

# Basic Arduino-Electronics Labs

ห้องปฏิบัติการระบบสมองกลฝังตัว (Embedded System Lab)  
สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ (ECE)  
คณะวิศวกรรมศาสตร์ มจพ.

เผยแพร่ครั้งแรก: 17 ธันวาคม 2554  
ปรับปรุงล่าสุด: 12 ตุลาคม 2557

# Objectives - วัตถุประสงค์

---

- เรียนรู้การเขียนโปรแกรมโดยใช้ภาษาระดับสูง (ภาษา C) สำหรับไมโครคอนโทรลเลอร์ ร่วมกับวงจรอิเล็กทรอนิกส์พื้นฐาน โดยใช้ “Arduino”
- เรียนรู้การต่อวงจรบนbreadบอร์ด โดยใช้อุปกรณ์หรือชิ้นส่วนอิเล็กทรอนิกส์พื้นฐานต่างๆ
- เรียนรู้การทำงานร่วมกันระหว่างซอฟต์แวร์ (โค้ดในไมโครคอนโทรลเลอร์) และฮาร์ดแวร์ (วงจรอิเล็กทรอนิกส์บนbreadบอร์ด)
- เรียนรู้การทดสอบและวัดสัญญาณทางไฟฟ้า เช่น แรงดันและกระแสในวงจร เพื่อตรวจสอบความถูกต้อง และทำความเข้าใจพฤติกรรมการทำงานของวงจร

# Lab List - รายการการทดลอง (1/3)

---

- Lab 1: Make LEDs Blinking!
- Lab 2: Press Button To Turn On LED!
- Lab 3: Press Button To Toggle LED!
- Lab 4: Press Two Buttons To Turn On LED!
- Lab 5: Use Trimpot To Turn On/Off LED!
- Lab 6: Show Voltage Level With 3 LEDs!
- Lab 7: Turn On LED Light When It Gets Dark!
- Lab 8: Turn On Cooling Fan When It Gets Warm!
- Lab 9: Make Dimming LED Light!
- Lab 10: Dim LED Light With Trimpot!
- Lab 11: Create Colored Light with RGB LED!
- Lab 12: Make Random Light with RGB LED!

# Lab List – รายการทดลอง (2/3)

---

- Lab 13: Make Random Light with 8 LEDs!
- Lab 14: Create Running LED Patterns!
- Lab 15: Create Pausable Running LEDs!
- Lab 16: Count Up In Binary!
- Lab 17: Count Up or Down In Binary!
- Lab 18: Turn On LEDs with DIP Switch!
- Lab 19: Make An LED Selector Switch!
- Lab 20: Count Up From 0 to 9 in Decimal!
- Lab 21: Adjust Motor Speed with Trimpot!
- Lab 22: Turn On/Off Motor with NPN Transistor!
- Lab 23: Turn On/Off Motor with N-MOSFET!
- Lab 24: Turn On/Off Motor with P-MOSFET!

# Lab List – รายการทดลอง (3/3)

---

- Lab 25: Build Your Optical Proximity Sensor!
- Lab 26: Build Your Optical Switch!
- Lab 27: Toggle Red/Green LEDs with Relay!
- Lab 28: Make Countdown Timer with Buzzer Alarm!

# คำแนะนำในการต่อวงจรบนบอร์ด

---

- ศึกษาและทำความเข้าใจผังวงจร (Schematic / Circuit Diagram) และเลือกใช้อุปกรณ์ให้ถูกต้อง
- ต่อวงจรตามผังวงจรที่ให้ไว้ แต่ยังไม่ต้องต่อแรงดันไฟเลี้ยง (VCC, Gnd) ให้กับบอร์ด
- ป้อนแรงดันไฟเลี้ยง ก็ต่อเมื่อได้ตรวจสอบความถูกต้องของวงจรในเบื้องต้นอย่างละเอียดถี่ถ้วนแล้ว
- คอมไพล์และดาวน์โหลดโค้ดโปรแกรมตัวอย่างลงในตัวชิปบนบอร์ด Arduino
- ป้อนไฟเลี้ยง (VCC, GND) จากบอร์ด Arduino ให้วงจรบนบอร์ด

# โจทย์ที่มอบหมาย

---

- ให้ผู้เรียนทำการทดลอง โดยให้ทำเดี่ยว หรือทำเป็นกลุ่ม (2 คน)
- ทำตามโจทย์การทดลองทั้งหมด
- ลองใช้โปรแกรม **Fritzing** ออกแบบการต่อวงจรบนเบรด์บอร์ด  
(=> **Breadboard View**) ให้สอดคล้องกับผังวงจรที่ให้ไว้ในแต่ละการทดลอง
- ศึกษาการทำงานของโค้ด Arduino และเขียนคำอธิบายโค้ดตามที่ตนเองเข้าใจ
- ต่อวงจรจริงบนเบรด์บอร์ด และทดลองโค้ด Arduino ที่ให้ไว้
- สังเกตพฤติกรรมการทำงานของวงจร และจดบันทึก

# Lab 1: Make LEDs Blinking!

---

- **วัตถุประสงค์**
  - ต่่วงจรโดยใช้ ไดโอดเปล่งแสง (LED) และตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดิจิทัล หมายเลข 8) สำหรับบอร์ด Arduino
  - ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรมอนิเตอร์บอร์ด
- **คำอธิบายโค้ดสำหรับ Arduino**
  - โค้ดทำหน้าที่สร้างเอาต์พุตแบบดิจิทัล และสลับค่าทางลอจิกระหว่าง 'Low' และ 'High' ไปมา โดยเว้นระยะเวลาคงที่ (ใช้คำสั่ง delay)
- **คำแนะนำ:** การต่่วงจรสำหรับ LED ทำได้สองแบบ คือ
  - แบบที่ 1: เอาต์พุตของ Arduino จะต้องเป็น 1 (High) จึงจะทำให้ LED สว่าง
  - แบบที่ 2: เอาต์พุตของ Arduino จะต้องเป็น 0 (Low) จึงจะทำให้ LED สว่าง
  - สามารถใช้ LED สีแดงและสีเขียว (พร้อมตัวต้านทานอีกสองตัว) ได้ในวงจรเดียวกัน

# Lab 1: Arduino Code

---

```
/*  
 * Example: Make LEDs Blinking.  
 */  
const byte LED_PIN = 8;      // use Digital Pin 8 (D8) for LED blinking  
int DELAY_MS = 500;         // set delay time (in msec) for LED blinking rate  
  
void setup() {  
    pinMode( LED_PIN, OUTPUT );    // use LED Pin as output  
}  
  
void loop() {  
    digitalWrite( LED_PIN, HIGH ); // output High: Turn on Red LED  
    delay( DELAY_MS );              // wait for a few hundred msec  
    digitalWrite( LED_PIN, LOW );  // output Low: Turn off Red LED  
    delay( DELAY_MS );              // wait for a few hundred msec  
}
```

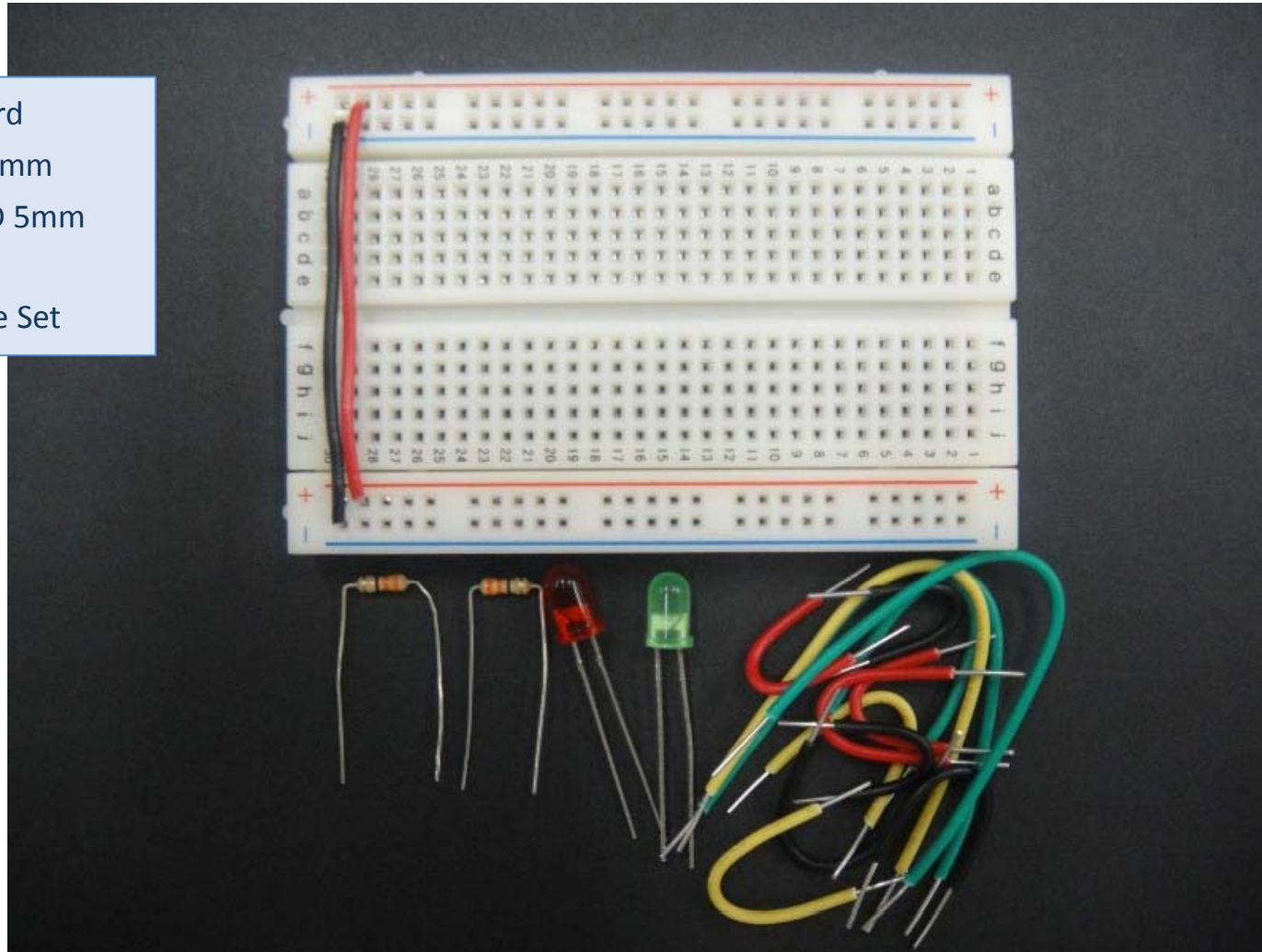
# Lab 1: Arduino Code

```
/*  
 * Example: Make LEDs Blinking.  
 */  
#define LED_PIN 8 // use Digital Pin 8 (D8) for LED blinking  
#define DELAY_MS 500 // set delay time (in msec) for LED blinking rate  
  
void setup() {  
    pinMode( LED_PIN, OUTPUT ); // use LED Pin as output  
}  
  
void loop() {  
    digitalWrite( LED_PIN, HIGH ); // output High: Turn on Red LED  
    delay( DELAY_MS ); // wait for a few hundred msec  
    digitalWrite( LED_PIN, LOW ); // output Low: Turn off Red LED  
    delay( DELAY_MS ); // wait for a few hundred msec  
}
```

คำสั่ง #define เป็นการกำหนดชื่อและค่าให้สัญลักษณ์ (ในภาษาซี เรียกว่า macro)

# Lab 1: Part List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x Green LED 5mm
- 2x 330 Ohm
- 1x Jump Wire Set



# Lab 1: Schematic – ผังวงจร

1 = Gnd

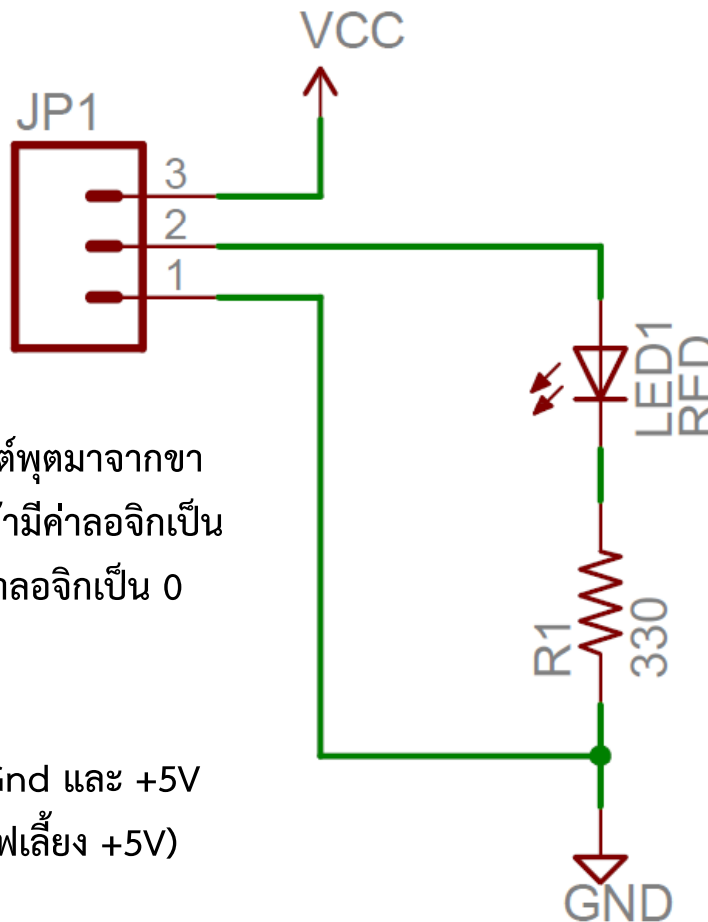
2 = Arduino Digital Pin 8

3 = VCC (+5V)

จุดเชื่อมต่อสัญญาณและ  
ไฟเลี้ยงไปยังบอร์ด Arduino

ให้ต่อขาที่ 2 ของ JP1 เพื่อรับสัญญาณเอาต์พุตมาจากขา  
ดิจิตอลหมายเลข 8 ของบอร์ด Arduino ถ้ามีค่าลอจิกเป็น  
1 (High) จะทำให้ LED “สว่าง” แต่ถ้ามีค่าลอจิกเป็น 0  
(Low) จะทำให้ LED “ดับ”

สำหรับขาที่ 1 และ 3 ของ JP1 ให้ต่อกับ Gnd และ +5V  
ตามลำดับ (สำหรับบอร์ด Arduino ที่ใช้ไฟเลี้ยง +5V)



# Lab 1: Schematic – ผังวงจร

ให้ต่อขาที่ 2 ของ JP1 เพื่อรับสัญญาณมาจากขาติจิตอล หมายเลข 8 ของบอร์ด Arduino ถ้ามีค่าลอจิกเป็น 1 (High) จะทำให้ LED “ดับ” แต่ถ้ามีค่าลอจิกเป็น 0 (Low) จะทำให้ LED “สว่าง”

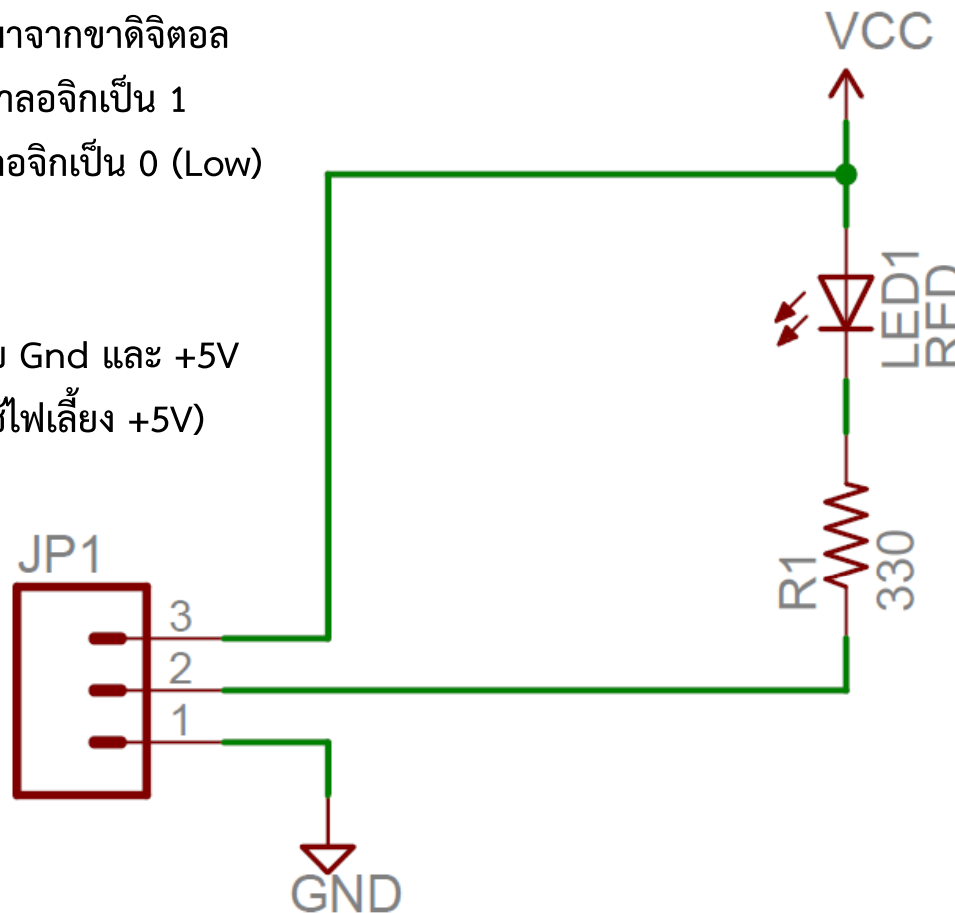
สำหรับขาที่ 1 และ 3 ของ JP1 ให้ต่อกับ Gnd และ +5V ตามลำดับ (สำหรับบอร์ด Arduino ที่ใช้ไฟเลี้ยง +5V)

จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

1 = Gnd

2 = Arduino Digital Pin 8

3 = VCC (+5V)



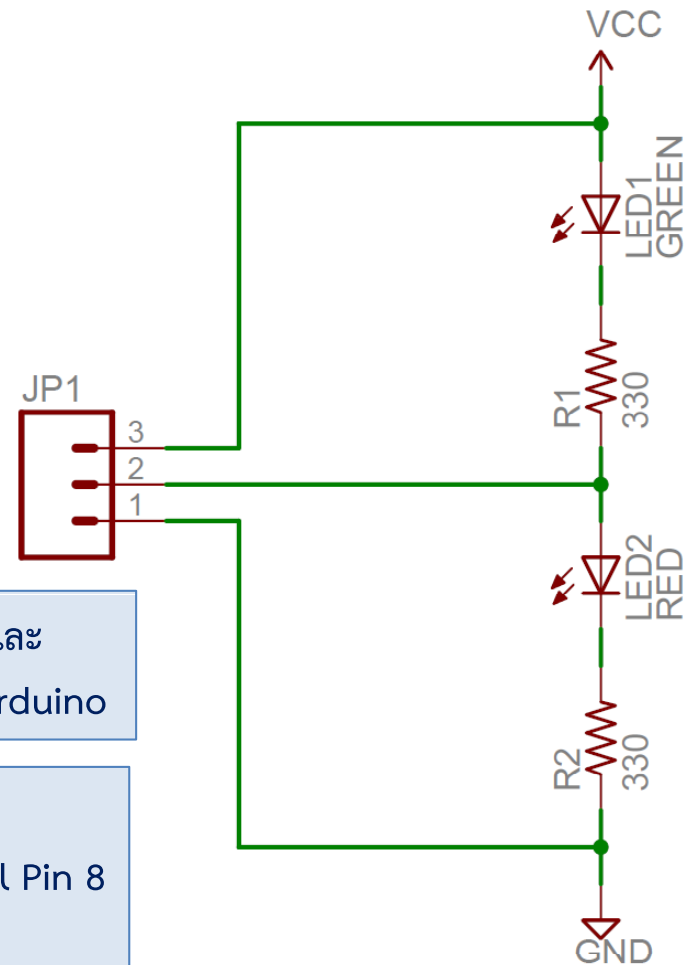
# Lab 1: Schematic – ผังวงจร

ให้ต่อขาที่ 2 ของ JP1 เพื่อรับสัญญาณมาจากขาติจิตอล หมายเลข 8 ของบอร์ด Arduino ถ้ามีค่าลอจิกเป็น 1 (High) จะทำให้ LED สีเขียว “ดับ” และ สีแดง “สว่าง” แต่ถ้ามีค่าลอจิกเป็น 0 (Low) จะทำให้ LED สีเขียว “สว่าง” และ สีแดง “ดับ”

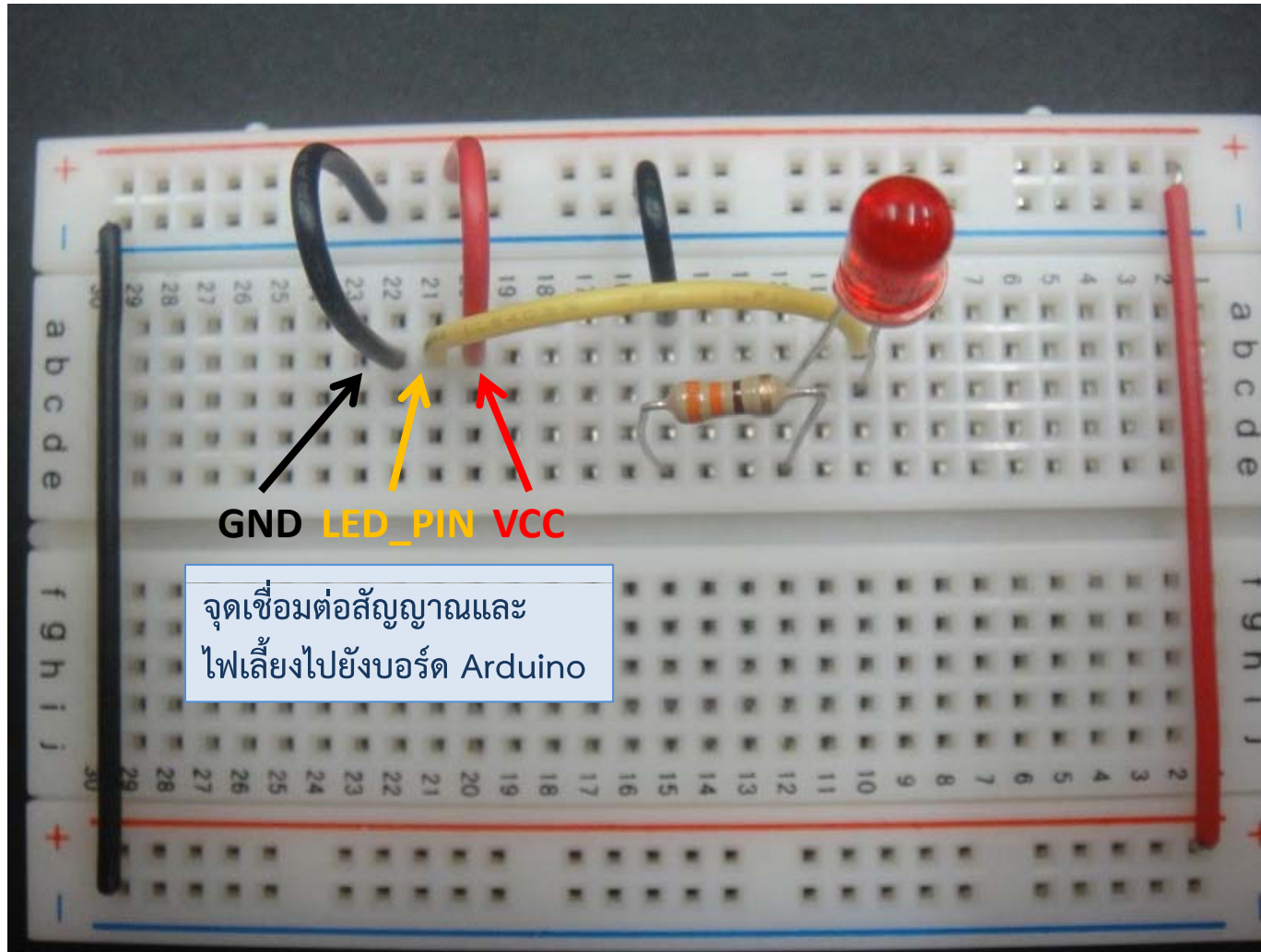
สำหรับขาที่ 1 และ 3 ของ JP1 ให้ต่อกับ Gnd และ +5V ตามลำดับ (สำหรับบอร์ด Arduino ที่ใช้ไฟเลี้ยง +5V)

จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

- 1 = Gnd
- 2 = Arduino Digital Pin 8
- 3 = VCC (+5V)



# Lab 1: ตัวอย่างการต่อวงจรบนbreadบอร์ด







## Lab 2: Press The Button To Turn On LED!

---

- **วัสดุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาติจิตอล หมายเลข 8) และปุ่มกด + ตัวต้านทาน 10k โอห์ม เป็นอินพุตแบบดิจิทัล (ขาติจิตอล หมายเลข 9) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนเบรด์บอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตจากปุ่มกด กดปุ่มทำงานแบบ Active-Low โดยจะอ่านค่าได้ลอจิก 0 (Low) และถ้าไม่กดปุ่ม จะอ่านค่าได้ลอจิก 1 (High) แล้วนำค่าที่อ่านได้ไปกำหนดสถานะทางลอจิกให้ LED ที่เป็นเอาต์พุต

## Lab 2: Arduino Code

---

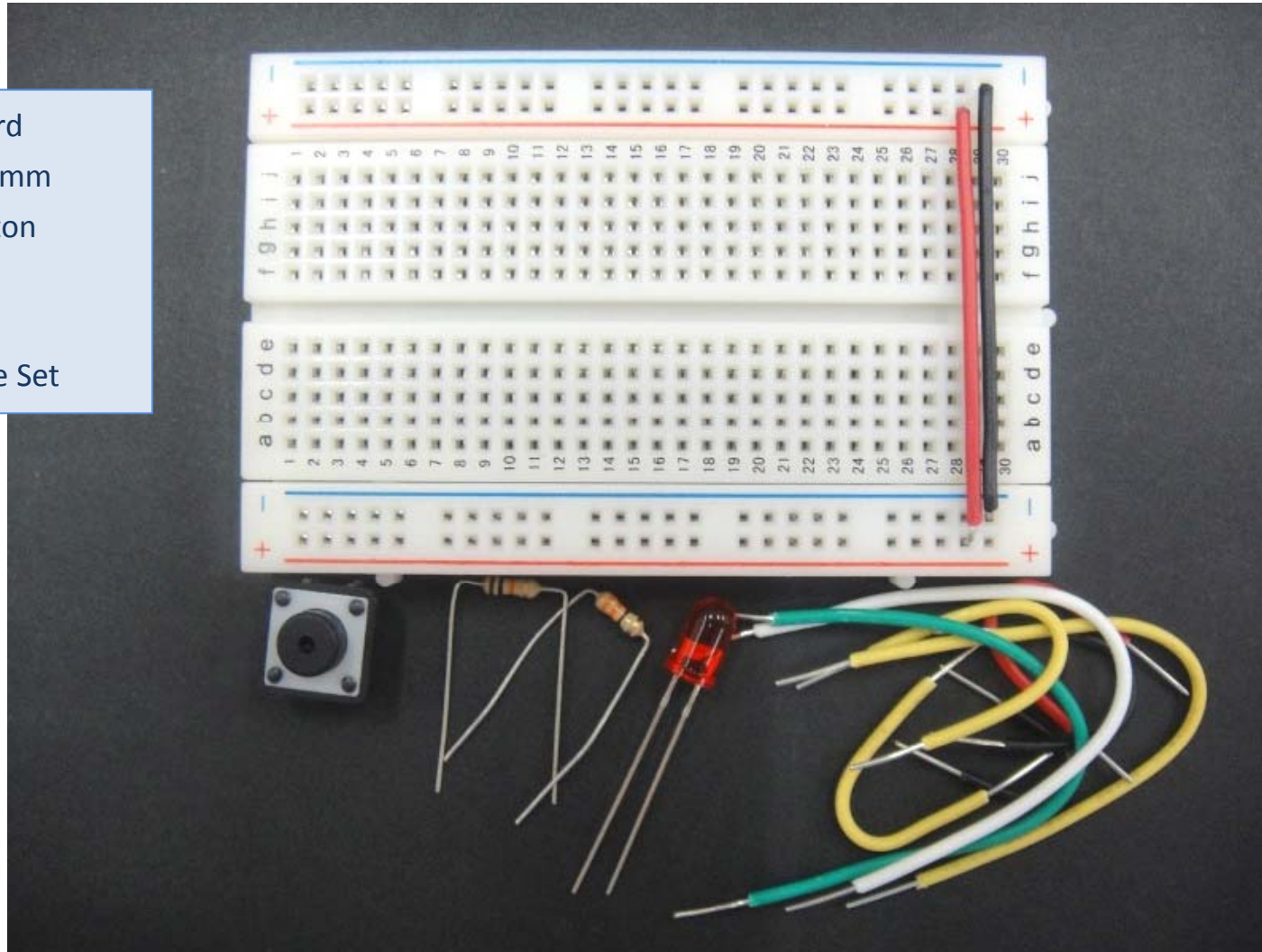
```
/* Example: Press and Hold Down Button To Turn On LED.
 * Hold the button pressed => the red LED is turned ON.
 * Release the button => the LED is OFF.
 */
const byte LED_PIN = 8;      // use Digital Pin 8 for LED Output
const byte BUTTON_PIN = 9;  // use Digital Pin 9 for Push Button Input

void setup() {
    pinMode( LED_PIN, OUTPUT );      // use LED Pin as output
    pinMode( BUTTON_PIN, INPUT );    // use Button Pin as input (active-Low)
}

void loop() {
    int value = digitalRead( BUTTON_PIN );    // read digital input from button
    digitalWrite( LED_PIN, !value );         // output negated value to LED
    delay( 100 );                            // wait for 100 msec before next reading
}
```

## Lab 2: Part List – รายการอุปกรณ์

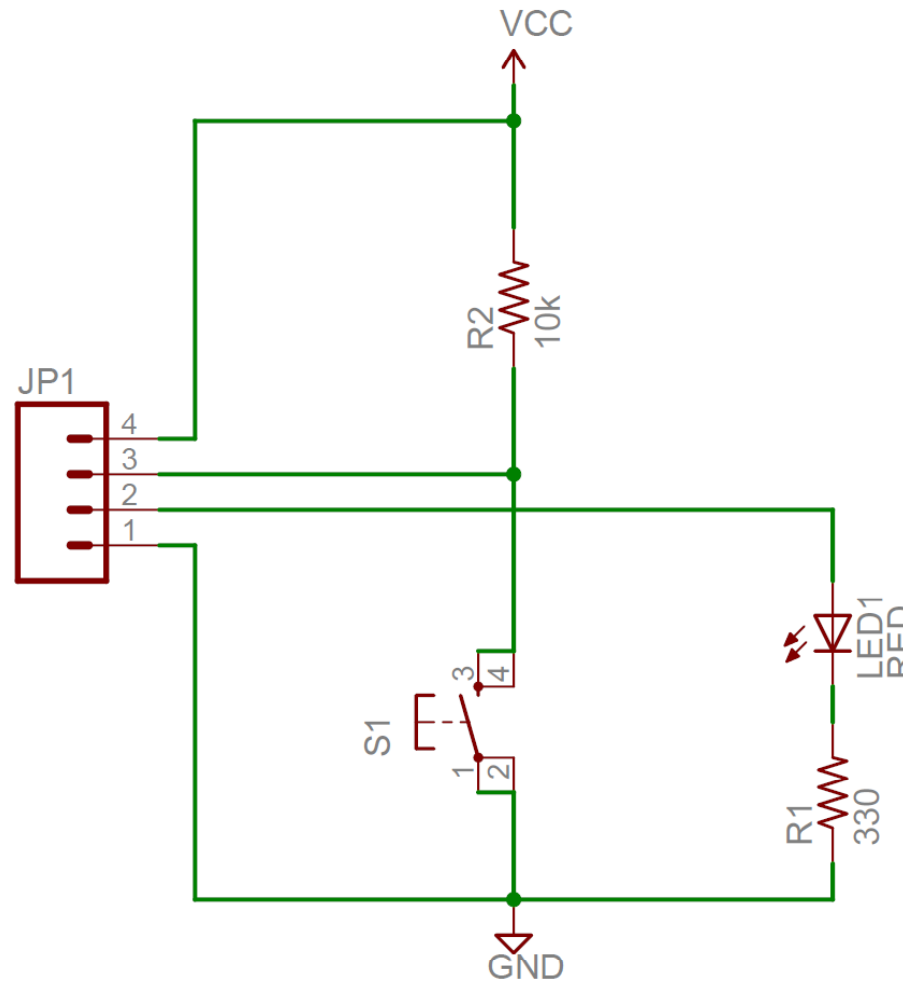
- 1x Breadboard
- 1x Red LED 5mm
- 1 x Push Button
- 1x 330 Ohm
- 1x 10k Ohm
- 1x Jump Wire Set



# Lab 2: Schematic – ผังวงจร

จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

- 1 = Gnd
- 2 = Arduino Digital Pin 8
- 3 = Arduino Digital Pin 9
- 4 = VCC (+5V)



# Lab 3: Press The Button To Toggle LED!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาติจิตอล หมายเลข 8) และปุ่มกด + ตัวต้านทาน 10k โอห์ม เป็นอินพุตแบบดิจิทัล (ขาติจิตอล หมายเลข 9) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตจากปุ่มกด (กดปุ่มทำงานแบบ Active-Low) เมื่อกดปุ่มแล้วปล่อยในแต่ละครั้ง จะทำให้เกิดการเปลี่ยนสถานะของการติดหรือดับที่ LED สลับกันระหว่าง “ติด” และ “ดับ”

- **คำแนะนำ:** การต่อบอร์ดสำหรับ LED ทั้งสีเขียวและสีแดง

- เอาต์พุตของ Arduino จะต้องเป็น 1 (High) จึงจะทำให้ LED สีเขียว “ติด” และ สีแดง “ดับ” แต่ถ้าเอาต์พุตเป็น 0 (Low) จะให้ผลตรงข้าม

# Lab 3: Arduino Code

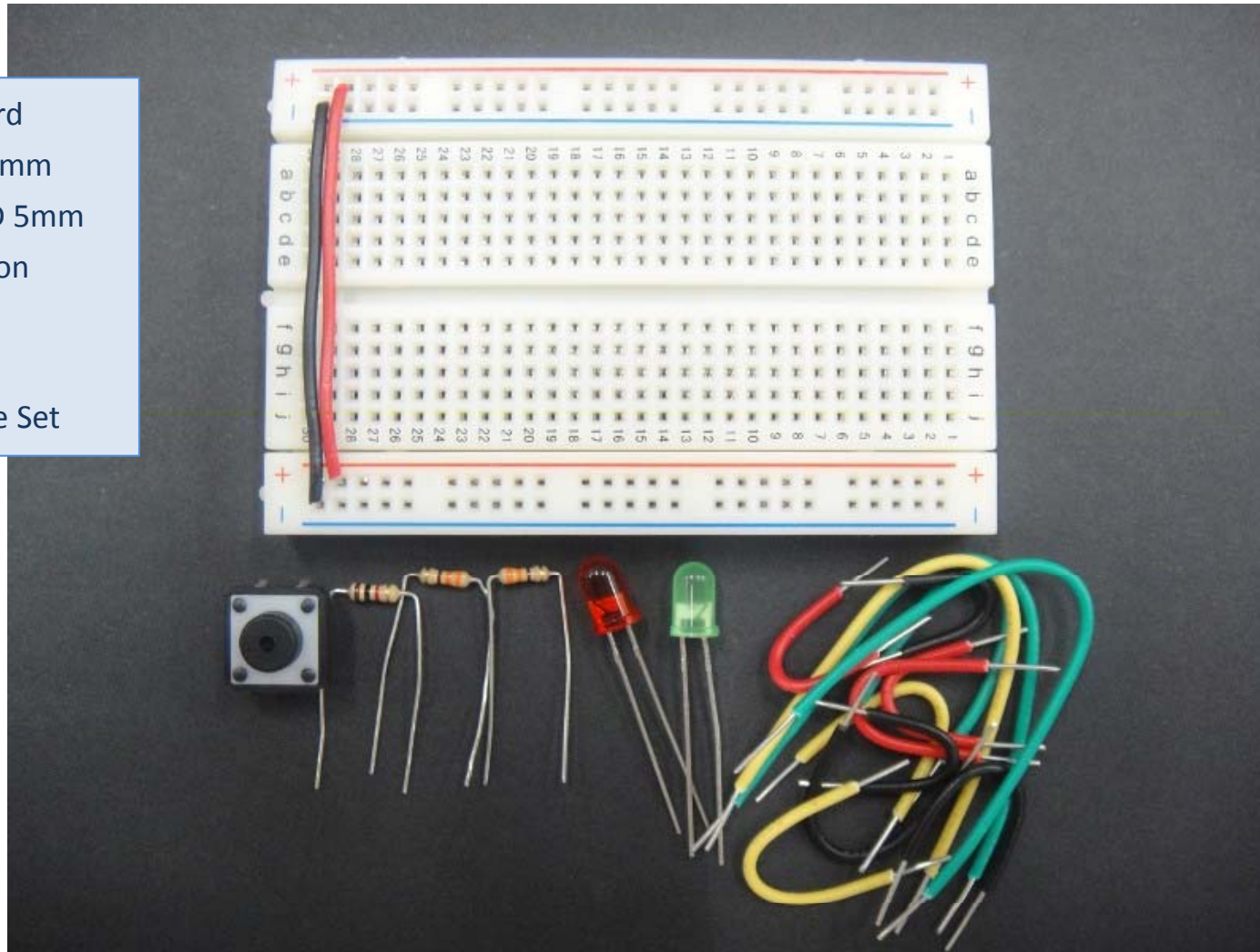
```
/* Example: Push Button To Toggle LED.
 * Each time the button is pressed and then released, the LED state is toggled.
 */
const byte LED_PIN = 8;      // use Digital Pin 8 (D8) for LED Output
const byte BUTTON_PIN = 9;  // use Digital Pin 9 (D9) for Push Button Input
bool state = false;        // LED state (initially OFF)

void setup() {
  pinMode( LED_PIN, OUTPUT );    // use LED Pin as output
  pinMode( BUTTON_PIN, INPUT );  // use Button Pin as input (active-Low)
  digitalWrite( LED_PIN, state ); // set LED to its initial state
}

void loop() {
  // wait until the button is pressed.
  while ( digitalRead( BUTTON_PIN ) == HIGH ) delay( 10 );
  // wait until the button is released.
  while ( digitalRead( BUTTON_PIN ) == LOW ) delay( 10 );
  state = !state;              // toggle the state
  digitalWrite( LED_PIN, state ); // use the state as output to LED pin
}
```

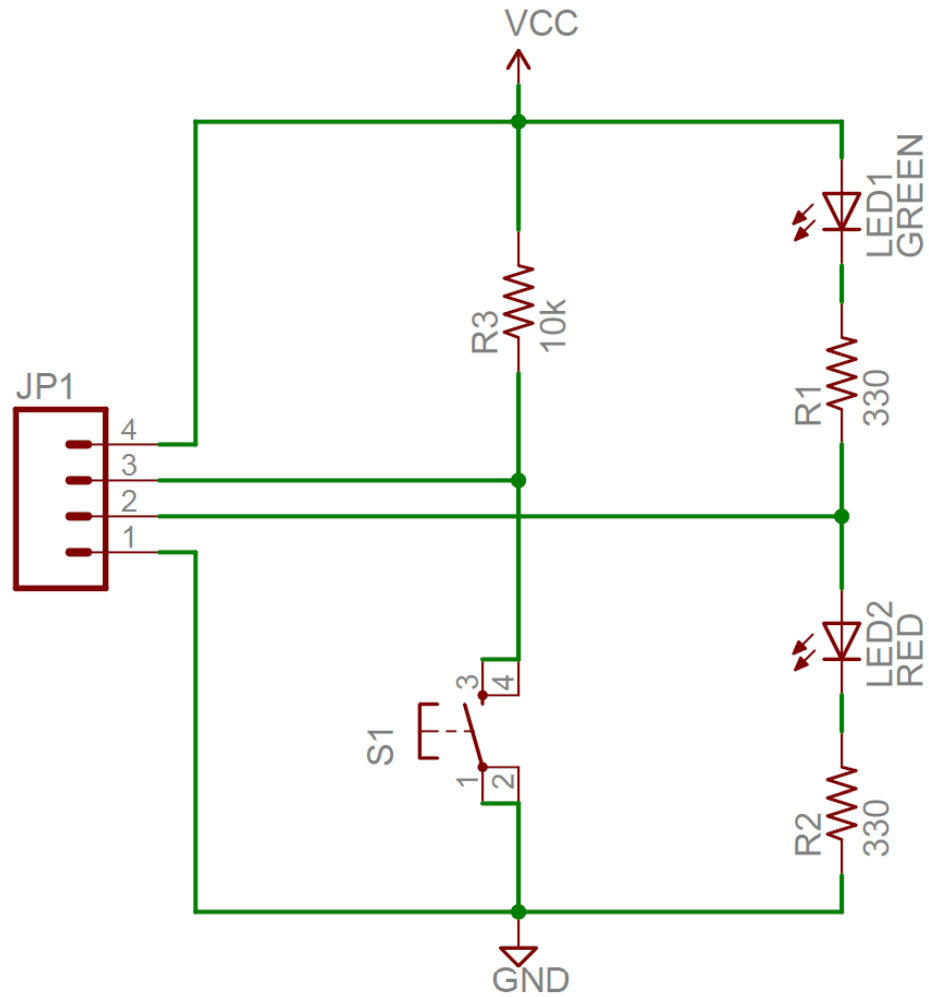
# Lab 3: Part List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x Green LED 5mm
- 1x Push Button
- 2x 330 Ohm
- 1x 10k Ohm
- 1x Jump Wire Set



# Lab 3: Schematic – ผังวงจร

- จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino
- 1 = Gnd
- 2 = Arduino Digital Pin 8
- 3 = Arduino Digital Pin 9
- 4 = VCC (+5V)



# Lab 4: Press Two Buttons To Turn On LED!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดีจิตอล หมายเลข 8) และปุ่มกด + ตัวต้านทาน 10k โอห์ม เป็นอินพุตแบบดิจิทัล จำนวน 2 ชุด (ขาดีจิตอล หมายเลข 9 และ 10) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- ทำหน้าที่อ่านอินพุตจากปุ่มกด (กดปุ่มทำงานแบบ Active-Low) ทั้งสองปุ่มตามลำดับ และนำมาตรวจสอบ ถ้ามีการกดปุ่มทั้งสองค้างไว้ จะทำให้เอาต์พุตซึ่งเป็น LED “ติด” แต่ถ้าไม่เป็นไปตามเงื่อนไข จะทำให้ LED “ดับ”

# Lab 4: Arduino Code

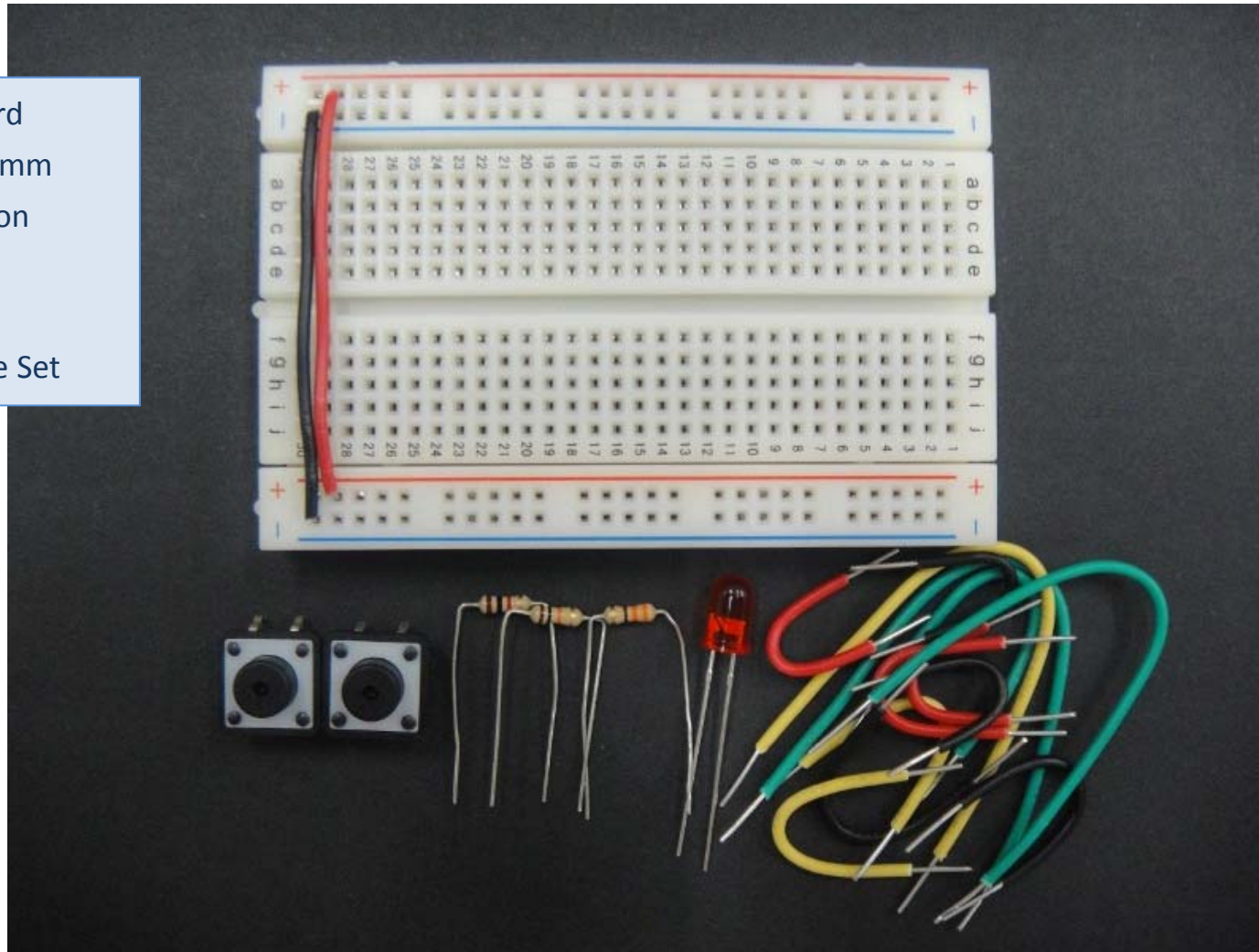
```
/*
 * Example: Press two buttons simultaneously and hold them down to turn on LED.
 */
const byte LED_PIN = 8;      // use Digital Pin 8 (D8) for LED Output
const byte BUTTON1_PIN = 9;  // use Digital Pin 9 (D9) for Push Button 1 Input
const byte BUTTON2_PIN = 10; // use Digital Pin 10 (D10) for Push Button 2 Input

void setup() {
  pinMode( LED_PIN, OUTPUT );      // use LED Pin as output
  pinMode( BUTTON1_PIN, INPUT );   // use Button 1 Pin as input (active-low)
  pinMode( BUTTON2_PIN, INPUT );   // use Button 2 Pin as input (active-low)
  digitalWrite( LED_PIN, LOW );    // turn LED off
}

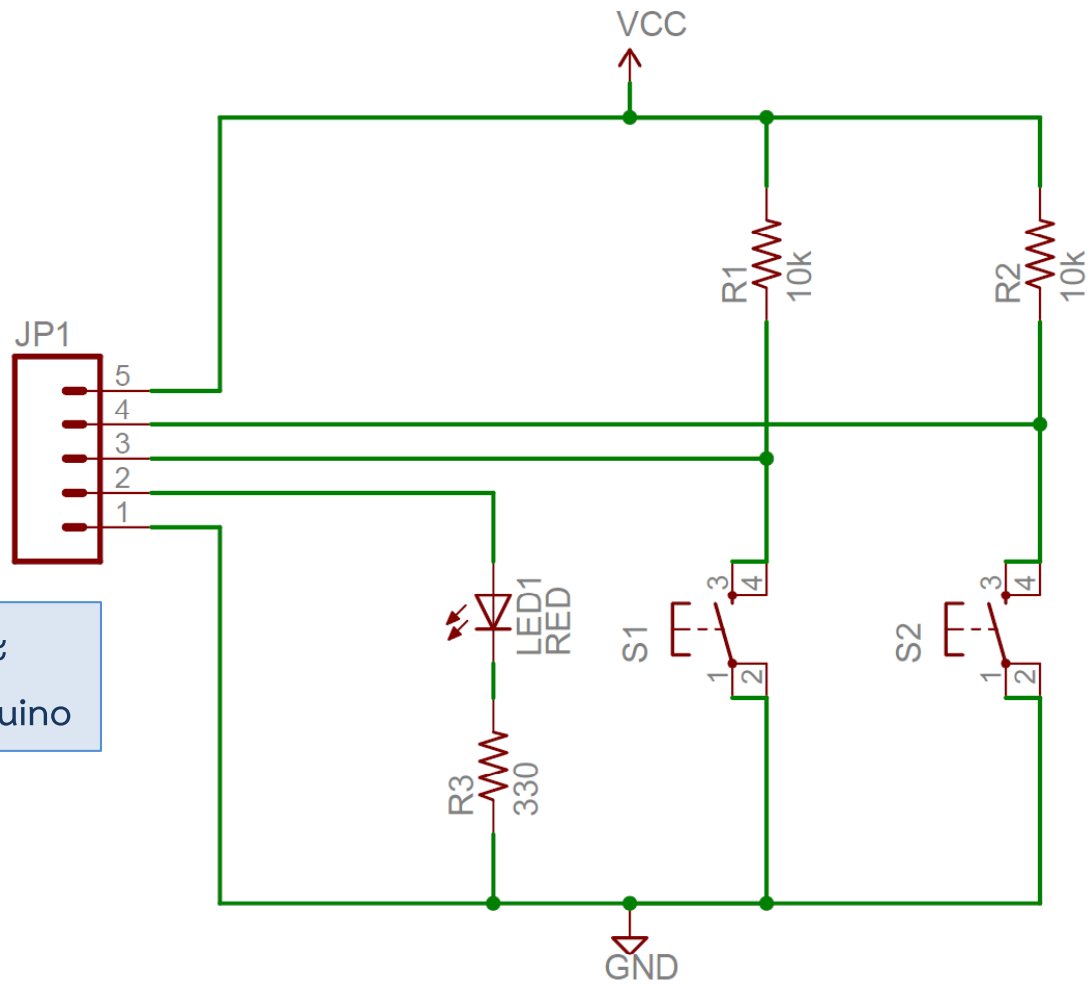
void loop() {
  if ( digitalRead( BUTTON1_PIN ) == LOW
      && digitalRead( BUTTON2_PIN ) == LOW ) // Are both buttons pressed?
    digitalWrite( LED_PIN, HIGH ); // turn on LED
  else
    digitalWrite( LED_PIN, LOW ); // turn off LED
}
```

# Lab 4: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 2x Push Button
- 1x 330 Ohm
- 2x 10k Ohm
- 1x Jump Wire Set



# Lab 4: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 5: Use Trimpot To Turn On/Off LED!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดิจิทัล หมายเลข 8) และตัวต้านทานปรับค่าได้ (Trimpot) 10k โอห์ม (หรือสูงกว่า) เป็นอินพุตแบบแอนะล็อก (ขาแอนะล็อก หมายเลข 0) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

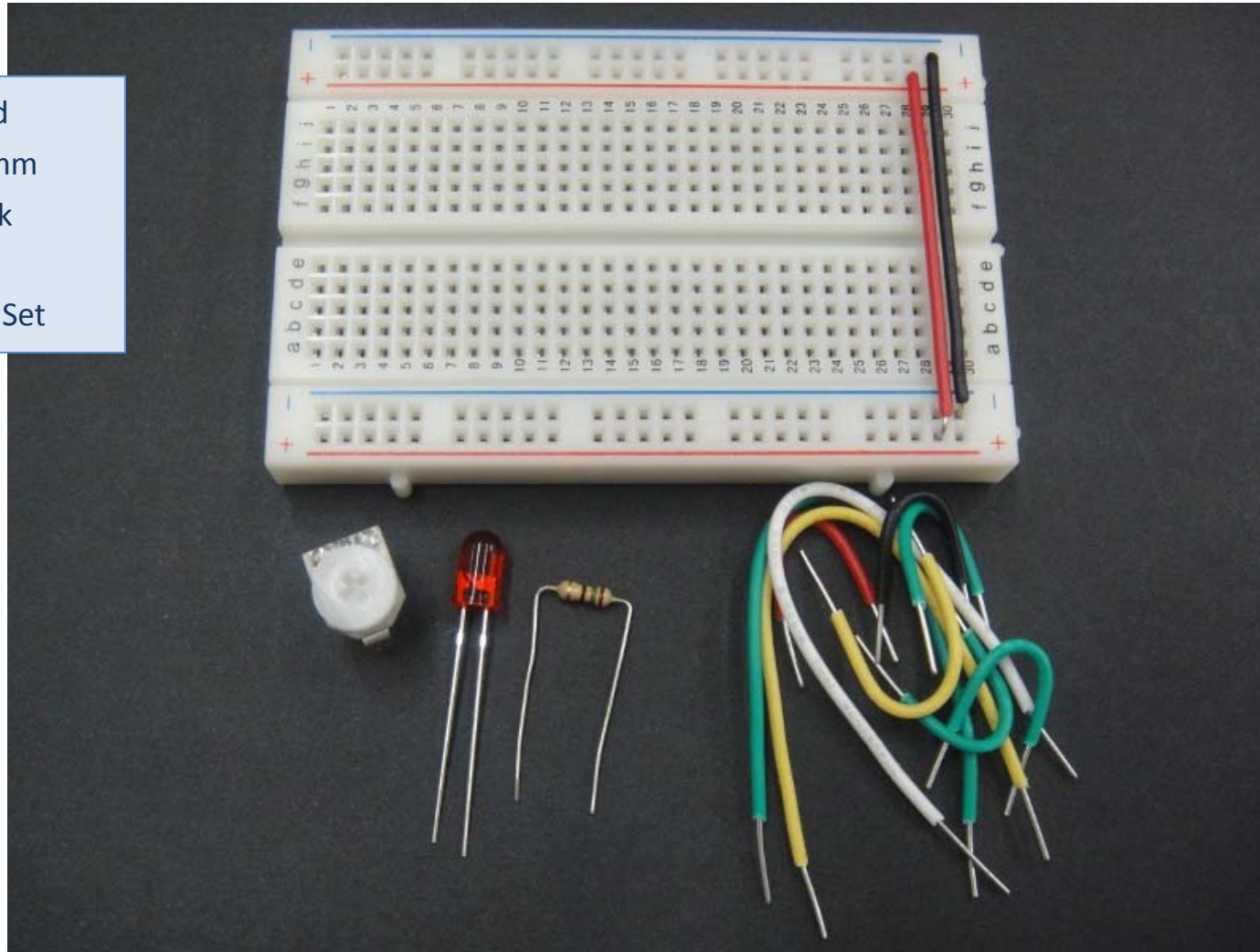
- โค้ดทำหน้าที่อ่านอินพุตแบบแอนะล็อก แล้วนำค่าที่ได้ (อยู่ในช่วง 0 - 1023) มาเปรียบเทียบกับค่าที่กำหนดไว้คือ 511 ซึ่งหมายถึงแรงดันอินพุต ประมาณ 2.5V ถ้าอินพุตมีค่ามากกว่า จะทำให้ LED “ติด” แต่ถ้าไม่ใช่ จะทำให้ LED “ดับ”
- หมายเหตุ: ไฟเลี้ยงที่ป้อนให้วงจบบอร์ดบอร์ด (VCC) คือ +5V ซึ่งตรงกับไฟเลี้ยงของบอร์ด Arduino ที่ใช้ในการทดลอง

# Lab 5: Arduino Code

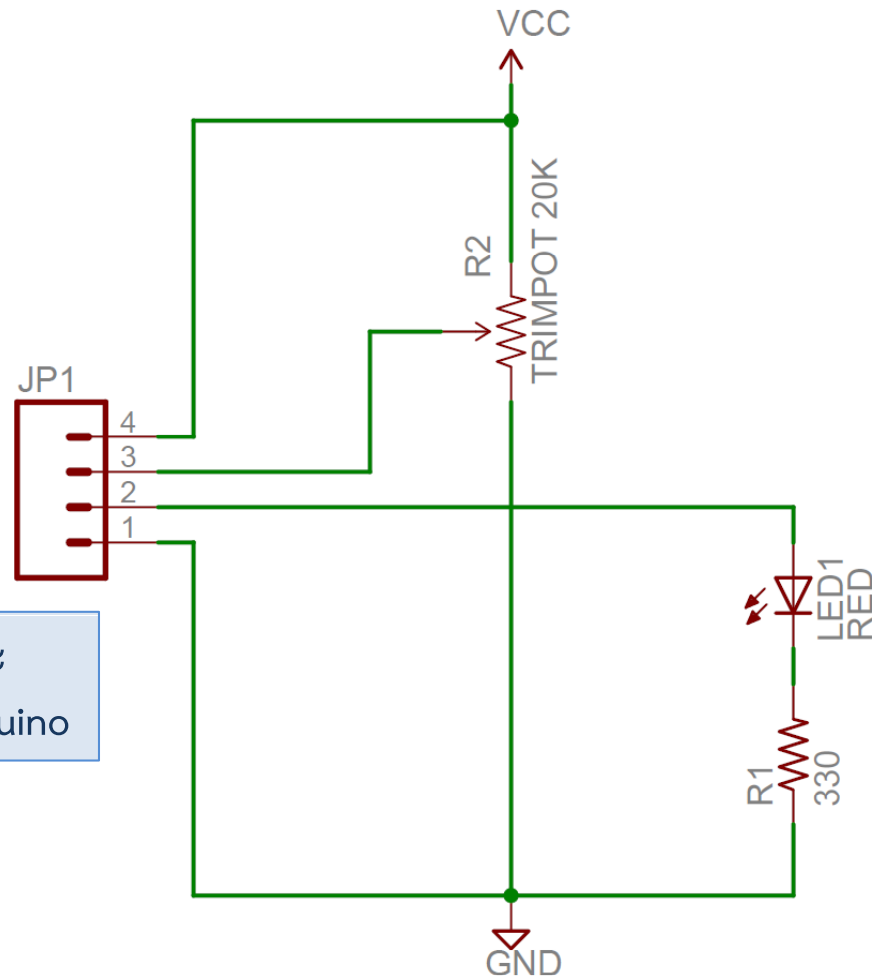
```
/*  
 * Example: Read analog input (voltage) and turn on LED if voltage > 2.5V  
 */  
const byte LED_PIN = 8;      // use Digital Pin 8 (D8) for LED Output  
const byte A0_PIN = 0;      // use Analog Pin 0 (A0) for Analog Input  
  
void setup() {  
  pinMode( LED_PIN, OUTPUT ); // use LED Pin as output  
  analogReference( DEFAULT ); // use +5V voltage reference  
}  
  
void loop() {  
  int value = analogRead( A0_PIN ); // read input value from analog input  
  if ( value > 511 )                // check whether input voltage > 2.5V  
    digitalWrite( LED_PIN, HIGH ); // turn on LED  
  else  
    digitalWrite( LED_PIN, LOW );  // turn off LED  
  delay( 100 );                     // wait for 100 msec before continuing  
}
```

# Lab 5: Parts List – รายการอุปกรณ์

1x Breadboard  
1x Red LED 5mm  
1x Trimpot 10k  
1x 330 Ohm  
1x Jump Wire Set



# Lab 5: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

## Lab 6: Show Voltage Level With 3 LEDs!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล 3 ชุด (ขาดีจิตอล หมายเลข 8, 9, 10) และตัวต้านทานปรับค่าได้ (Trimpot) 10k โอห์ม (หรือสูงกว่า) เป็นอินพุตแบบแอนะล็อก (ขาแอนะล็อก หมายเลข 0) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรมอนิเตอร์บอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตแบบแอนะล็อก แล้วนำค่าที่ได้ (อยู่ในช่วง 0 - 1023) มาเปรียบเทียบเพื่อแบ่งแรงดันออกเป็น 4 ช่วง (0V-1.25V, 1.25V-2.5V, 2.5V-3.75V และ 3.75V-5V) และใช้ LED จำนวน 3 ตัว ในการแสดงช่วงของแรงดันเป็นเอาต์พุต
- หมายเหตุ: ไฟเลี้ยงที่ป้อนให้วงจรมอนิเตอร์บอร์ด (VCC) คือ +5VDC ซึ่งตรงกับไฟเลี้ยงของบอร์ด Arduino ที่ใช้ในการทดลอง

## Lab 6: Arduino Code (1/2)

---

```
/*  
 * Example: Show Voltage Level With 3 LEDs  
 */  
const byte LED1_PIN = 8;    // use Digital Pin 8 (D8) for LED1 Output  
const byte LED2_PIN = 9;    // use Digital Pin 9 (D9) for LED2 Output  
const byte LED3_PIN = 10;   // use Digital Pin 10 (D10) for LED3 Output  
const byte A0_PIN = 0;      // use Analog Pin 0 (A0) for Analog Input  
  
void setup() {  
    pinMode( LED1_PIN, OUTPUT );    // use LED 1 Pin as output  
    pinMode( LED2_PIN, OUTPUT );    // use LED 2 Pin as output  
    pinMode( LED3_PIN, OUTPUT );    // use LED 3 Pin as output  
    analogReference( DEFAULT );     // use +5V voltage reference  
}  
  
// Code continues on the next page.
```

## Lab 6: Arduino Code (2/2)

---

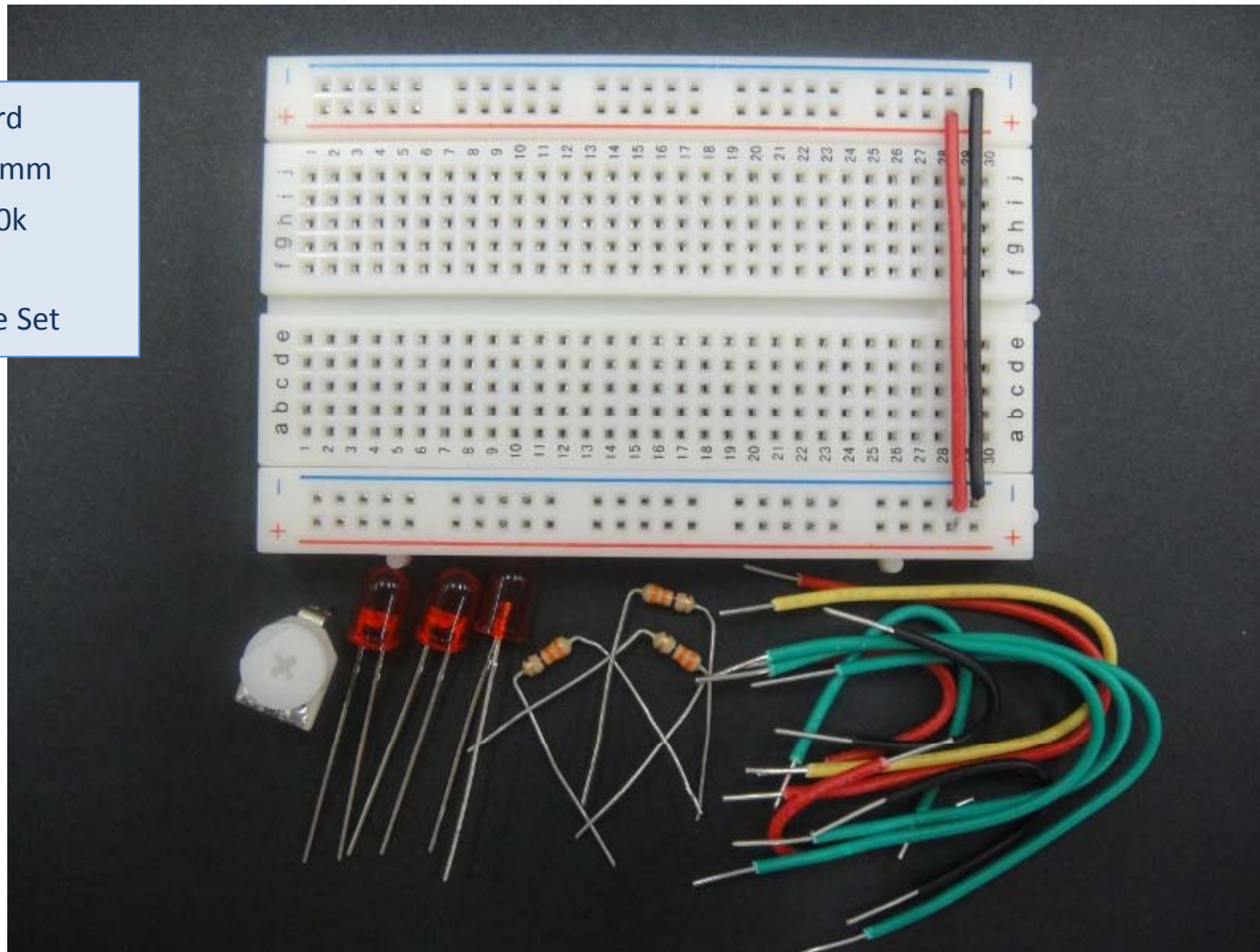
```
// Code continues from the previous page.

void loop() {
    int value = analogRead( A0_PIN ); // read value from analog input
                                        // (between 0..1023)

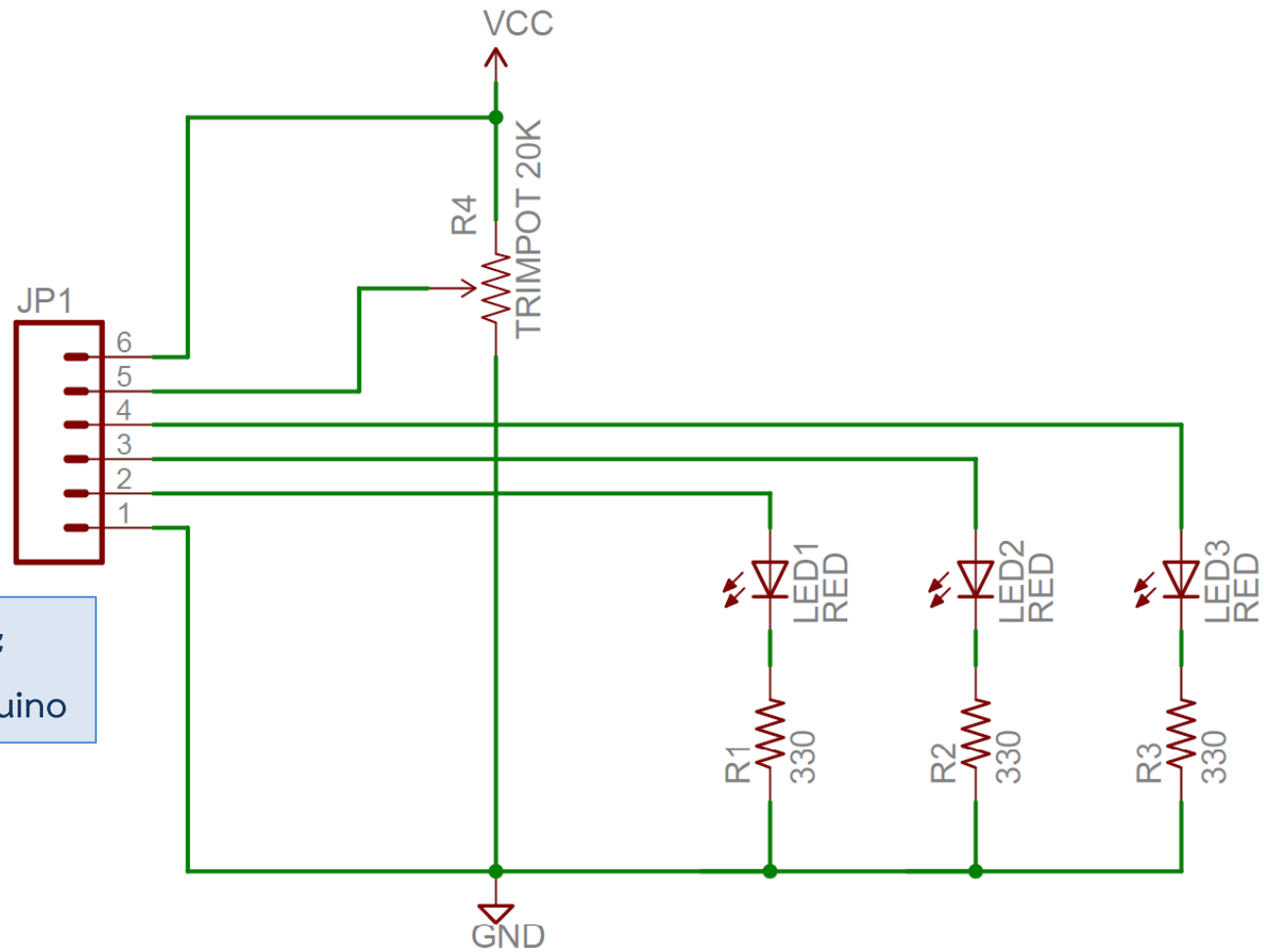
    digitalWrite( LED1_PIN, LOW ); // turn off LED1
    digitalWrite( LED2_PIN, LOW ); // turn off LED2
    digitalWrite( LED3_PIN, LOW ); // turn off LED3
    if ( value > 255 ) // input voltage > approx. 1.25V
        digitalWrite( LED1_PIN, HIGH ); // turn on LED1
    if ( value > 511 ) // input voltage > approx. 2.5 V
        digitalWrite( LED2_PIN, HIGH ); // turn on LED2
    if ( value > 768 ) // input voltage > approx. 3.75V
        digitalWrite( LED3_PIN, HIGH ); // turn on LED3
    delay( 100 ); // wait for 100 msec before continuing
}
```

# Lab 6: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 3x Red LED 5mm
- 1x Trimpot 10k
- 3x 330 Ohm
- 1x Jump Wire Set



# Lab 6: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 7: Turn On LED When It Gets Dark!

---

- **วัตถุประสงค์**

- ต่่วงจรโดยใช้ ตัวต้านทานไวแสง (LDR) + ตัวต้านทาน 10k โอห์ม เป็นอินพุตแบบแอนะล็อก (ขาแอนะล็อก หมายเลข 0) ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตดิจิทัล (ขาดิจิทัล หมายเลข 8) และวงจربัชเชอร์เสียงผ่านทรานซิสเตอร์แบบ NPN เป็นเอาต์พุต (ขาดิจิทัล หมายเลข 9) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านค่าจากอินพุตที่ขาแอนะล็อก แล้วนำค่าที่ได้ (อยู่ในช่วง 0 - 1023) มาเปรียบเทียบกับค่าที่กำหนดไว้ ถ้าได้ค่าที่มากกว่า จะทำให้ LED “ติด” และสร้างเสียงผ่านบัชเชอร์ เพื่อแจ้งเตือน แต่ถ้ามีค่าน้อยกว่า จะทำให้ LED “ดับ” และไม่มีเสียงจากบัชเชอร์
- หมายเหตุ: ไฟเลี้ยงที่ป้อนให้วงจบบอร์ด (VCC) คือ +5VDC ซึ่งตรงกับไฟเลี้ยงของบอร์ด Arduino ที่ใช้ในการทดลอง

## Lab 7: Arduino Code (1/2)

```
/* Example: Turn LED ON (with Alarm) when it gets dark.
 * Get the voltage input from a voltage divider (LDR on the low side,
 * 10k Ohm on the high side)
 */
const byte LED_PIN = 8;           // use Digital Pin 8 (D8) for LED Output
const byte BUZZER_PIN = 9;       // use Digital Pin 9 (D9) for Buzzer Output
const byte AN0_PIN = 0;         // use Analog Pin 0 (A0) for Analog Input (LDR)
int REF_VAULE = 400;            // reference value (between 0..1023)
bool state = false;             // internal state (initially set to OFF)

void setup() {
    pinMode( LED_PIN, OUTPUT );   // use LED Pin as output
    pinMode( BUZZER_PIN, OUTPUT ); // use Buzzer Pin as output
    analogReference( DEFAULT );   // use +5V voltage reference
    digitalWrite( LED_PIN, LOW ); // turn LED off
    digitalWrite( BUZZER_PIN, LOW ); // turn Piezo buzzer off
}
// Code continues on the next page.
```

## Lab 7: Arduino Code (2/2)

---

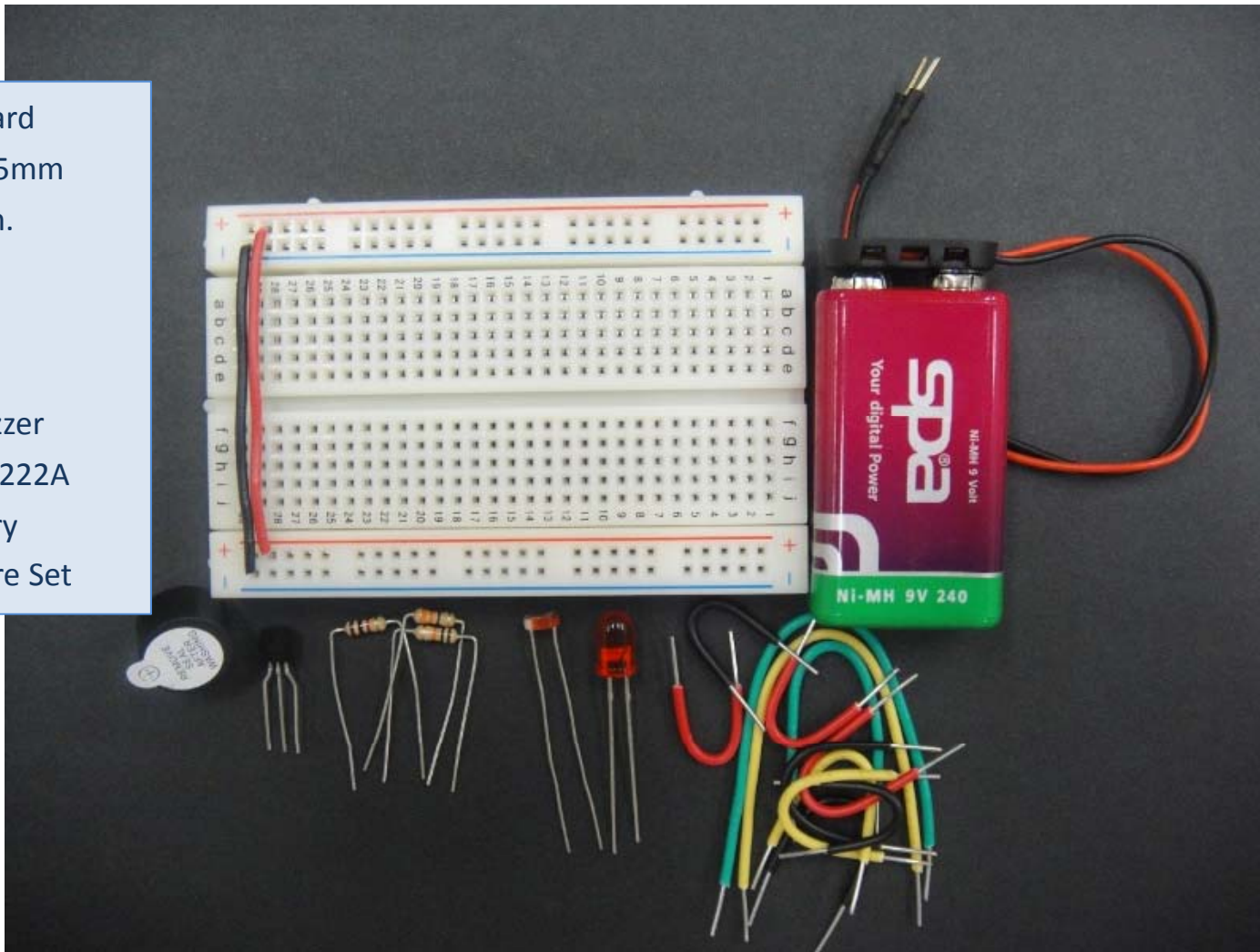
```
// Code continues from the previous page.

void loop() {
  // check whether the input value is greater than the reference value.
  while ( analogRead( AN0_PIN ) > REF_VAULE ) {
    state = !state;           // toggle current state
    digitalWrite( LED_PIN, state );   // output current state to the LED
    digitalWrite( BUZZER_PIN, state ); // output current state
                                         // to the Piezo buzzer

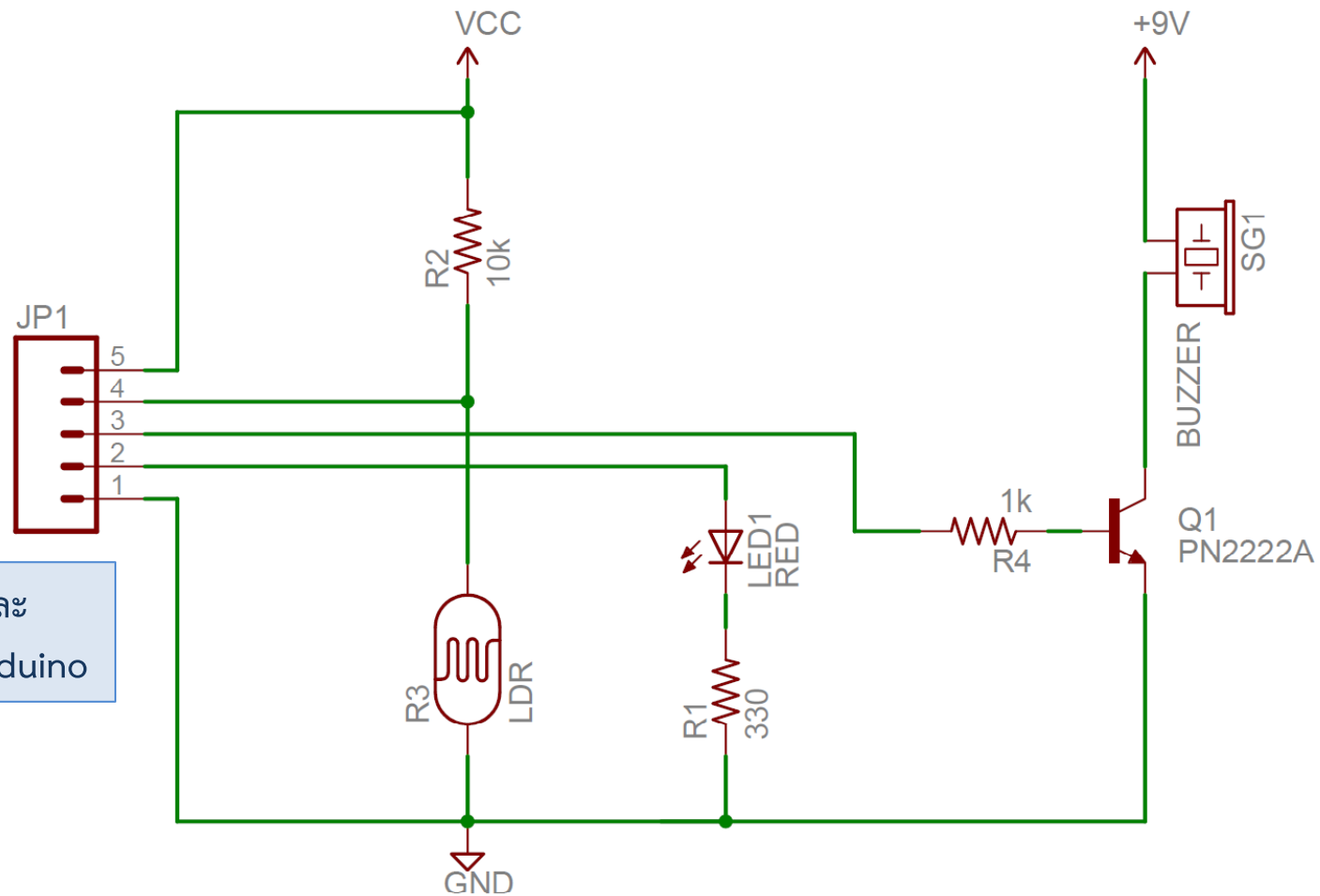
    delay( 100 );
  }
  digitalWrite( LED_PIN, LOW );   // turn LED off
  digitalWrite( BUZZER_PIN, LOW ); // turn Piezo buzzer off
}
```

# Lab 7: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x LDR 5mm.
- 1x 330 Ohm
- 1x 1k Ohm
- 1x 10k Ohm
- 1x Piezo Buzzer
- 1x NPN PN2222A
- 1x 9V Battery
- 1x Jump Wire Set



# Lab 7: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 8: Turn On Fan When It Gets Warm!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ เทอร์มิสเตอร์ (Thermistor) แบบ NTC + ตัวต้านทาน 10k โอห์ม เป็นอินพุตแบบแอนะล็อก (ขาแอนะล็อก หมายเลข 0) เพื่อวัดระดับอุณหภูมิในห้อง ใช้ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตดิจิทัล (ขาดีจิตอล หมายเลข 8) และมีวงจรควบคุมพัดลมระบายความร้อนที่ใช้มอเตอร์ไฟฟ้า โดยใช้ทรานซิสเตอร์แบบ NPN เป็นเอาต์พุต (ขาดีจิตอล หมายเลข 9) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านค่าจากอินพุตที่ขาแอนะล็อก แล้วนำค่าที่ได้ (อยู่ในช่วง 0 - 1023) มาเปรียบเทียบกับค่าที่กำหนดไว้ ถ้าได้ค่าที่มากกว่า จะทำให้ LED “ติด” และทำให้มอเตอร์ไฟฟ้าของพัดลมระบายความร้อนทำงาน แต่ถ้าไม่ใช่ จะทำให้ LED “ดับ” และพัดลมจะไม่หมุน

## Lab 8: Arduino Code (1/2)

```
/* Example: Turn on Cooling Fan when it gets dark.
 * Get the voltage input from a voltage divider (NTC Thermistor on the high side,
 * 10k Ohm on the low side).
 */
const byte LED_PIN = 8;           // use Digital Pin 8 (D8) for LED Output
const byte FAN_PIN = 9;          // use Digital Pin 9 (D9) for Cooling Fan Output
const byte AN0_PIN = 0;         // use Analog Pin 0 (A0) for Analog Input (Thermistor)
int REF_VAULE = 575;            // reference value (between 0 .. 1023)
bool state = false;             // internal state (initially set to OFF)

void setup() {
  pinMode( LED_PIN, OUTPUT );    // use LED Pin as output
  pinMode( FAN_PIN, OUTPUT );    // use Fan Pin as output
  analogReference( DEFAULT );    // use +5V voltage reference
  digitalWrite( LED_PIN, LOW );  // Turn off LED
  digitalWrite( FAN_PIN, LOW );  // Turn off Cooling Fan
}
// Code continues on the next page.
```

## Lab 8: Arduino Code (2/2)

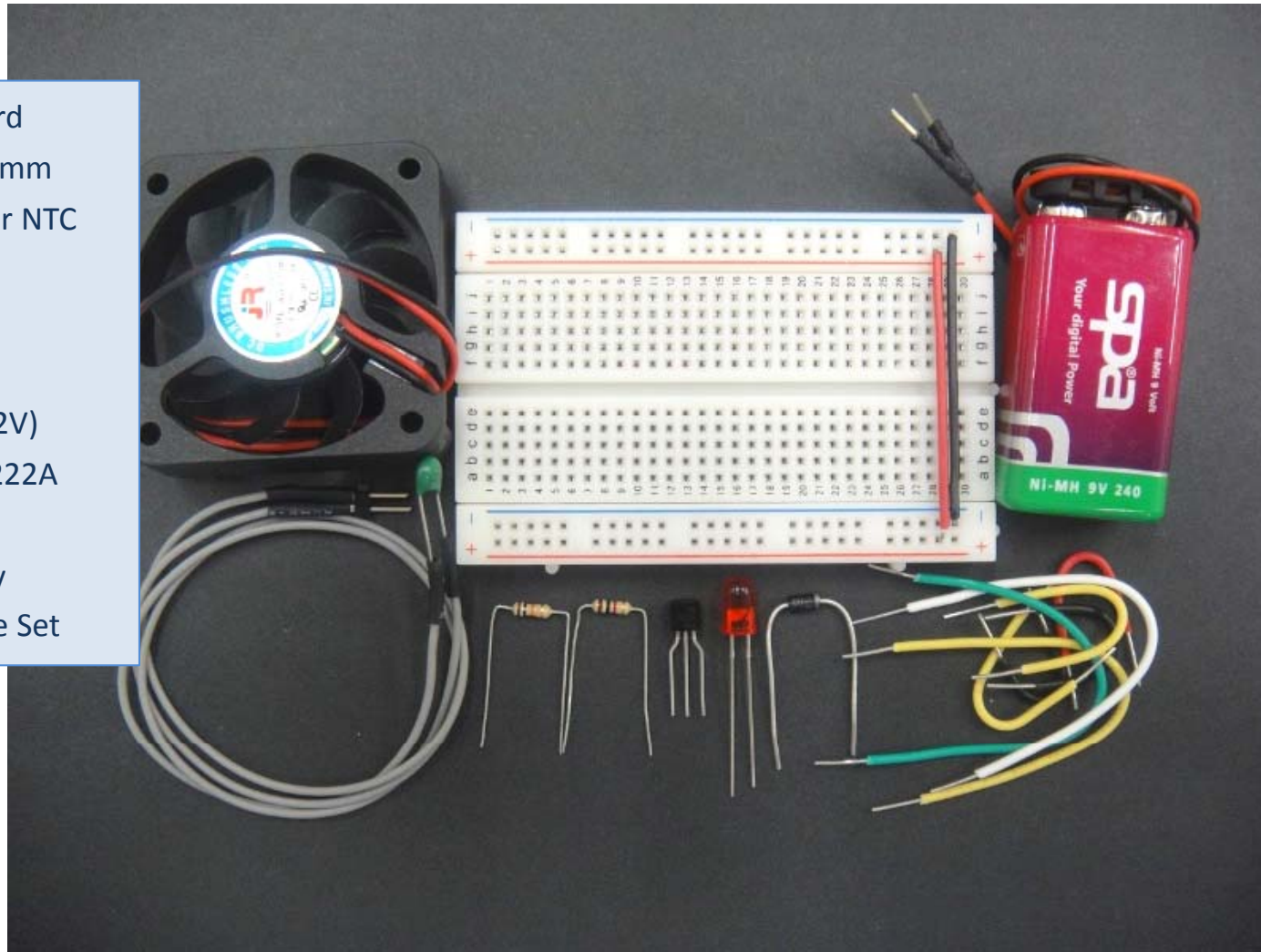
---

*// Code continues from the previous page.*

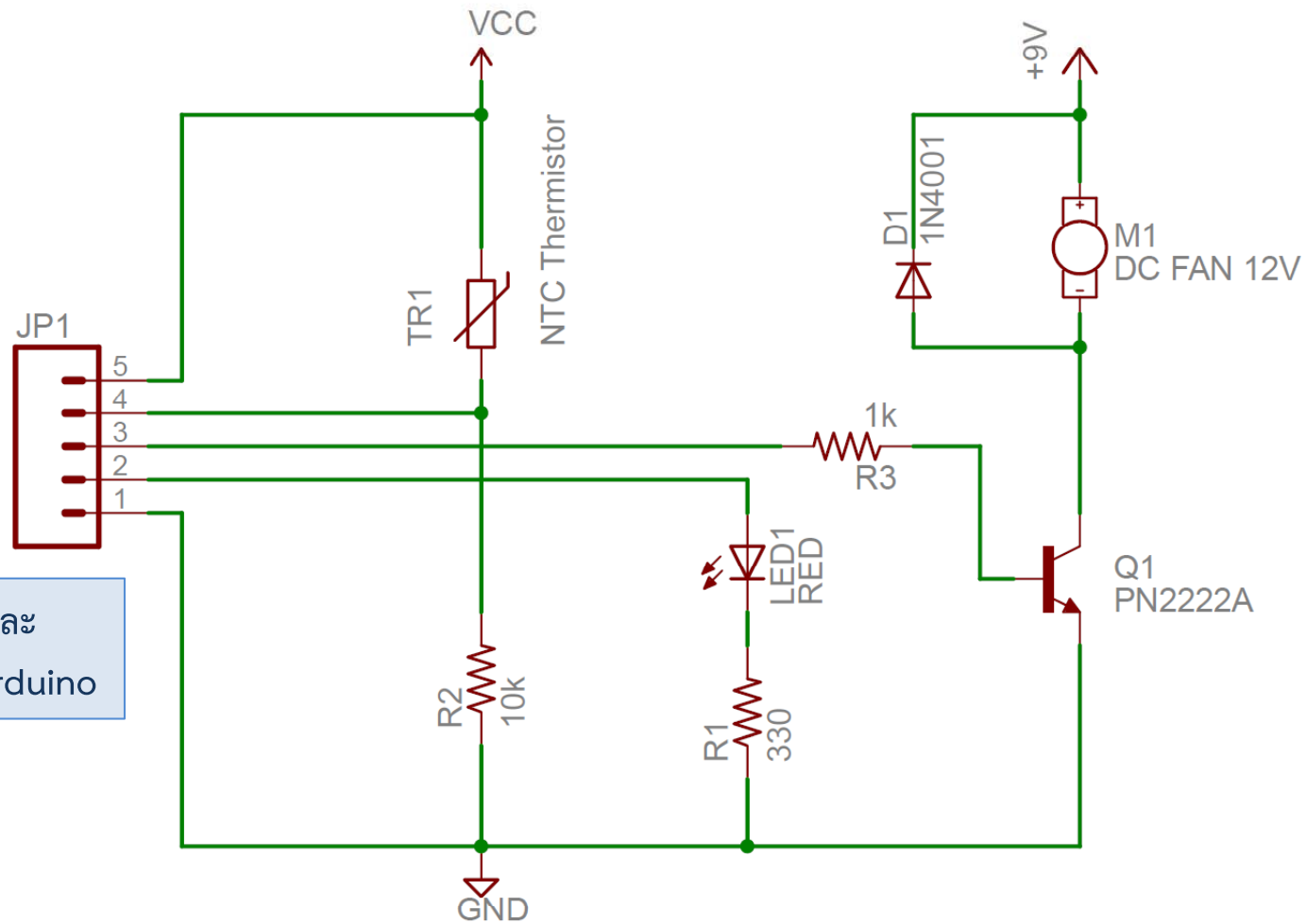
```
void loop() {  
    // check whether the input value is greater than the reference value.  
    while ( analogRead( AN0_PIN ) > REF_VAULE ) {  
        state = !state;           // toggle current state  
        digitalWrite( LED_PIN, state ); // output current state to the LED  
        digitalWrite( FAN_PIN, HIGH ); // turn on the Fan  
        delay( 100 );  
    }  
    digitalWrite( LED_PIN, LOW ); // turn off LED  
    digitalWrite( FAN_PIN, LOW ); // turn off Cooling Fan  
}
```

# Lab 8: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x Thermistor NTC
- 1x 330 Ohm
- 1x 1k Ohm
- 1x 10k Ohm
- 1x DC Fan (12V)
- 1x NPN PN2222A
- 1x 1N4001
- 1x 9V Battery
- 1x Jump Wire Set



# Lab 8: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 9: Make Dimming LED Light!

---

- **วัตถุประสงค์**

- ต่อบริการโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตแบบดิจิตอล (ขาดิจิตอล หมายเลข 9) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุตที่เป็นสัญญาณแบบ PWM (Pulse Width Modulation) โดยมีความถี่คงที่ แต่ปรับเปลี่ยนค่า Duty Cycle ได้ (0..255 ระดับ) ให้ผลเหมือนสร้างเอาต์พุตแบบแอนะล็อก และในโค้ดตัวอย่าง ค่า Duty Cycle จะถูกเพิ่มขึ้นทีละ 1 แล้ววนซ้ำไปเรื่อยๆ การเพิ่มค่า Duty Cycle นี้จะส่งผลให้ LED มีความสว่างขึ้นตามลำดับ

# Lab 9: Make Dimming LED Light!

---

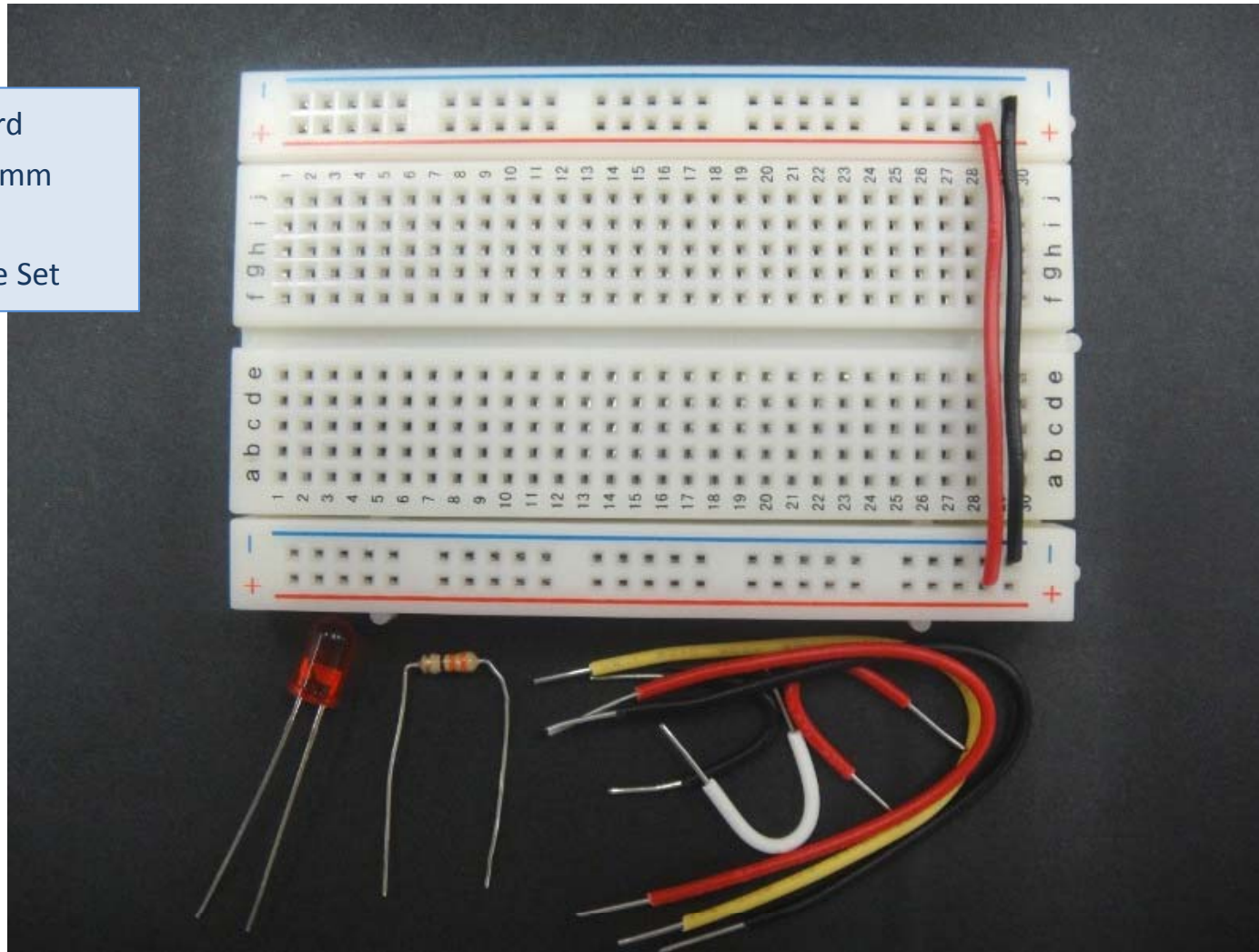
```
/* Example: Make Dimming LED Light.
*/
const byte LED_PIN = 9; // use Digital Pin 9 (D9) for LED Output
int brightness = 0; // brightness level between 0..255 (initially set to 0)

void setup() {
    pinMode( LED_PIN, OUTPUT ); // use LED Pin as output
}

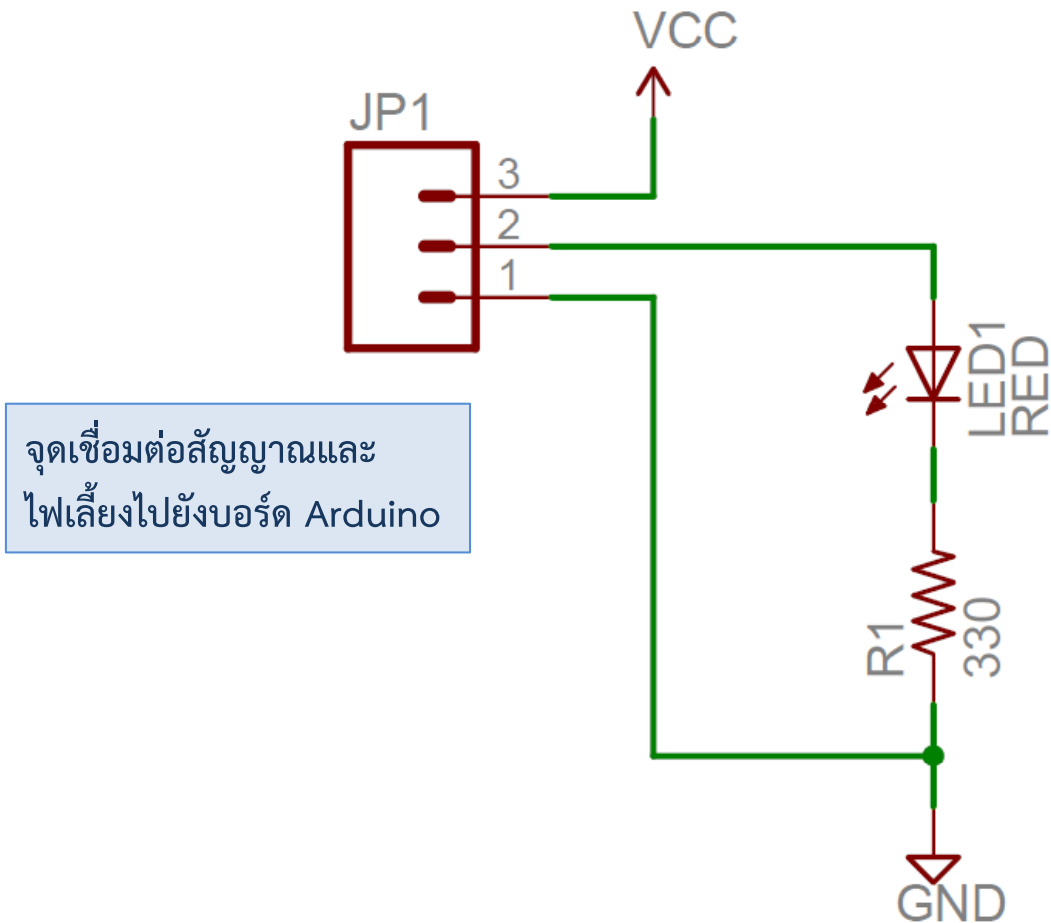
void loop() {
    analogWrite( LED_PIN, brightness ); // set the brightness for LED light
    brightness = (brightness+1) % 256; // The brightness value must be
                                        // between 0..255.
    delay(10); // wait for 10 msec before continuing
}
```

# Lab 9: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x 330 Ohm
- 1x Jump Wire Set



# Lab 9: Schematic – ผังวงจร



# Lab 10: Dim LED Light With Trimpot!

---

- **วัตถุประสงค์**

- ต่อบริเวณโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม เป็นเอาต์พุตดิจิทัล (ขาดีจิตอล หมายเลข 9) และตัวต้านทานปรับค่าได้ 10k โอห์ม (หรือสูงกว่า) เพื่อใช้เป็นอินพุตแอนะล็อก (ขาแอนะล็อก หมายเลข 0) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรมอนิเตอร์บอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตแบบแอนะล็อกจากตัวต้านทานปรับค่าได้ แล้วนำค่าที่ได้ (ซึ่งมีค่าอยู่ในช่วง 0 – 1023) มาย่อขนาดลงให้อยู่ในช่วง 0 - 255 แล้วนำไปใช้เป็นค่า Duty Cycle (0..255 ระดับ) เพื่อสร้างเอาต์พุตที่เป็นสัญญาณแบบ PWM

# Lab 10: Dim LED Light With Trimpot

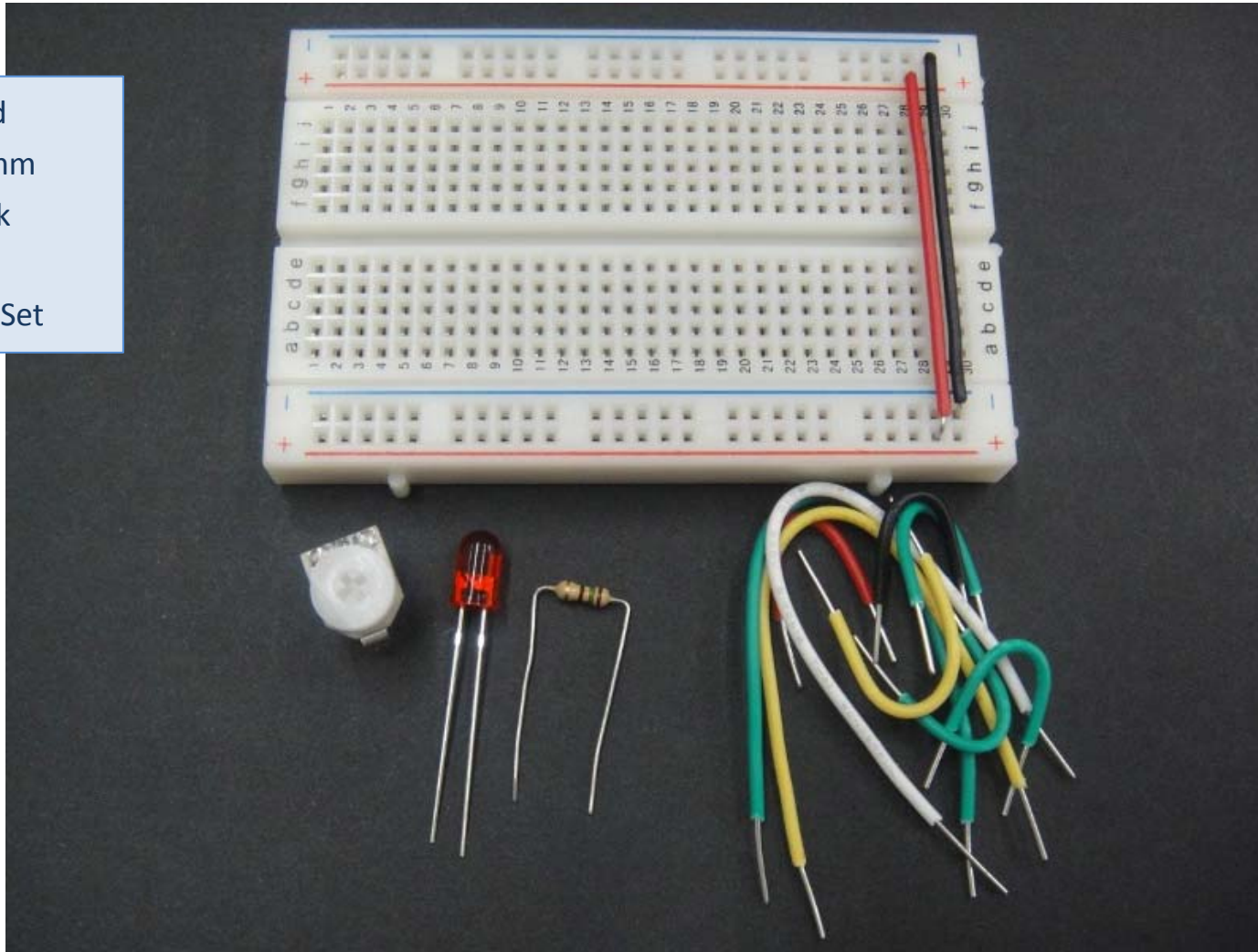
```
/* Example: Dim LED Light with Trimpot
*/
const byte LED_PIN = 9;           // use Digital Pin 9 for LED Output
const byte TRIMPOT_PIN = 0;      // use Analog Pin 0 (A0) for Analog Input

void setup() {
    pinMode( LED_PIN, OUTPUT );   // use LED Pin as output
    analogReference( DEFAULT );   // use +5V voltage reference
}

void loop() {
    int value = analogRead( TRIMPOT_PIN );    // read analog value from A0
    analogWrite( LED_PIN, value/4 );         // output analog value to LED
    delay(100);                             // wait for 100 msec before continuing
}
```

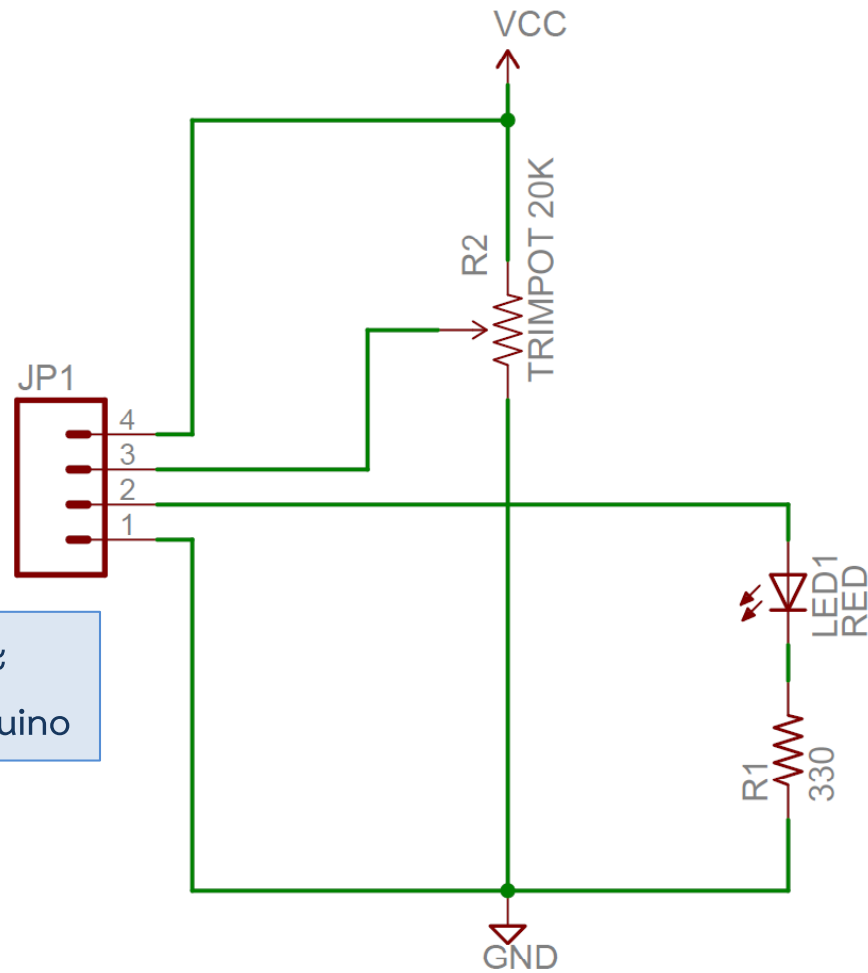
# Lab 10: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm
- 1x Trimpot 10k
- 1x 330 Ohm
- 1x Jump Wire Set



# Lab 10: Schematic – ผังวงจร

จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino



# Lab 11: Create Colored Light with RGB LED!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสงสามสี (RGB LED) + ตัวต้านทานอีก 3 ตัว เป็นเอาต์พุตแบบดิจิตอล (ขาดิจิตอล หมายเลข 9, 10, 11) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุตแบบ PWM สำหรับ RGB LED จำนวน 3 ช่องสัญญาณเป็นเอาต์พุตสำหรับสร้างแสงสีแดง (Red) สีเขียว (Green) และสีฟ้า (Blue) โดยกำหนดค่า Duty Cycle ที่แตกต่างกันไปตามที่กำหนดไว้ เพื่อสร้างแสงสีต่างๆอันเกิดจาก LED ทั้งสามสี

# Lab 11: Arduino Code (1/2)

---

```
/* Example: Create Colored Light with RGB LED.
*/
const byte R_LED_PIN = 9;    // use Digital Pin 9 (D9) for Red LED Output
const byte G_LED_PIN = 10;  // use Digital Pin 10 (D10) for Green LED Output
const byte B_LED_PIN = 11;  // use Digital Pin 11 (D11) for Blue LED Input

void setup() {
    pinMode( R_LED_PIN, OUTPUT );    // use Red LED Pin as output
    pinMode( G_LED_PIN, OUTPUT );    // use Green LED Pin as output
    pinMode( B_LED_PIN, OUTPUT );    // use Blue LED Pin as output
    digitalWrite( R_LED_PIN, HIGH ); // Turn off Red LED
    digitalWrite( G_LED_PIN, HIGH ); // Turn off Green LED
    digitalWrite( B_LED_PIN, HIGH ); // Turn off Blue LED
}

// Code continues on the next page.
```

## Lab 11: Arduino Code (2/2)

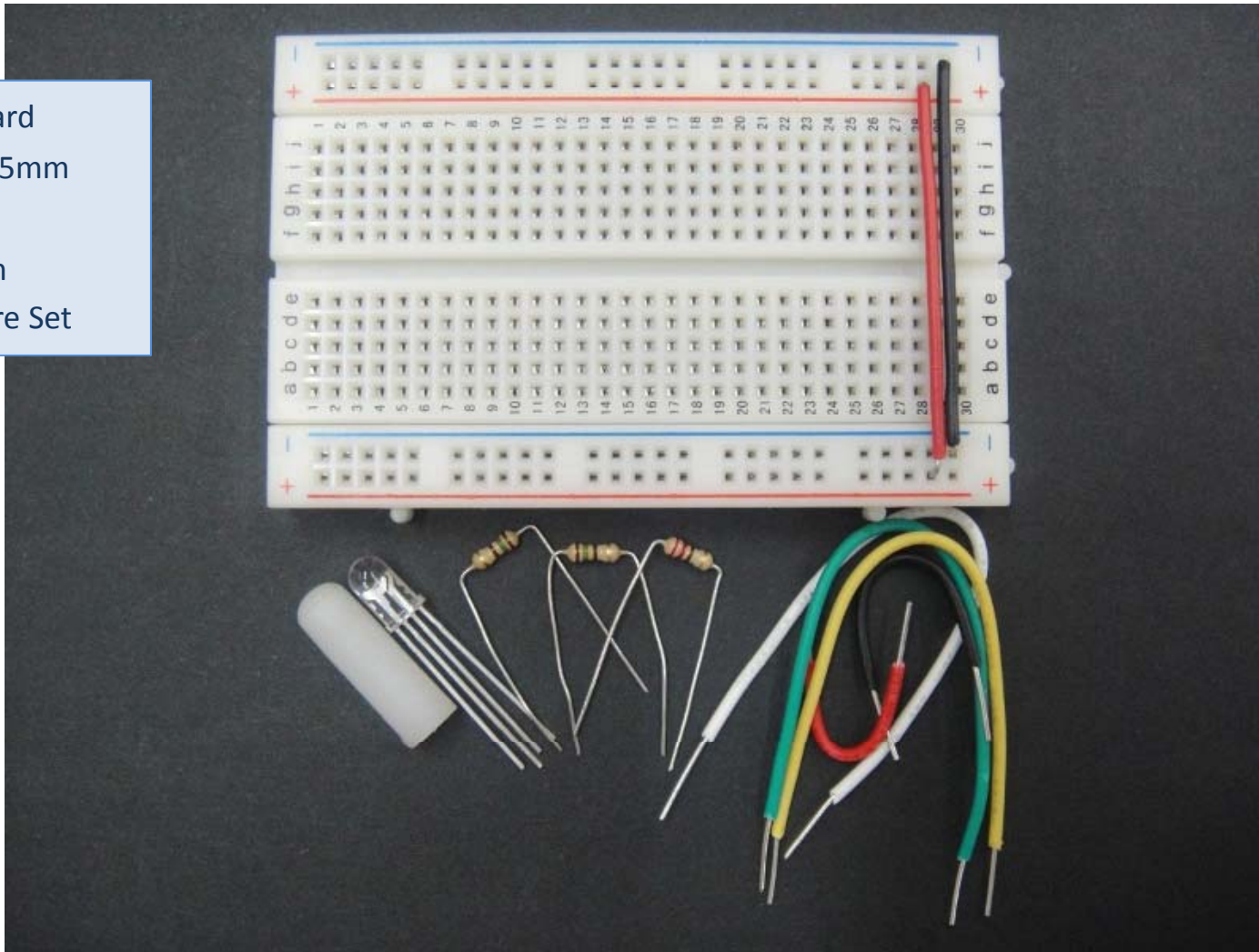
*// Code continues from the previous page.*

```
const byte values[8][3] = {
    {255,0,0}, {0,255,0}, {0,0,255}, {255,255,0},
    {0,255,255}, {255,0,255}, {255,255,255}, {0,0,0}  };

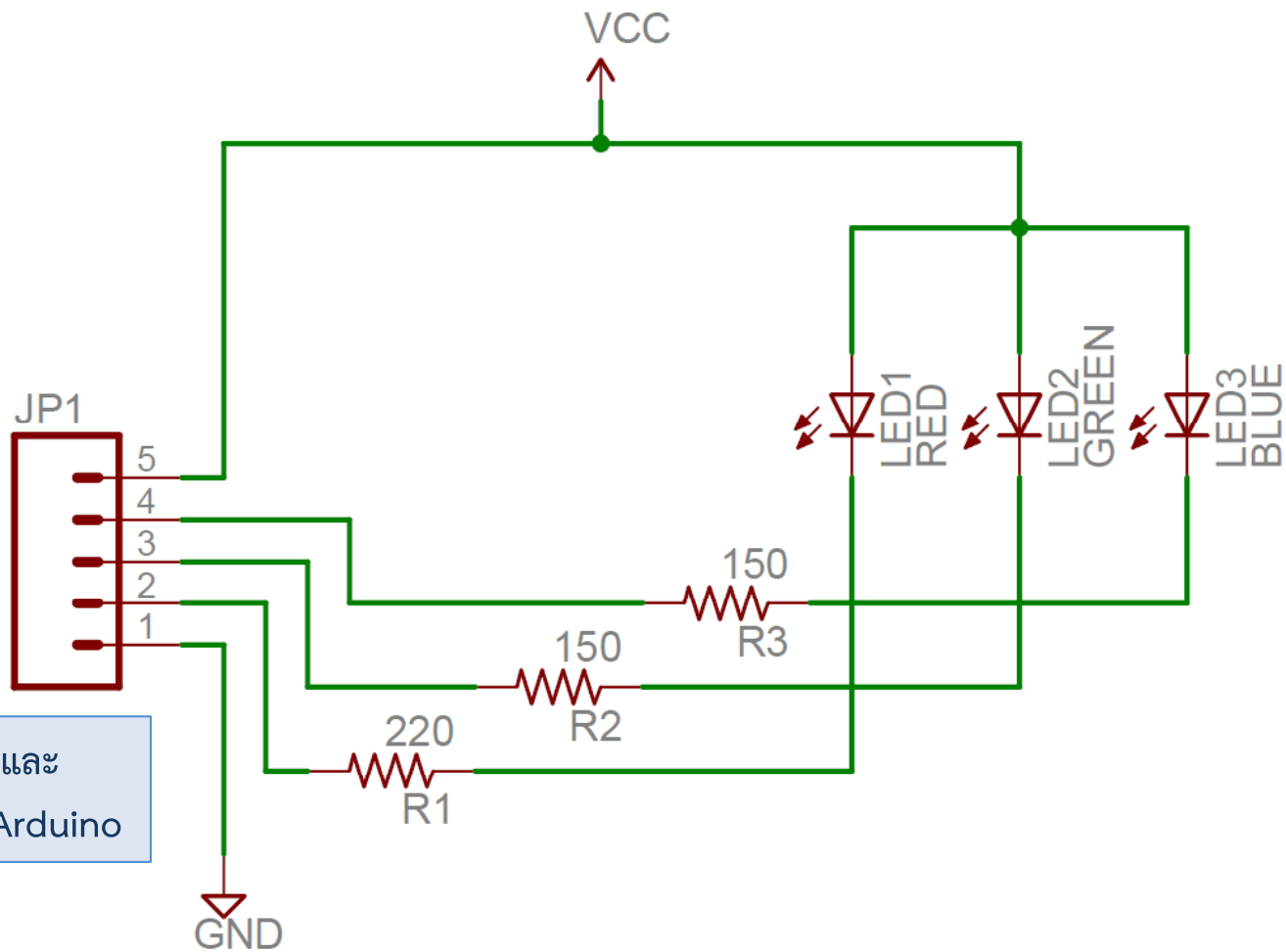
void loop() {
    for (int i=0; i < 8; i++) {
        analogWrite( R_LED_PIN, 255 - values[i][0] ); // set brightness for Red
        analogWrite( G_LED_PIN, 255 - values[i][1] ); // set brightness for Green
        analogWrite( B_LED_PIN, 255 - values[i][2] ); // set brightness for Blue
        delay(1000); // wait for 1000 msec before continuing
    }
}
```

# Lab 11: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x RGB LED 5mm
- 1x 220 Ohm
- 2x 150 Ohm
- 1x Jump Wire Set



# Lab 11: Schematic – ฟังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 12: Make Random Light with RGB LED!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสงสามสี (RGB LED) + ตัวต้านทานอีก 3 ตัว เป็นเอาต์พุตแบบดิจิตอล (ขาดิจิตอล หมายเลข 9, 10, 11) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุตแบบ PWM สำหรับ RGB LED จำนวน 3 ช่องสัญญาณเป็นเอาต์พุตสำหรับสร้างแสงสีแดง (Red) สีเขียว (Green) และสีฟ้า (Blue) โดยกำหนดค่า Duty Cycle ที่แตกต่างกันไปแบบสุ่ม เพื่อสร้างแสงสีต่างๆอันเกิดจาก LED ทั้งสามสี

# Lab 12: Arduino Code

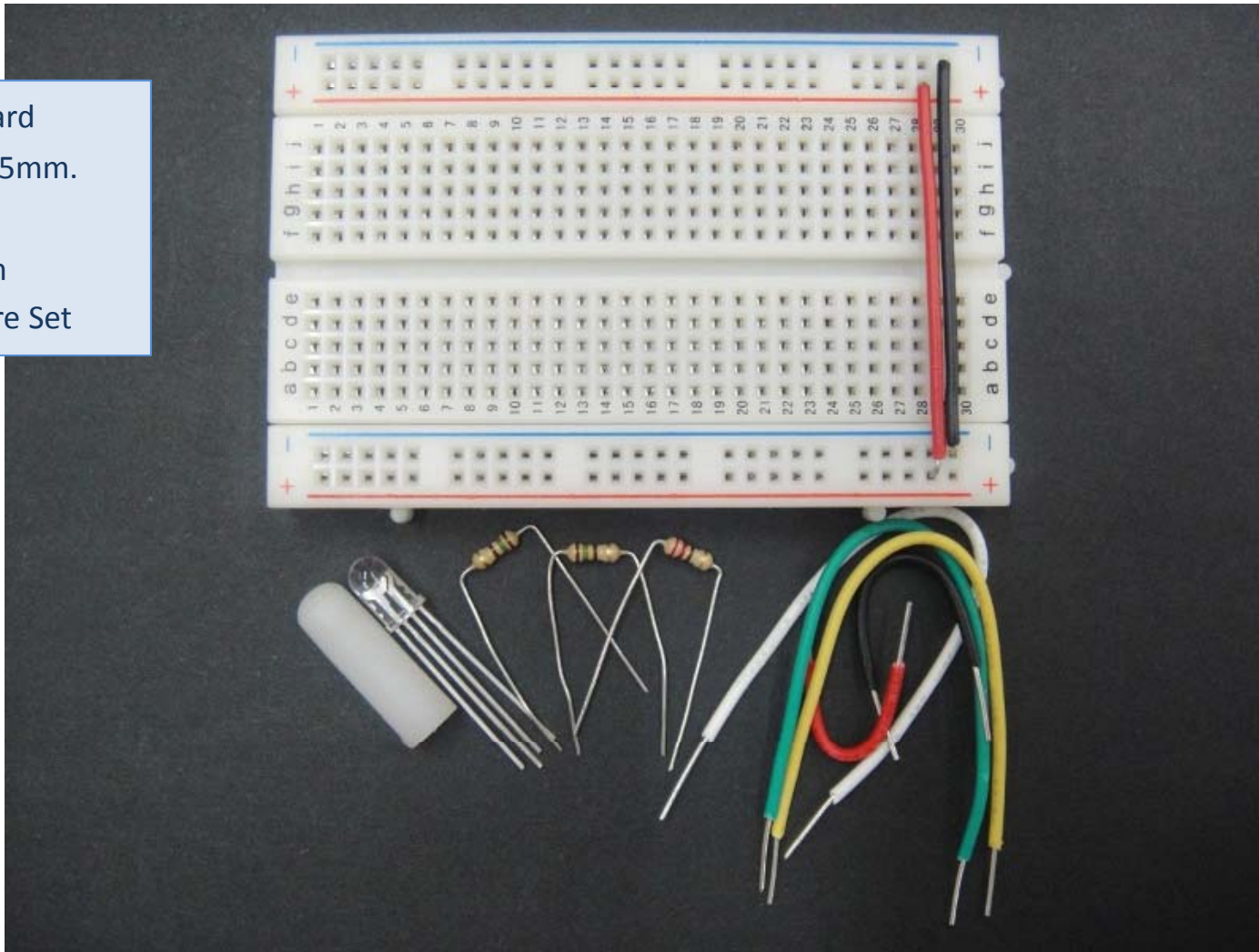
```
/* Example: Make Random Light with RGB LED (Common Anode) */
const byte R_LED_PIN = 9;   // use Digital Pin 9 (D9) for Red LED Output
const byte G_LED_PIN = 10;  // use Digital Pin 10 (D10) for Green LED Output
const byte B_LED_PIN = 11;  // use Digital Pin 11 (D11) for Blue LED Input

void setup() {
  pinMode( R_LED_PIN, OUTPUT );   // use Red LED Pin as output
  pinMode( G_LED_PIN, OUTPUT );   // use Green LED Pin as output
  pinMode( B_LED_PIN, OUTPUT );   // use Blue LED Pin as output
  randomSeed( analogRead(0) );    // Read analog value from floating input A0
                                  // and use it as the seed number for random().
}

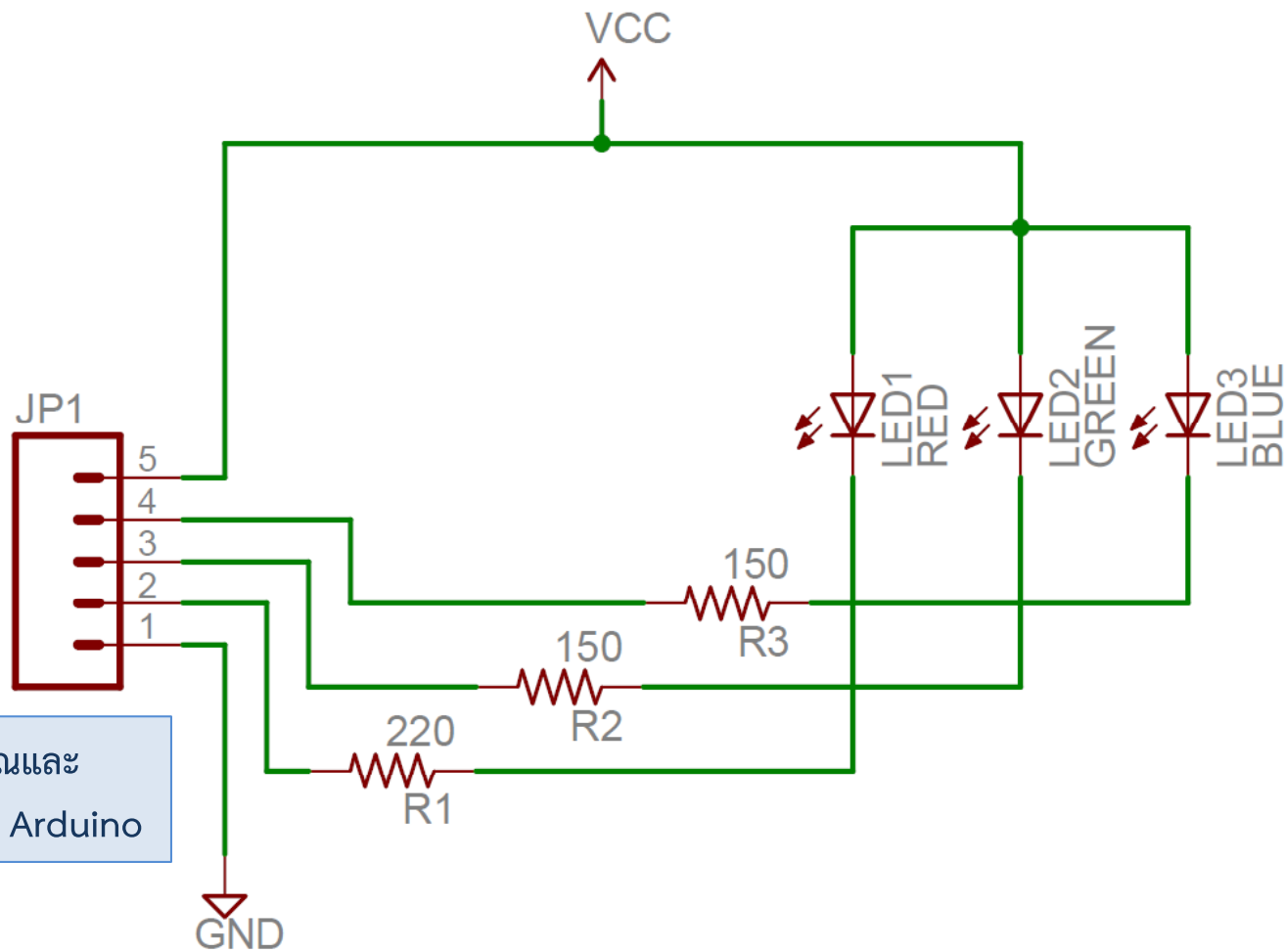
void loop() {   // generate a random value between 0..255 and use it
               // as PWM duty cycle for analog output
  analogWrite( R_LED_PIN, random(0,256) );   // set brightness for Red
  analogWrite( G_LED_PIN, random(0,256) );   // set brightness for Green
  analogWrite( B_LED_PIN, random(0,256) );   // set brightness for Blue
  delay(1000);
}
```

# Lab 12: Parts List – รายการอุปกรณ์

1x Breadboard  
1x RGB LED 5mm.  
1x 220 Ohm  
2x 150 Ohm  
1x Jump Wire Set



# Lab 12: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 13: Make Random Light with 8 LEDs!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 8 ตัว + ตัวต้านทานแบบชุด (R-Pack) 9 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดิจิทัล หมายเลข 5,6,7,8,9,10,11,12) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 8 ตัว ขนาด 330 โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

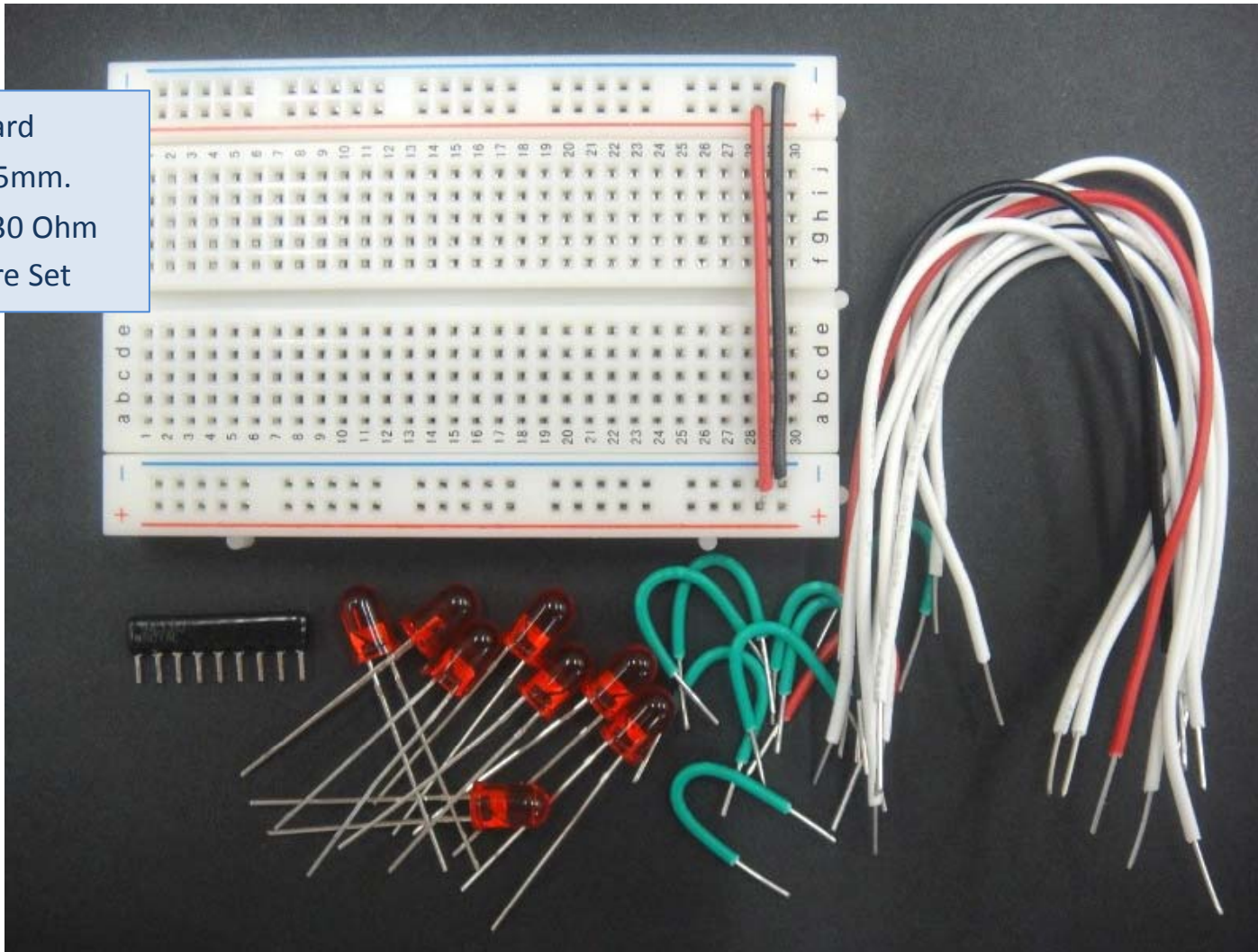
- โค้ดทำหน้าที่สร้างเอาต์พุต 8 บิต ที่ให้ค่าแบบสุ่ม สำหรับ LED ทั้ง 8 ตัว และเปลี่ยนไปเรื่อยๆ

# Lab 13: Arduino Code

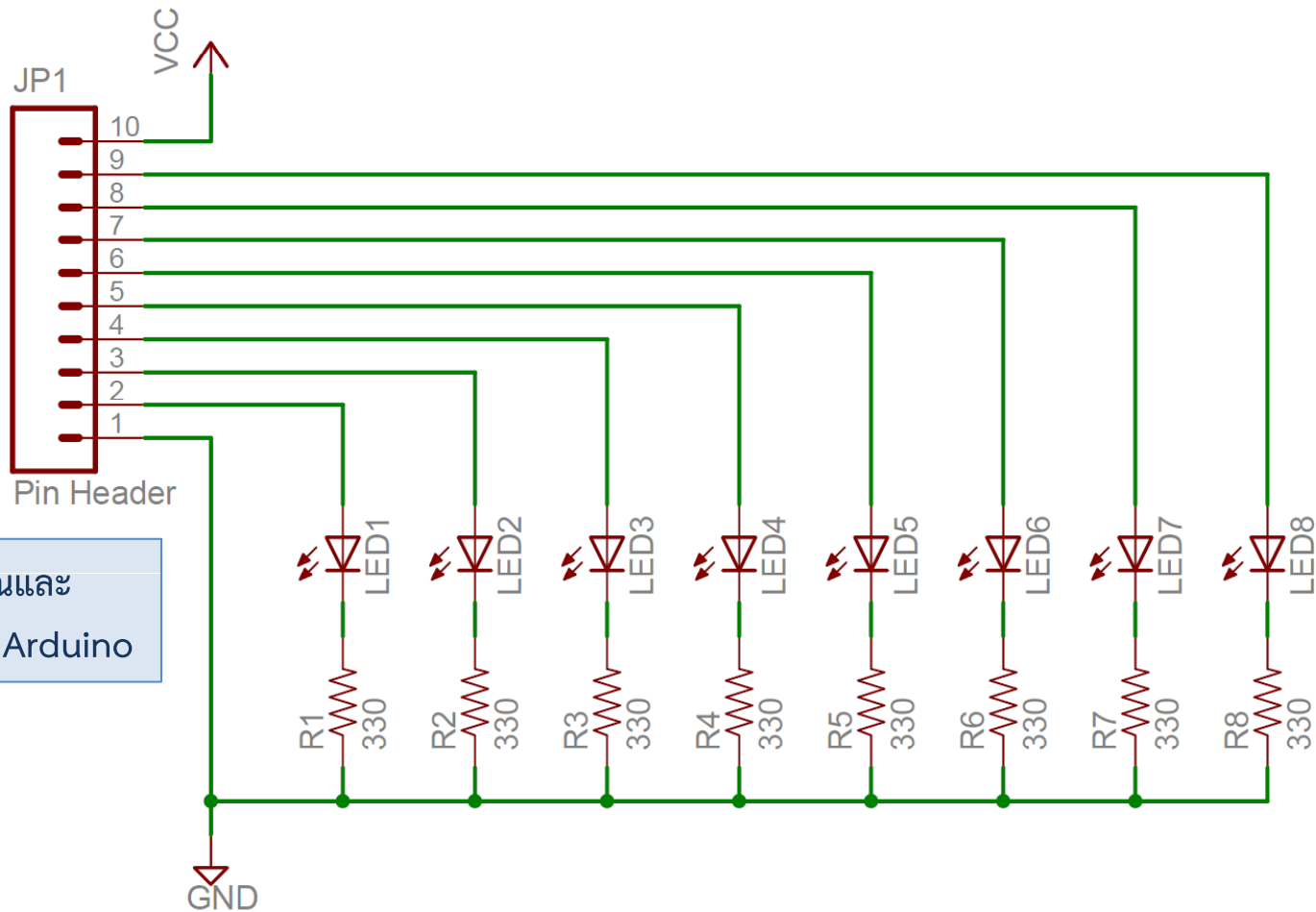
```
/* Example: Make Random Light with 8 LEDs!  
*/  
const byte LED_PINS [8]  
    = { 5,6,7,8,9,10,11,12 }; // Define Digital Pins for 8-Bit LED Output  
bool state = true;           // internal state (initially set to TRUE)  
  
void setup() {  
    for (int i=0; i < 8; i++) // for each LED pin do  
        pinMode( LED_PINS[i], OUTPUT ); // use this LED Pin as output  
    randomSeed( analogRead( A0 ) ); // set a seed number for random() function  
}  
void loop() {  
    byte t = (byte)random(0,256); // generate a random number between 0..255  
    for (int i=0; i < 8; i++) {  
        digitalWrite( LED_PINS[i], t & 1 ); // set LED output state at position i  
        t >>= 1;  
    }  
    delay(100); // wait for 100 msec before continuing  
}
```

# Lab 13: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 8x Red LED 5mm.
- 1x R-Pack 330 Ohm
- 1x Jump Wire Set



# Lab 13: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 14: Create Running LED Patterns!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 8 ตัว + ตัวต้านทานแบบชุด (R-Pack) 9 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดิจิทัล หมายเลข 5,6,7,8,9,10,11,12) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 8 ตัว ขนาด 330 โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุต 8 บิต สำหรับ LED ทั้ง 8 ตัว โดยมีเพียงหนึ่งดวงที่ “ติด” และจะเลื่อนตำแหน่งของดวงที่ “ติด” ไปเรื่อยๆทีละหนึ่งตำแหน่ง

# Lab 14: Arduino Code (Version A)

```
/* Example: Create Running LED Patterns (Version A) */
const byte LED_PINS [8] = { 5,6,7,8,9,10,11,12 };
const byte PATTERNS [8] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

byte index = 0;    // used as a counter between 0 to 7.

void setup() {
  for (int i=0; i < 8; i++)          // for each LED pin do
    pinMode( LED_PINS[i], OUTPUT ); // use this LED Pin as output
}

void loop() {
  byte t = PATTERNS[index]; // get byte value from array at current index
  for (int i=0; i < 8; i++) { // for each bit in byte value do ..
    digitalWrite( LED_PINS[i], t & 1 ); // write current bit to LED output
    t >>= 1;          // shift-bit to left by 1
  }
  index = (index+1) % 8; // increment index by 1
  delay(50);           // delay for 50 msec
}
```

## Lab 14: Arduino Code (Version B)

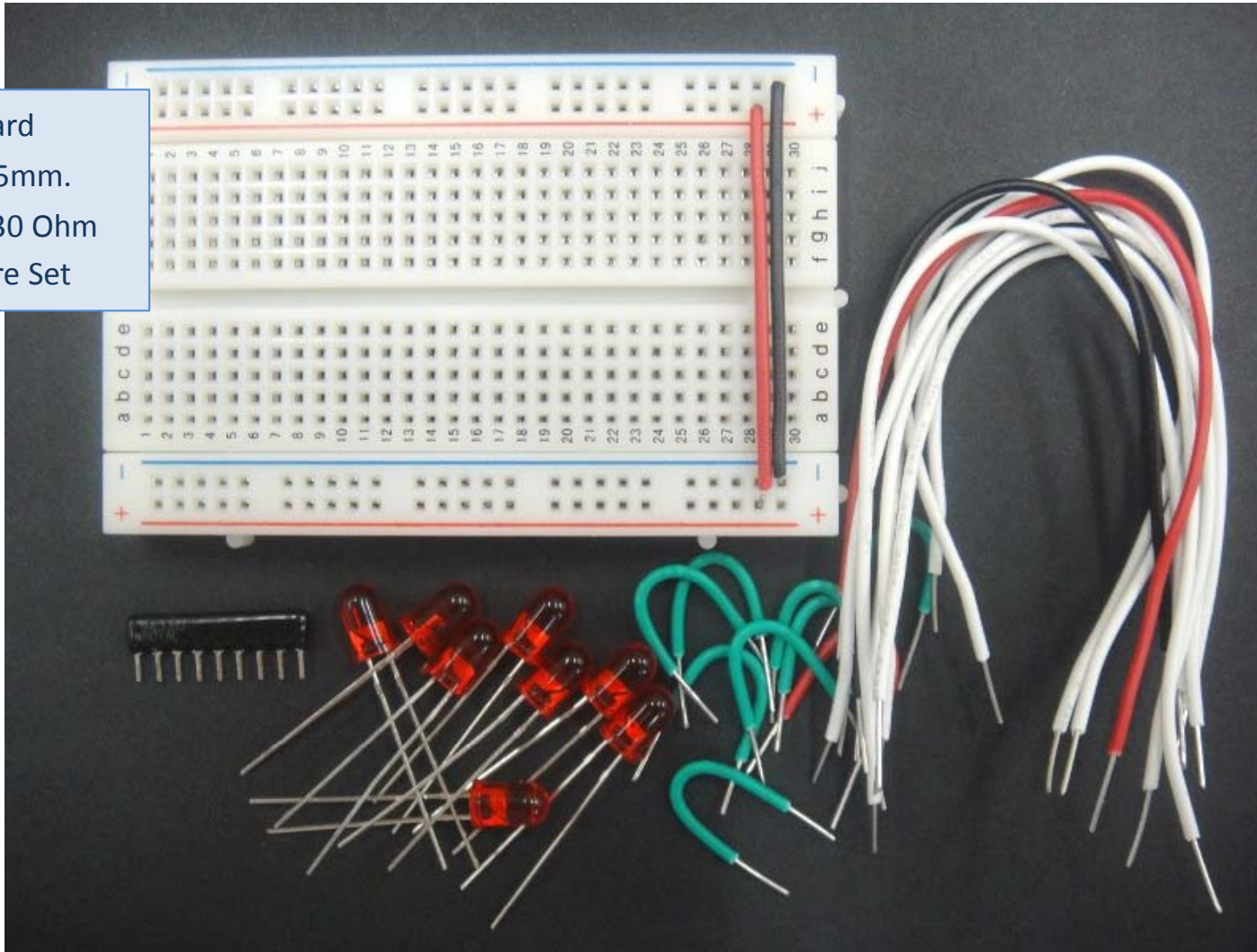
```
/* Example: Create Running LED Patterns (Version B) */
const byte LED_PINS [8] = { 5,6,7,8,9,10,11,12 };
const byte PATTERNS [14] = { 0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,
                             0x40,0x20,0x10,0x08,0x04,0x02 };

byte index = 0; // used as a counter between 0 to 7.
void setup() {
    for (int i=0; i < 8; i++)
        pinMode( LED_PINS[i], OUTPUT ); // use all LED Pins as output
}
void loop() {
    byte t = PATTERNS[index]; // get byte value from array at current index
    for (int i=0; i < 8; i++) {
        digitalWrite( LED_PINS[i], t & 1 ); // write current bit to LED output
                                                // at position i

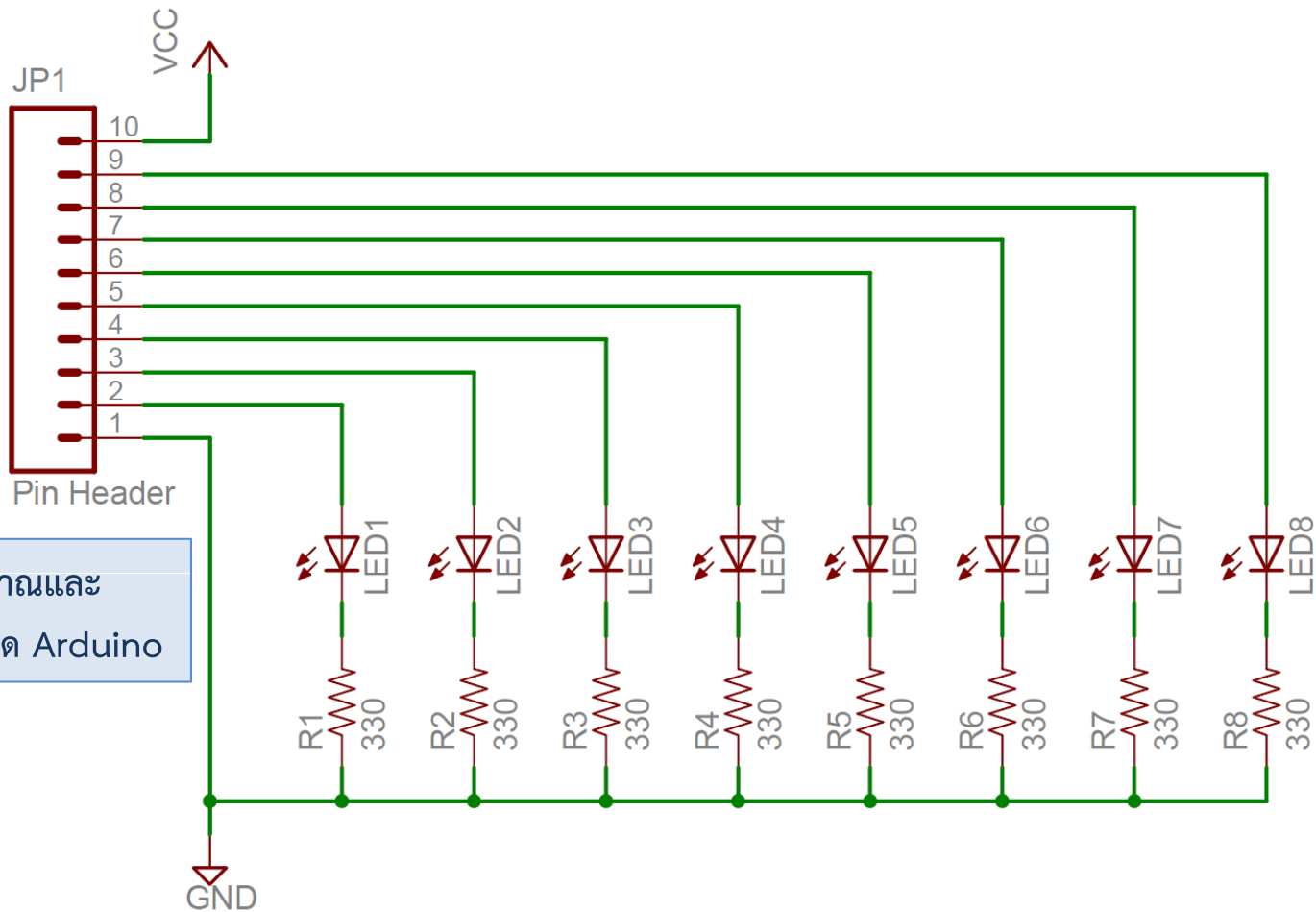
        t >>= 1;
    }
    index = (index+1) % 14; // increment index by 1
    delay( 50 ); // wait for 50 msec
}
```

# Lab 14: Parts List – รายการอุปกรณ์

1x Breadboard  
8x Red LED 5mm.  
1x R-Pack 330 Ohm  
1x Jump Wire Set



# Lab 14: Schematic – ผังวงจร



จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino

# Lab 15: Create Pausable Running LEDs!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 8 ตัว + ตัวต้านทานแบบชุด (R-Pack) 9 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาติจิตอล หมายเลข 5,6,7,8,9,10,11,12) และมีปุ่มกด + ตัวต้านทาน 10k โอห์ม 1 ชุด เพื่อใช้เป็นอินพุตแบบดิจิทัล (ขาติจิตอล หมายเลข 3) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 8 ตัว ขนาด 330 โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุต 8 บิต สำหรับ LED ทั้ง 8 ตัว โดยมีเพียงหนึ่งดวงที่ “ติด” และจะเลื่อนตำแหน่งของดวงที่ “ติด” ไปเรื่อยๆ ทีละหนึ่งตำแหน่ง แต่ถ้ามีการกดปุ่มหนึ่งครั้ง จะสามารถหยุดการเปลี่ยนตำแหน่ง (Stop) และถ้ากดอีกครั้ง ก็จะทำให้มีการเลื่อนตำแหน่งต่อไปได้ (Start)

## Lab 15: Arduino Code (1/3)

---

```
/* Example: Create Pausable Running LEDs. */
const byte START_STOP_PIN = 3;           // define digital pin for Start/Stop button
const byte LED_PINS [8] = { 5,6,7,8,9,10,11,12 }; // define digital pins for LEDs

const byte PATTERNS [14] = { 0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,
                             0x40,0x20,0x10,0x08,0x04,0x02 };

bool state = true;                       // internal state (initially set to TRUE)
byte index = 0;                           // counting variable between 0..255

void setup() {
    pinMode( START_STOP_PIN, INPUT );     // use Start/Stop Pin as input
    for (int i=0; i < 8; i++)             // for each LED Pin do
        pinMode( LED_PINS[i], OUTPUT );  // output bit to current LED pin
}

// Code continues on the next page.
```

## Lab 15: Arduino Code (2/3)

*// Code continues from the previous page.*

```
void loop() {  
    byte t = PATTERNS[index];  
    if ( isButtonClicked( START_STOP_PIN ) ) // Start/Stop button clicked?  
        state = !state; // toggle the state  
    if ( state ) {  
        for (int i=0; i < 8; i++) { // for each LED pin do  
            digitalWrite( LED_PINS[i], t & 1 ); // output bit to current LED pin  
            t >>= 1;  
        }  
        index = (index+1) % 14; // increment the index by 1  
        delay(100);  
    }  
}
```

*// Code continues on the next page.*

## Lab 15: Arduino Code (3/3)

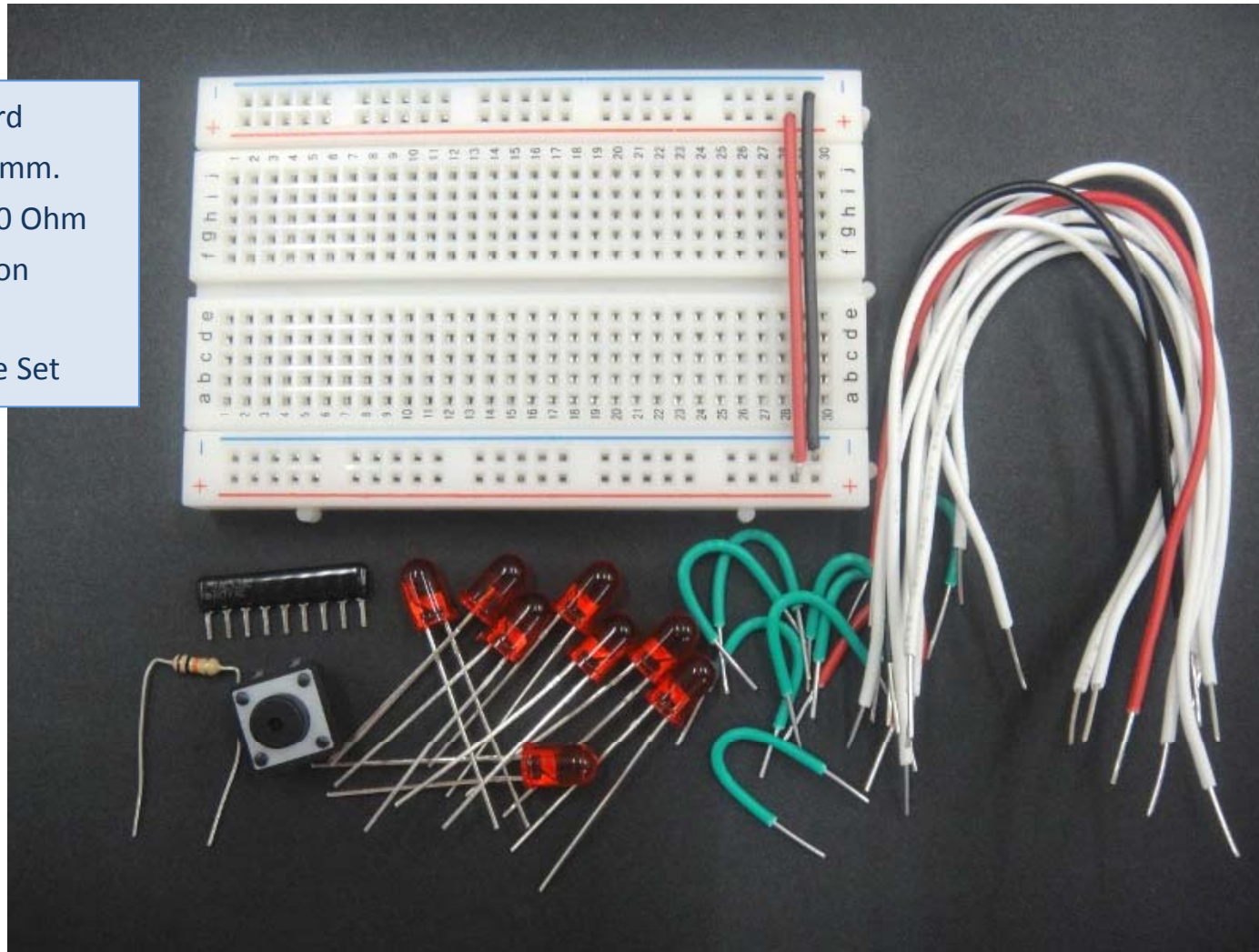
---

*// Code continues from the previous page.*

```
bool isButtonClicked( byte pinNumber ) {
    byte t = 5;
    while ( (digitalRead( pinNumber ) == LOW) && --t )
        delay(1);
    if ( t==0 ) {
        while ( digitalRead( pinNumber ) == LOW );
        return true;
    }
    return false;
}
```

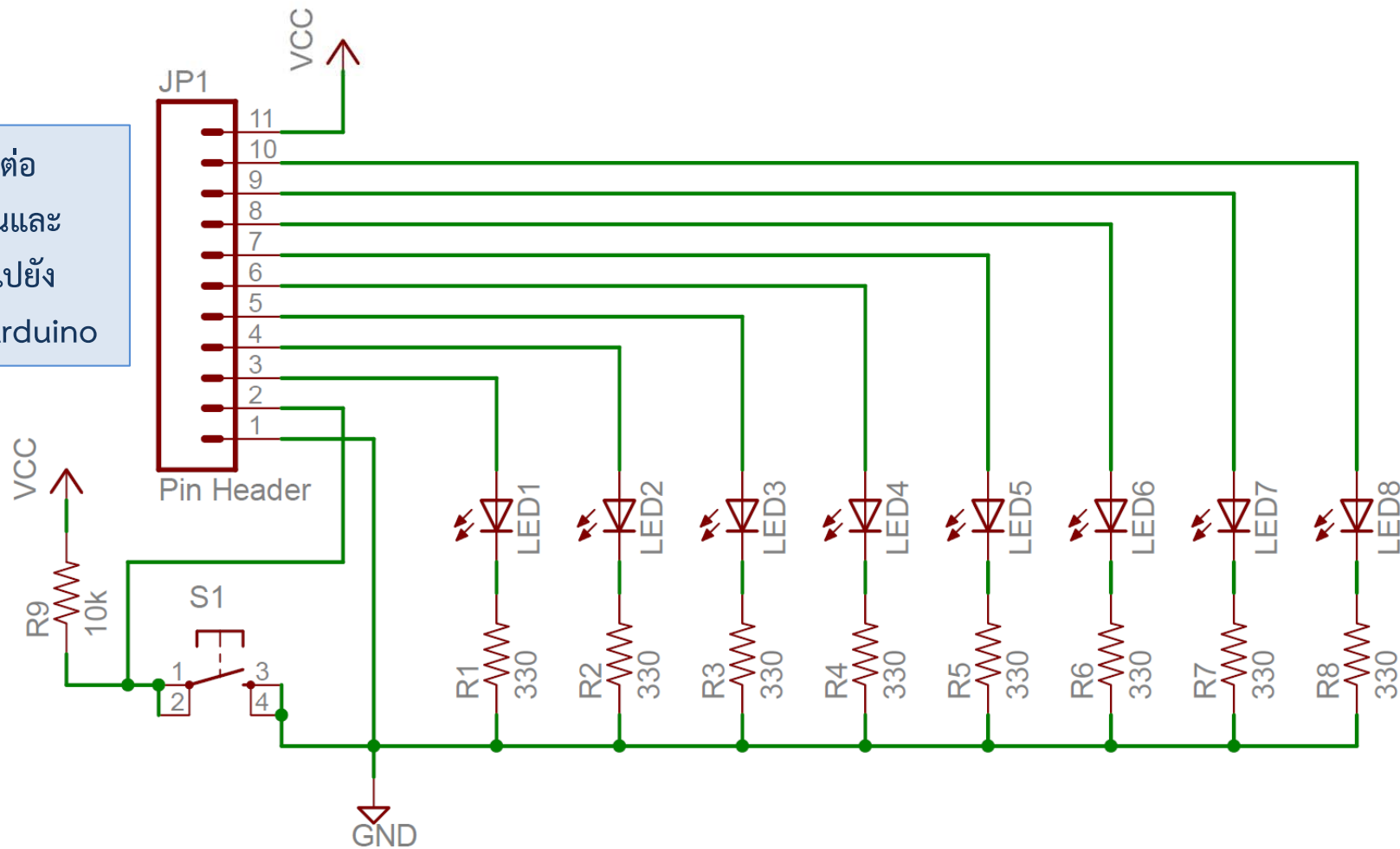
# Lab 15: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 8x Red LED 5mm.
- 1x R-Pack 330 Ohm
- 1x Push Button
- 1x 10k Ohm
- 1x Jump Wire Set



# Lab 15: Schematic – ฟังวงจร

จุดเชื่อมต่อ  
สัญญาณและ  
ไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 16: Count-Up In Binary!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 8 ตัว + ตัวต้านทานแบบชุด (R-Pack) 9 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดีจิตอล หมายเลข 5,6,7,8,9,10,11,12) และมีปุ่มกด + ตัวต้านทาน 10k โอห์ม 2 ชุด (ขาดีจิตอล หมายเลข 3, 4) เพื่อใช้เป็นอินพุตดิจิทัลสำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 8 ตัว ขนาด 330 โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุต 8 บิต สำหรับ LED ทั้ง 8 ตัว เพื่อแสดงค่าของตัวนับแบบไบนารี (Binary Counter) ขนาด 8 บิต เริ่มต้นการนับที่ 0 เมื่อกดปุ่ม “Up” แต่ละครึ่ง จะเพิ่มค่าที่ละหนึ่ง แต่ถ้ากดปุ่ม “Reset” จะทำให้ตัวนับถูกรีเซ็ตเป็น 0

## Lab 16: Arduino Code (1/3)

```
/* Example: Count-Up In Binary.
*/
const byte UP_BUTTON_PIN = 3;           // define digital pin for Up button
const byte RESET_BUTTON_PIN = 4;       // define digital pin for Reset button
const byte LED_PINS [8] = { 5,6,7,8,9,10,11,12 }; // define digital pins for LEDs
byte count = 0;                         // a counting variable (between 0..255)

void setup() {
    pinMode( UP_BUTTON_PIN, INPUT );     // use Up Button Pin as input
    pinMode( RESET_BUTTON_PIN, INPUT ); // use Reset Button Pin as input
    for (int i=0; i < 8; i++)
        pinMode( LED_PINS[i], OUTPUT ); // use all LED Pins as output
}
void loop() {
    checkButtons();
    displayLEDs();
}
// Code continues from the next page.
```

## Lab 16: Arduino Code (2/3)

---

*// Code continues from the previous page.*

```
void checkButtons() {
  if ( isButtonClicked( UP_BUTTON_PIN ) ) // Has the Up Button been clicked?
    count = count + 1;      // increment count by 1
  if ( isButtonClicked( RESET_BUTTON_PIN ) ) // Has the Reset Button been clicked?
    count = B00000000;     // reset value to 0 (in binary number representation)
}

void displayLEDs() {
  byte t = count;
  for (int i=0; i < 8; i++) {
    digitalWrite( LED_PINS[i], t & 1 );
    t >>= 1;
  }
}
```

*// Code continues on the next page.*

## Lab 16: Arduino Code (3/3)

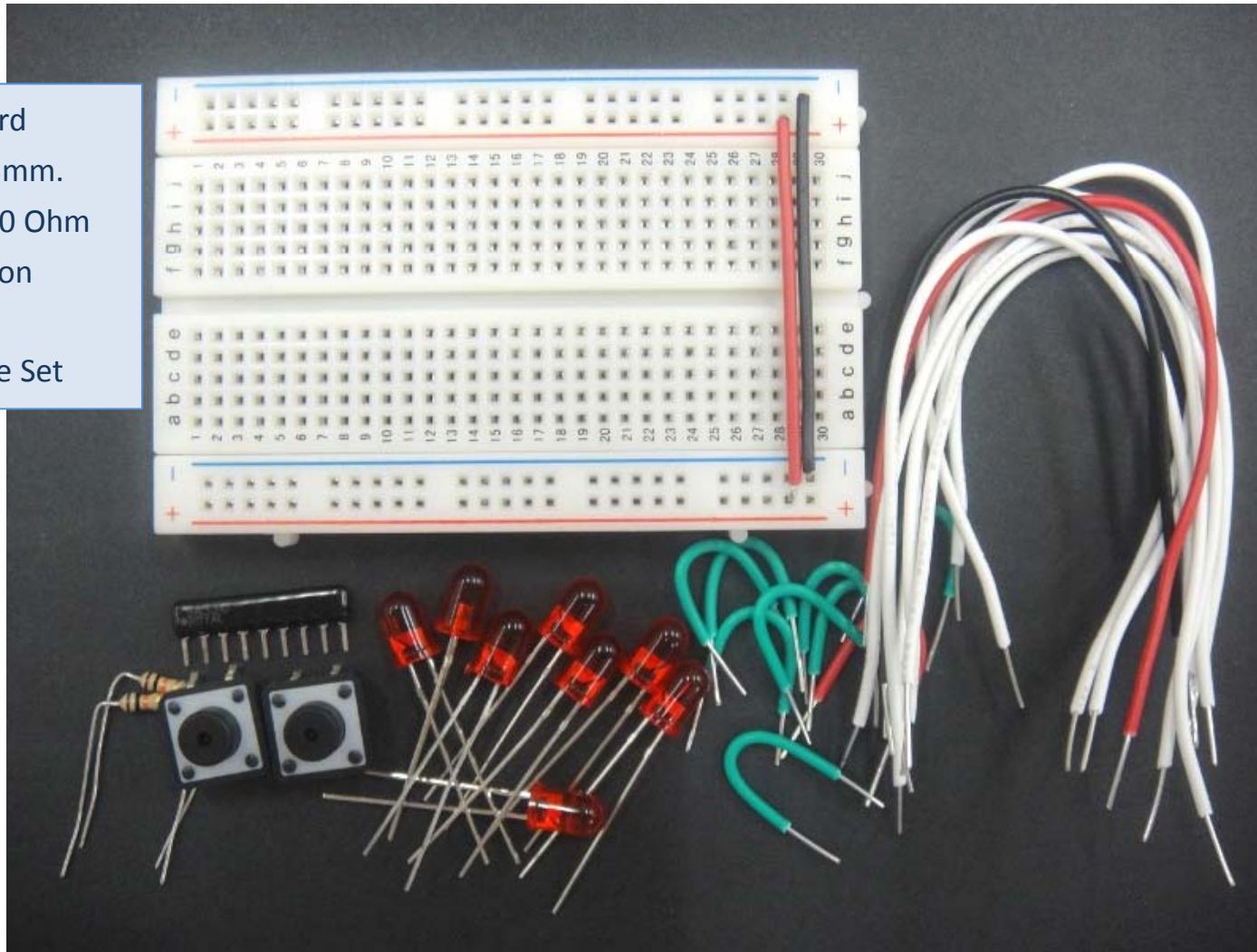
---

*// Code continues from the previous page.*

```
bool isButtonClicked( byte pinNumber ) {
    byte t = 5;
    while ( (digitalRead( pinNumber ) == LOW) && --t )
        delay(1);
    if ( t==0 ) {
        while ( digitalRead( pinNumber ) == LOW );
        return true;
    }
    return false;
}
```

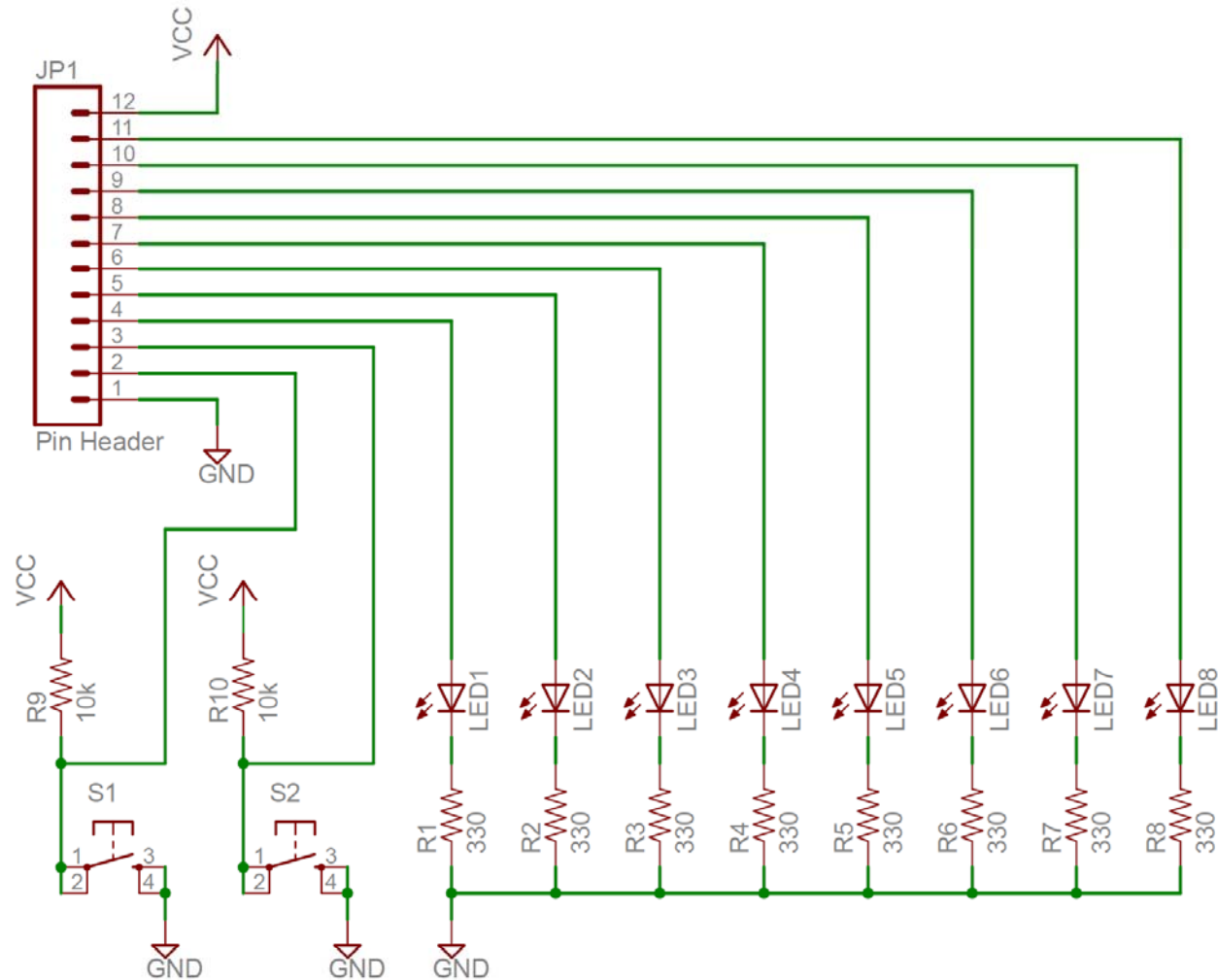
# Lab 16: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 8x Red LED 5mm.
- 1x R-Pack 330 Ohm
- 2x Push Button
- 2x 10k Ohm
- 1x Jump Wire Set



# Lab 16: Schematic – ฟังวงจร

จุดเชื่อมต่อ  
สัญญาณและ  
ไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 17: Count Up or Down In Binary!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 8 ตัว + ตัวต้านทานแบบชุด (R-Pack) 9 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิตอล (ขาติจิตอล หมายเลข 5,6,7,8,9,10,11,12) และมีปุ่มกด + ตัวต้านทาน 10k โอห์ม 2 ชุด (ขาติจิตอล หมายเลข 3, 4) เพื่อใช้เป็นอินพุตดิจิตอล สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 8 ตัว ขนาด 330 โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างเอาต์พุต 8 บิต สำหรับ LED ทั้ง 8 ตัว เพื่อแสดงค่าของตัวนับแบบไบนารี ขนาด 8 บิต เริ่มต้นการนับที่ 0 เมื่อกดปุ่ม “Up” แต่ละครั้ง จะเพิ่มค่าที่ละหนึ่ง แต่ถ้ากดปุ่ม “Down” จะทำให้ลดค่าลงทีละหนึ่ง

## Lab 17: Arduino Code (1/3)

```
/* Example: Count Up or Down In Binary.
*/
const byte UP_BUTTON_PIN = 3;           // define digital pin for Up button
const byte DOWN_BUTTON_PIN = 4;        // define digital pin for Down button
const byte LED_PINS [8] = { 5,6,7,8,9,10,11,12 }; // define digital pins for LEDs
byte count = 0;           // a counting variable (between 0..255)

void setup() {
    pinMode( UP_BUTTON_PIN, INPUT );    // use Up Button Pin as input
    pinMode( DOWN_BUTTON_PIN, INPUT ); // use Down Button Pin as input
    for (int i=0; i < 8; i++)           // for each LED pin do
        pinMode( LED_PINS[i], OUTPUT ); // use current LED Pin as output
}
void loop() {
    checkButtons();
    displayLEDs();
}
// Code continues on the next page.
```

## Lab 17: Arduino Code (2/3)

---

*// Code continues from the previous page.*

```
void checkButtons() {
    if ( isButtonClicked( UP_BUTTON_PIN ) ) // Has the Up Button been clicked?
        count = count + 1;    // increment count by 1
    if ( isButtonClicked( DOWN_BUTTON_PIN ) ) // Has the Down Button been clicked?
        count = count - 1;    // decrement count by 1
}
```

```
void displayLEDs() {
    byte t = count;
    for (int i=0; i < 8; i++) {           // for each LED pin do
        digitalWrite( LED_PINS[i], t & 1 ); // output bit to current LED pin
        t >>= 1;
    }
}
```

*// Code continues on the next page.*

## Lab 17: Arduino Code (3/3)

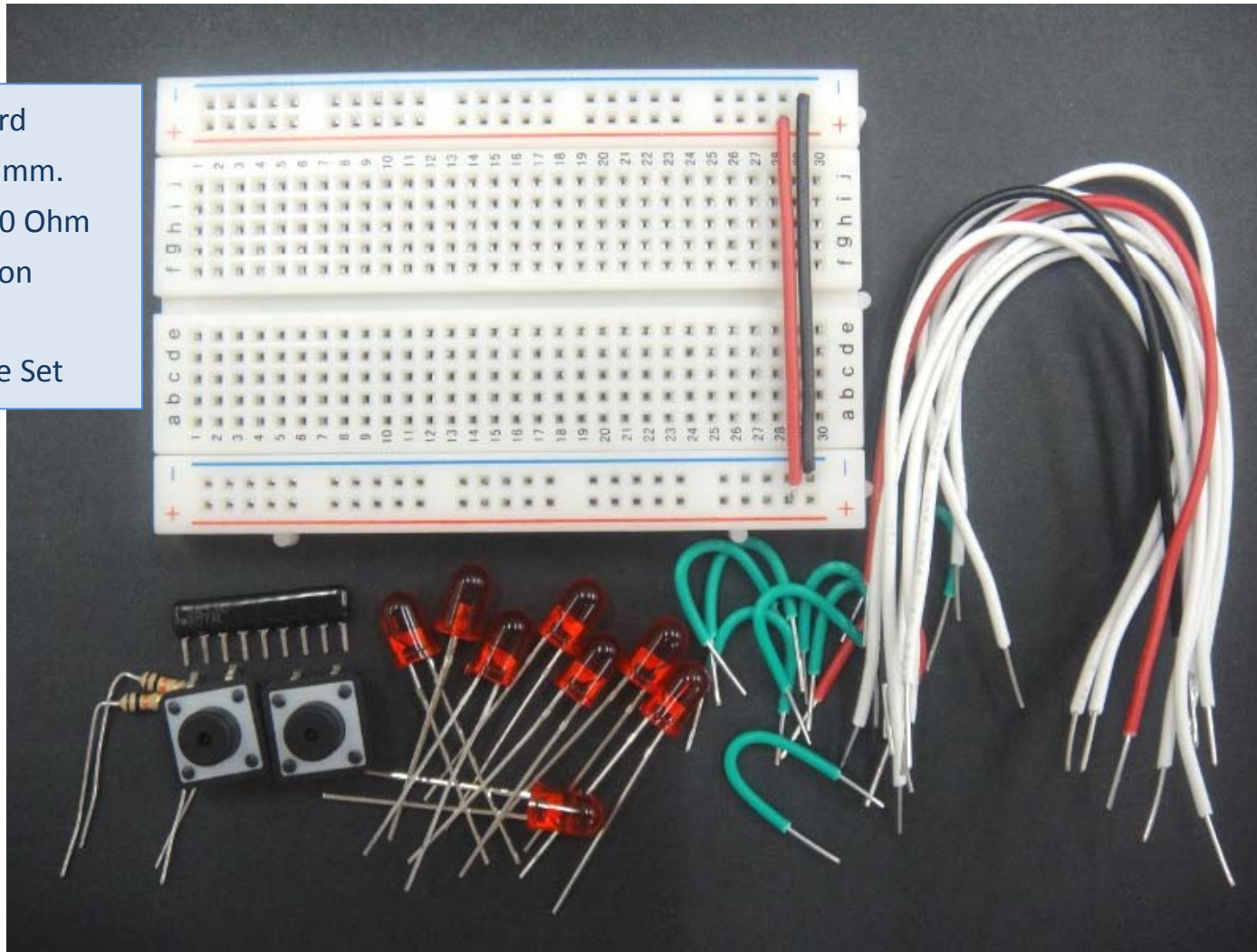
---

*// Code continues from the previous page.*

```
bool isButtonClicked( byte pinNumber ) {
  byte t = 5;
  while ( (digitalRead( pinNumber ) == LOW) && --t )
    delay(1);
  if ( t==0 ) {
    while ( digitalRead( pinNumber ) == LOW );
    return true;
  }
  return false;
}
```

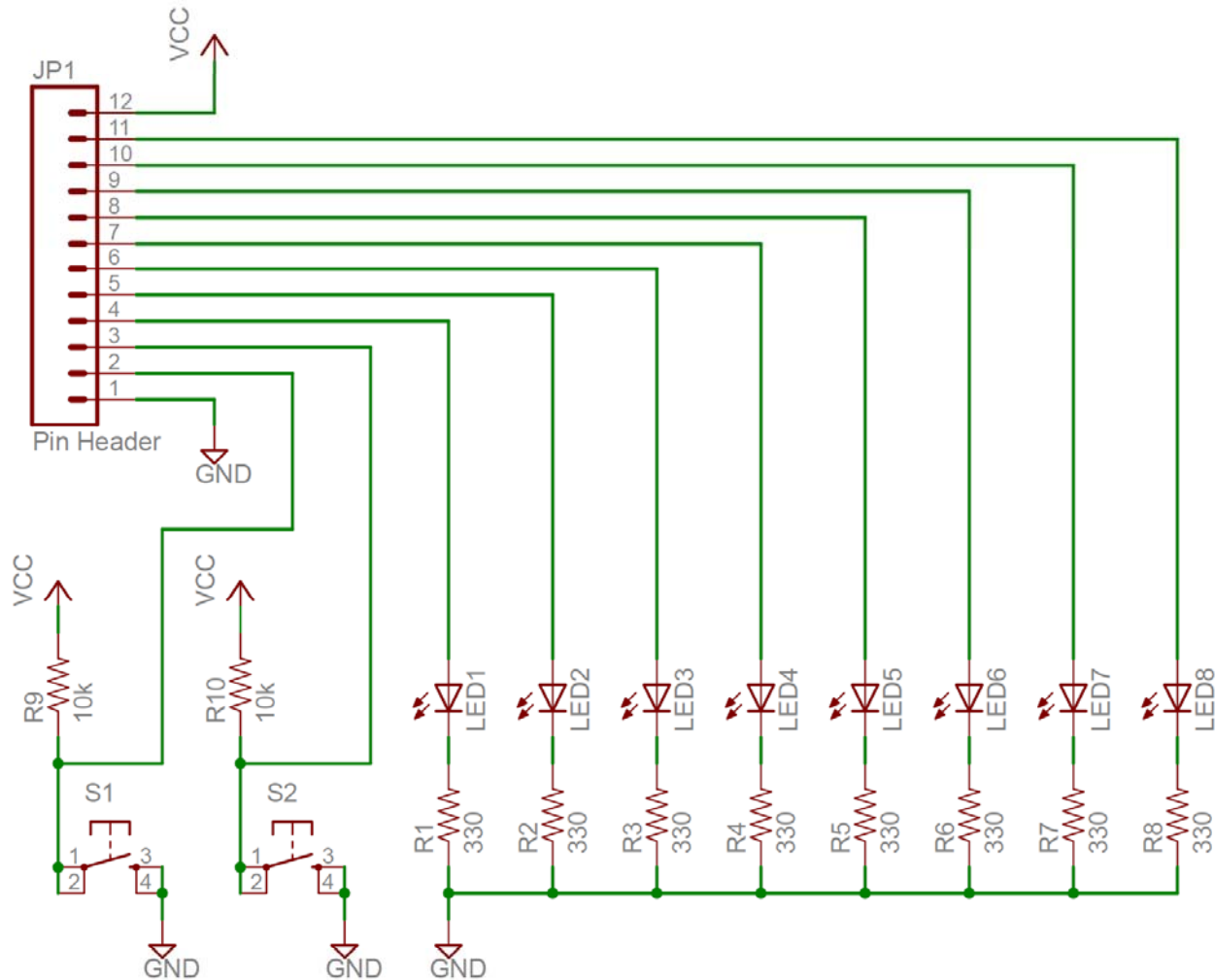
# Lab 17: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 8x Red LED 5mm.
- 1x R-Pack 330 Ohm
- 2x Push Button
- 2x 10k Ohm
- 1x Jump Wire Set



# Lab 17: Schematic – ฟังวงจร

จุดเชื่อมต่อ  
สัญญาณและ  
ไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 18: Turn On LEDs with DIP Switch!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ไดโอดเปล่งแสง (LED) 4 ตัว + ตัวต้านทานแบบชุด (R-Pack) 5 ขา 330 โอห์ม เป็นเอาต์พุตแบบดิจิทัล (ขาดีจิตอล หมายเลข 9,10,11,12) และสวิตช์เลื่อน (DIP Switch) แบบ 4 ช่อง + ตัวต้านทานแบบชุด 5 ขา 10k โอห์ม (ขาดีจิตอล หมายเลข 5,6,7,8) เพื่อใช้เป็นอินพุตดิจิทัล สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ด
- หมายเหตุ: ถ้าไม่มีตัวต้านทานแบบชุด ให้ใช้ตัวต้านทาน 4 ตัว 330 โอห์ม และอีก 4 ตัว 10k โอห์ม

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตจากสวิตช์เลื่อนทั้ง 4 ช่อง (สวิตช์แต่ละช่องทำงานแบบ Active-Low เมื่ออยู่ในตำแหน่ง ON จะได้ค่าเป็น 0 แต่ถ้าอยู่ในตำแหน่ง OFF จะได้ค่าเป็น 1) แล้วนำค่าที่ได้ไปสร้างเอาต์พุตสำหรับ LED ทั้ง 4 ตัว เรียงตามลำดับไป ถ้าเลื่อนสวิตช์ในช่องใดไปยังตำแหน่ง ON จะทำให้ LED ที่อยู่ในลำดับที่ตรงกัน “ติด” และถ้าอยู่ในตำแหน่งที่ OFF จะทำให้ LED “ดับ”

# Lab 18: Arduino Code

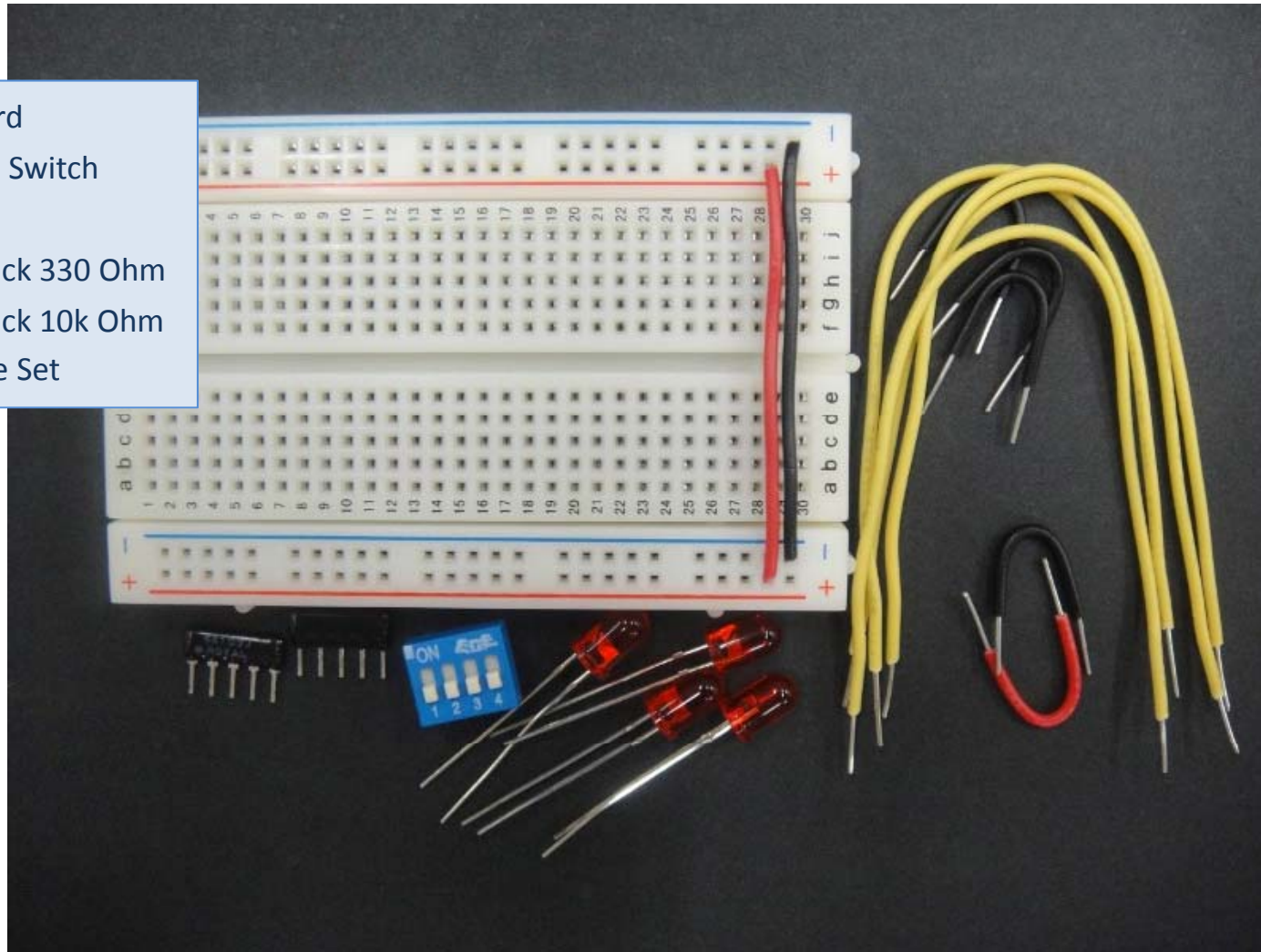
```
/* Example: Turn On LEDs with DIP Switch.
*/
const byte SW_PINS [8] = { 5,6,7,8 }; // define digital pins for DIP Switches
const byte LED_PINS [8] = { 9,10,11,12 }; // define digital pins for LEDs

void setup() {
    for (int i=0; i < 4; i++) { // for each position 0..3 do
        pinMode( SW_PINS[i], INPUT ); // use DIP Switch Pins as input
        pinMode( LED_PINS[i], OUTPUT ); // use LED Pins as output
    }
}

void loop() {
    int value;
    for (int i=0; i < 4; i++) { // for each position 0..3 do
        value = digitalRead( SW_PINS[i] ); // read a value from Switch input
        digitalWrite( LED_PINS[i], !value ); // write the (negated) value to LED
    }
    delay(100);
}
```

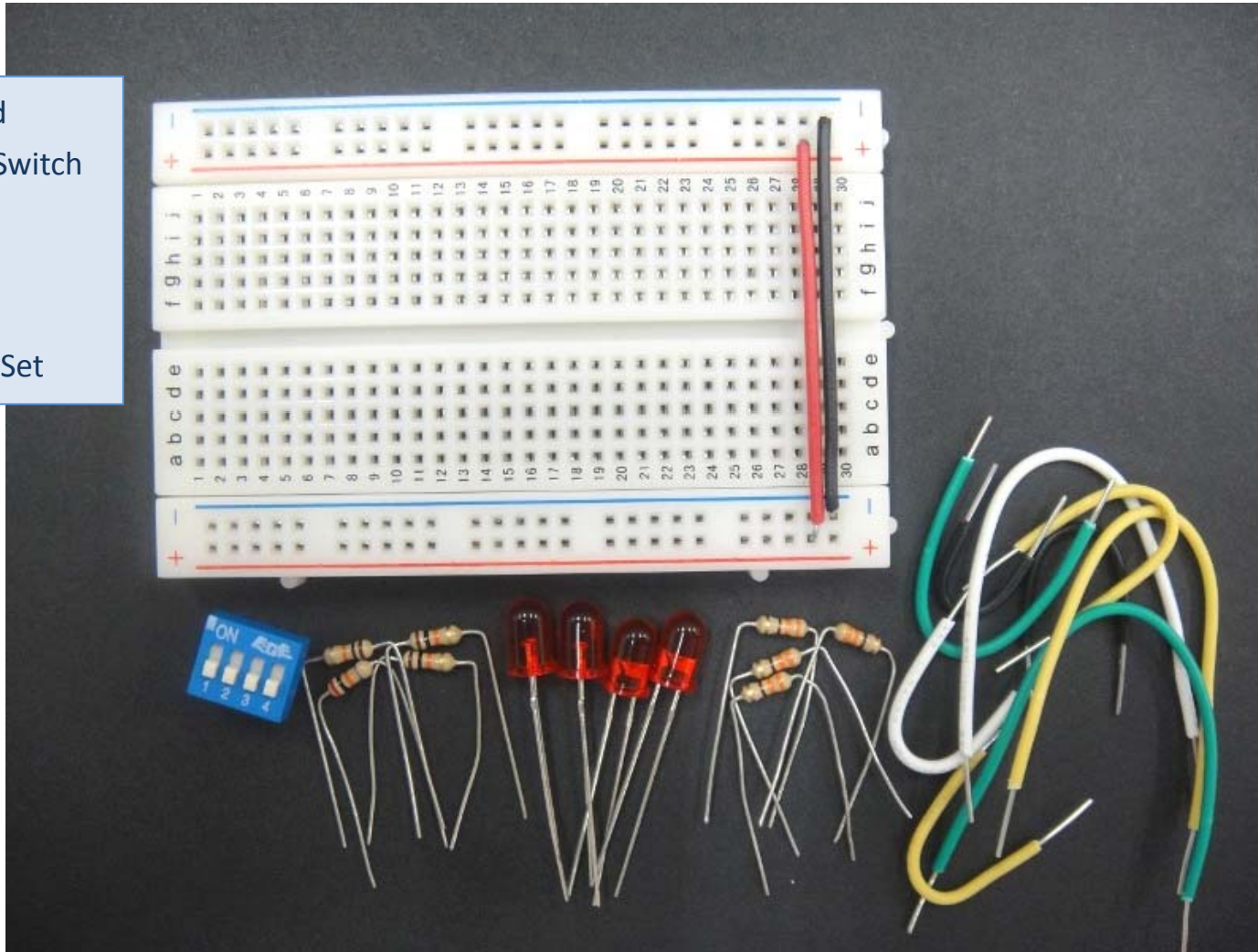
# Lab 18: Part List – รายการอุปกรณ์

- 1x Breadboard
- 1x 4-pos. DIP Switch
- 4x LED 5mm.
- 1x 5-Pin R-Pack 330 Ohm
- 1x 5-Pin R-Pack 10k Ohm
- 1x Jump Wire Set

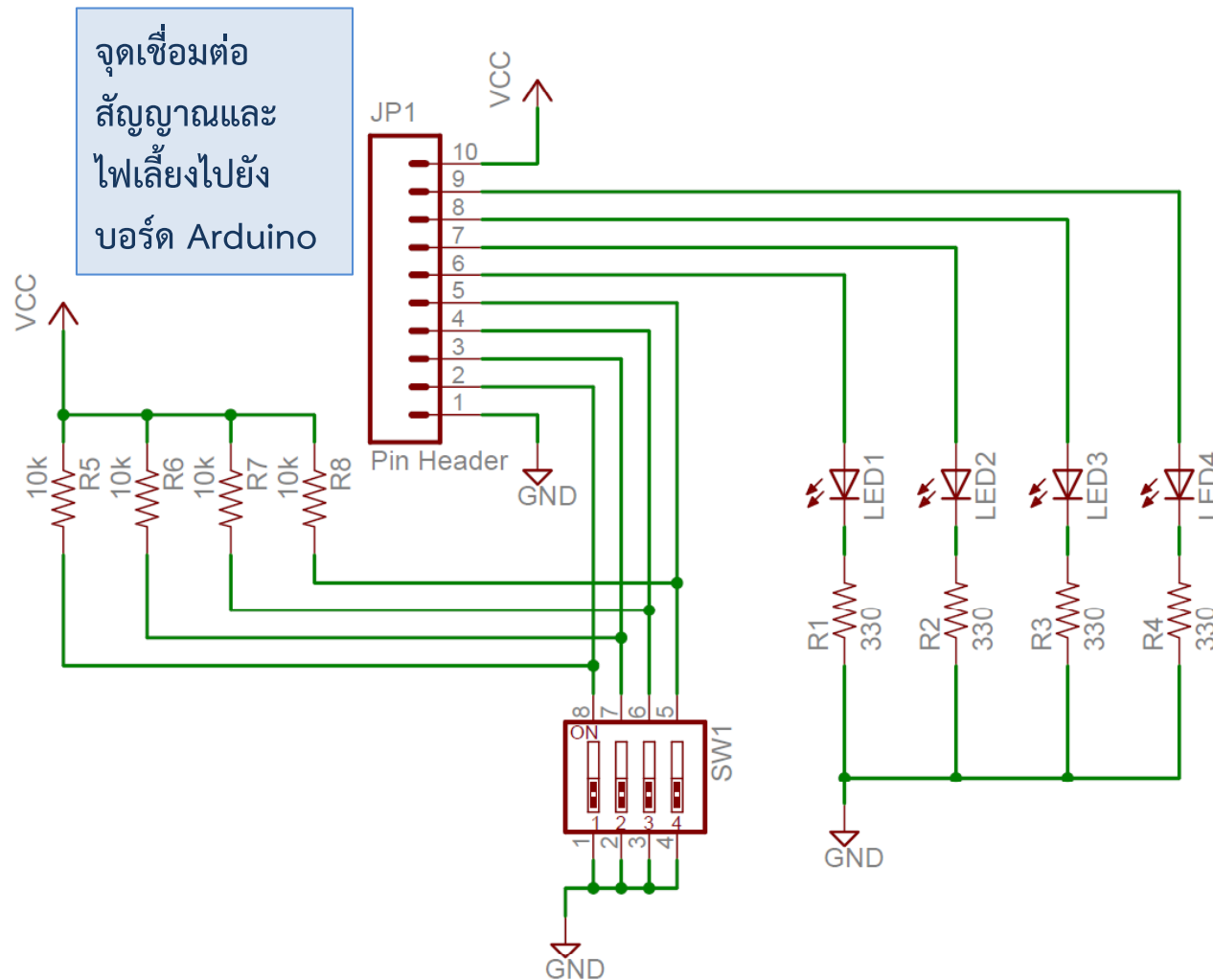


# Lab 18: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x 4-pos. DIP Switch
- 4x LED 5mm.
- 4x 330 Ohm
- 4x 10k Ohm
- 1x Jump Wire Set



# Lab 18: Schematic – ผังวงจร



# Lab 19: Make An LED Selector Switch!

---

- **วัสดุประสงค์**

- ต่อบริเวณโดยใช้ ไดโอดเปล่งแสง (LED) + ตัวต้านทาน 330 โอห์ม 3 ชุด เป็นเอาต์พุตแบบดิจิทัล (ขาดีจิตอล หมายเลข 8,9,10) และมีปุ่มกด + ตัวต้านทานแบบชุด 10k โอห์ม 3 ชุด (ขาดีจิตอล หมายเลข 5,6,7) เพื่อใช้เป็นอินพุตแบบดิจิทัล สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านอินพุตจากปุ่มกดแต่ละช่อง และตรวจสอบดูว่า ถ้ามีการกดปุ่มที่ตำแหน่งใด จะทำให้ LED ที่อยู่ในตำแหน่งที่ตรงกัน “ติด” และทำให้ LED ในตำแหน่งอื่นๆ “ดับ” แต่ถ้าไม่มีการกดปุ่มใดๆ จะ LED จะไม่มีการเปลี่ยนสถานะ

## Lab 19: Arduino Code (1/3)

---

```
/* Example: Make an LED Selector Switch.
*/
const byte BUTTON_PINS [3] = { 5,6,7 }; // define digital pins for buttons
const byte LED_PINS [3] = { 8,9,10 }; // define digital pins for LEDs

bool button_states [3];

void setup() {
  for (int i=0; i < 3; i++) {           // for each position 0..3 do
    pinMode( BUTTON_PINS[i], INPUT ); // use Button Pin as input
    pinMode( LED_PINS[i], OUTPUT ); // use LED Pin as output
  }
  digitalWrite( LED_PINS[0], HIGH ); // turn on the LED at position 0
}

// Code continues on the next page.
```

## Lab 19: Arduino Code (2/3)

---

```
// Code continues from the previous page.
void loop() {
  byte count = 0;
  for (int i=0; i < 3; i++) { // for each position do
    button_states[i] = false; // reset state of this button to false
    if ( isButtonClicked( BUTTON_PINS[i] ) ) {
      button_states[i] = true; // change state of this button to true (clicked)
      count++; // increment the number of clicked buttons
    }
  }
  if ( count == 1 ) { // if only one button has been clicked.
    for (int i=0; i < 3; i++) // for each position 0..3 do
      digitalWrite( LED_PINS[i],
                    (button_states[i]) ? HIGH : LOW ); // update LED state
    }
    delay(10);
  }
// Code continues from the next page.
}
```

## Lab 19: Arduino Code (3/3)

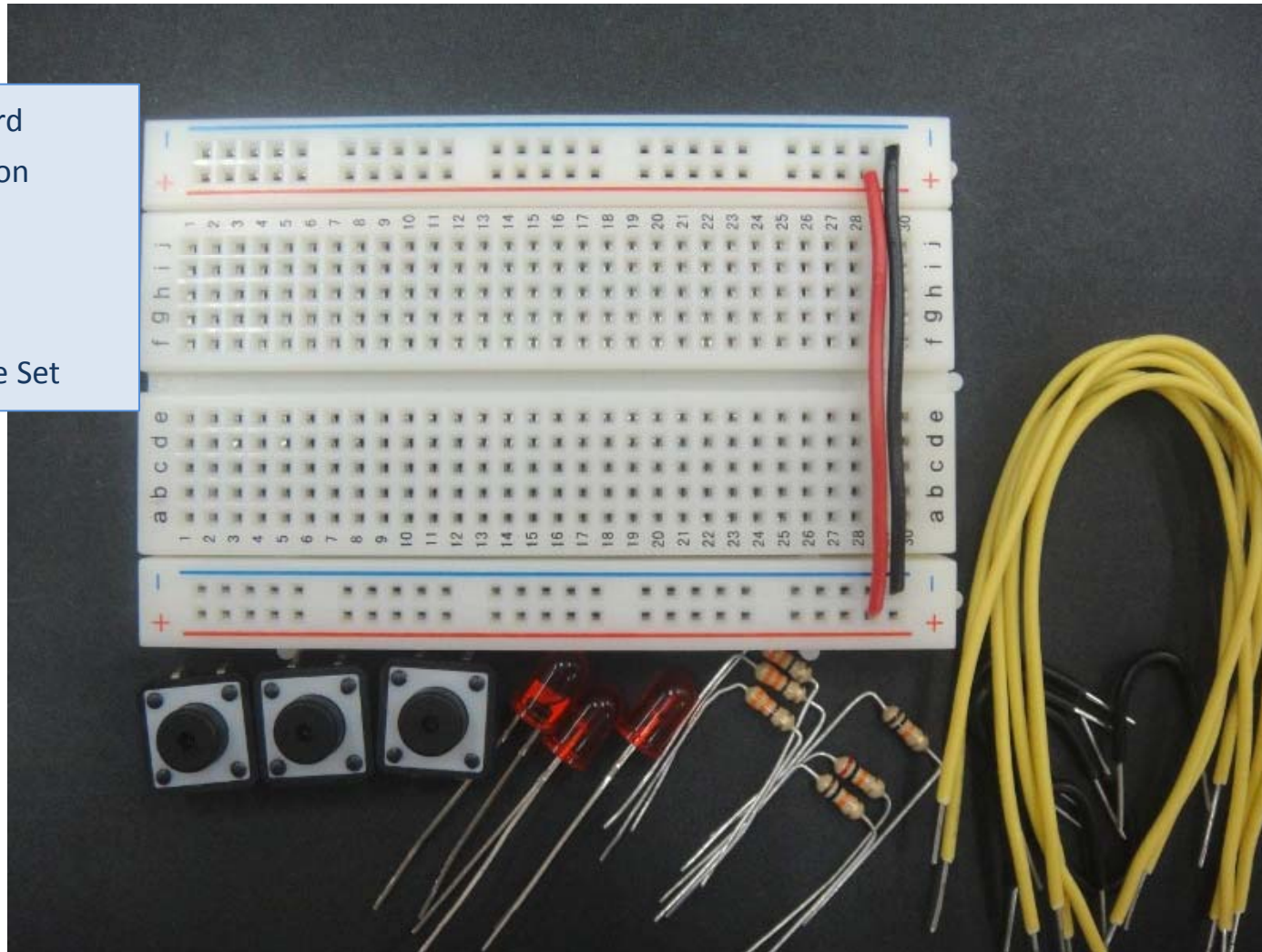
---

```
// Code continues from the previous page.

// check whether a button has been clicked
bool isButtonClicked( byte pinNumber ) {
    byte t = 5;
    while ( (digitalRead( pinNumber ) == LOW) && --t )
        delay(1);
    if ( t==0 ) {
        while ( digitalRead( pinNumber ) == LOW );
        return true;
    }
    return false;
}
```

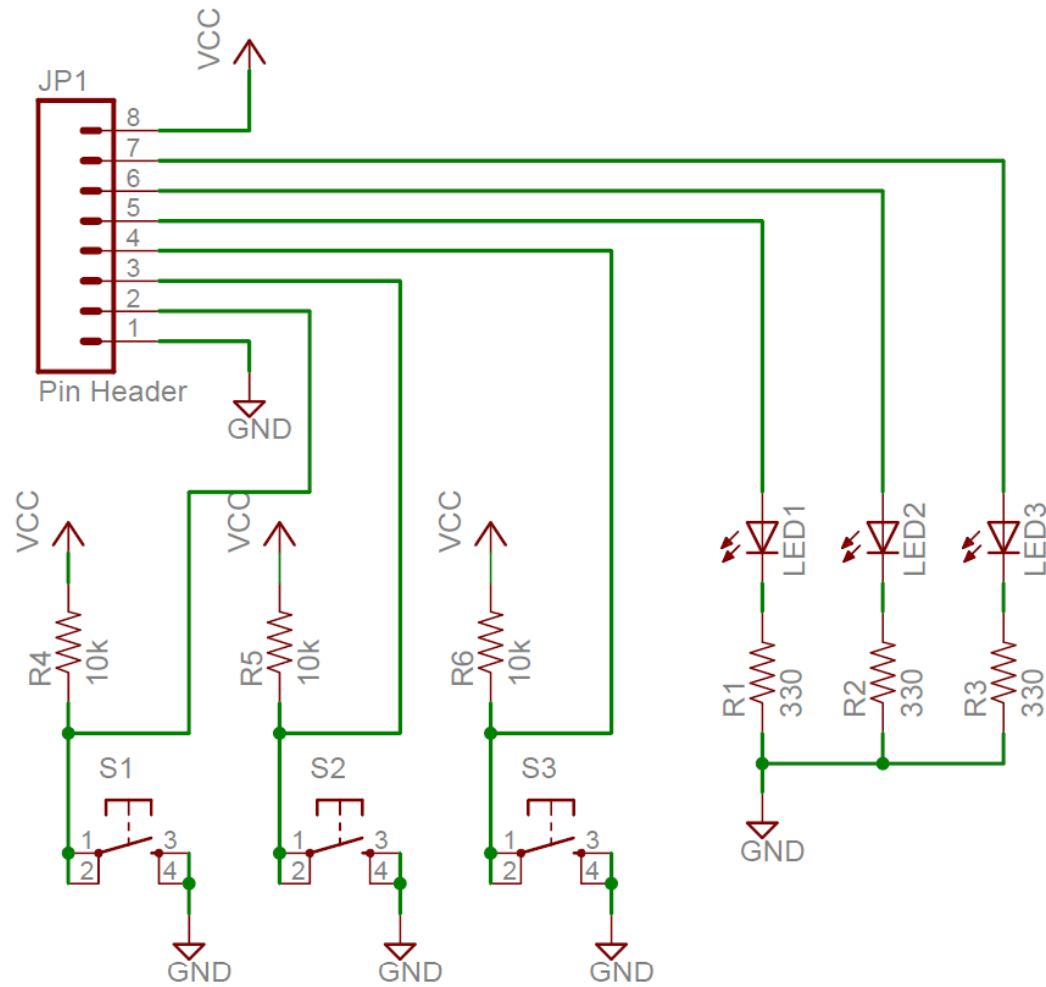
# Lab 19: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 3x Push Button
- 3x LED 5mm.
- 3x 330 Ohm
- 3x 10k Ohm
- 1x Jump Wire Set



# Lab 19: Schematic – ผังวงจร

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 20: Count Up From 0 to 9 in Decimal!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ โมดูล 7-Segment Display (Common Cathode) 1 หลักรวม + ตัวต้านทาน 330 โอห์ม 8 ตัว เป็นเอาต์พุตแบบดิจิตอล (ขาดิจิตอล หมายเลข 5,6,7,8,9,10,11,12) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรมอนิเตอร์บอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่เป็นตัวนับ โดยนับเพิ่มขึ้นทีละ 1 จาก 0 ถึง 9 แล้ววนซ้ำไปเรื่อยๆ ค่าของตัวนับจะถูกแปลงให้เป็นข้อมูล 8 บิต แล้วนำไปสร้างเอาต์พุตสำหรับ 7-Segment เพื่อแสดงเป็นตัวเลข 0-9 ที่ตรงกับค่าของตัวนับในขณะนั้น

## Lab 20: Arduino Code (1/2)

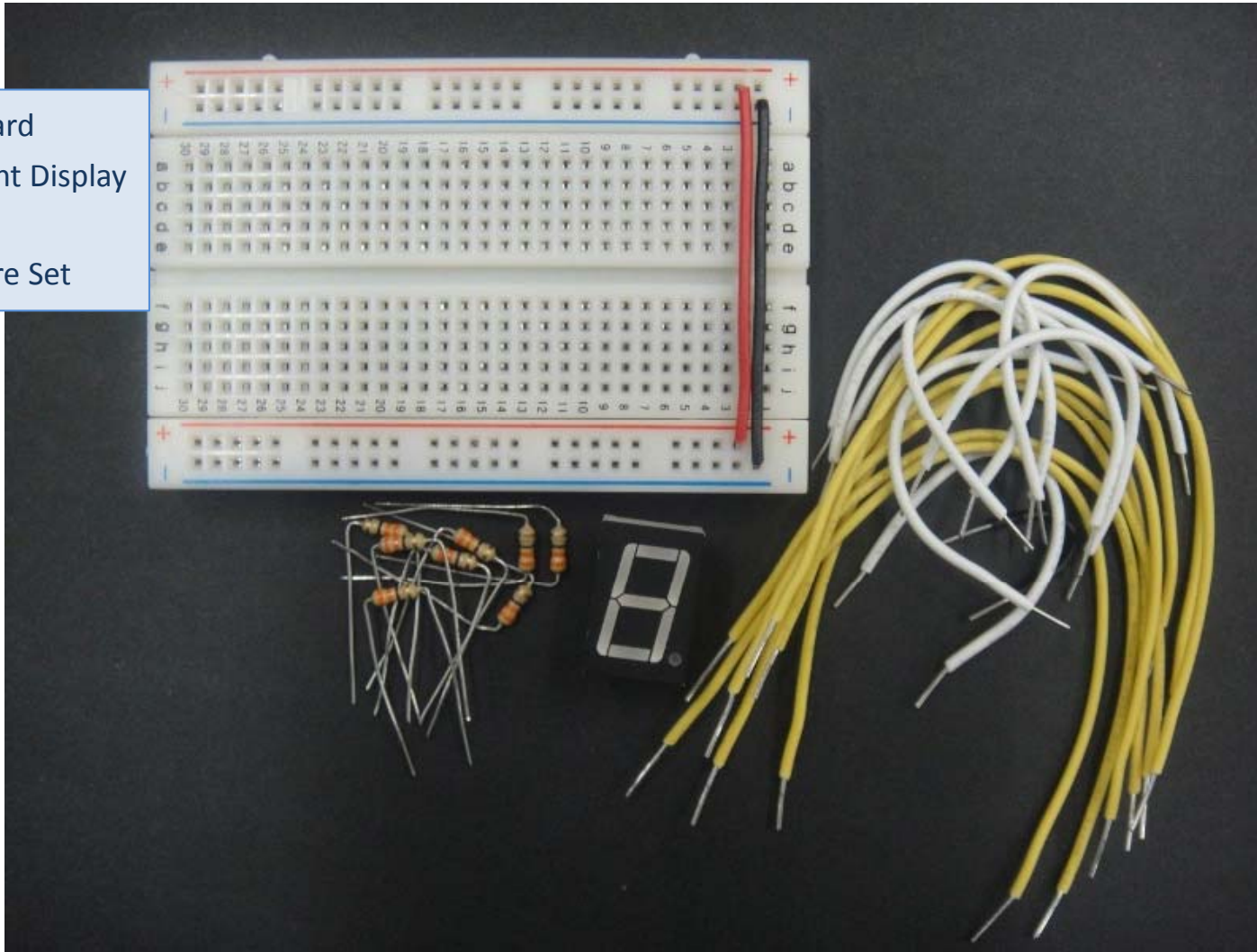
```
/* Example: Count Up From 0 to 9 in Decimal!
*/
const byte LED_PINS[] = {5,6,7,8,9,10,11,12};
                // define pins for 7-segment LEDs (a,b,c,d,e,f,g,h)
byte count = 0;
const byte DIGIT_7SEG[] = {// Look-up table for Common-Cathode 7-Segment Display
/*hgfedcba*/
B00111111,      /*0*/
B00000110,      /*1*/
B01011011,      /*2*/
B01001111,      /*3*/
B01100110,      /*4*/
B01101101,      /*5*/
B01111100,      /*6*/
B00000111,      /*7*/
B01111111,      /*8*/
B01101111       /*9*/
};
// Code continues on the next page.
```

## Lab 20: Arduino Code (2/2)

```
// Code continues from the previous page.
void setup() {
    for (int i=0; i < 8; i++) // for each position 0..7 do
        pinMode( LED_PINS[i], OUTPUT ); // use LED Pin as output
    }
void loop(){
    displayDigit( count ); // display current counter's value
    count = (count+1) % 10; // count up by 1 (modulo 10)
    delay(1000);
}
void displayDigit( byte value ) {
    if ( 0 <= value && value < 10 ) {
        byte value = DIGIT_7SEG[ count ]; // decode
        for (int i=0; i < 8; i++) { // foreach LED segment 0..7 do
            digitalWrite( LED_PINS[i], (value & 1) ); // write bit to LED
            value >>= 1;
        }
    }
}
```

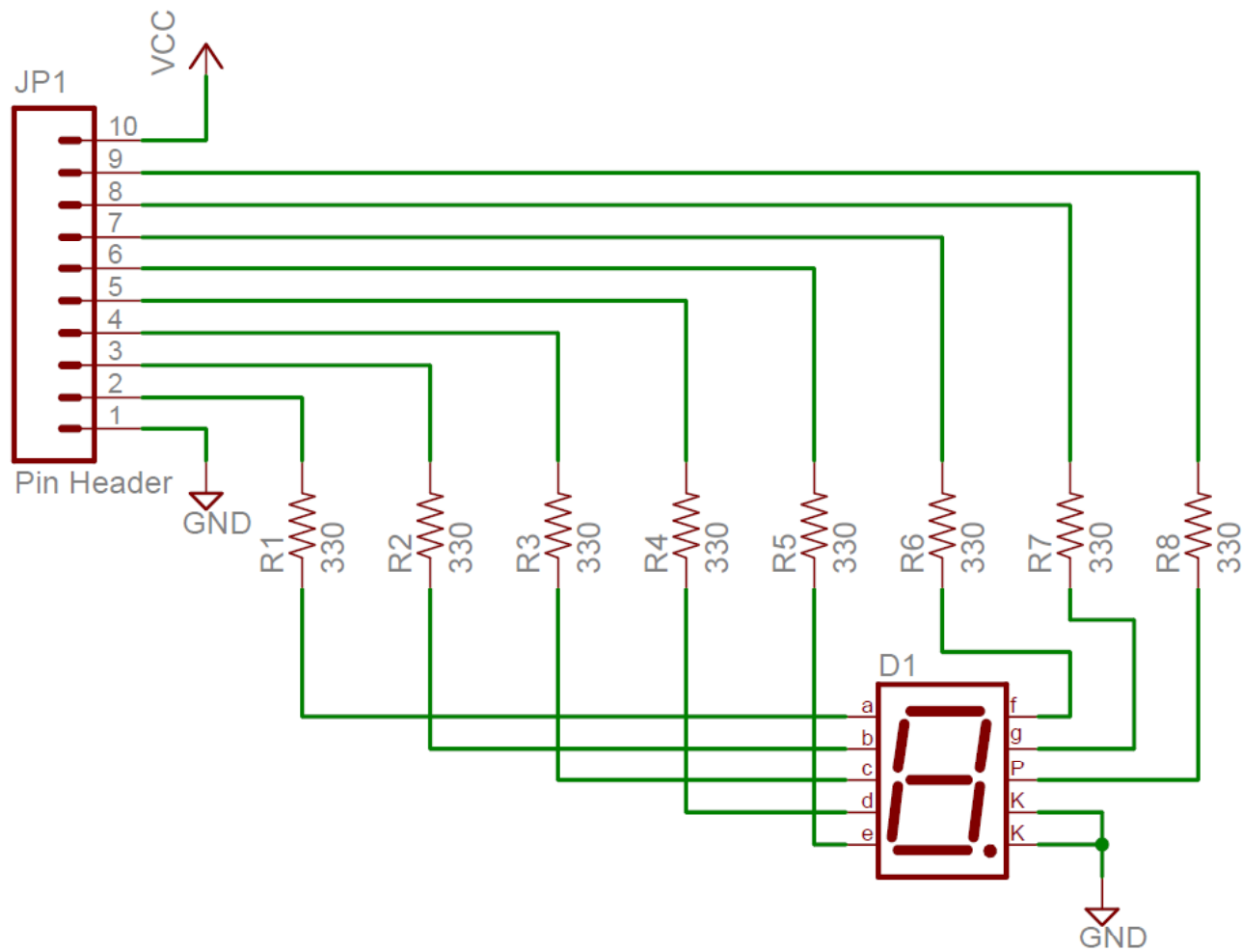
# Lab 20: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x 7-Segment Display
- 8x 330 Ohm
- 1x Jump Wire Set



# Lab 20: Schematic – ฟังก์ชัน

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



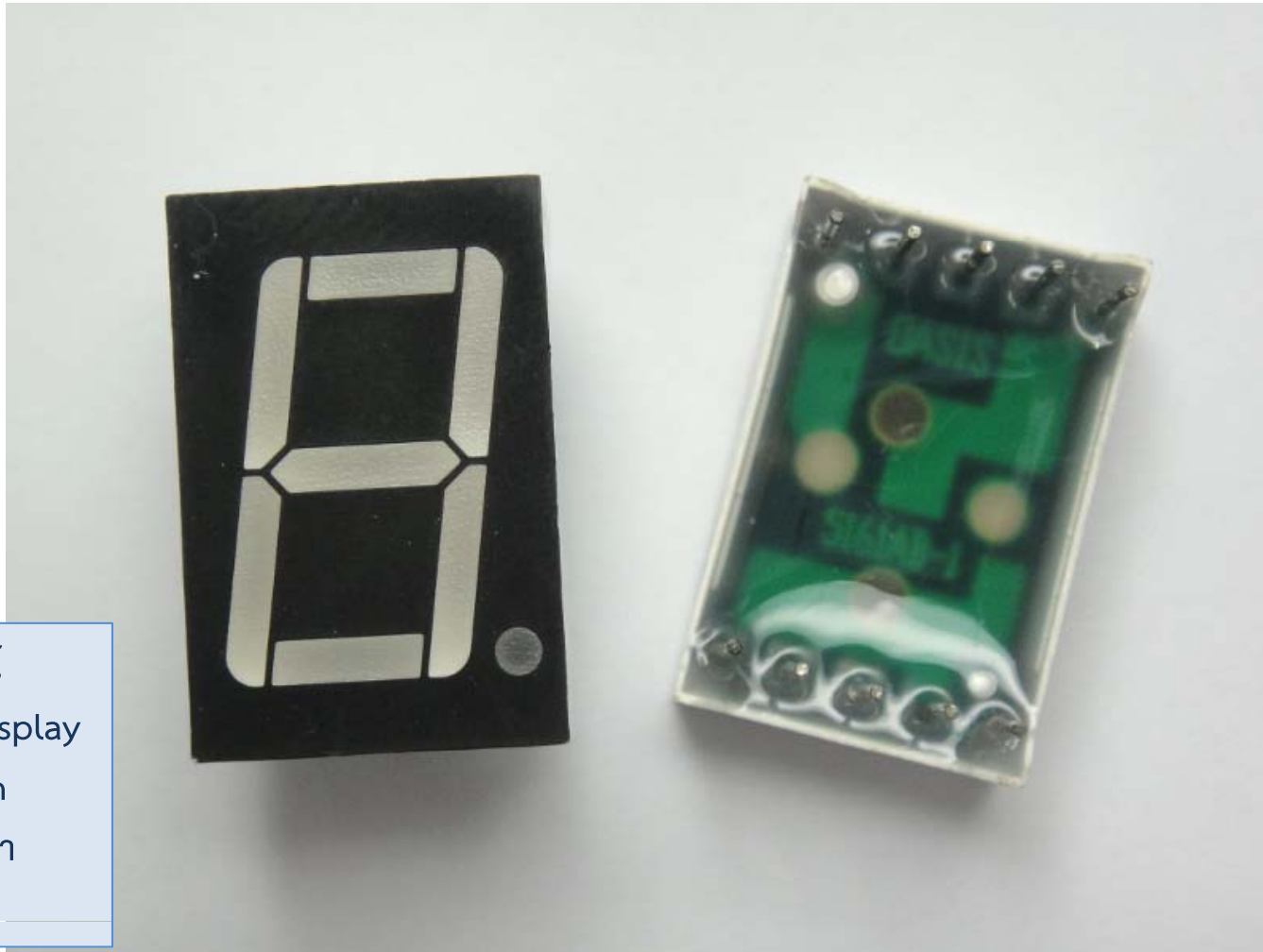
## Lab 20: การทดลองต่อ 7-Segment Display

---

- อุปกรณ์ 7-Segment Display สำหรับการทดลอง เป็นแบบ Common Cathode (CC) ที่ทั้งหมด 8 ขา แบ่งเป็นสองด้าน (บนและล่าง) ด้านละ 4 ขา
- ในการต่อวงจร จะต้องรู้ตำแหน่งและความหมายของขาแต่ละขาก่อน ซึ่งสามารถดูได้จากดาต้าชีท (Datasheet) ของผู้ผลิต หรือสามารถหาได้จากการทดลอง
  - ขั้นตอนแรก ให้ใช้มัลติมิเตอร์ ตรวจสอบหาขา COM ซึ่งมี 2 ขา และทั้งสองขานี้เชื่อมต่อเข้าด้วยกันทางไฟฟ้า
  - เมื่อรู้ตำแหน่งของขา COM ทั้งสองแล้ว ให้ทดลองต่อวงจรโดยจ่ายไฟกระแสตรง เช่น +3V หรือ +5V ผ่านตัวต้านทาน เพื่อจำกัดปริมาณกระแส (เช่น 330 โอห์ม) ให้กับขาต่างๆที่ไม่ใช่ขา COM แล้วดูว่าส่วนใดของ 7-Segment ให้แสงสว่าง ให้ทำขั้นตอนสำหรับขาอื่นๆที่เหลือ

## Lab 20: การทดลองต่อ 7-Segment Display

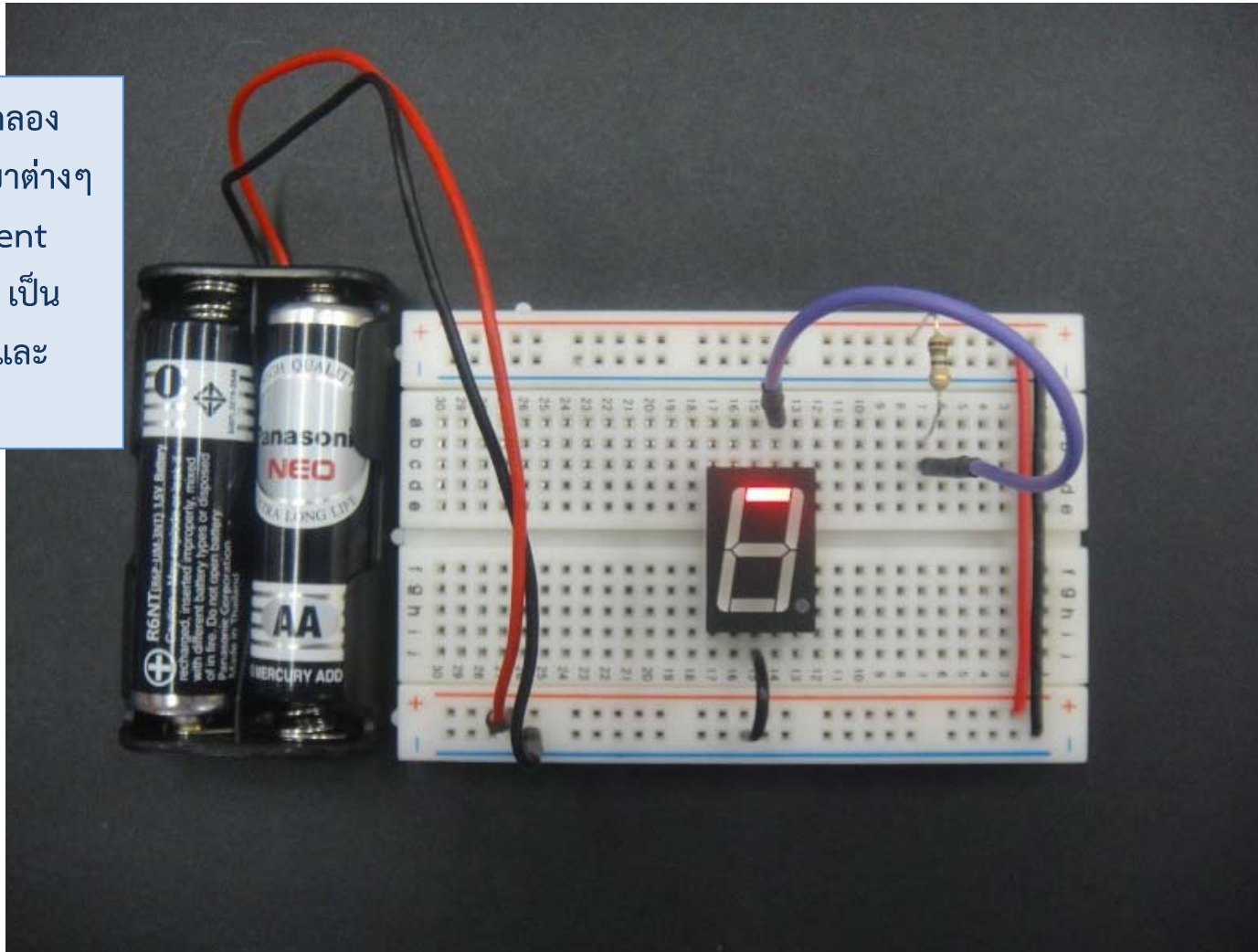
---



ตัวอย่างอุปกรณ์  
7-Segment Display  
แบบ Common  
Cathode (ใช้ขา  
แคโทดร่วมกัน)

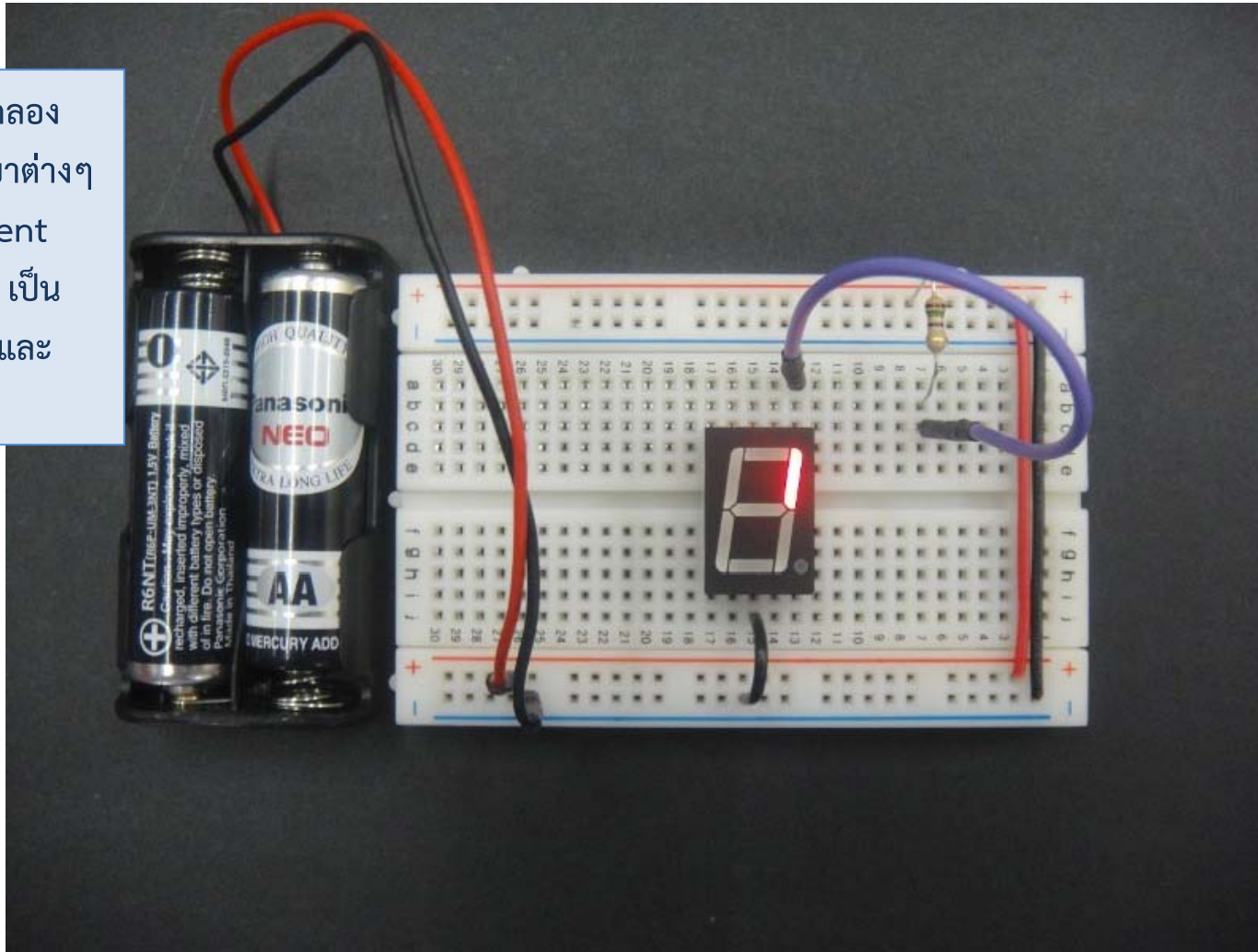
# Lab 20: การทดลองต่อ 7-Segment Display

ตัวอย่างการทดลอง  
จ่ายกระแสให้ขาต่างๆ  
ของ 7-Segment  
(ตัวอย่างในรูป เป็น  
การต่อที่ขา a และ  
COM)



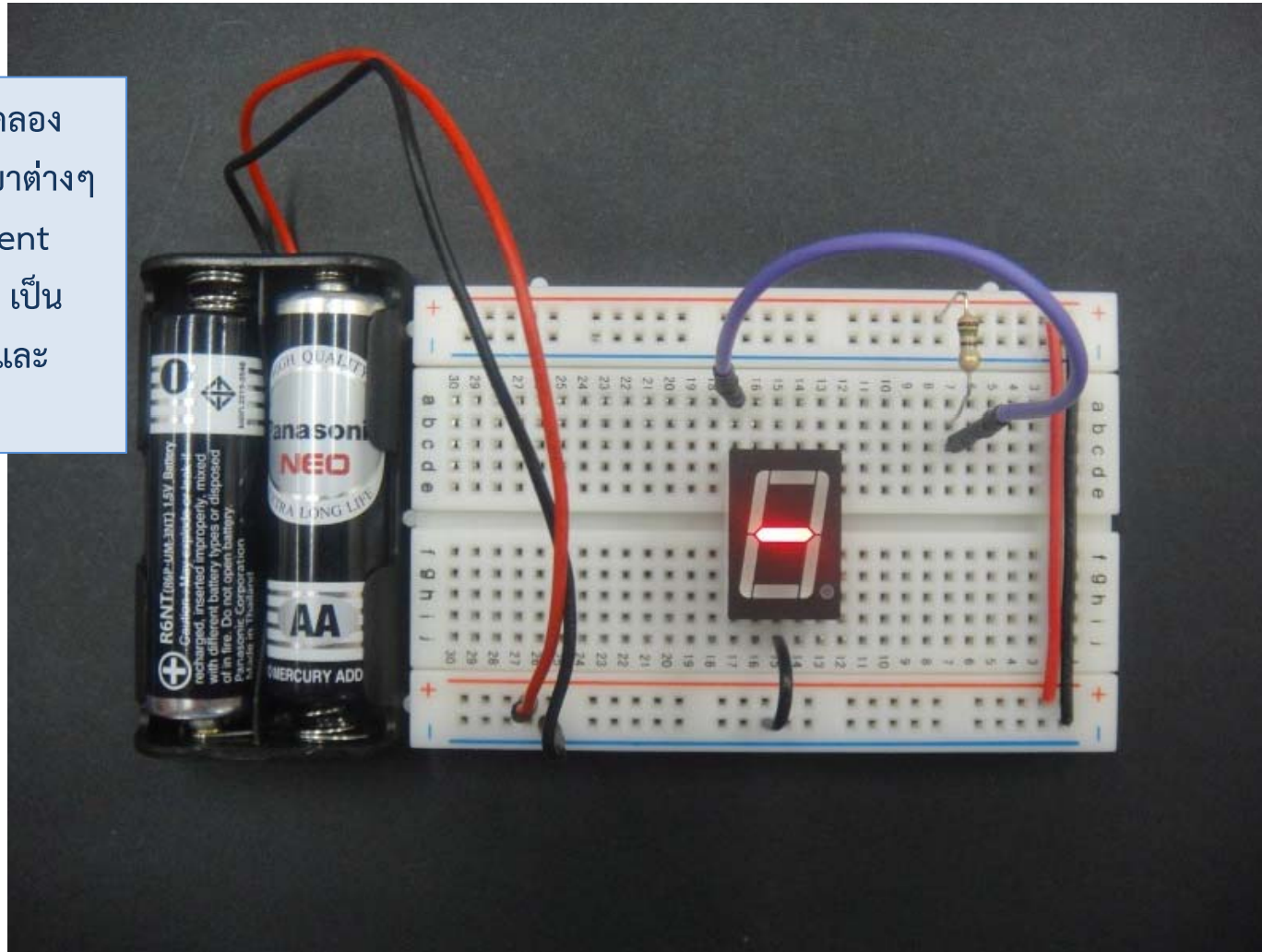
# Lab 20: การทดลองต่อ 7-Segment Display

ตัวอย่างการทดลอง  
จ่ายกระแสให้ขาต่างๆ  
ของ 7-Segment  
(ตัวอย่างในรูป เป็น  
การต่อที่ขา b และ  
COM)



# Lab 20: การทดลองต่อ 7-Segment Display

ตัวอย่างการทดลอง  
จ่ายกระแสให้ขาต่างๆ  
ของ 7-Segment  
(ตัวอย่างในรูป เป็น  
การต่อที่ขา g และ  
COM)



# Lab 21: Adjust Motor Speed with Trimpot!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้ ชิพ L293D ซึ่งเป็นไอซีสำหรับควบคุมและจ่ายกระแสให้มอเตอร์ไฟฟ้ากระแสตรงได้ 2 ตัว แต่จะเลือกใช้เพียงช่องเดียว (ทดลองกับมอเตอร์ 1 ตัว ทำงานได้ที่ +5VDC) โดยจะต้องสร้างสัญญาณ In2, In1, En ให้ชิพตัวนี้ (ขาคิจิตอล หมายเลข 5,6,9) มีปุ่มกด + ตัวต้านทาน 10k โอห์ม เพื่อใช้เป็นอินพุต (ขาคิจิตอล หมายเลข 3) และตัวต้านทานปรับค่าได้ (Trimpot) เพื่อสร้างแรงดันเป็นอินพุตแบบแอนะล็อก (ขาแอนะล็อก หมายเลข 0) สำหรับบอร์ด Arduino
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจรบนเบรตบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตเพื่อป้อนให้ชิพ L293D สำหรับมอเตอร์ช่องที่ 1 และสามารถปรับความเร็วของมอเตอร์ได้ โดยการหมุนปรับค่าที่ตัวต้านทานปรับค่าได้ นอกจากนั้นยังสามารถกดปุ่มได้เพื่อเปลี่ยนทิศทางการหมุนของมอเตอร์ได้

## Lab 21: Arduino Code (1/3)

```
/* Example: Adjust Motor Speed with Trimpot.
*/
const byte BUTTON_PIN = 3;           // define Button pin (digital pin 3)
const byte L293D_IN2_PIN = 5;        // define L293D In2 pin (digital pin 5)
const byte L293D_IN1_PIN = 6;        // define L293D In1 pin (digital pin 6)
const byte L293D_EN_PIN = 9;         // define L293D En pin (digital pin 9)
const byte TRIMPOT_PIN = 0;          // define Trimpot pin (analog pin 0)
bool forward = true;

void setup() {
    pinMode( BUTTON_PIN, INPUT );     // used Button pin as input
    pinMode( L293D_IN1_PIN, OUTPUT ); // used as output to L293D In1
    pinMode( L293D_IN2_PIN, OUTPUT ); // used as output to L293D In2
    pinMode( L293D_EN_PIN, OUTPUT );  // used as output to L293D En
    digitalWrite( L293D_IN1_PIN, HIGH );
    digitalWrite( L293D_IN2_PIN, HIGH );
    analogWrite( L293D_EN_PIN, 0 );   // disable motor run
}
// Code continues on the next page.
```

## Lab 21: Arduino Code (2/3)

---

*// Code continues from the previous page.*

```
void loop() {
  int value = analogRead( TRIMPOT_PIN );    // read analog input
  value = value/4; // scale down from 0..1023 to 0..255
  analogWrite( L293D_EN_PIN, value );    // set PWM duty cycle for the motor
  if ( isButtonClicked( BUTTON_PIN ) ) { // if button has been clicked
    forward = !forward;                  // change motor direction
    analogWrite(L293D_EN_PIN, 0 );       // disable motor temporarily
    delay(200);
  }
  digitalWrite( L293D_IN1_PIN, forward ? HIGH:LOW ); // output to L293D IN1 pin
  digitalWrite( L293D_IN2_PIN, forward ? LOW:HIGH ); // output to L293D IN2 pin
  delay(50);
}
```

*// Code continues on the next page.*

## Lab 21: Arduino Code (3/3)

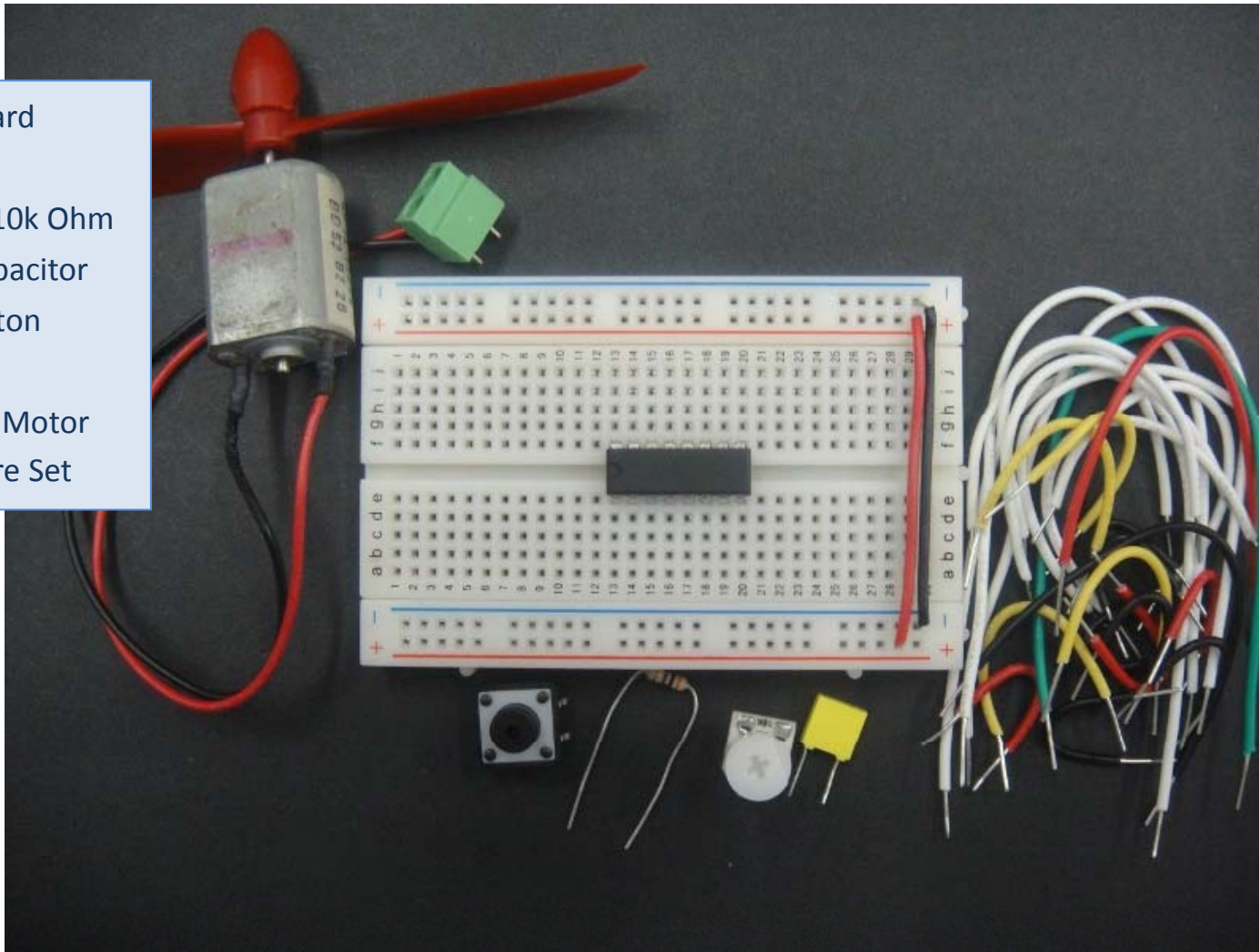
---

```
// Code continues from the previous page.

bool isButtonClicked( byte pinNumber ) {
    byte t = 5;
    while ( (digitalRead( pinNumber ) == LOW) && --t )
        delay(1);
    if ( t==0 ) {
        while ( digitalRead( pinNumber ) == LOW );
        return true;
    }
    return false;
}
```

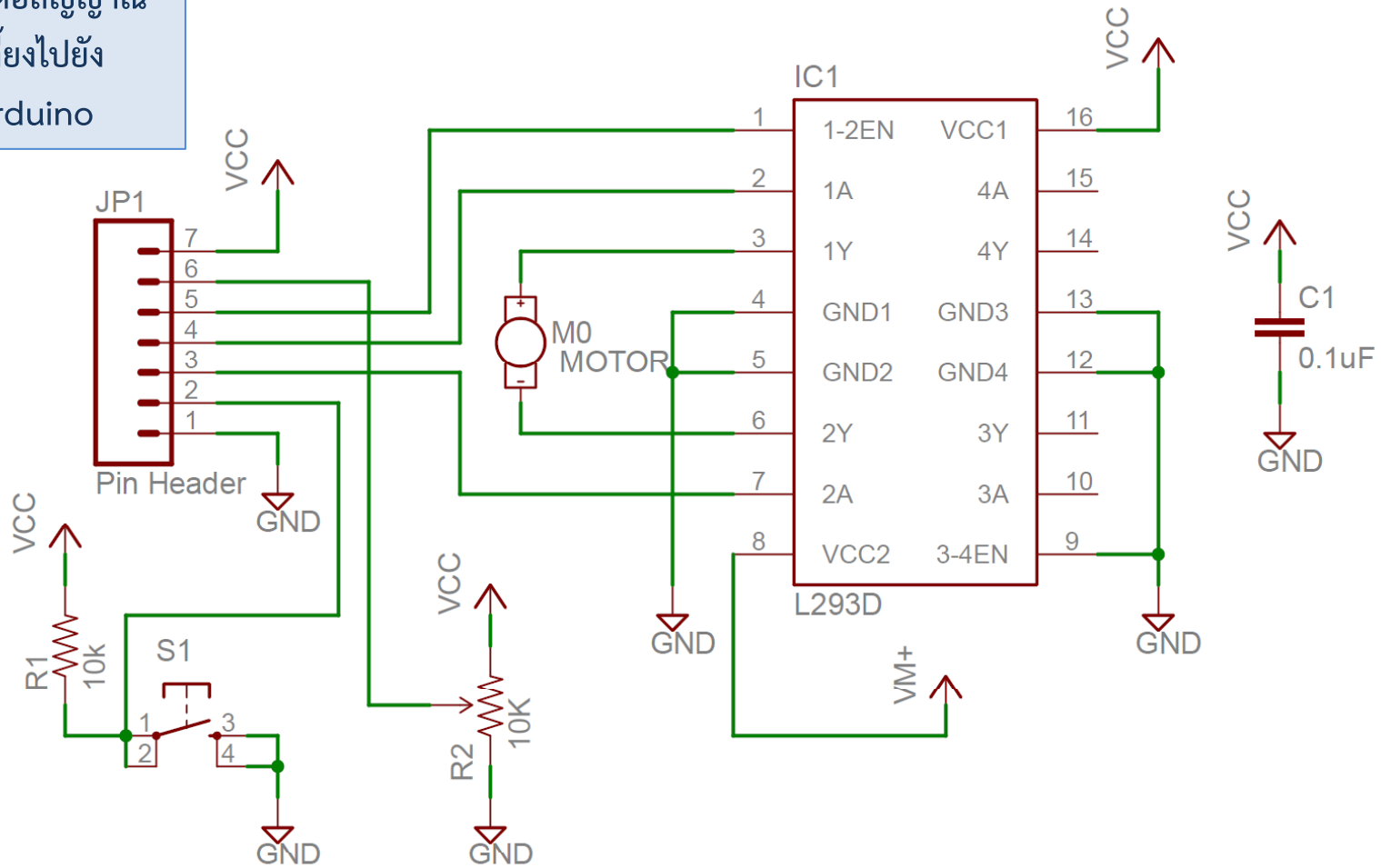
# Lab 21: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x L293D IC
- 1x Trimpot 10k Ohm
- 1x 0.1uF Capacitor
- 1x Push Button
- 1x 10k Ohm
- 1x Small DC Motor
- 1x Jump Wire Set



# Lab 21: Schematic – ฟังวงจร

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 22: Turn On/Off Motor with NPN Transistor!

---

- **วัตถุประสงค์**

- ต่่วงจรโดยใช้ อุปกรณ์แยกสัญญาณด้วยแสง หรือที่เรียกว่า **Optocoupler** และทรานซิสเตอร์ **BJT** แบบ **NPN** ในการควบคุมการหมุน (เปิด/ปิด) ของมอเตอร์กระแสตรงขนาดเล็กที่ใช้กระแสไม่มาก (ต่ำกว่า 500mA)
- สำหรับบอร์ด Arduino มีเอาต์พุตดิจิทัล (ขาดิจิทัล หมายเลข 5) เพื่อควบคุมการทำงานของมอเตอร์ และมีอินพุตดิจิทัลจากปุ่มกด (ขาดิจิทัล หมายเลข 6)
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตดิจิทัล เพื่อควบคุมการเปิดปิดกระแสให้มอเตอร์ไฟฟ้ากระแสตรง ตามสถานะกดปุ่ม เมื่อกดปุ่มค้างไว้จะทำให้มอเตอร์หมุน แต่ถ้าปล่อย จะทำให้มอเตอร์หยุดหมุน

## Lab 22: Arduino Code

---

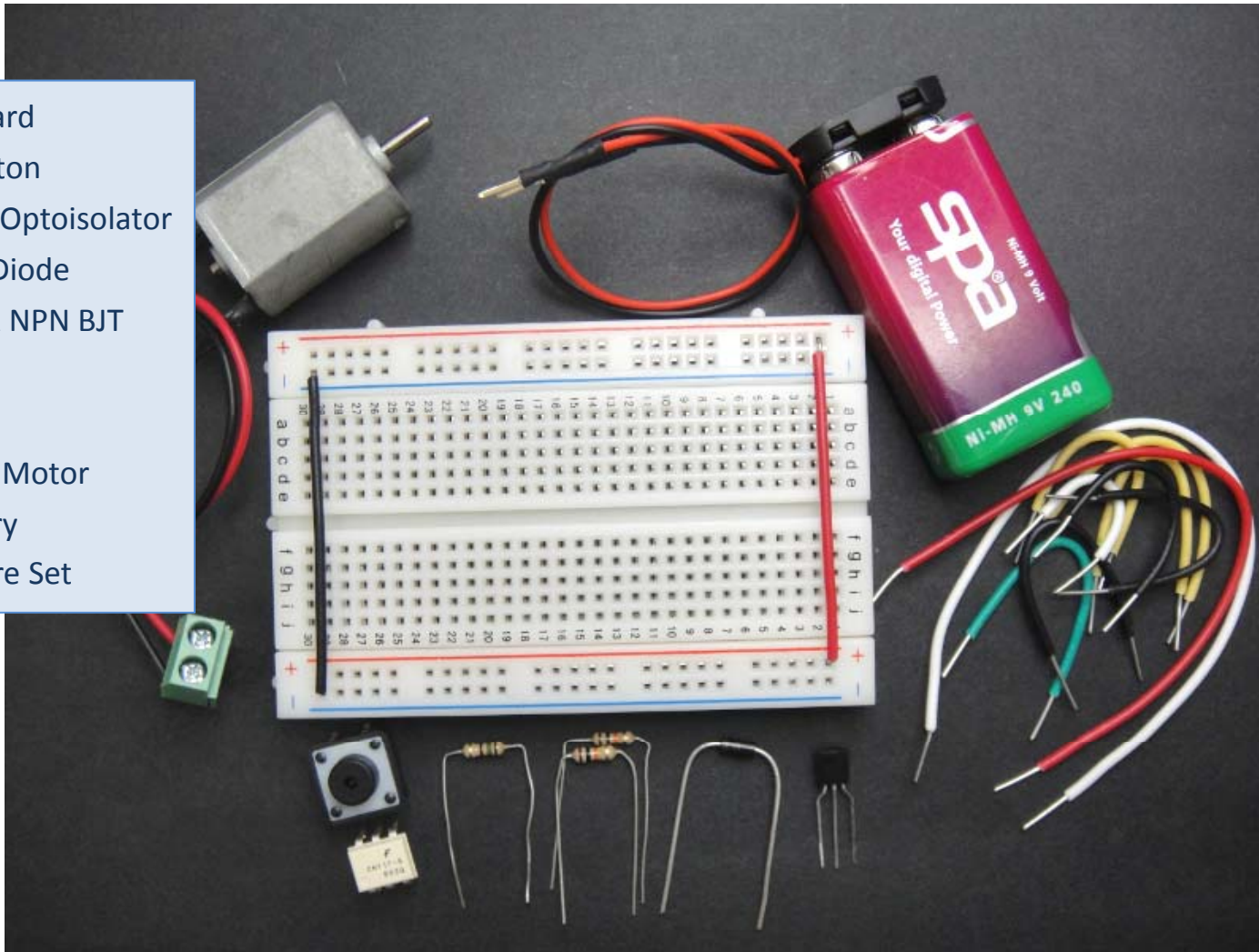
```
/* Example: Turn On/Off Motor with NPN Transistor and Optocoupler!
*/
const byte MOTOR_PIN = 5;  // use Digital Pin 5 (D5) to turn on/off motor
const byte BUTTON_PIN = 6; // use Digital Pin 6 (D6) to read input from
button

void setup() {
  pinMode( MOTOR_PIN, OUTPUT );  // use MOTOR Pin as output
  pinMode( BUTTON_PIN, INPUT );  // use BUTTON Pin as input
}

void loop() {
  if ( digitalRead( BUTTON_PIN ) == LOW ) // Button is pressed (active-low).
    digitalWrite( MOTOR_PIN, HIGH ); // output High: turn on motor
  else
    digitalWrite( MOTOR_PIN, LOW );  // output Low: turn off motor
  delay( 10 );  // wait for 10 msec
}
```

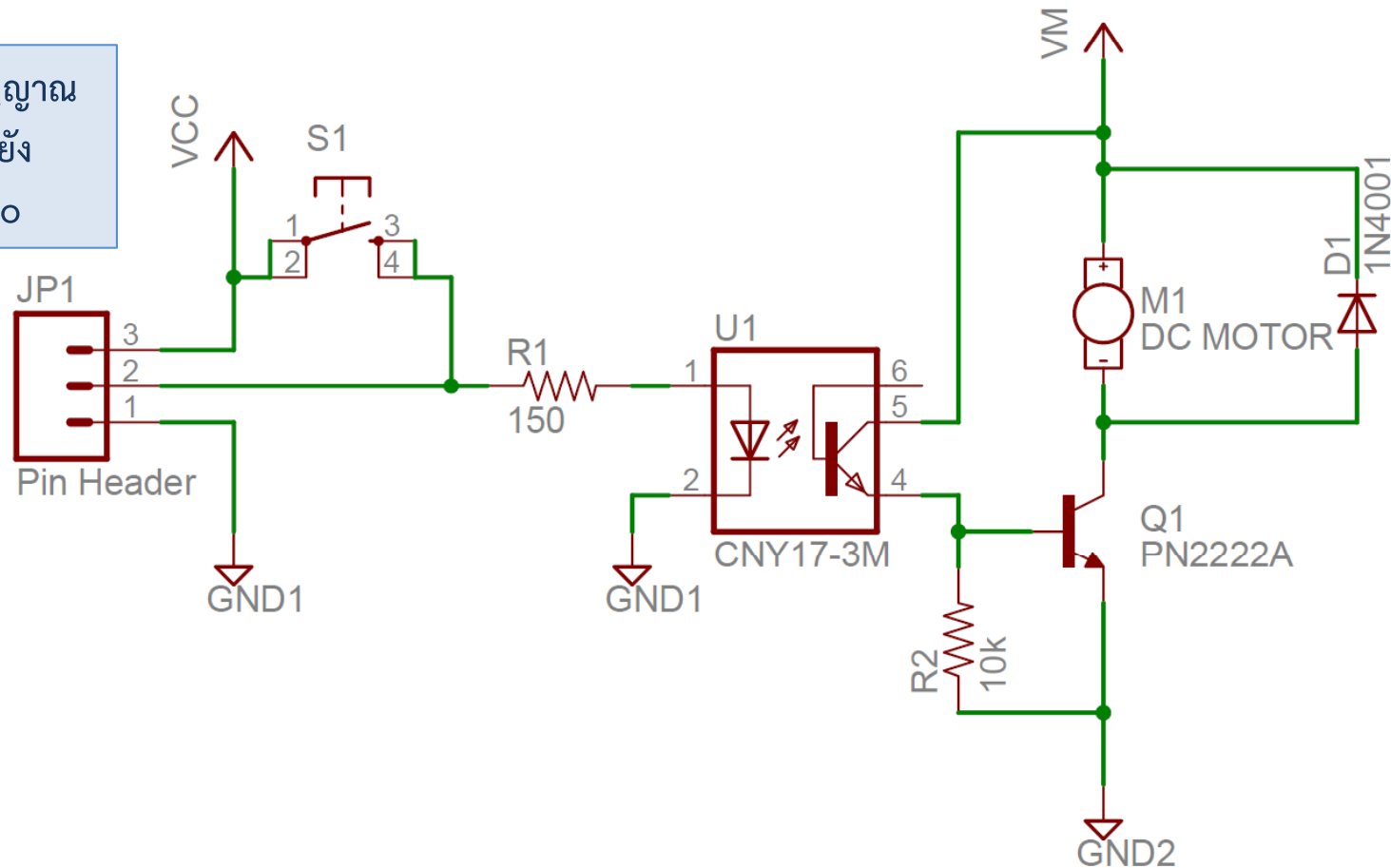
# Lab 22: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Push Button
- 1x CNY17-3 Optoisolator
- 1x 1N4001 Diode
- 1x PN2222A NPN BJT
- 1x 150 Ohm
- 2x 10k Ohm
- 1x Small DC Motor
- 1x 9V Battery
- 1x Jump Wire Set



# Lab 22: Schematic – ฟังวงจร

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino

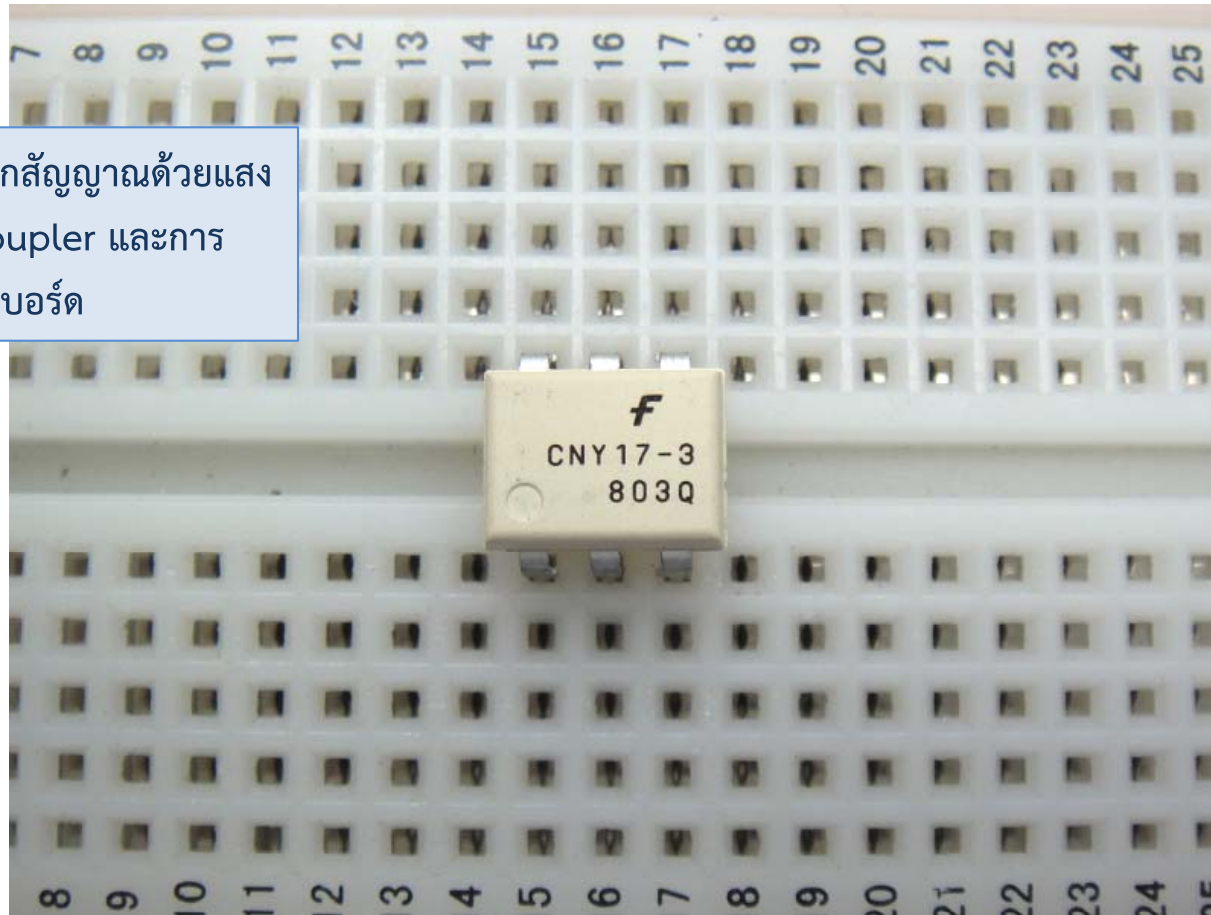


สังเกตว่า มีการแยกกราวนด์ในวงจร แบ่งเป็น GND1 และ GND2

## Lab 22: ตัวอย่างอุปกรณ์ - Optocoupler

---

ตัวอย่างอุปกรณ์แยกสัญญาณด้วยแสง  
CNY17-3 Optocoupler และการ  
เสียบขาลงบนเบอร์ดบอร์ด



# Lab 23: Turn On/Off Motor with N-MOSFET!

---

- **วัตถุประสงค์**

- ต่่วงจรโดยใช้ อุปกรณ์แยกสัญญาณด้วยแสง หรือที่เรียกว่า **Optocoupler** และทรานซิสเตอร์แบบ **N-channel MOSFET** ในการควบคุมการหมุน (เปิด/ปิด) ของมอเตอร์กระแสตรง
- ต่่วงจรเข้ากับบอร์ด Arduino โดยมีเอาต์พุตดิจิทัล (ขาดีจิตอล หมายเลข 5) เพื่อควบคุมการทำงานของมอเตอร์ และมีอินพุตดิจิทัลจากปุ่มกด (ขาดีจิตอล หมายเลข 6)
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ดบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตดิจิทัล เพื่อควบคุมการเปิดปิดกระแสให้มอเตอร์ไฟฟ้ากระแสตรง ผ่านวงจรมแยกสัญญาณทางไฟฟ้าและทรานซิสเตอร์เป็นตัวเปิด-ปิดกระแสให้มอเตอร์

## Lab 23: Arduino Code

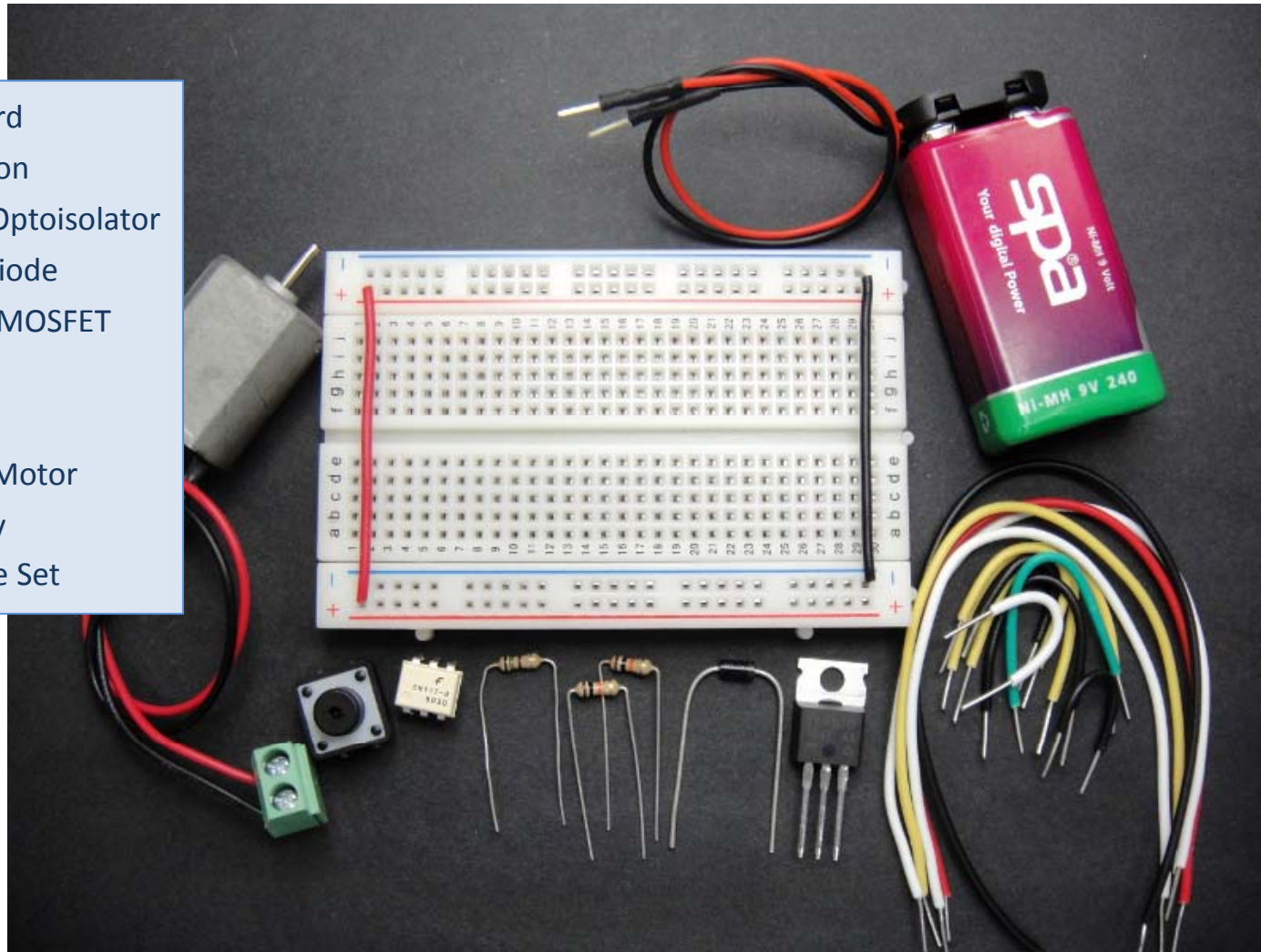
```
/* Example: Turn On/Off Motor with N-channel MOSFET and Optocoupler!
*/
const byte MOTOR_PIN = 5;    // use Digital Pin 5 (D5) to turn on/off motor
const byte BUTTON_PIN = 6;   // use Digital Pin 6 (D6) to read input
from button

void setup() {
    pinMode( MOTOR_PIN, OUTPUT );    // use MOTOR Pin as output
    pinMode( BUTTON_PIN, INPUT );    // use BUTTON Pin as input
}

void loop() {
    if ( digitalRead( BUTTON_PIN ) == LOW ) // Button is pressed (active-low).
        digitalWrite( MOTOR_PIN, HIGH );    // output High: turn on motor
    else
        digitalWrite( MOTOR_PIN, LOW );    // output Low: turn off motor
    delay( 10 );    // wait for 10 msec
}
```

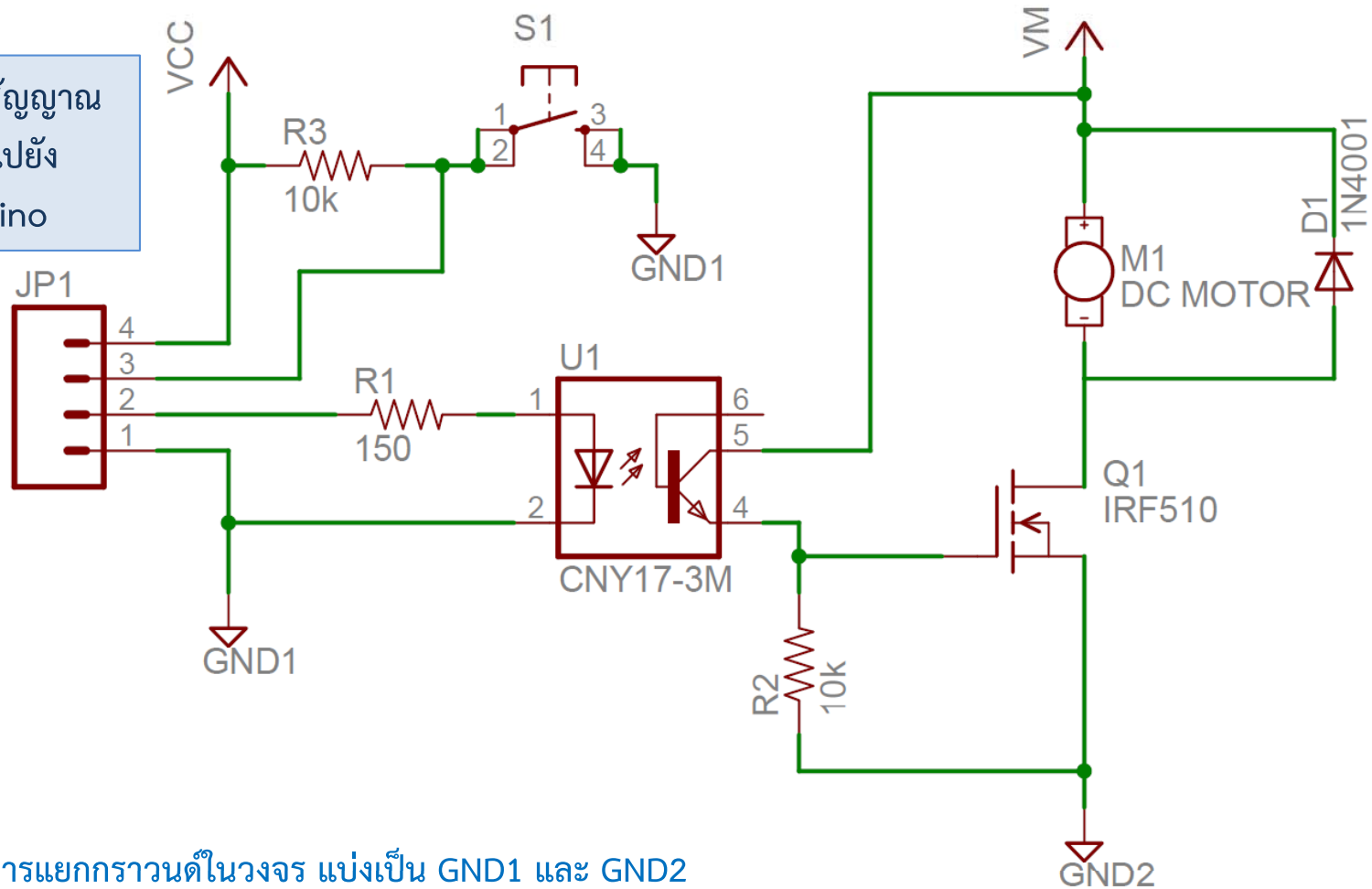
# Lab 23: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Push Button
- 1x CNY17-3 Optoisolator
- 1x 1N4001 Diode
- 1x IRF510 N-MOSFET
- 1x 150 Ohm
- 2x 10k Ohm
- 1x Small DC Motor
- 1x 9V Battery
- 1x Jump Wire Set



# Lab 23: Schematic – ฟังวงจร

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



สังเกตว่า มีการแยกกราวด์ในวงจร แบ่งเป็น GND1 และ GND2

# Lab 24: Turn On/Off Motor with P-MOSFET!

---

- **วัตถุประสงค์**

- ต่อย่างจรงโดยใช้ อุปกรณ์แยกสัญญาณด้วยแสง หรือที่เรียกว่า **Optocoupler** และทรานซิสเตอร์แบบ **P-channel MOSFET** ในการควบคุมการหมุน (เปิด/ปิด) ของมอเตอร์กระแสตรง
- ต่อย่างจรงเข้ากับบอร์ด Arduino โดยมีเอาต์พุตดิจิตอล (ขาติจิตอล หมายเลข 5) เพื่อควบคุมการทำงานของมอเตอร์ และมีอินพุตดิจิตอลจากปุ่มกด (ขาติจิตอล หมายเลข 6)
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนเบรตบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตดิจิตอลเพื่อควบคุมการเปิด-ปิดกระแสให้มอเตอร์ไฟฟ้ากระแสตรง ผ่านวงจรแยกสัญญาณทางไฟฟ้าและทรานซิสเตอร์เป็นตัวเปิด/ปิดกระแสให้มอเตอร์

# Lab 24: Arduino Code

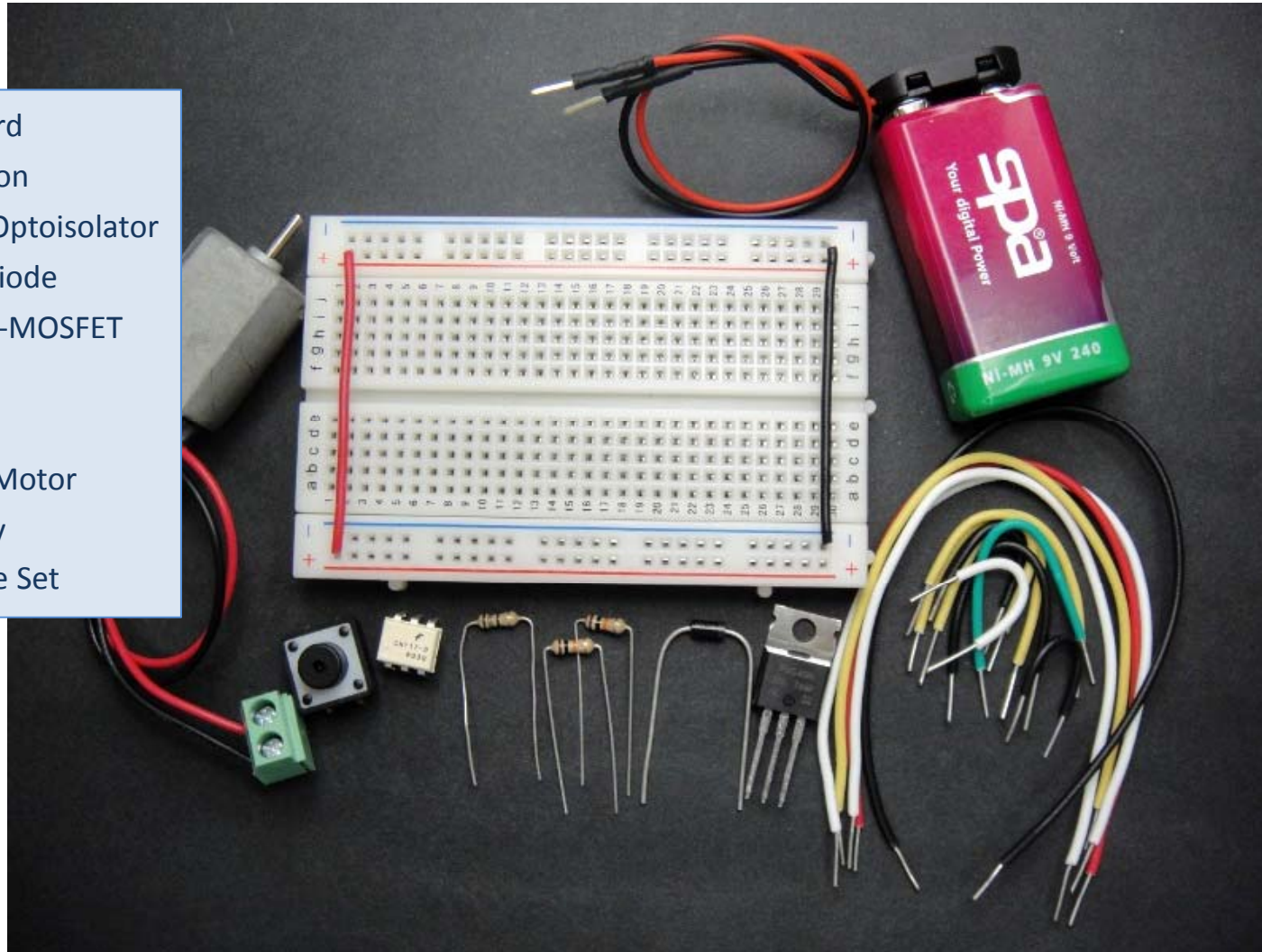
```
/* Example: Turn On/Off Motor with P-channel MOSFET and Optocoupler!
*/
const byte MOTOR_PIN = 5;    // use Digital Pin 5 (D5) to turn on/off motor
const byte BUTTON_PIN = 6;   // use Digital Pin 6 (D6) to read input from
button

void setup() {
  pinMode( MOTOR_PIN, OUTPUT );    // use MOTOR Pin as output
  pinMode( BUTTON_PIN, INPUT );    // use BUTTON Pin as input
}

void loop() {
  if ( digitalRead( BUTTON_PIN ) == LOW ) // Button is pressed (active-low).
    digitalWrite( MOTOR_PIN, HIGH ); // output High: turn on motor
  else
    digitalWrite( MOTOR_PIN, LOW ); // output Low: turn off motor
  delay( 10 );    // wait for 10 msec
}
```

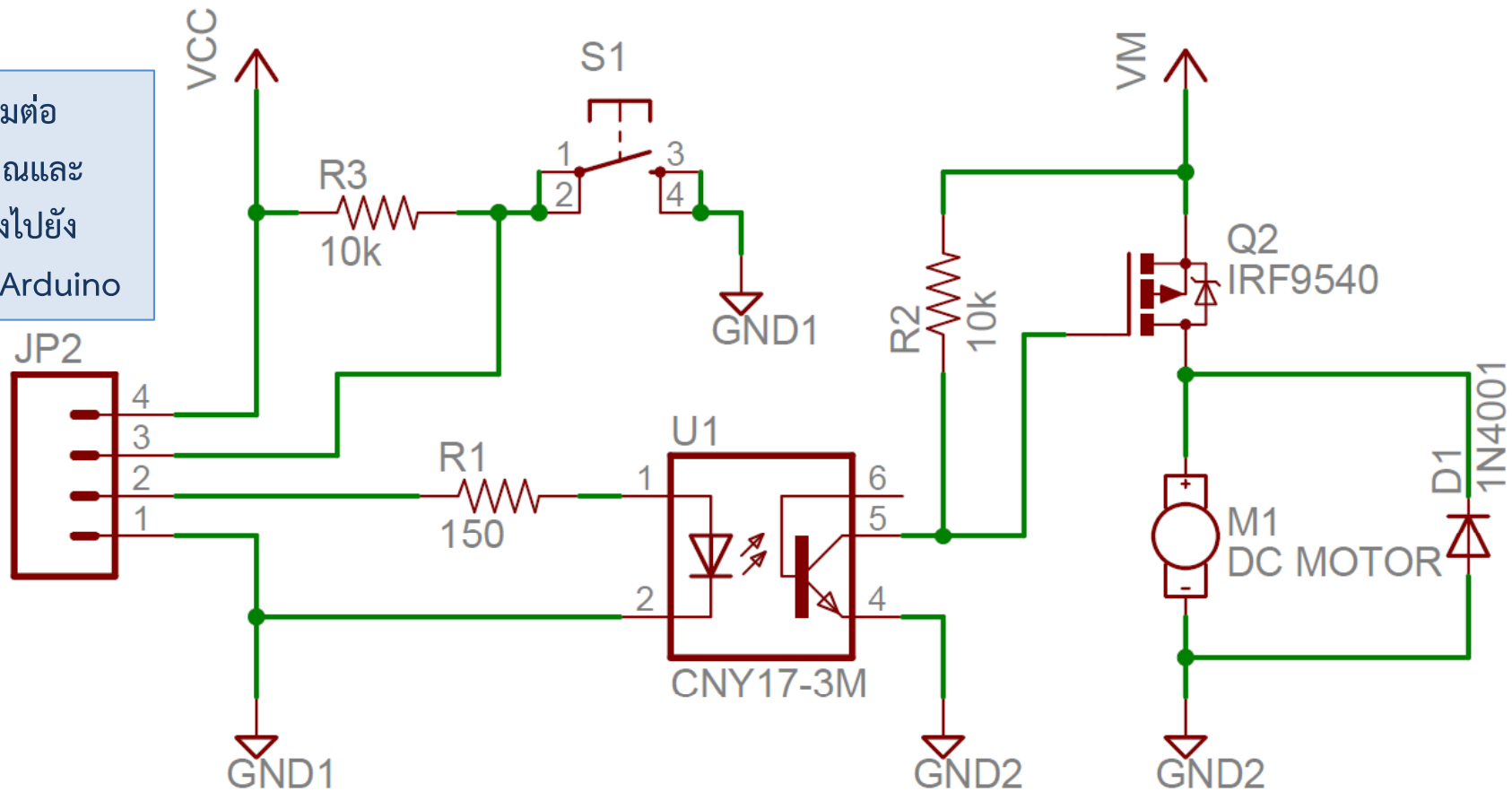
# Lab 24: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Push Button
- 1x CNY17-3 Optoisolator
- 1x 1N4001 Diode
- 1x IRF9540 P-MOSFET
- 1x 150 Ohm
- 2x 10k Ohm
- 1x Small DC Motor
- 1x 9V Battery
- 1x Jump Wire Set



# Lab 24: Schematic – ฟังวงจร

จุดเชื่อมต่อ  
สัญญาณและ  
ไฟเลี้ยงไปยัง  
บอร์ด Arduino



สังเกตว่า มีการแยกกราวด์ในวงจร แบ่งเป็น GND1 และ GND2

# Lab 25: Build Your Optical Proximity Sensor!

---

- **วัตถุประสงค์**

- ต่อบางวงจรโดยใช้อุปกรณ์ที่ทำงานโดยใช้แสง ซึ่งประกอบด้วย โฟโตทรานซิสเตอร์ที่ทำหน้าที่เป็น “ตัวรับแสง” และไดโอดเปล่งแสง (อินฟราเรด) ที่ทำหน้าที่เป็น “ตัวให้แสง”
- ต่อบางวงจรเข้ากับบอร์ด Arduino โดยใช้สัญญาณดิจิทัลเป็นเอาต์พุต (**ขาคิจิตอล หมายเลข 5**) เพื่อเปิด/ปิดตัวให้แสง ใช้สัญญาณดิจิทัลเป็นเอาต์พุต (**ขาคิจิตอล หมายเลข 6**) เพื่อเปิด-ปิด LED และรับสัญญาณแอนะล็อก (**ขาแอนะล็อก หมายเลข 0**) จากตัวรับแสง
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนเบรด์บอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตดิจิทัลเพื่อให้ตัวให้แสงปล่อยแสงอินฟราเรดออกมา ถ้ามีวัตถุเข้าใกล้ จะทำให้อินพุตแอนะล็อกที่ได้จากตัวรับแสงมีค่าเปลี่ยนแปลงไป และนำค่านี้มาเปรียบเทียบและตรวจจับวัตถุในระยะใกล้ได้

# Lab 25: Arduino Code

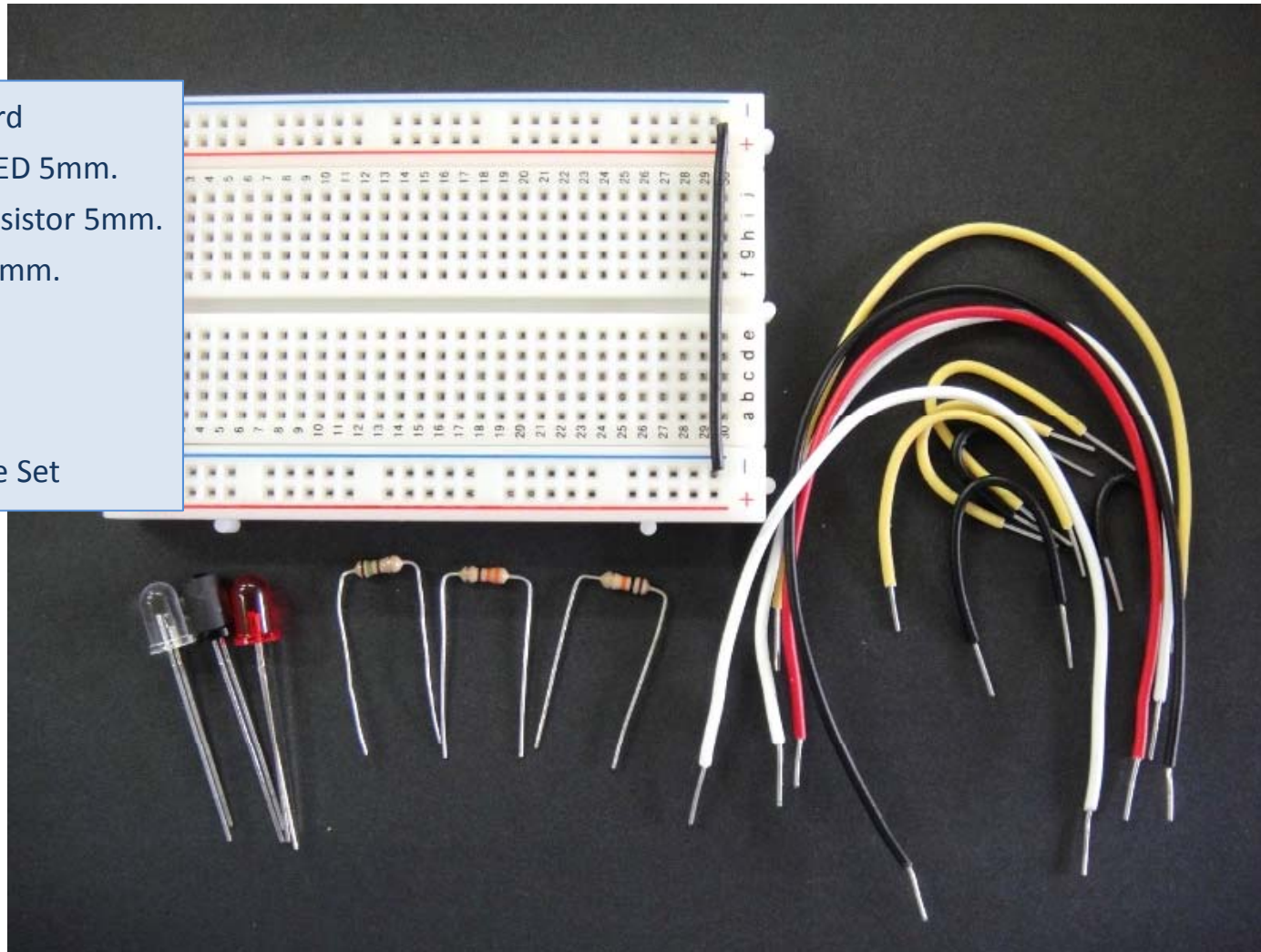
```
/* Example: Build Your Optical Proximity Sensor!
*/
const byte IR_LED_PIN = 5;    // use Digital Pin 5 (D5) to turn on/off IR LED
const byte ALARM_LED_PIN = 6; // use Digital Pin 6 (D6) to turn on/off Alarm LED
const byte AIN_PIN = 0;      // use Analog Pin 0 (A0) as input
const int REF_VALUE = 250;

void setup() {
    pinMode( IR_LED_PIN, OUTPUT );    // use IR_LED Pin as output
    pinMode( ALARM_LED_PIN, OUTPUT ); // use ALARM_LED Pin as output
    analogReference( DEFAULT );       // use +5V voltage reference
    digitalWrite( IR_LED_PIN, HIGH ); // turn on IR LED
}

void loop() {
    boolean b = ( analogRead( AIN_PIN ) < REF_VALUE ); // read analog and compare
    digitalWrite( ALARM_LED_PIN, b ? HIGH : LOW ); // update Alarm LED status
    delay(10); // wait for 10 msec before continuing
}
```

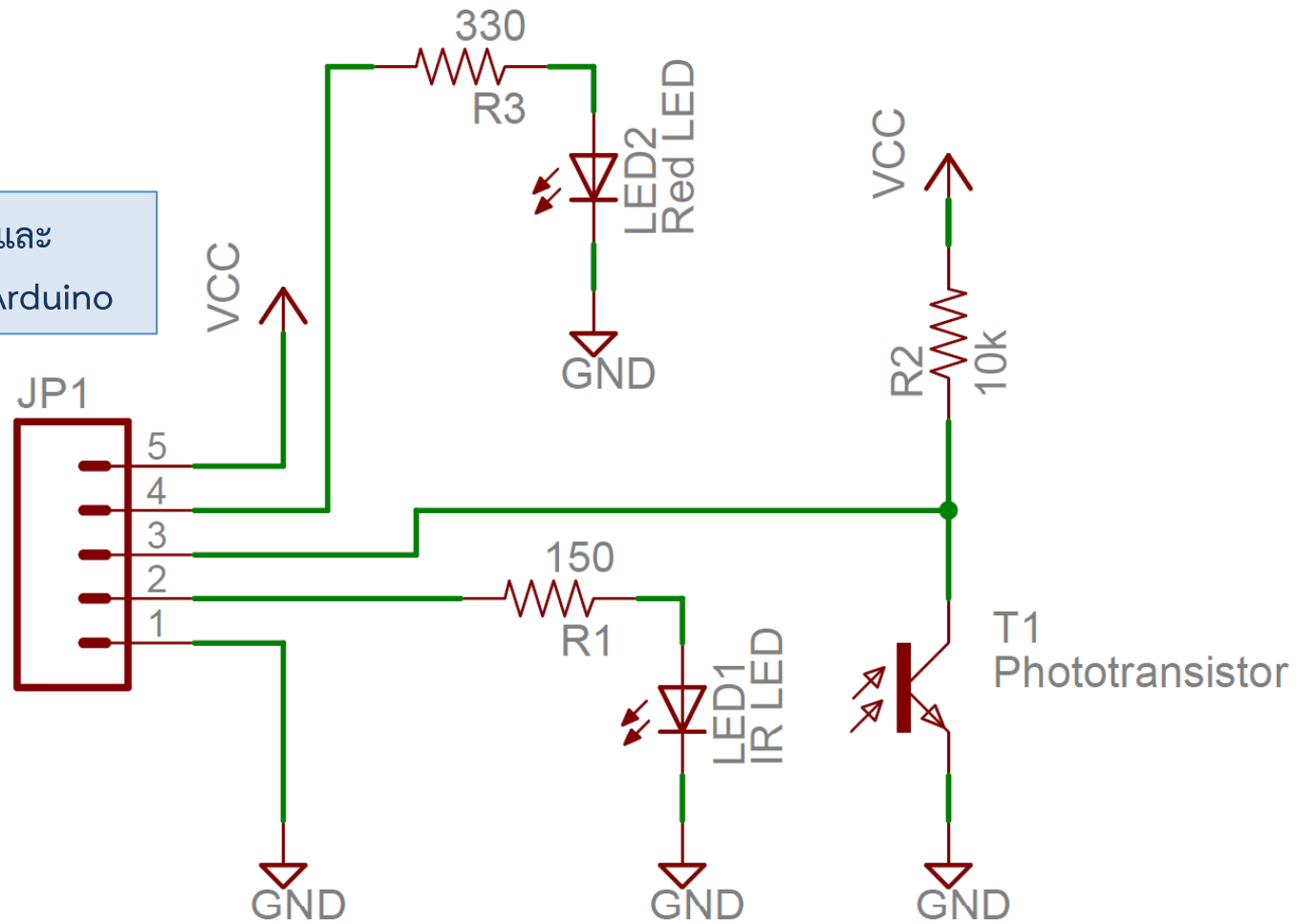
# Lab 25: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Infrared LED 5mm.
- 1x Phototransistor 5mm.
- 1x Red LED 5mm.
- 1x 10k Ohm
- 1x 330 Ohm
- 1x 150 Ohm
- 1x Jump Wire Set



# Lab 25: Schematic – ฟังวงจร

จุดเชื่อมต่อสัญญาณและไฟเลี้ยงไปยังบอร์ด Arduino



# Lab 26: Build Your Optical Switch!

---

- **วัตถุประสงค์**

- ต่อบรรจุโดยใช้สวิตช์แสงที่เรียกว่า **Opto-Interrupter** ซึ่งภายในประกอบด้วยสองส่วนที่สำคัญคือ โฟโตทรานซิสเตอร์ ที่ทำหน้าที่เป็น “ตัวรับแสง” และไดโอดเปล่งแสง (อินฟราเรด) ที่ทำหน้าที่เป็น “ตัวให้แสง”
- ต่อบรรจุกับบอร์ด Arduino โดยมีขาอินพุตดิจิทัล (**ขาดิจิทัล หมายเลข 3**) จากสวิตช์แสง และมีขาเอาต์พุต (**ขาดิจิทัล หมายเลข 5**) สำหรับ LED
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่อ่านค่าจากอินพุตดิจิทัล และกำหนดค่าให้เอาต์พุต โดยจะทำให้ LED “ติด” เมื่อมีวัตถุมายังช่องรับแสง และจะทำให้ LED “ไม่ติด” เมื่อไม่มีวัตถุมายังช่อง

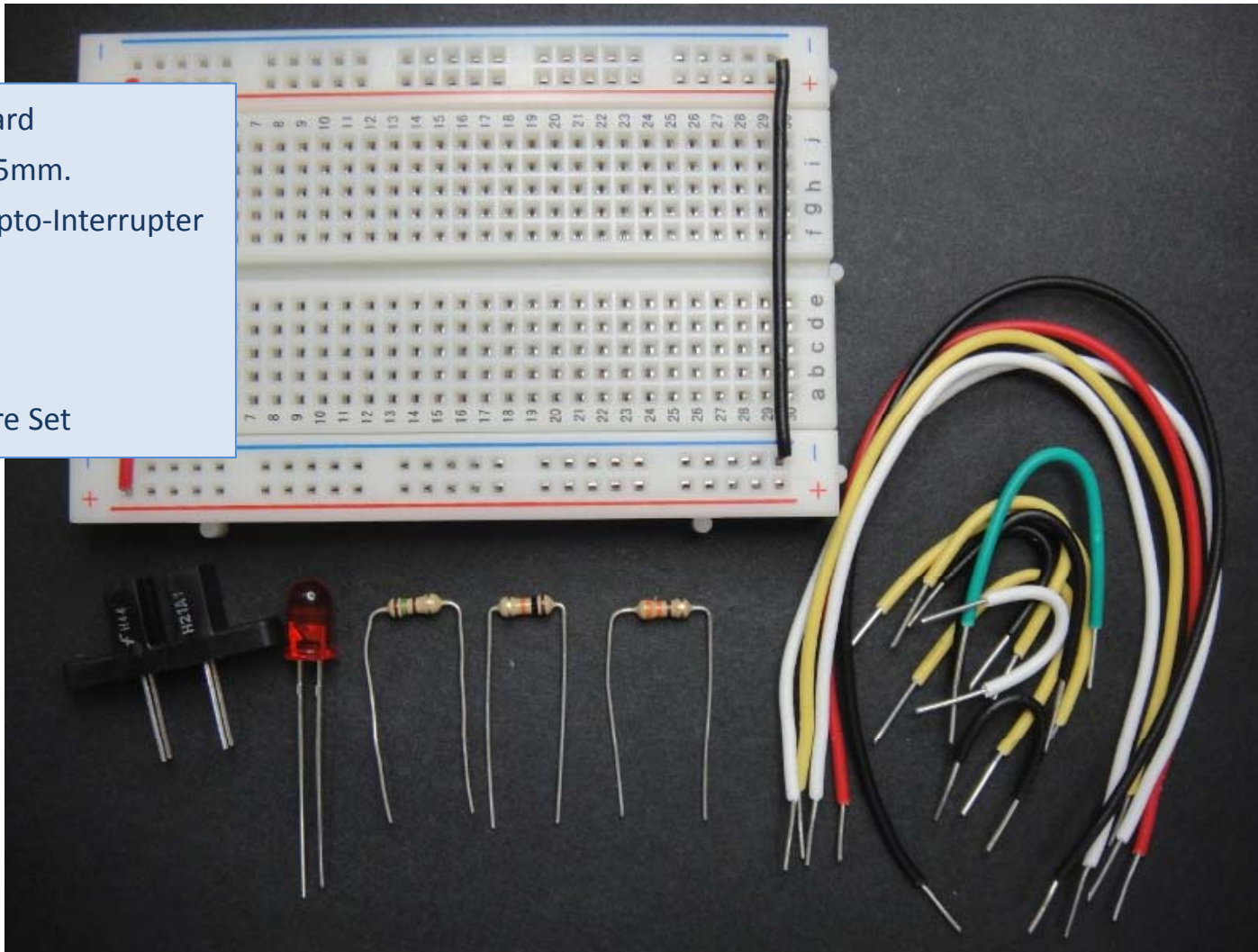
## Lab 26: Arduino Code

---

```
/* Example: Build Your Optical Switch with Opto-Interrupter!  
*/  
const byte SENSOR_PIN = 3; //use Digital Pin 3 (D3) to read input from sensor  
const byte LED_PIN = 5;    //use Digital Pin 5 (D5) to turn on/off LED  
  
void setup() {  
    pinMode( LED_PIN, OUTPUT );    // use LED Pin as output  
    pinMode( SENSOR_PIN, INPUT ); // use SENSOR Pin as input  
}  
  
void loop() {  
    if ( digitalRead( SENSOR_PIN ) == HIGH ) // Light beam is interrupted.  
        digitalWrite( LED_PIN, HIGH );    // output High: turn on LED  
    else  
        digitalWrite( LED_PIN, LOW );    // output Low: turn off LED  
    delay( 10 ); // wait for 10 msec  
}
```

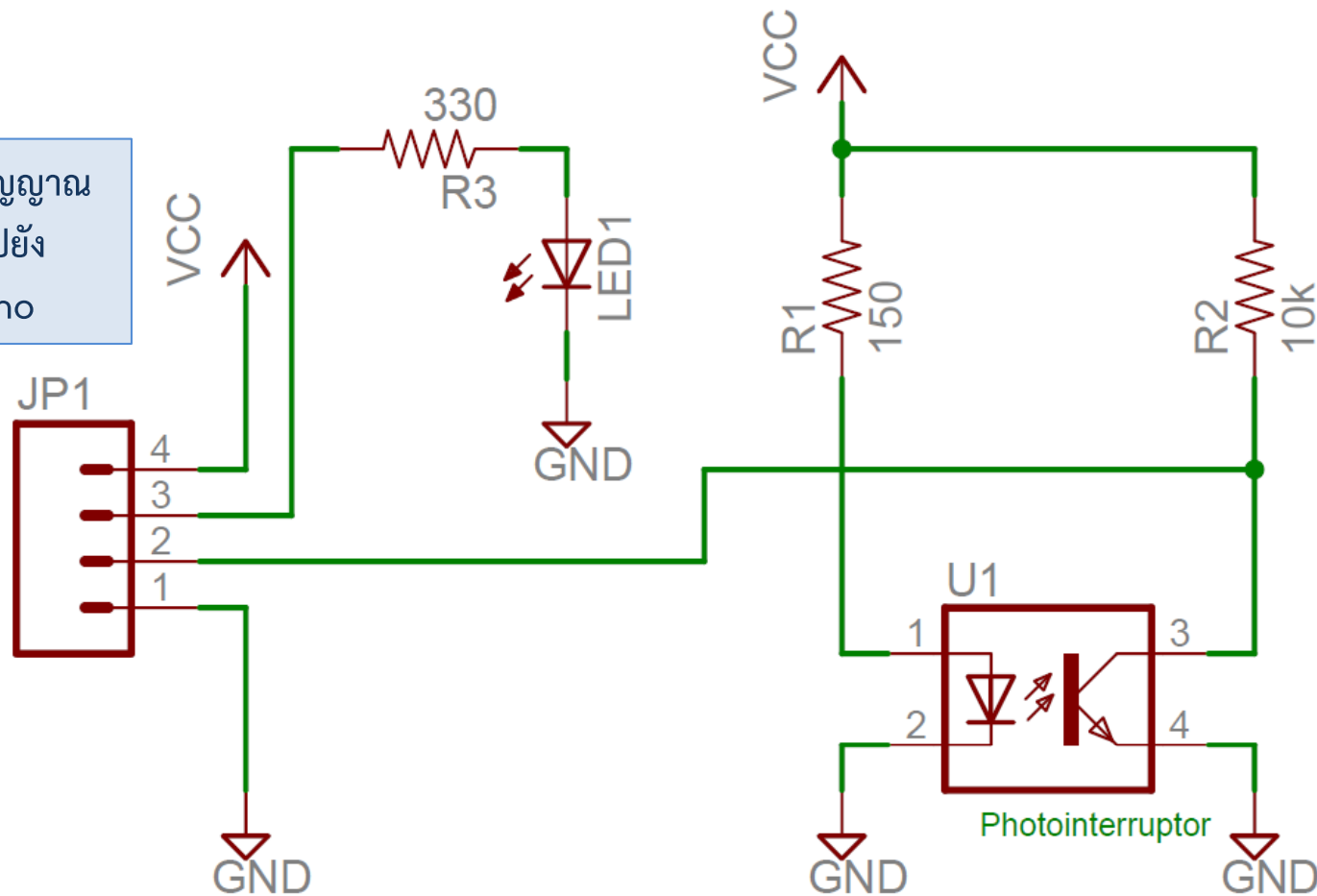
# Lab 26: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm.
- 1x H21A1 Opto-Interrupter
- 1x 10k Ohm
- 1x 330 Ohm
- 1x 150 Ohm
- 1x Jump Wire Set



# Lab 26: Schematic – ฟังวงจร

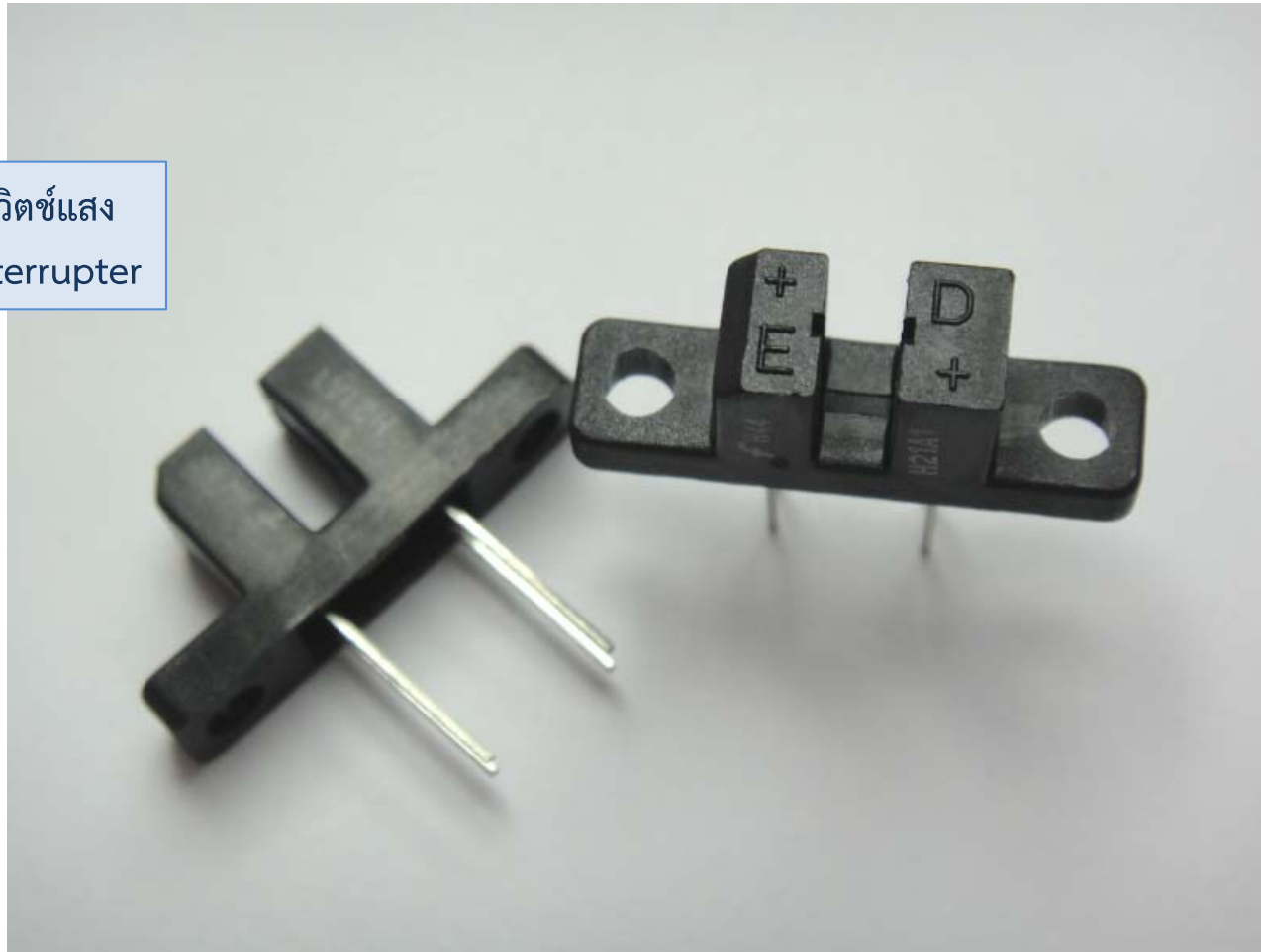
จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



## Lab 26: ตัวอย่างอุปกรณ์ Opto-Interrupter (1/2)

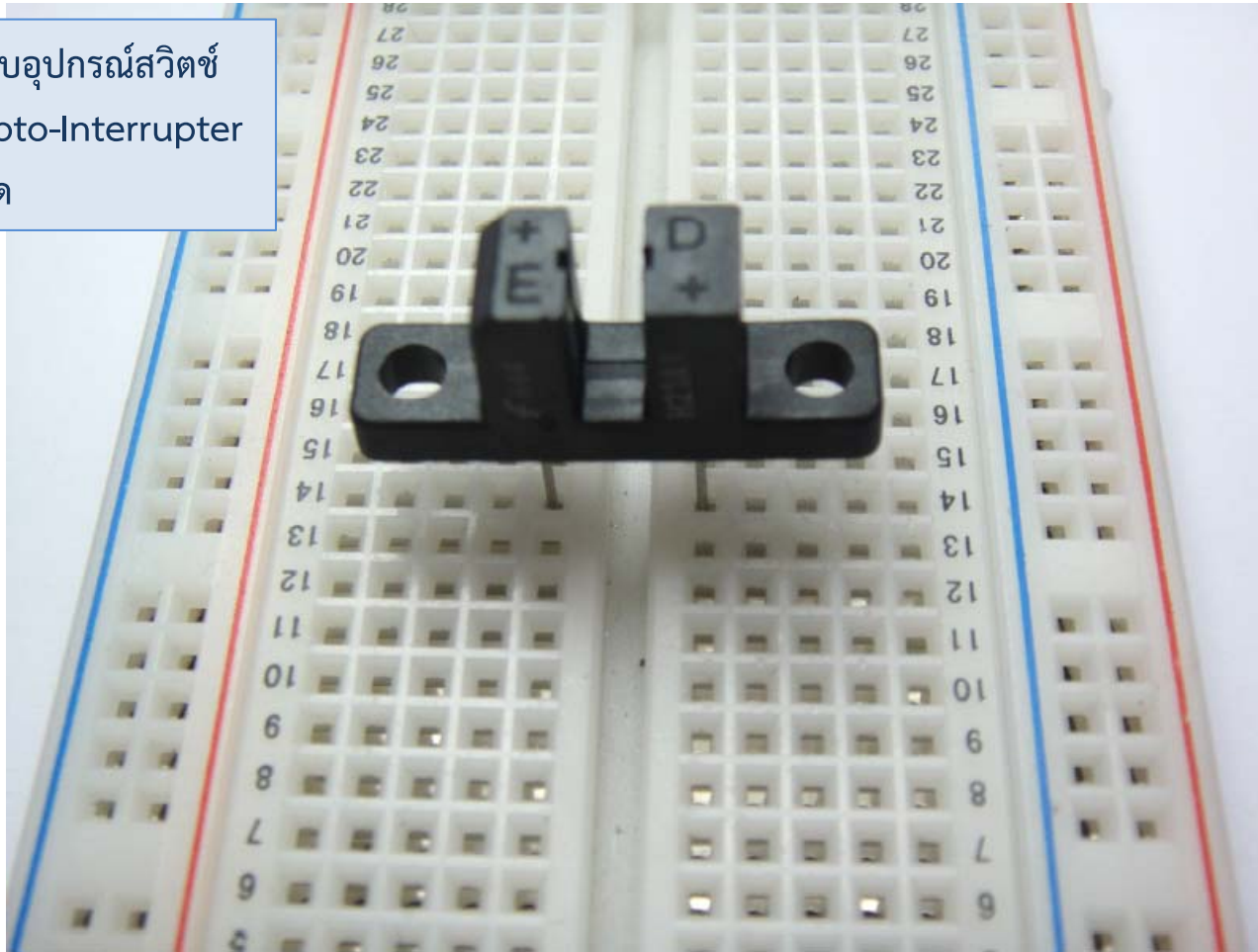
---

ตัวอย่างอุปกรณ์สวิทซ์แสง  
H21A1 Opto-Interrupter



## Lab 26: ตัวอย่างอุปกรณ์ Opto-Interrupter (2/2)

ตัวอย่างการเสียบอุปกรณ์สวิทช์  
แสง H21A1 Opto-Interrupter  
ลงบนเบรตบอร์ด



# Lab 27: Toggle Red/Green LEDs with Relay!

---

- **วัตถุประสงค์**

- ต่อบอร์ดโดยใช้รีเลย์ เพื่อใช้เป็นสวิตช์ไฟฟ้าในการเปิด-ปิดไดโอดเปล่งแสงสีแดงและสีเขียว
- ต่อบอร์ดเข้ากับบอร์ด Arduino โดยมีเอาต์พุตดิจิทัล (ขาดีจิตอล หมายเลข 5) เพื่อควบคุมการทำงานของรีเลย์ โดยผ่านทรานซิสเตอร์แบบ NPN และมีอินพุตดิจิทัลจากปุ่มกด (ขาดีจิตอล หมายเลข 3)
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบนบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่สร้างสัญญาณเอาต์พุตดิจิทัลเพื่อควบคุมการทำงานของรีเลย์

## Lab 27: Arduino Code (Version A)

---

```
/* Example: Toggle Red/Green LEDs with Relay!  
*/  
const byte BUTTON_PIN = 3; // use Digital Pin 3 for Button input  
const byte RELAY_PIN = 5; // use Digital Pin 5 for output to Relay  
  
void setup() {  
    pinMode( BUTTON_PIN, INPUT ); // use BUTTON Pin as input  
    pinMode( RELAY_PIN, OUTPUT ); // use RELAY Pin as output  
}  
  
void loop() {  
    boolean b = ( digitalRead( BUTTON_PIN )==LOW ); // read input from button  
    digitalWrite( RELAY_PIN, b ); // update output state  
}
```

## Lab 27: Arduino Code (Version B)

---

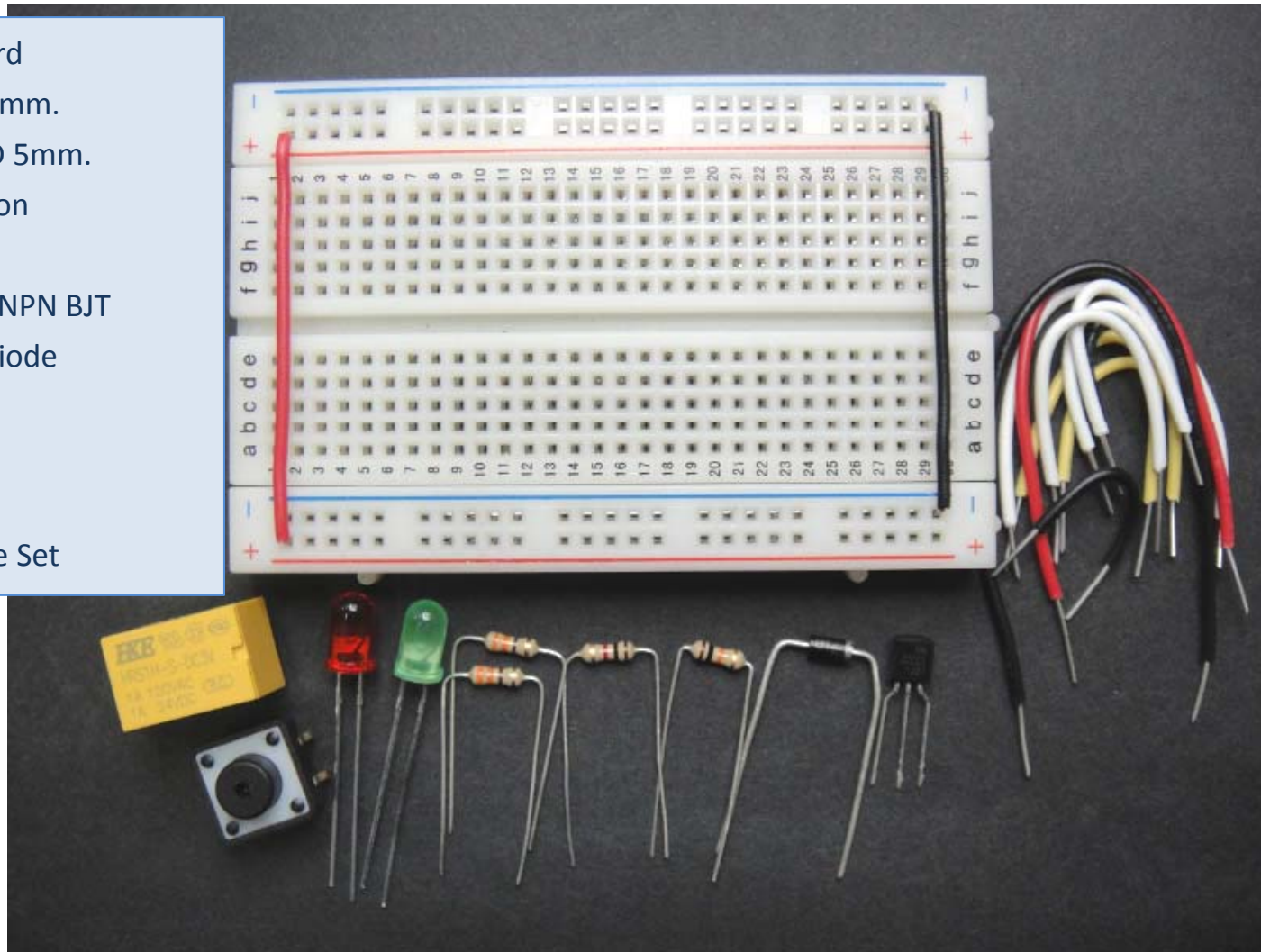
```
/* Example: Toggle Red/Green LEDs with Relay! */
const byte BUTTON_PIN = 3; // use Digital Pin 3 for Button input
const byte RELAY_PIN = 5; // use Digital Pin 5 for output to Relay
boolean state = false;

void setup() {
    pinMode( BUTTON_PIN, INPUT ); // use BUTTON Pin as input
    pinMode( RELAY_PIN, OUTPUT ); // use RELAY Pin as output
}

void loop() {
    while ( digitalRead( BUTTON_PIN ) )
        delay(10); // read input until the button is pressed.
    state ^= true; // toggle the state
    digitalWrite( RELAY_PIN, state ? HIGH:LOW ); // update output
    while ( !digitalRead( BUTTON_PIN ) )
        delay(10); // read input until the button is released.
}
```

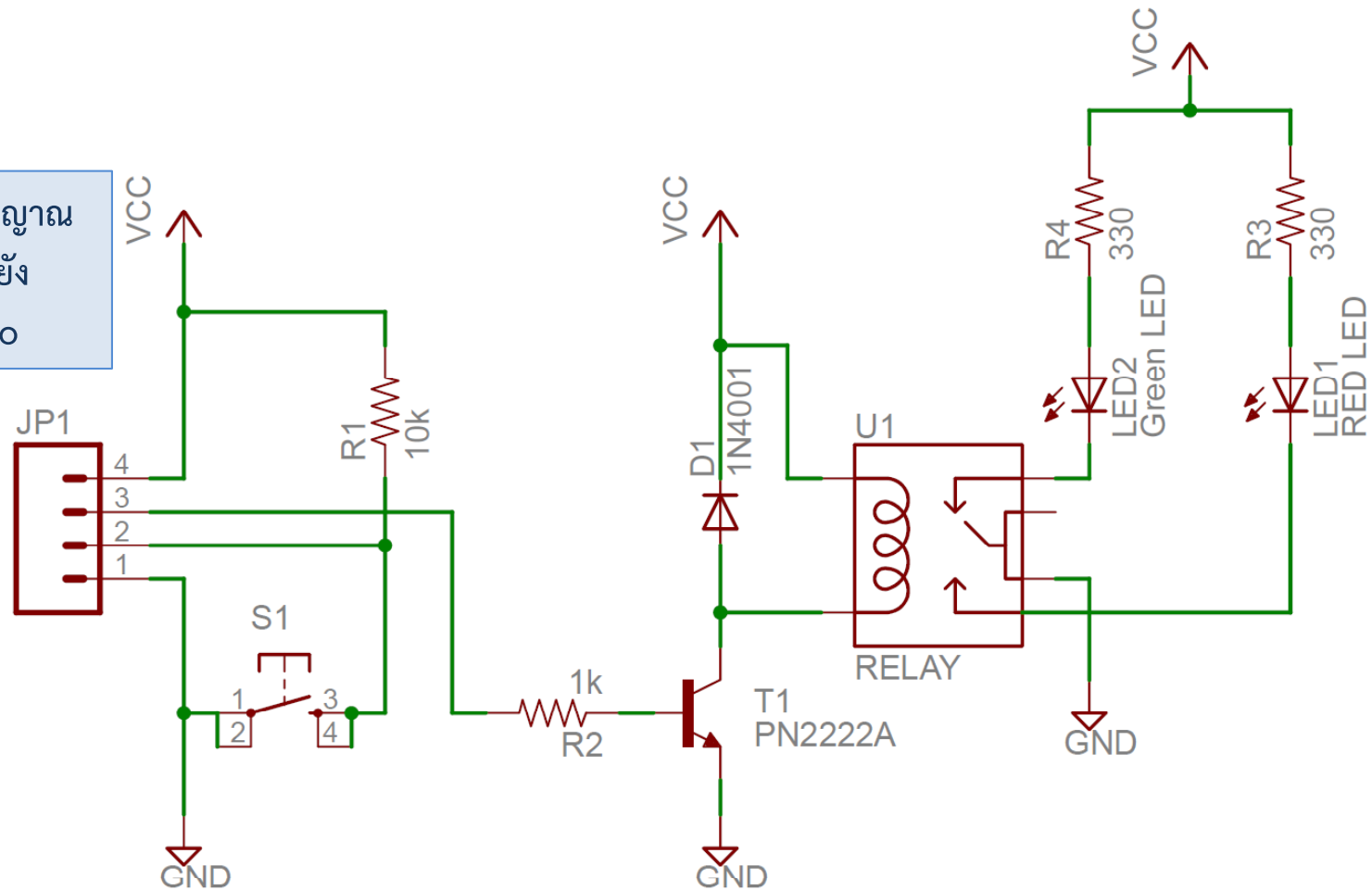
# Lab 27: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm.
- 1x Green LED 5mm.
- 1x Push Button
- 1x 3V Relay
- 1x PN2222A NPN BJT
- 1x 1N4001 Diode
- 1x 10k Ohm
- 1x 1k Ohm
- 2x 330 Ohm
- 1x Jump Wire Set



# Lab 27: Schematic – ฟังวงจร

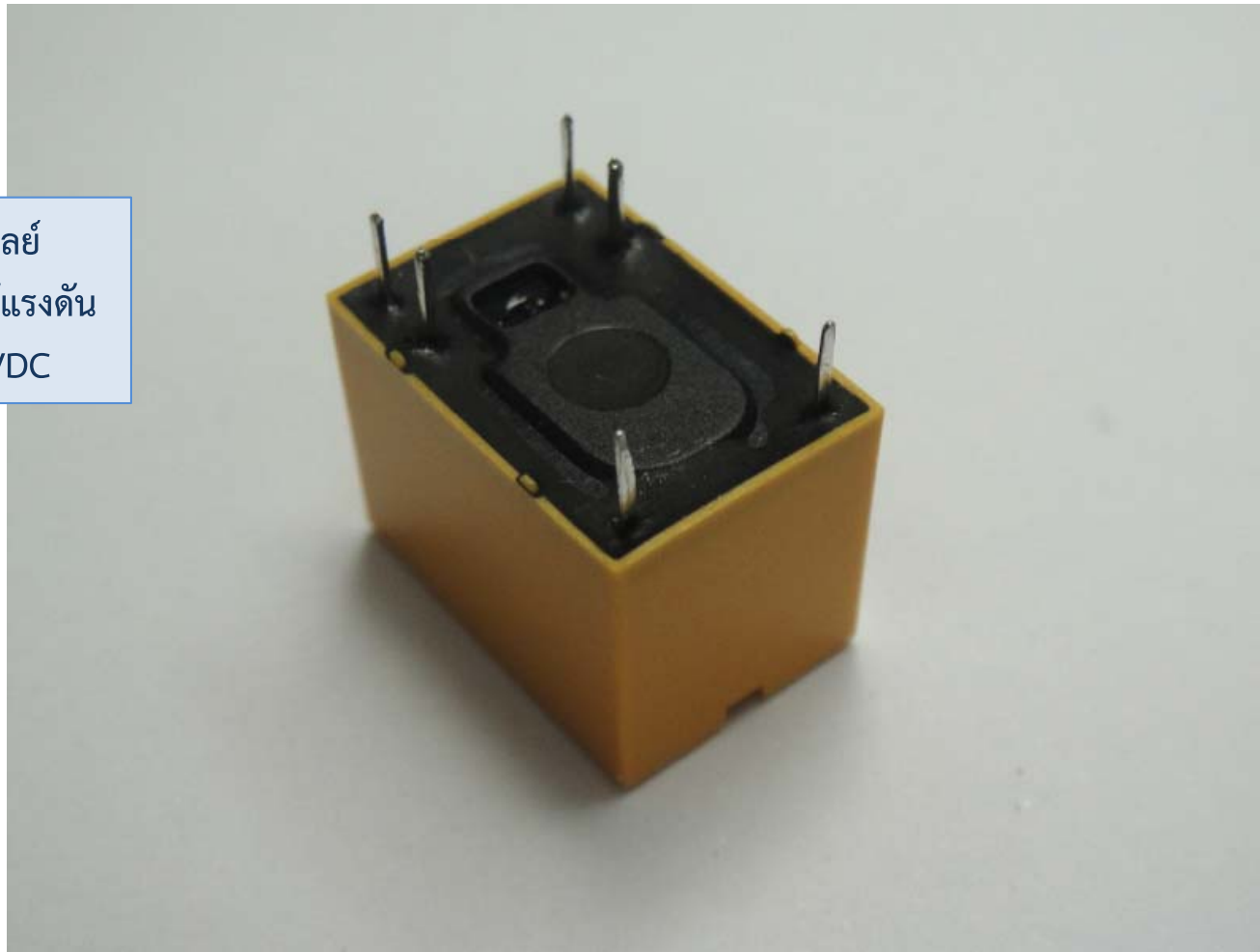
จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



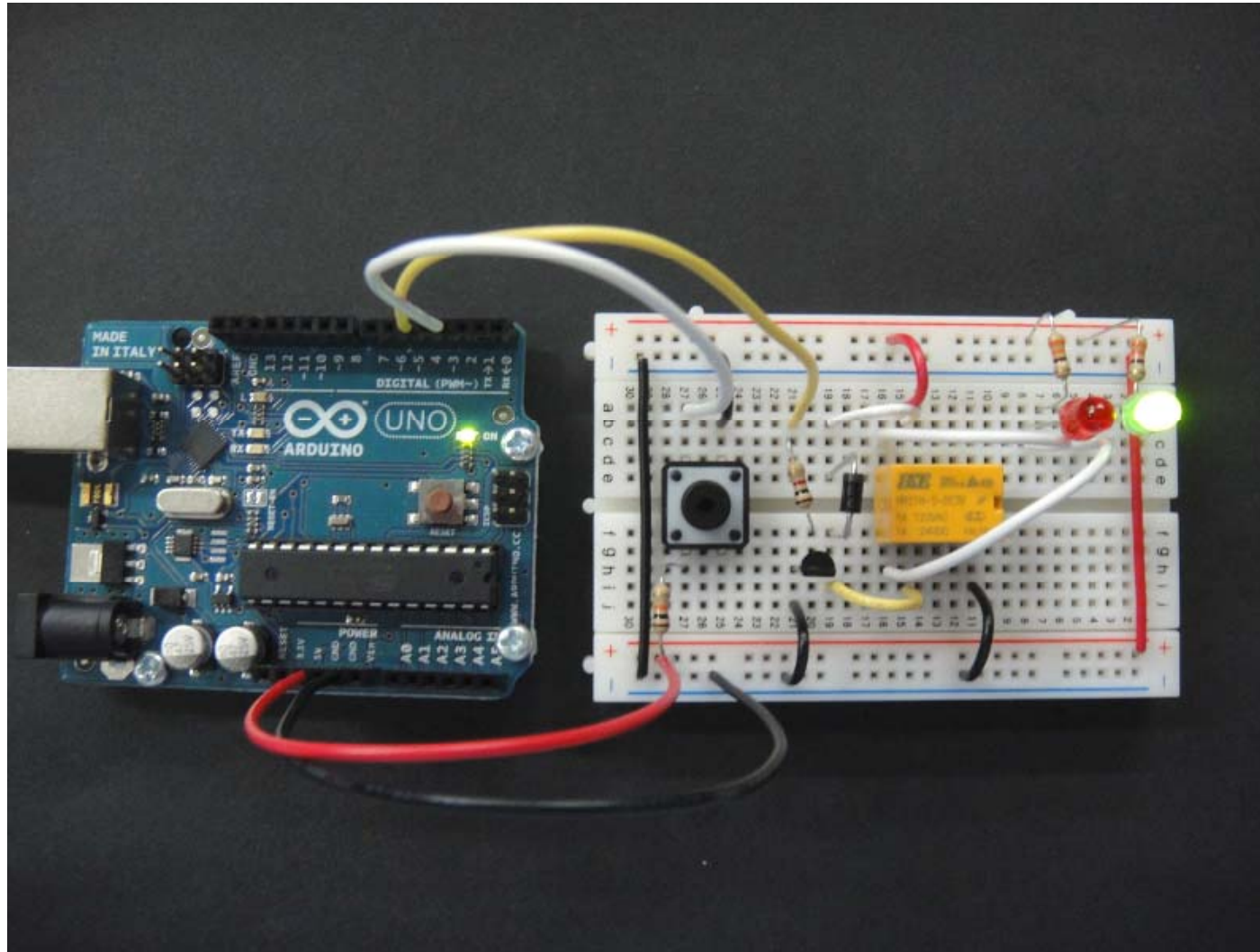
## Lab 27: ตัวอย่างอุปกรณ์ Relay – รีเลย์

---

ตัวอย่างอุปกรณ์รีเลย์  
แบบ 6 ขา และใช้แรงดัน  
สำหรับคอยล์ที่ 3VDC



## Lab 27: Breadboard Circuit - วงจรบนเบรตบอร์ด



# Lab 28: Make Countdown Timer + Buzzer Alarm!

---

- **วัตถุประสงค์**

- ต่อบังคับโดยใช้บัสเซอร์เสียงเพื่อสร้างสัญญาณเสียงเตือน และต่อบังคับปุ่มกดเพื่อให้หยุดการทำงานของบัสเซอร์เมื่อกดปุ่ม
- ต่อบังคับเข้ากับบอร์ด Arduino โดยมีเอาต์พุตดิจิทัล (ขาดีจิตอล หมายเลข 5) เพื่อควบคุมการทำงานของบัสเซอร์ และไดโอดเปล่งแสง (ขาดีจิตอล หมายเลข 6) และมีอินพุตดิจิทัลจากปุ่มกด (ขาดีจิตอล หมายเลข 7)
- ศึกษาการทำงานของโค้ดตัวอย่างสำหรับ Arduino ร่วมกับวงจบบอร์ด

- **คำอธิบายโค้ดสำหรับ Arduino**

- โค้ดทำหน้าที่นับถอยหลังประมาณ 5 วินาที ในช่วงของการนับถอยหลังถ้าไม่มีการกดปุ่ม จะสร้างสัญญาณเอาต์พุตดิจิทัลที่ทำให้บัสเซอร์มีเสียงเตือน แต่ถ้ามีการกดปุ่ม จะเริ่มนับถอยหลังใหม่อีก 5 วินาที

## Lab 28: Arduino Code (1/2)

---

```
/* Example: Make Countdown Timer with Buzzer Alarm!  
*/  
const byte BUZZER_PIN = 5; // use Digital Pin 5 (D5) for Buzzer output pin  
const byte LED_PIN    = 6; // use Digital Pin 6 (D6) for LED output pin  
const byte BUTTON_PIN = 7; // use Digital Pin 7 (D7) for Button input pin  
byte count;  
  
void setup() {  
    pinMode( BUZZER_PIN, OUTPUT ); // use BUZZER pin as output  
    pinMode( LED_PIN, OUTPUT );   // use LED pin as output  
    pinMode( BUTTON_PIN, INPUT ); // use BUTTON pin as input  
}  
  
// Code continues on the next page.
```

## Lab 28: Arduino Code (2/2)

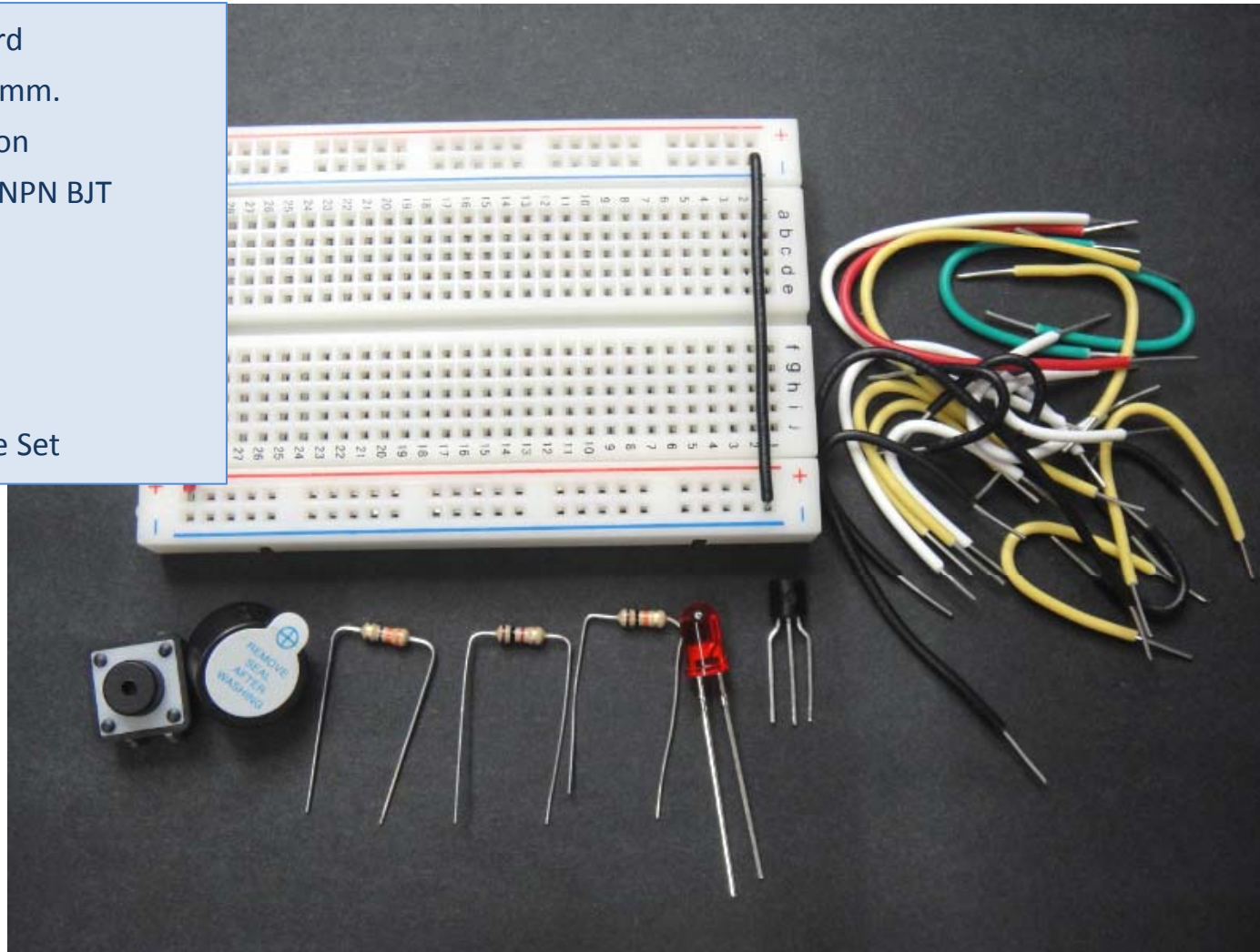
---

*// Code continues from the previous page.*

```
void loop() {
  count = 50;           // reset count to its initial value
  digitalWrite( BUZZER_PIN, LOW ); // turn off Buzzer
  digitalWrite( LED_PIN, LOW );   // turn off LED
  while (count-- > 0) { // count down if counter is greater than zero
    if ( digitalRead( BUTTON_PIN ) == LOW ) // reset counter when pressed
      count = 50;
    delay(100); // delay for 100 msec
  }
  digitalWrite( BUZZER_PIN, HIGH ); // turn on Buzzer
  digitalWrite( LED_PIN, HIGH );   // turn on LED
  while ( digitalRead( BUTTON_PIN ) == HIGH )
    delay(10); // wait until button is released.
}
```

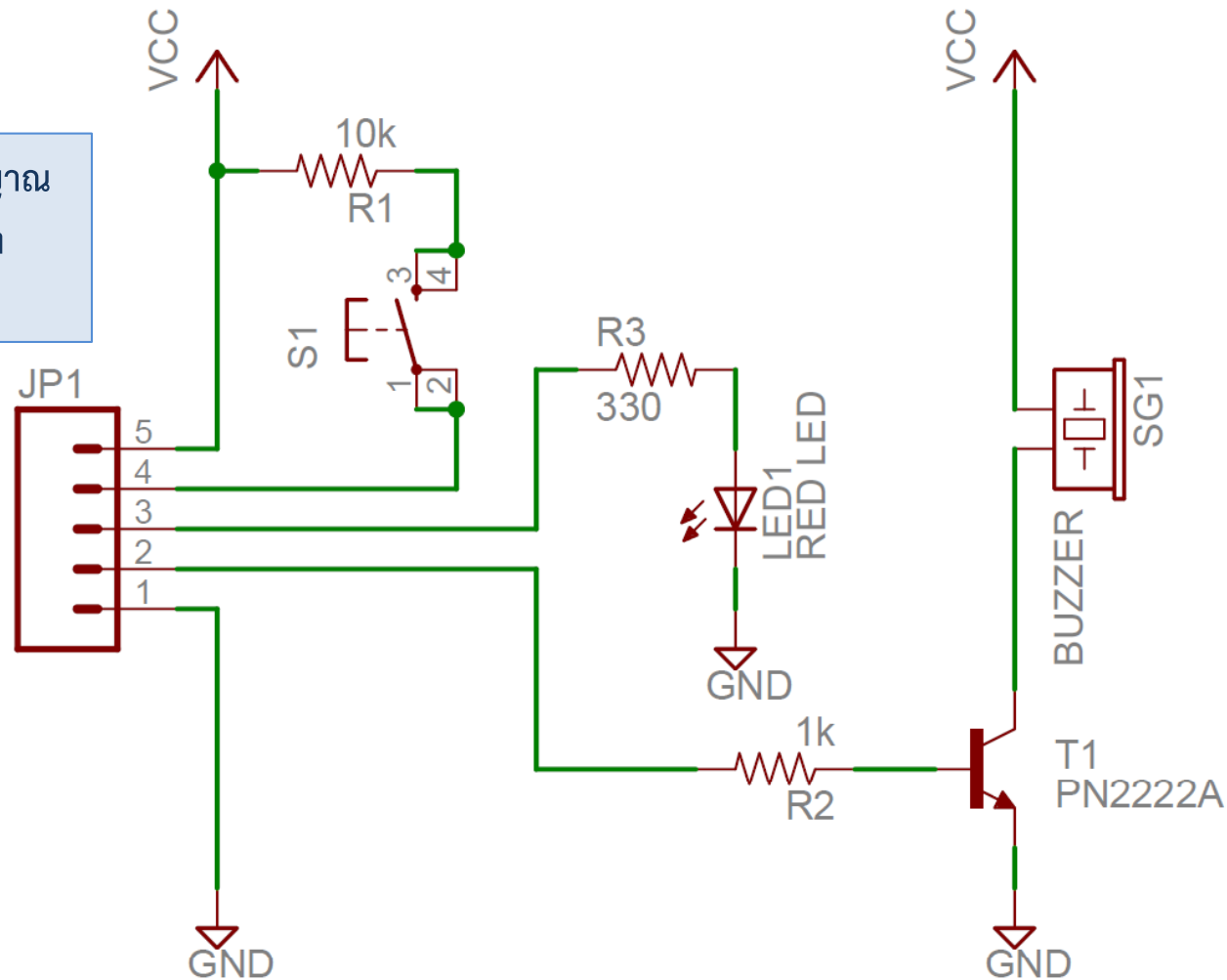
# Lab 28: Parts List – รายการอุปกรณ์

- 1x Breadboard
- 1x Red LED 5mm.
- 1x Push Button
- 1x PN2222A NPN BJT
- 1x Buzzer
- 1x 10k Ohm
- 1x 1k Ohm
- 1x 330 Ohm
- 1x Jump Wire Set



# Lab 28: Schematic – ฟังวงจร

จุดเชื่อมต่อสัญญาณ  
และไฟเลี้ยงไปยัง  
บอร์ด Arduino



# Lab 28: Breadboard Circuit - วงจรบนเบรตบอร์ด

