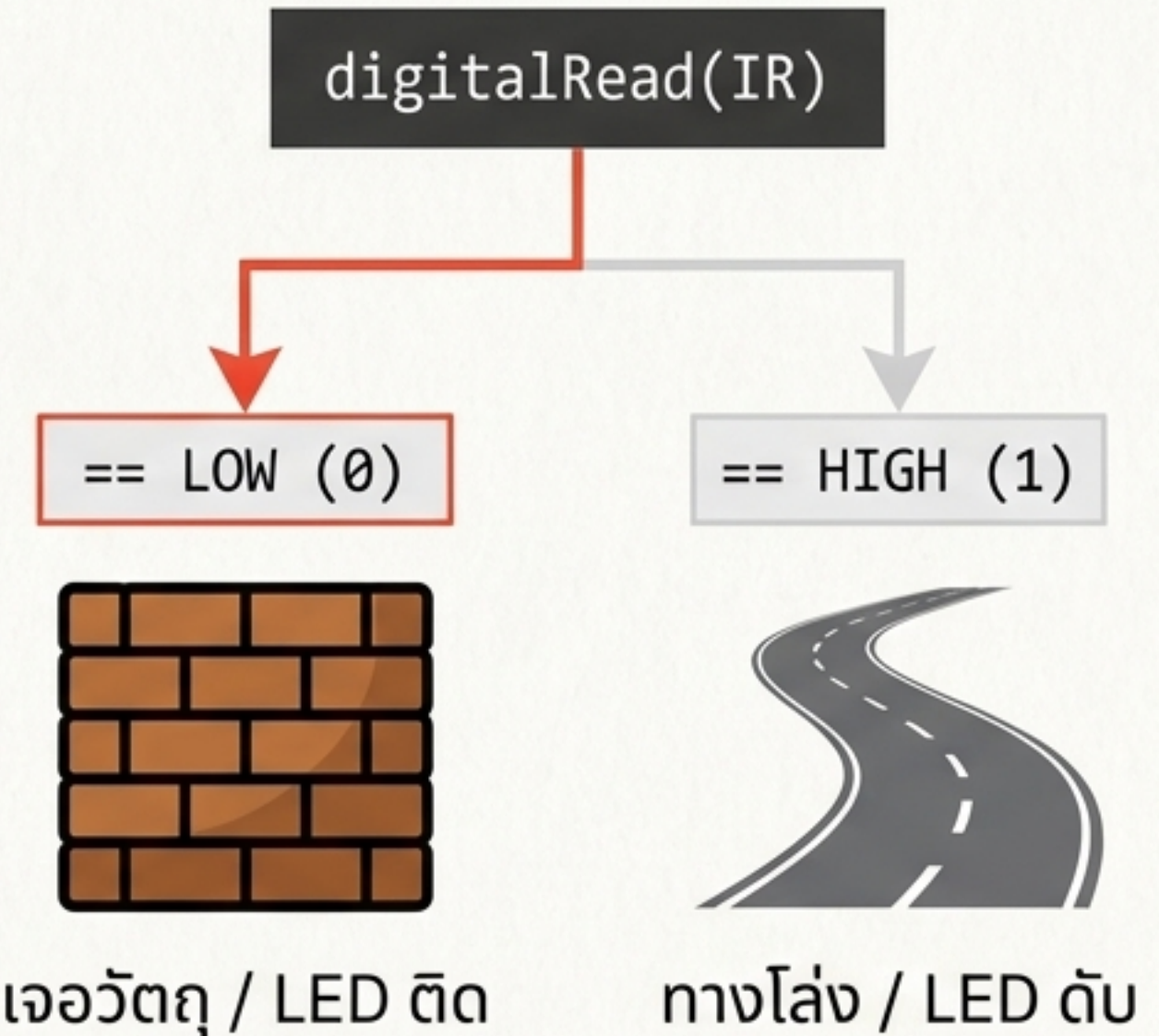
Note: มีตัวต้านทาน Pull-up 10K ภายในวงจรแล้ว ไม่ต้องต่อเพิ่มภายนอก
สัญญาณทำงานแบบ Digital TTL (0 = 0V, 1 = 5V)

ตรรกะการเขียนโปรแกรม (IR Code Logic)

```
#define IR 4 // ขาสัญญาณเซนเซอร์

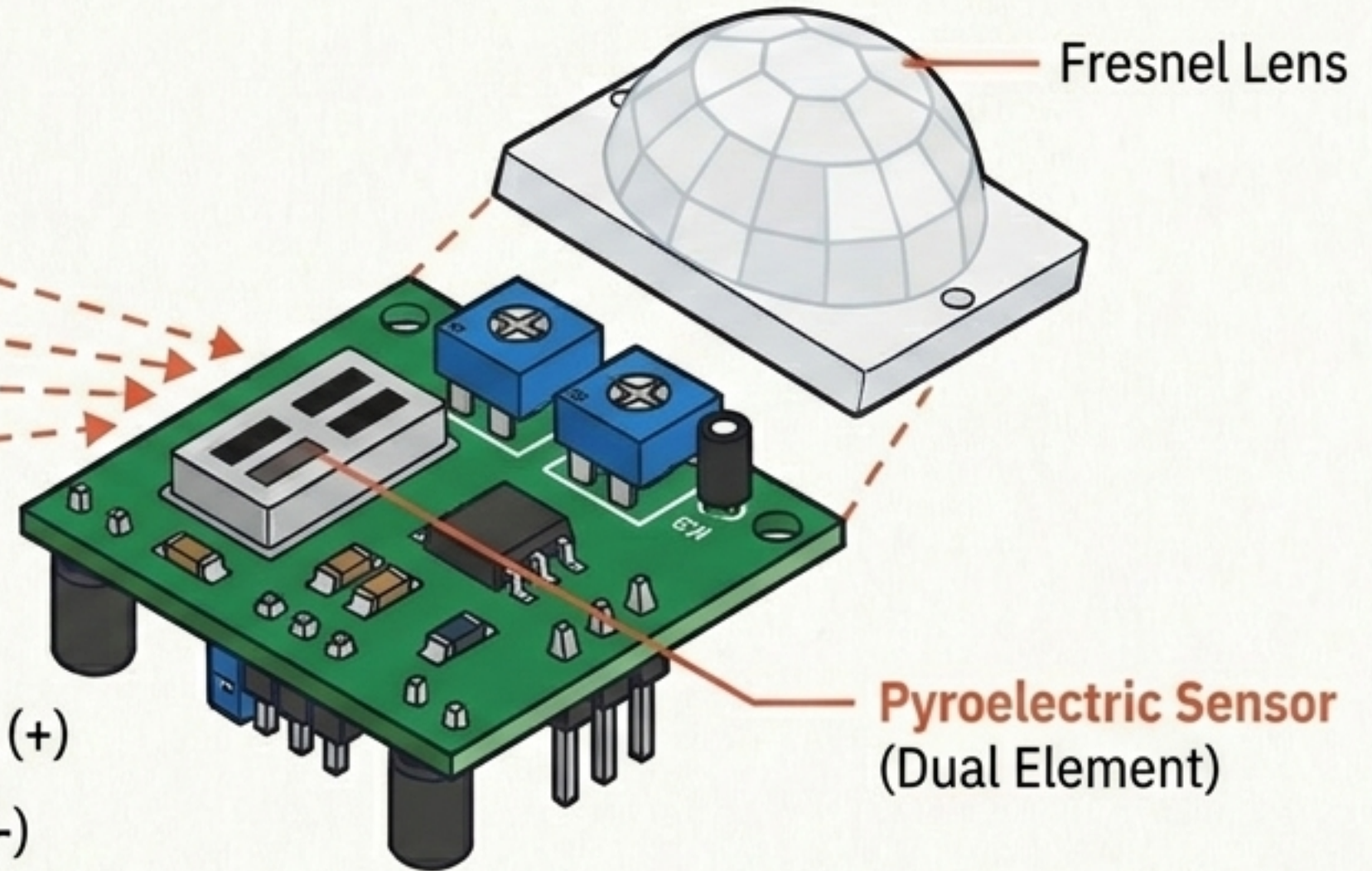
void setup() {
  pinMode(IR, INPUT);
}

void loop() {
  if (digitalRead(IR) == LOW) {
    // วัตถุถูกตรวจจับ (Object Detected)
    Serial.println("0: เจอสิ่งกีดขวาง!");
  }
}
```



Key Function: `digitalRead()` คือหัวใจหลัก ใช้สำหรับอ่านค่าสถานะดิจิทัลของเซนเซอร์

เครื่องมือที่ 2: เซนเซอร์จับความเคลื่อนไหว HC-SR501 (PIR)



🔥 **Step 1:** ความร้อนผ่านşikซ้าย -> แรงดันไฟฟ้าเพิ่มขึ้น (+)

🔥 **Step 2:** ความร้อนผ่านşikขวา -> แรงดันไฟฟ้าลดลง (-)

ทำไมถึงจับ 'ความเคลื่อนไหว'? (How Motion is Detected)

เซนเซอร์ไม่ได้จับความร้อนที่อยู่นิ่ง แต่จับ 'ความแตกต่าง' ของรังสีอินฟราเรด (ความร้อนจากสิ่งมีชีวิต) ที่ตัดผ่านจากเซนเซอร์şikหนึ่งไปยังอีกşikหนึ่ง เลนส์เฟรสเนล (Fresnel Lens) ช่วยขยายและโฟกัสพื้นที่รับสัญญาณให้กว้างถึง 140 องศา (ระยะ 3-7 เมตร)

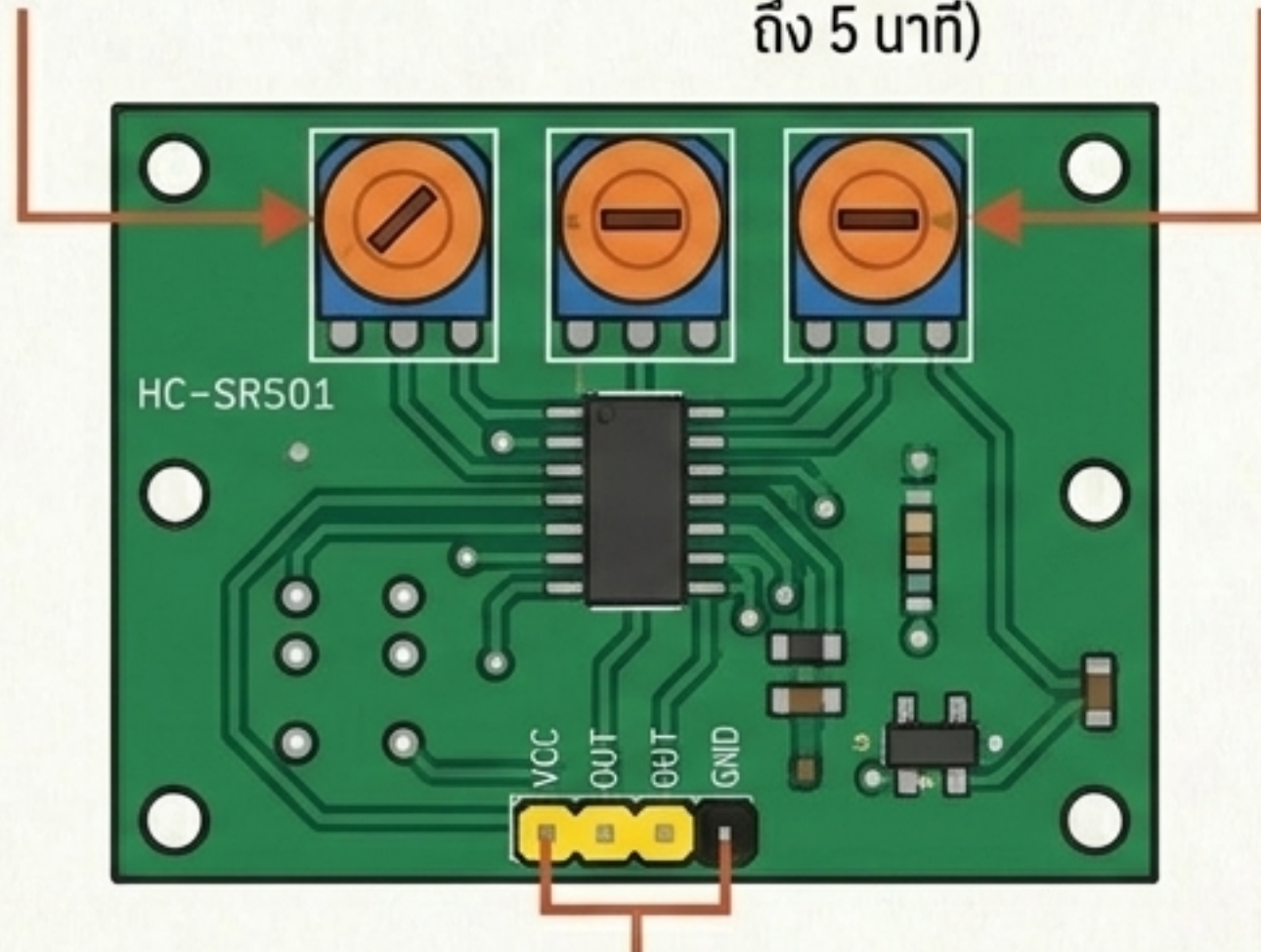
กายวิภาคและการต่อสายไฟ (PIR Sensor Wiring)

1. Sensitivity Adjust

(ปรับความไว/ระยะทาง: 3-7 เมตร)

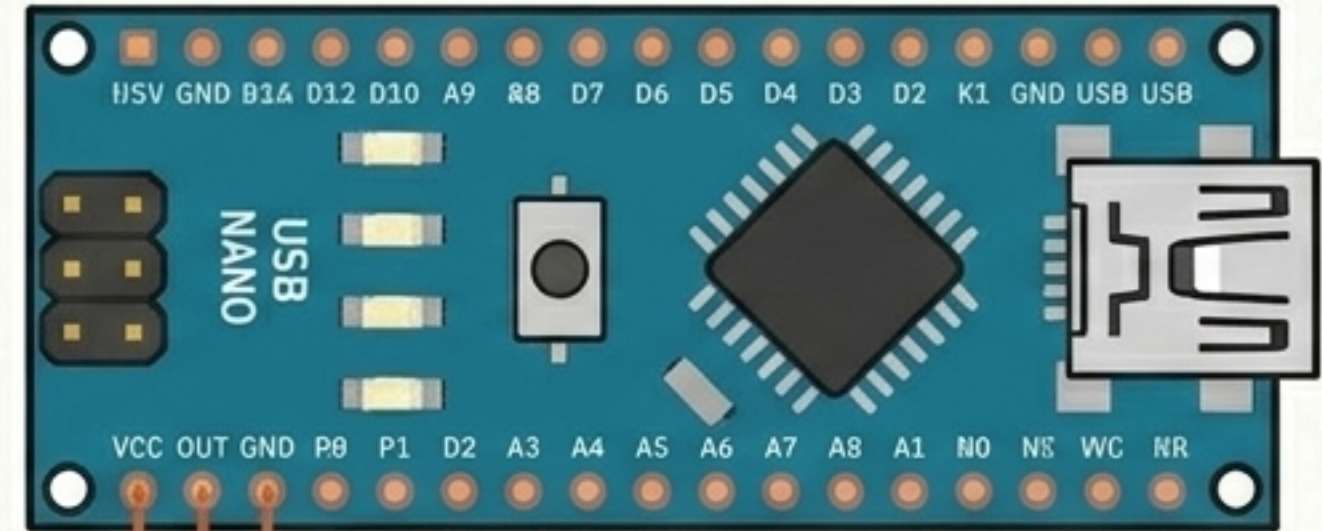
2. Time Delay Adjust

(ปรับหน่วงเวลา: 0.3 วินาที ถึง 5 นาที)



3. Jumper

(โหมด L = ทริกเกอร์ครั้งเดียว,
โหมด H = ทริกเกอร์ซ้ำต่อเนื่อง)



VCC

→ ต่อเข้า 5V
(รองรับแรงดัน 4.5V - 20V)

OUT

→ ต่อเข้า Digital Pin (เช่น D2)
- ปลอยสัญญาณ 3.3V TTL
(Arduino อ่านค่า HIGH ได้)

GND

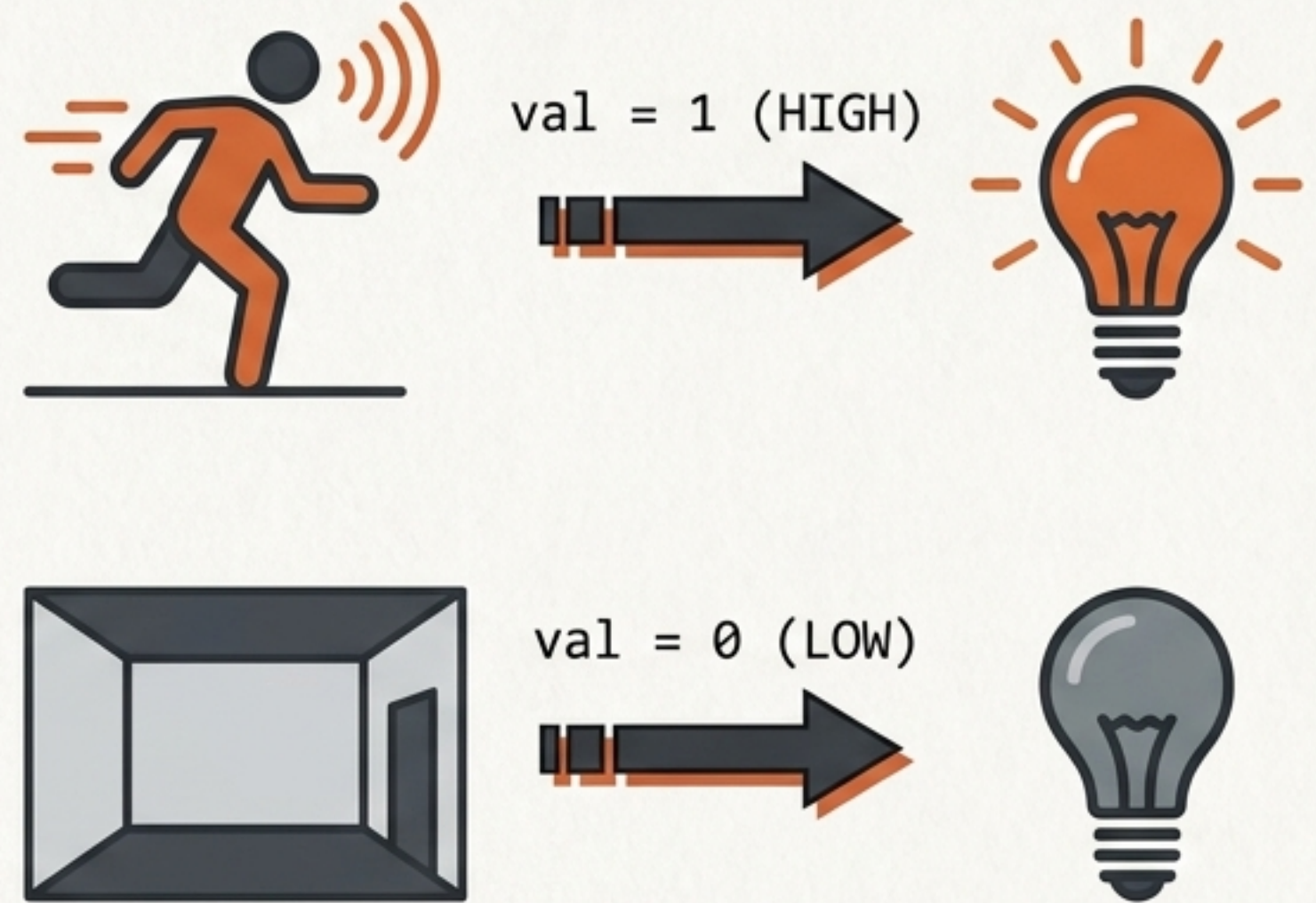
→ ต่อเข้า GND

ตรรกะการเขียนโปรแกรม (PIR Code Logic)

```
#define PIR 2
#define LED 13

void setup() {
  pinMode(PIR, INPUT);
  pinMode(LED, OUTPUT);
}

void loop() {
  int val = digitalRead(PIR);
  if (val == HIGH) { // 1 = เจอการเคลื่อนไหว
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

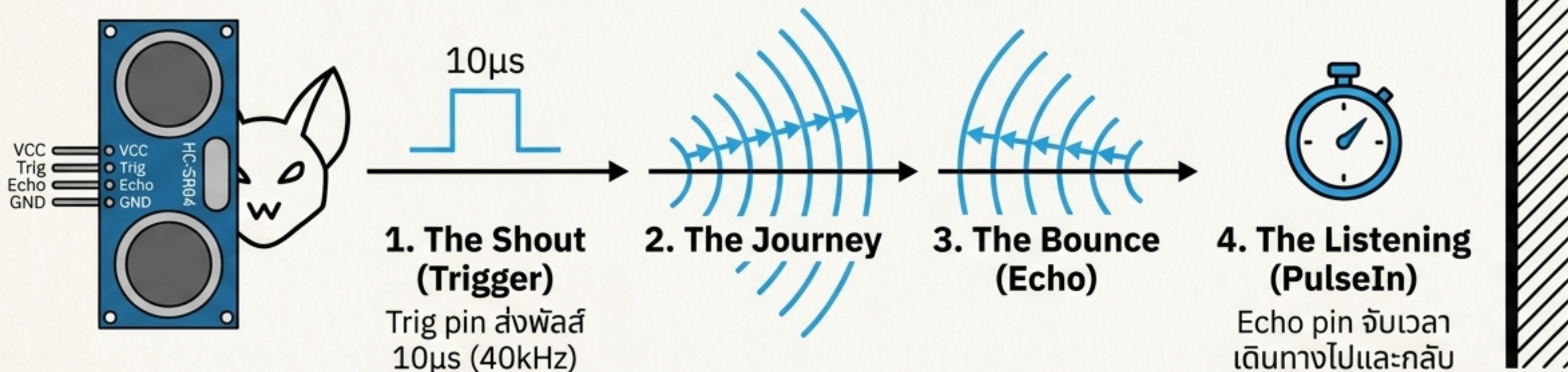


Note: แตกต่างจาก IR Sensor, เซนเซอร์ PIR จะให้ค่าเป็น HIGH (1) เมื่อตรวจพบการเคลื่อนไหว (Active HIGH)

เครื่องมือที่ 3: เซนเซอร์อัลตราโซนิก HC-SR04 (Ultrasonic)

Echolocation Mapping

หลักการทำงาน: โซนาร์ดิจิทัล (Echolocation Principle)



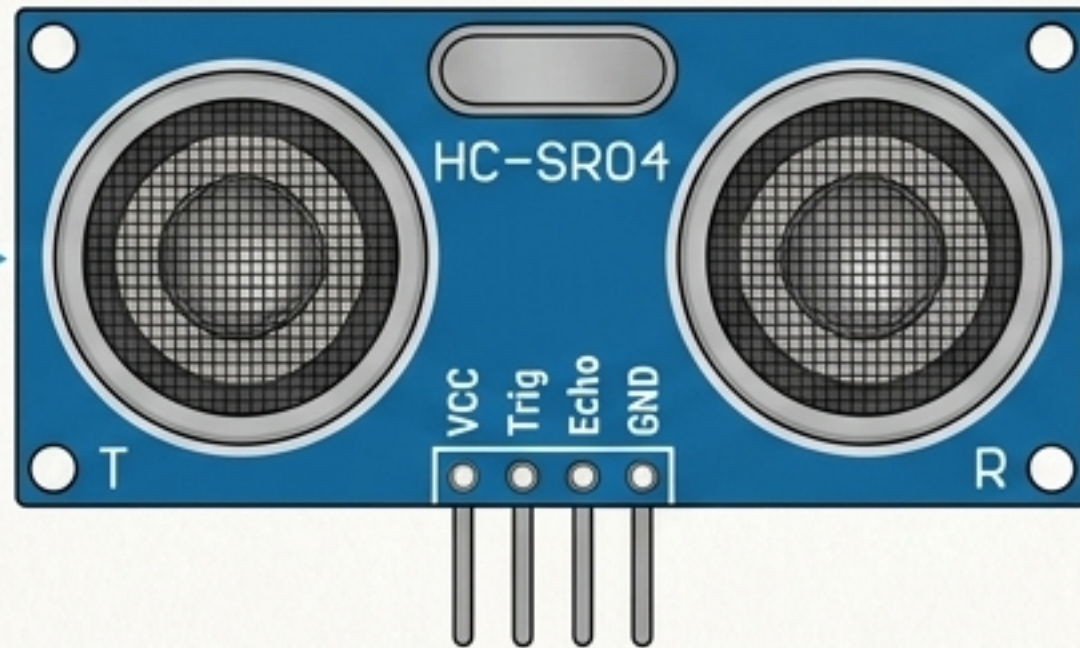
Key Concept

โซนาร์ในโลกดิจิทัล (Digital Sonar)

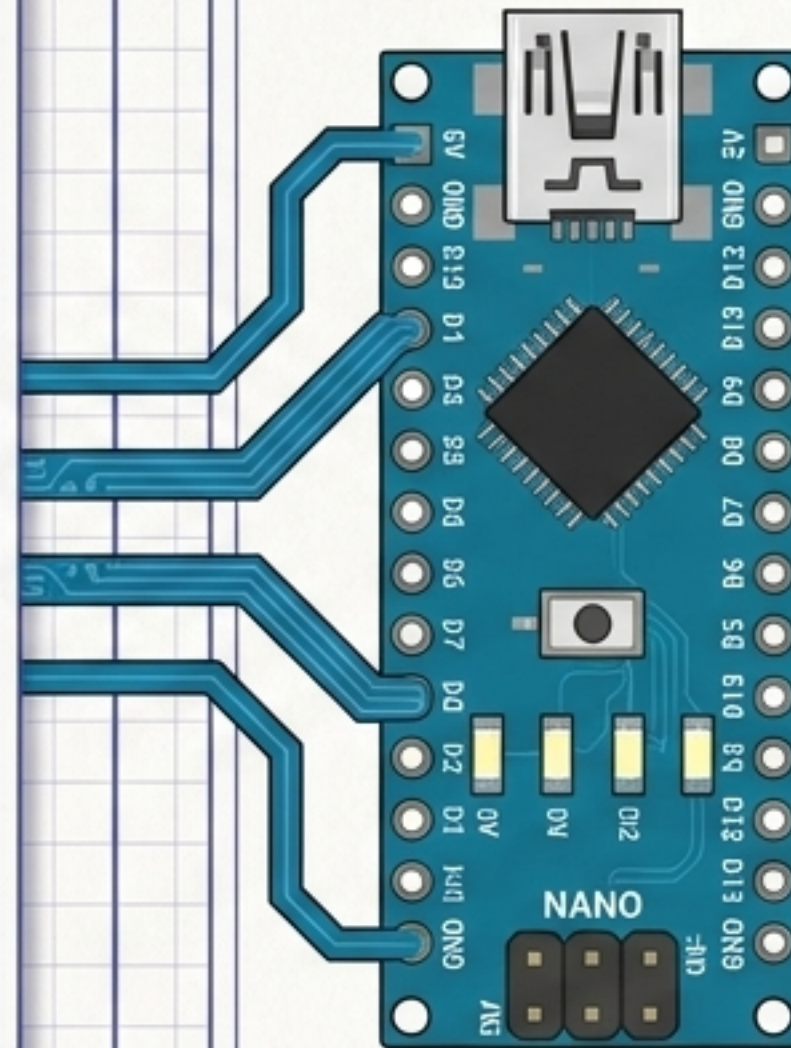
ใช้คลื่นเสียงความถี่สูง (40 kHz) ไปกระทบวัตถุและสะท้อนกลับ
ระยะเวลาของการสะท้อนกลับจะเป็นสัดส่วนตรงกับระยะห่างระหว่างวัตถุ (วัดได้ 2-400 ซม.)

กายวิภาคและการต่อสายไฟ (Ultrasonic Wiring)

Transmitter (T) - ตัวส่งคลื่น



Receiver (R) - ตัวรับคลื่นสะท้อน



VCC -> ต่อเข้า 5V

Trig (ทริกเกอร์)

-> ต่อเข้า Digital Pin (เช่น D8)
- ขานี้รับคำสั่งให้เริ่ม 'ตะโกน'

Echo (เอคโค่)

-> ต่อเข้า Digital Pin (เช่น D13)
- ขานี้รอ 'ฟัง' และจับเวลาการสะท้อนกลับ

GND -> ต่อเข้า GND

$$\text{Distance} = \frac{(\text{Duration} * 0.0343)}{2}$$

เวลาที่วัดได้จากคำสั่ง pulseIn()
(หน่วยเป็นไมโครวินาที)

ความเร็วเสียงในอากาศ
(343 เมตร/วินาที หรือ
0.0343 ซม./ไมโครวินาที)

หารสอง เพราะคลื่นเสียงเดินทาง
'ไป' และ 'กลับ' (Round trip)

```
digitalWrite(Trig_Pin, HIGH);  
delayMicroseconds(10);           // ส่งพัลส์ 10µs  
digitalWrite(Trig_Pin, LOW);  
  
long duration = pulseIn(Echo_Pin, HIGH); // จับเวลาสะท้อนกลับ  
int distance = (duration * 0.0343) / 2;   // แปลงเป็นเซนติเมตร
```

สามารถใช้ Library HCSR04.h หรือ NewPing เพื่อยุบโค้ดการคำนวณให้เหลือเพียง hc.dist() ได้

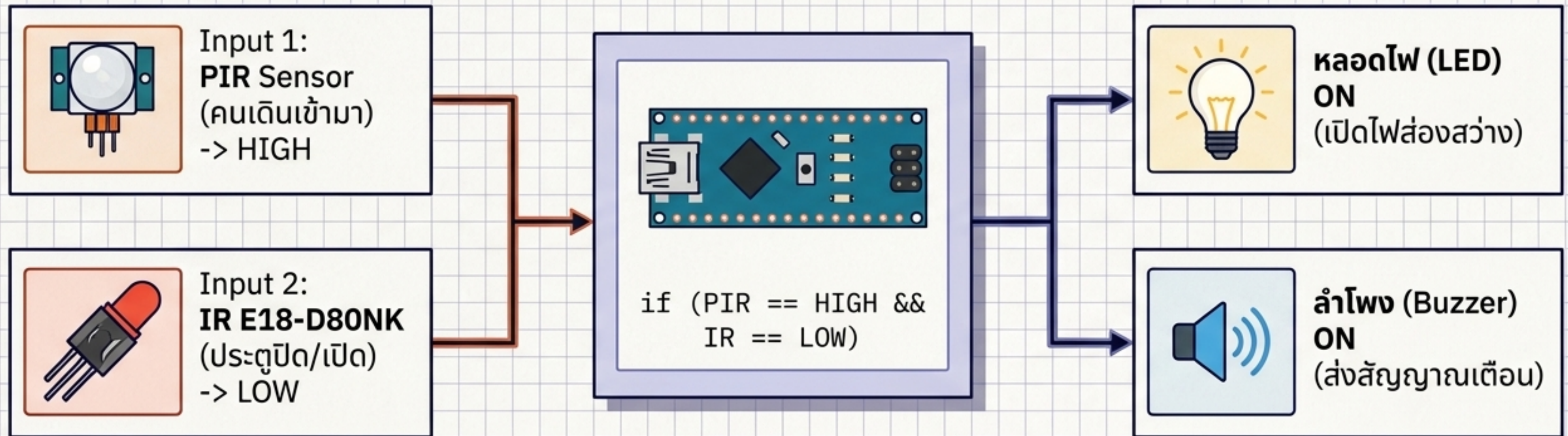
สรุปการเลือกใช้งาน

(Sensor Comparison Matrix)

คุณสมบัติ (Feature)	E18-D80NK (IR)	HC-SR501 (PIR)	HC-SR04 (Ultrasonic)
ฟิสิกส์หลัก (Physics)	แสงอินฟราเรด (Light)	รังสีความร้อน (Heat)	คลื่นเสียง (Sound)
ระยะการทำงาน (Range)	3 - 80 ซม. (ใกล้)	3 - 7 เมตร (ปานกลาง)	2 - 400 ซม. (ไกล/แม่นยำ)
ตรรกะ Output (Logic)	0 (LOW) = เจอวัตถุ	1 (HIGH) = เจอความเคลื่อนไหว	เวลา (μ s) -> ระยะทาง
จุดเด่น (Best For)	หลบหลีกสิ่งกีดขวาง, เดินตามเส้น	กันขโมย, เปิด-ปิดไฟอัตโนมัติ	วัดระดับน้ำ, วัดระยะห่างจุดจอดรถ
ฟังก์ชัน (Code)	<code>digitalRead()</code>	<code>digitalRead()</code>	<code>pulseIn()</code>

การบูรณาการระบบ (System Integration: Lab 4.2)

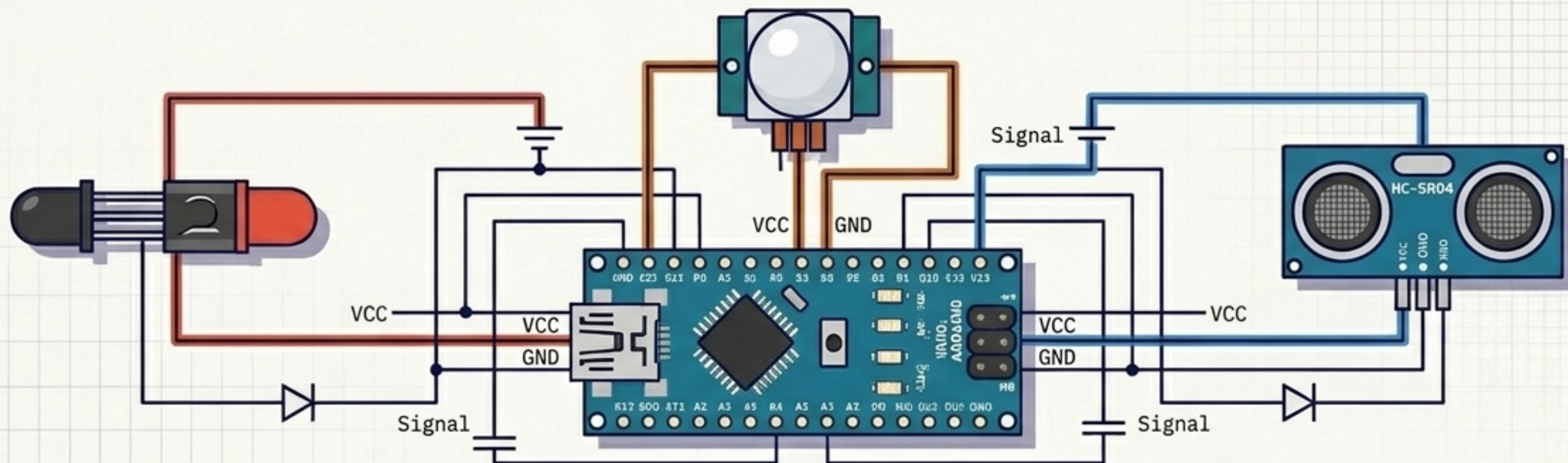
นำ 'การจับความเคลื่อนไหว (PIR)' มารวมกับ 'การจับความใกล้ชิด (IR)'



Insight

Sensor Fusion: การรวมเซนเซอร์หลายชนิดเข้าด้วยกัน ช่วยลดข้อผิดพลาด (False Alarms) และทำให้หุ่นยนต์ตัดสินใจได้ซับซ้อนและแม่นยำยิ่งขึ้น

จากเซนเซอร์... สู่การรับรู้ (From Sensors to Perception)



ไมโครคอนโทรลเลอร์เป็นเพียงสมองที่ถูกปิดตา
การเลือกใช้ฟิสิกส์ที่ถูกต้อง—แสง (IR) สำหรับการสัมผัส, ความร้อน (PIR) สำหรับการตรวจจับ
ชีวิต, และเสียง (Ultrasonic) สำหรับมิติพื้นที่—
คือการมอบประสาทสัมผัสให้เครื่องจักร

เครื่องมือแต่ละชิ้นมีข้อจำกัดทางฟิสิกส์ของตัวเอง
แต่วิศวกรที่เข้าใจข้อจำกัดเหล่านั้น
จะสามารถผสานเซนเซอร์เข้าด้วยกัน
เพื่อสร้างระบบอัจฉริยะที่ตอบสนองต่อโลกความ
จริงได้อย่างไร้ขีดจำกัด