



อ้างอิง arduino.cc
เรียนรู้เข้าใจ ใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino ,
สำนักพิมพ์ ETT จำกัด, เอกชัย มะกรุ

AGENDA

- Introduction to Arduino
- Arduino Micro-controller Architecture
- LED
- Workshop LED
- Temp & Humidity
- Workshop DHT22
- Q & A

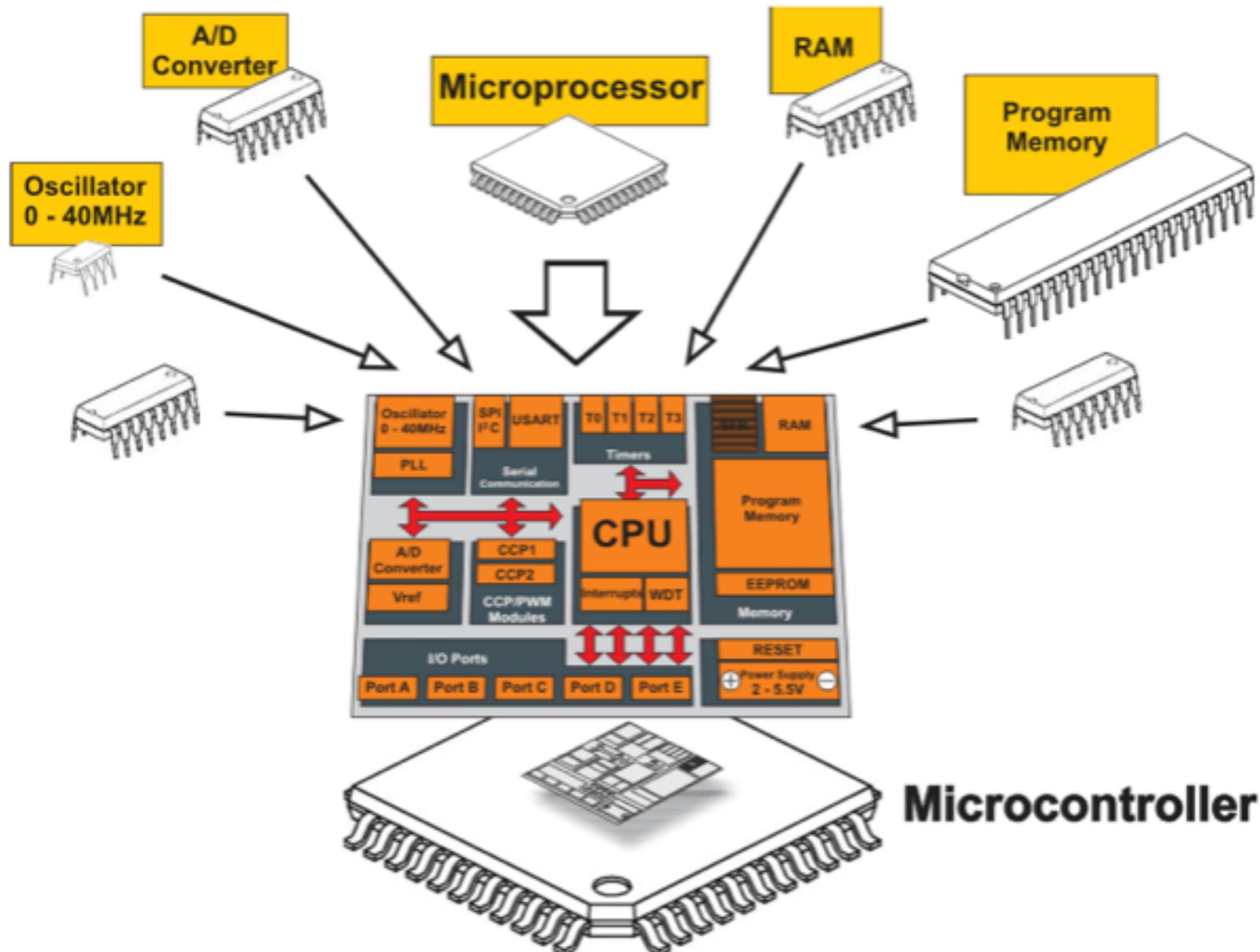
INTRODUCTION TO ARDUINO

INTRODUCTION TO ARDUINO

ไมโครคอนโทรลเลอร์ (Microcontroller, μC, uC หรือ MCU)

- คืออุปกรณ์ควบคุมขนาดเล็กถูกพัฒนามาเพื่อตอบสนองความต้องการทางการตลาด ในการออกแบบระบบไมโครprocessor ให้มีต้นทุนต่ำลง
- ไมโครคอนโทรลเลอร์ ได้รวมเอา CPU, ROM, RAM, Analog I/O, Digital I/O, วงจรกำเนิดสัญญาณนาฬิกา และพอร์ตการสื่อสาร ทำการบรรจุเข้าไว้ในตัวถังเดียว กันหรือ Single chip
- ซึ่งทำให้สามารถใช้งานได้โดยไม่ต้องมีส่วนประกอบเพื่อใช้งานมาก เหมือนไมโครprocessor

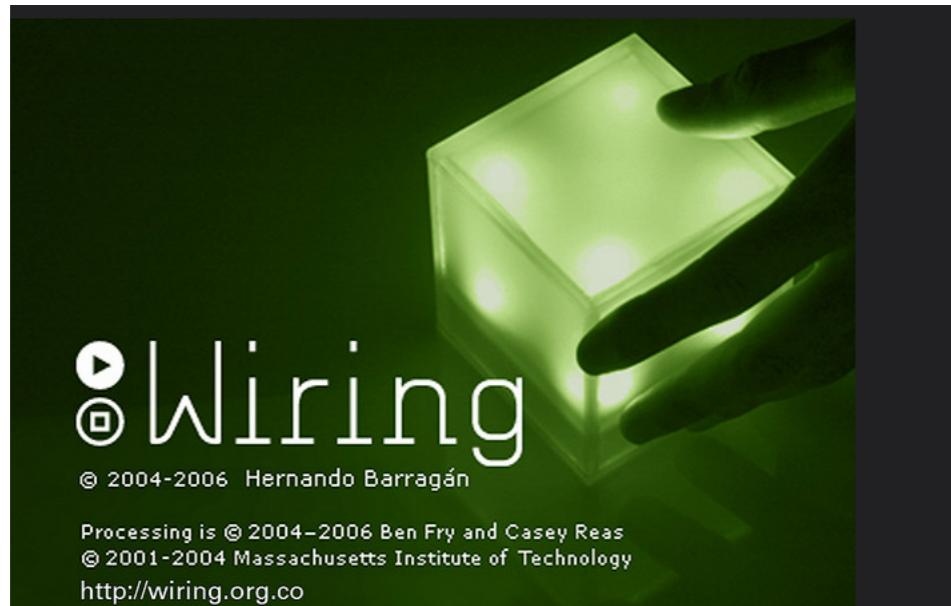
INTRODUCTION TO ARDUINO



ภาพไมโครคอนโทรลเลอร์

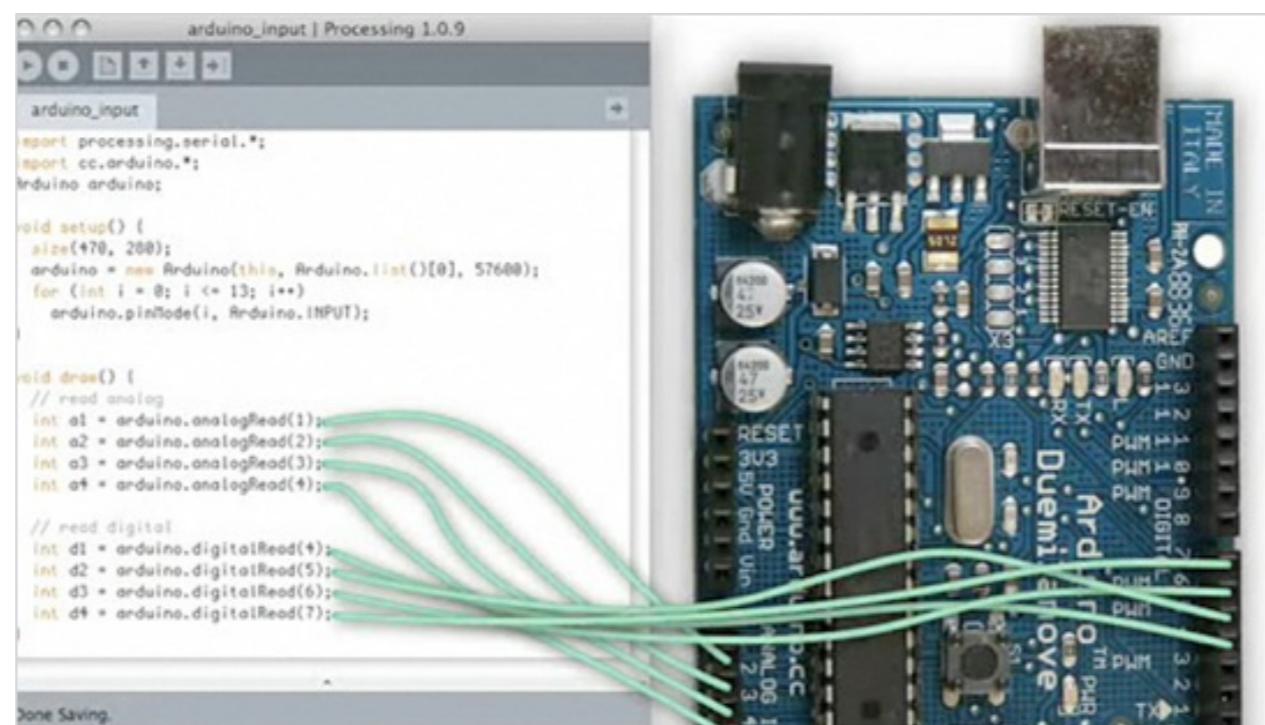
INTRODUCTION TO ARDUINO

- Arduino มีที่มาจากการที่อาจารย์สอนด้านเทคโนโลยีประดิษฐ์ได้รวมกลุ่มกันกับนักเรียนและวิศวกรประดิษฐ์ขึ้นเพื่อให้นักเรียนได้ใช้ไมโครคอนโทรลเลอร์ในการทำโครงการ
- จากเดิมใช้ไมโครคอนโทรลเลอร์ Basic Stamp ซึ่งมีราคาแพง และต้นแบบของ Arduino คือ Wiring ที่ออกแบบใหม่ให้ต้นทุนต่ำลงเหมาะสมสำหรับนักเรียน ซึ่งได้นำไปเผยแพร่แพร่แวงจรรวมทั้งโปรแกรมแก่สาธารณะทางออนไลน์ที่ arduino.cc



INTRODUCTION TO ARDUINO

- จากนั้นก็แพร่หลายมาสู่นักประดิษฐ์ ศิลปิน และนักพัฒนาทั่วโลก
- สิ่งที่ทำให้ Arduino ประสบความสำเร็จเป็นอย่างมากคือบอร์ด Arduino นั้นถูกออกแบบภายใต้ลิขสูตรแบบ Creative Commons License ทำให้เกิดการต่อยอดและสร้างบอร์ดทางเลือกออกมาได้อย่างอิสระ



INTRODUCTION TO ARDUINO



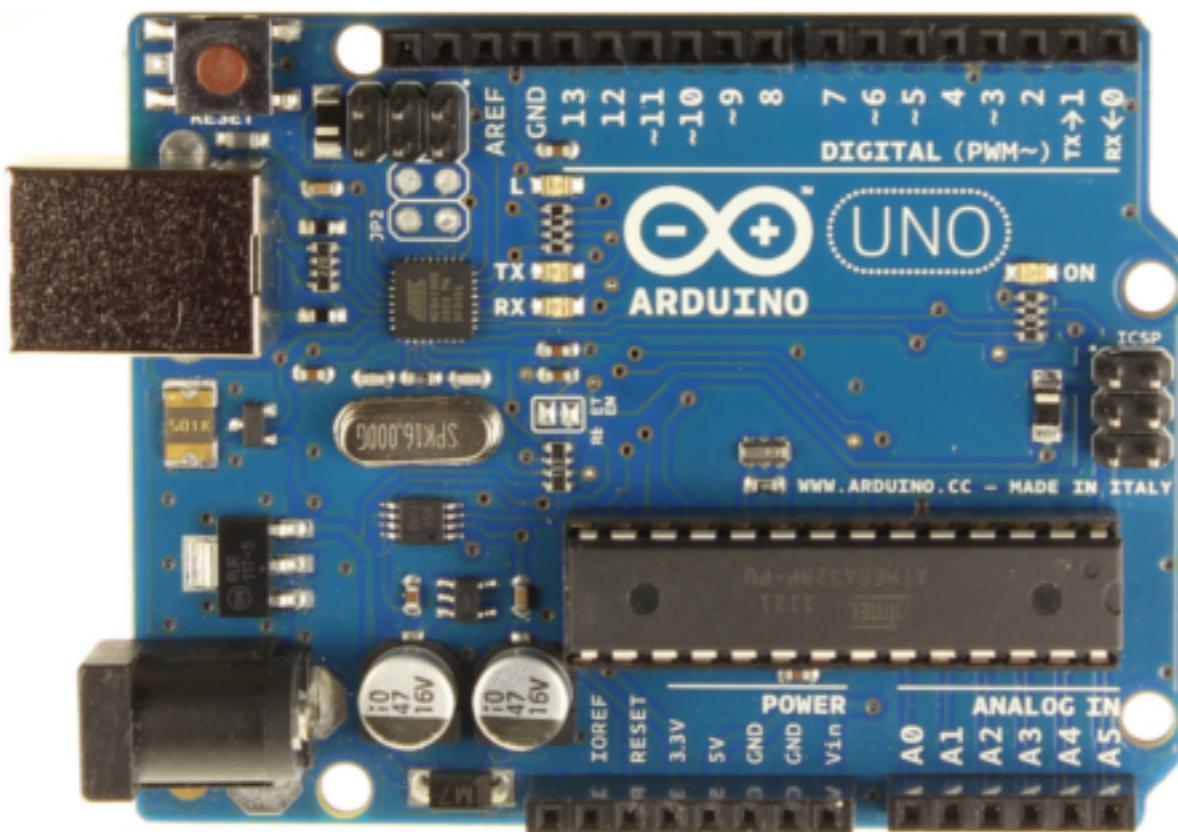
The Arduino core team จากซ้าย David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis และ Massimo Banzi จากงาน Maker Faire จัดที่นิวยอร์ค ถ่ายโดย Randi Silberman Klett

ARDUINO MICRO-CONTROLLER ARCHITECTURE

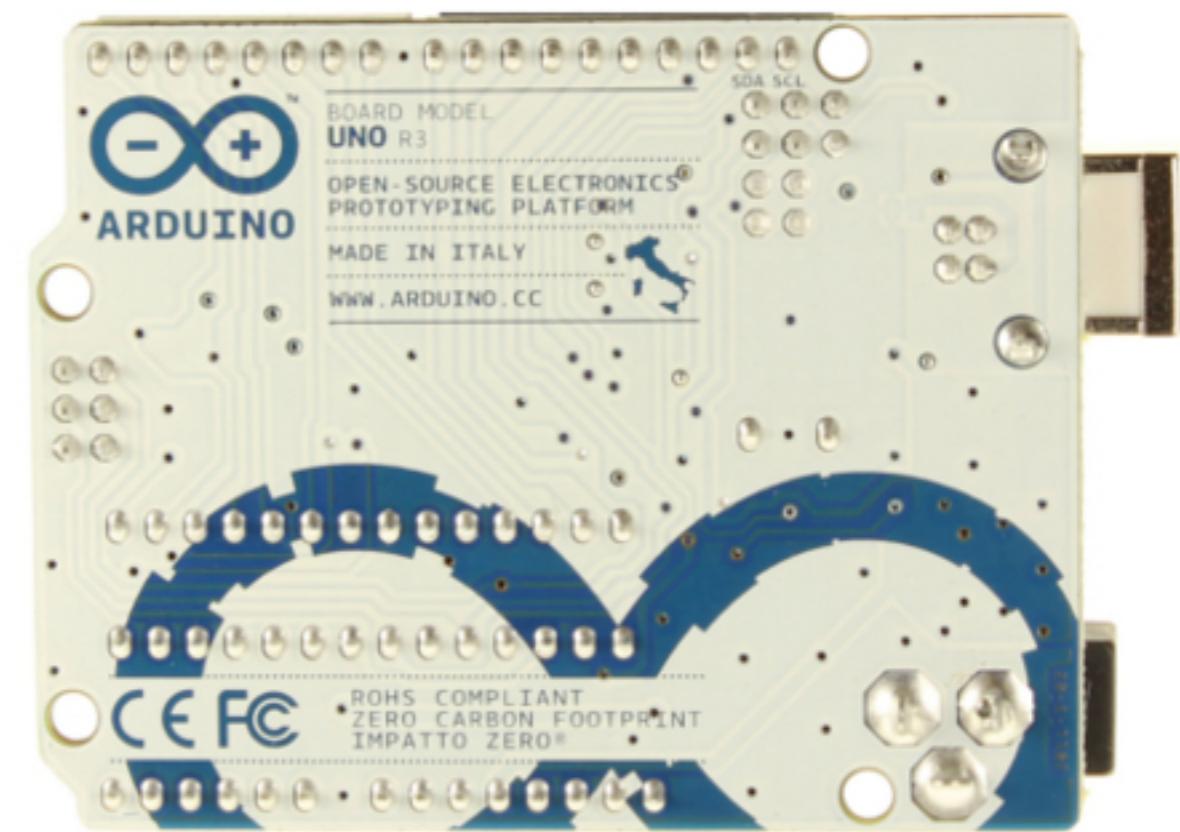
ARDUINO MICRO-CONTROLLER ARCHITECTURE

ในการใช้งานขอให้ผู้ศึกษาเริ่มต้นด้วย Arduino Uno รุ่น 3 ซึ่งเป็นแบบมาตรฐาน

Arduino Uno



Arduino Uno R3 Front



Arduino Uno R3 Back

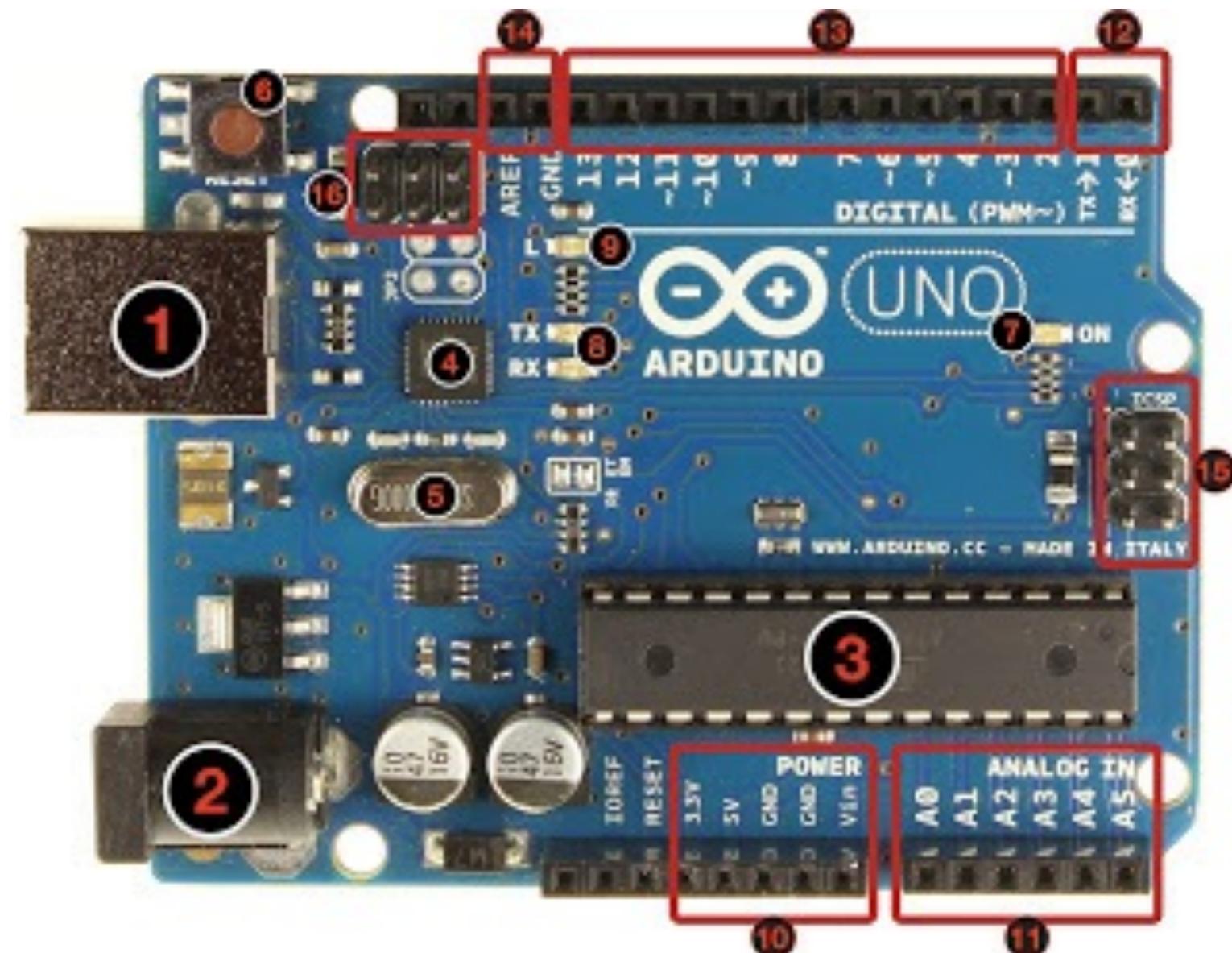
ARDUINO MICRO-CONTROLLER ARCHITECTURE

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

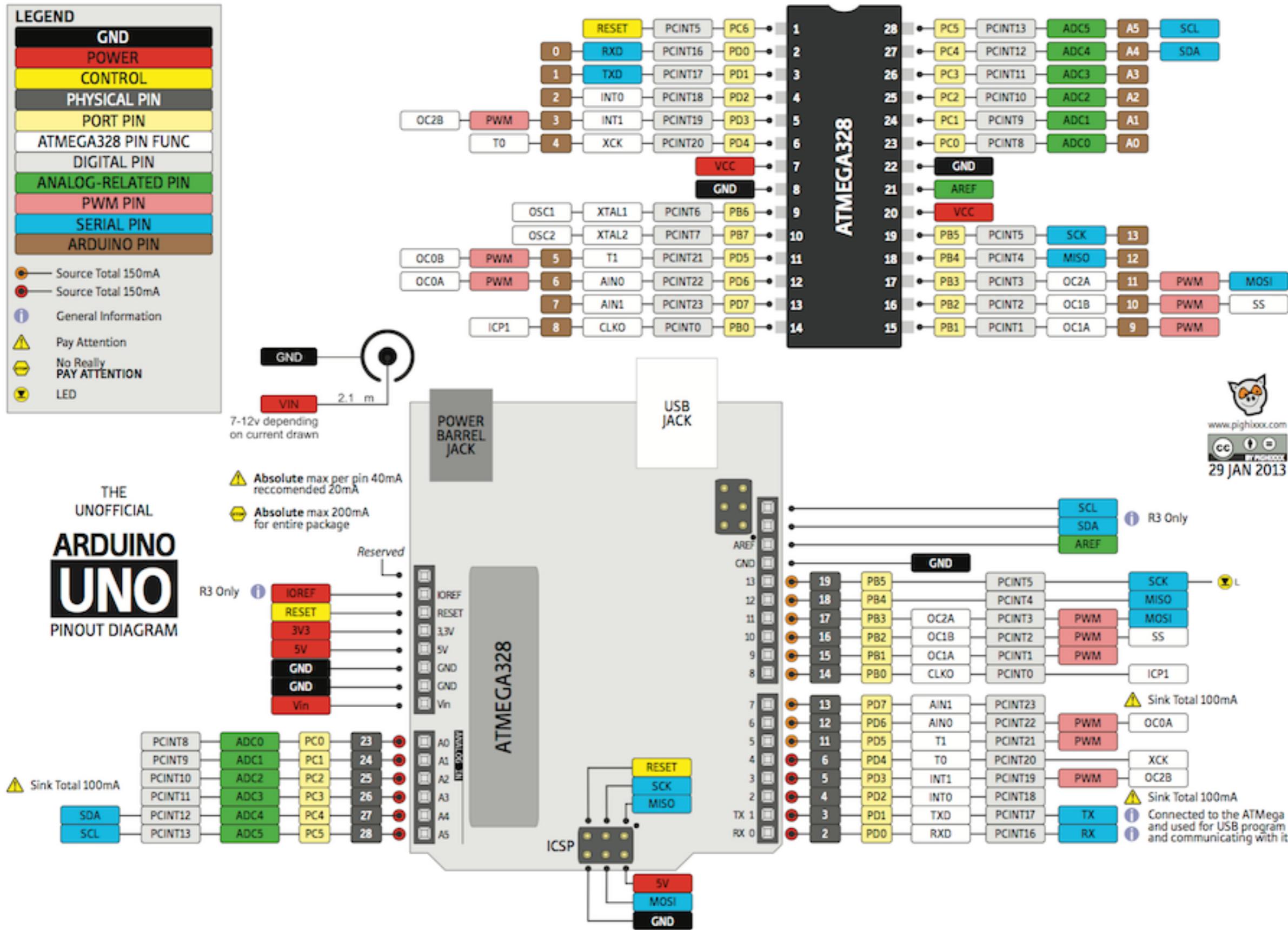
ARDUINO MICRO-CONTROLLER ARCHITECTURE

1. USB jack
2. power jack
3. processor
4. communication chip
5. 16 MHz crystal
6. reset button
7. Power On led
8. TX/NX LED
9. LED pin 13
10. Power supply pins
11. Analog Inputs pins
12. TX and RX pins (shouldn't use)
13. Digital Inputs/outputs. The " ~ " in front of the numbers are for PWM outputs. *
14. Ground and AREF pins.
15. ICSP for Atmega328 & USB interface

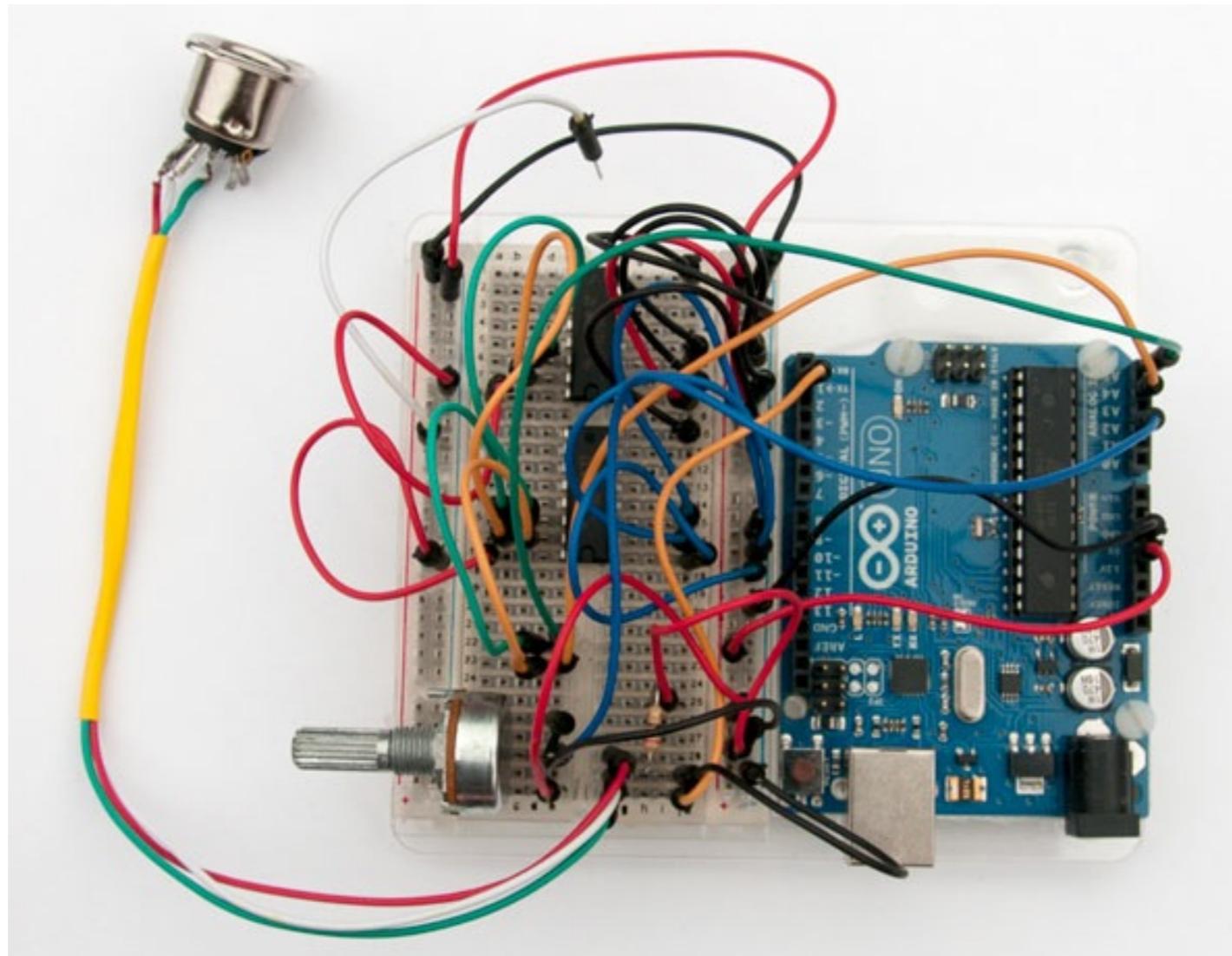


*PWM pin = 3, 5, 6, 9, 10, และ 11

ARDUINO MICRO-CONTROLLER ARCHITECTURE



ARDUINO MICRO-CONTROLLER ARCHITECTURE



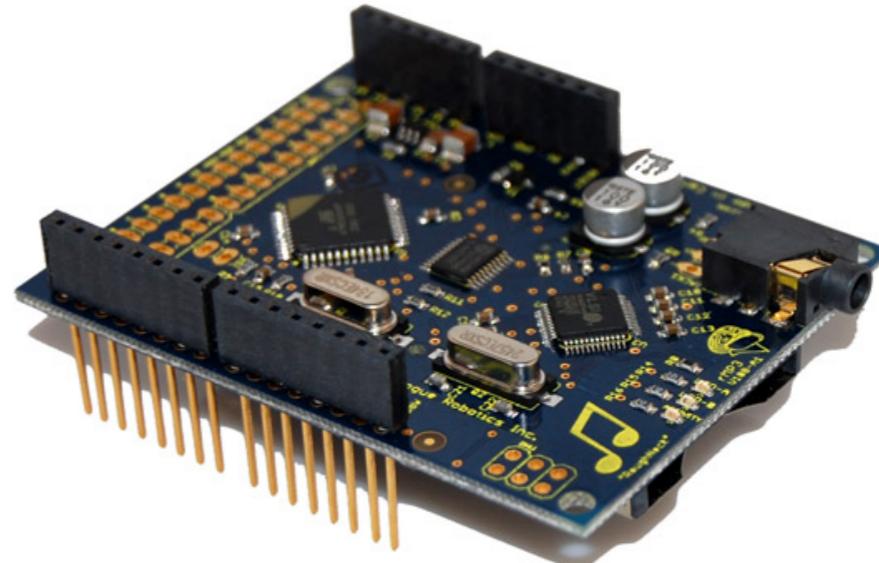
ทุกวันนี้นอกจากบอร์ด Arduino แล้วเรายังได้เห็นการต่อยอดของชาร์ดแวร์ออกมามากในรูปแบบของ Shield (ชีลด์) ซึ่งเป็นบอร์ดเสริมช่วยเพิ่มความสามารถให้กับ Arduino และช่วยลดการต้องบัดกรีอุปกรณ์ได้

ทำให้การพัฒนาหรือประดิษฐ์อุปกรณ์ด้วยไมโครคอนโทรลเลอร์ง่ายขึ้นไปอีกขั้น

ARDUINO MICRO-CONTROLLER ARCHITECTURE



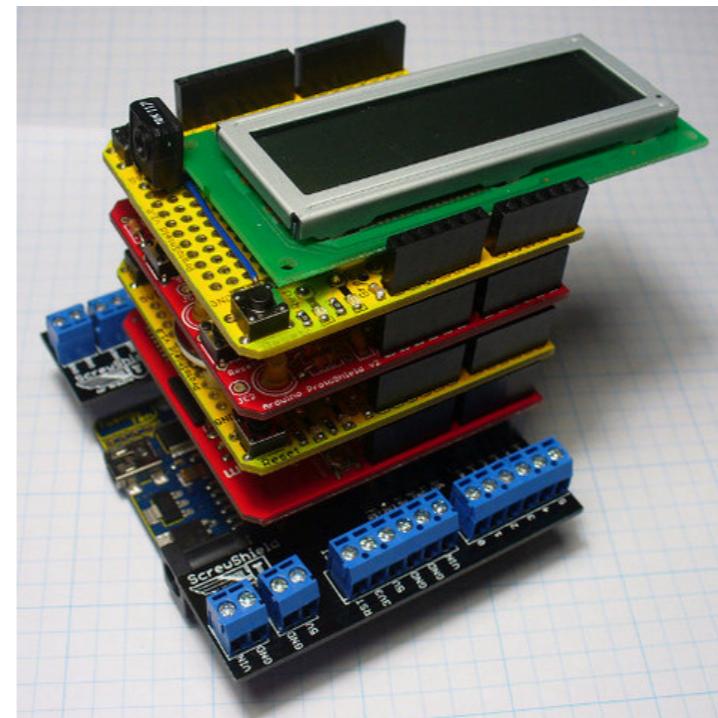
Relay Shield



Audio Shield



Ethernet Shield



Stacking Shield

อุปกรณ์หลอดไฟ LED

หลอดไฟ LED

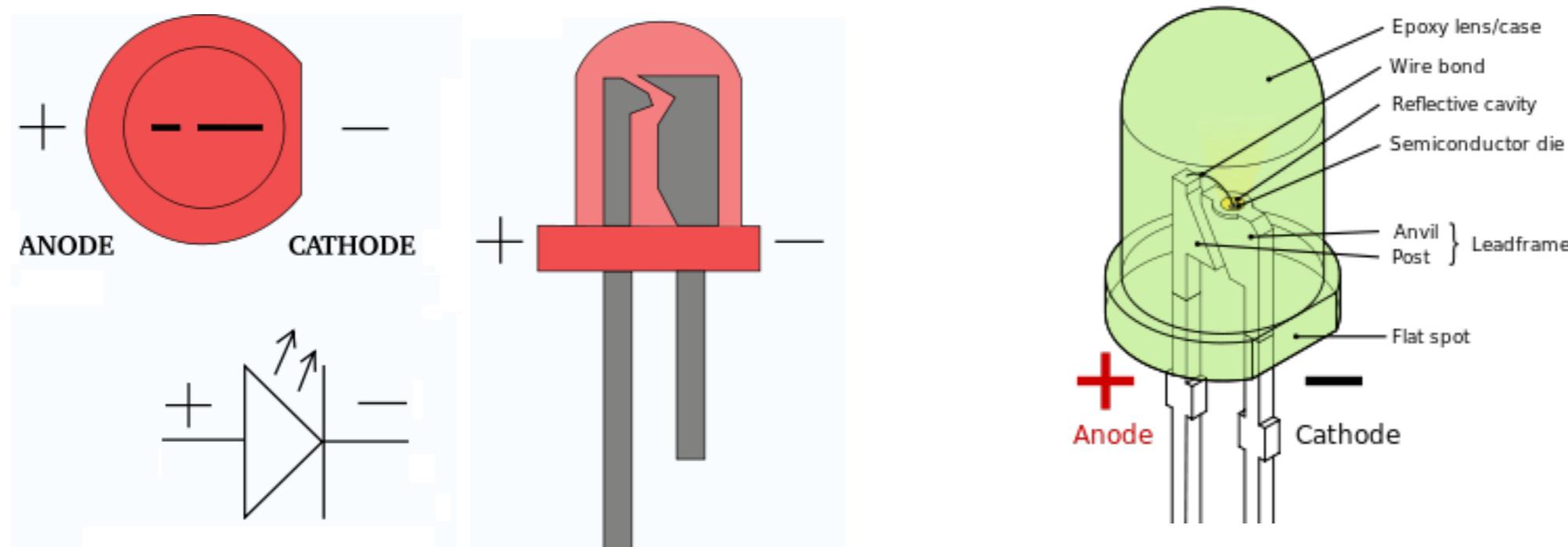
LED คือ ไนโตรดเปล่งแสง ย่อมาจากคำว่า (light-emitting diode) ซึ่งสามารถเปล่งแสงออกมากได้ แสงที่เปล่งออกมาระบบทด้วยคลื่นความถี่เดียวและเพลสต์อเนื่องกัน ซึ่งต่างกับแสงธรรมชาติที่ตาคนมองเห็น

โดยหลอด LED สามารถเปล่งแสงได้เมื่อจ่ายกระแสไฟฟ้าเข้าเพียงเล็กน้อยเท่านั้น และประสิทธิภาพในการให้แสงสว่างมากกว่าหลอด พลุออเรสเซ่น



หลักการทำงานของหลอด LED

- หลอด LED หรือไดโอดเปล่งแสง โครงสร้างประกอบไปด้วยสารกึ่งตัวนำสองชนิด (สารกึ่งตัวนำชนิด N และสารกึ่งตัวนำชนิด P) ประจุบเข้าด้วยกัน มีผิวข้างหนึ่งเรียบคล้ายกระจก
- เมื่อจ่ายไฟฟ้ากระแสตรงผ่านตัว LED โดยจ่ายไฟบวกให้ขาแอนโอด (A) จ่ายไฟลบให้ขาแคทโอด (K) ทำให้อิเล็กตรอนที่สารกึ่งตัวนำชนิด N มีพลังงานสูงขึ้น จนสามารถวิ่งข้ามรอยต่อจากสารชนิด N ไปรวมกับโอลในสารชนิด P
- การท่ออิเล็กตรอนเคลื่อนที่ผ่านรอยต่อ PN ทำให้เกิดกระแสไฟล์ เป็นผลให้ระดับพลังงานของอิเล็กตรอนเปลี่ยนไปและคายพลังงานออกมานิรูปคลื่นแสง



ตัวต้านทาน หรือ รีซิสเตอร์ (resistor หรือ R)

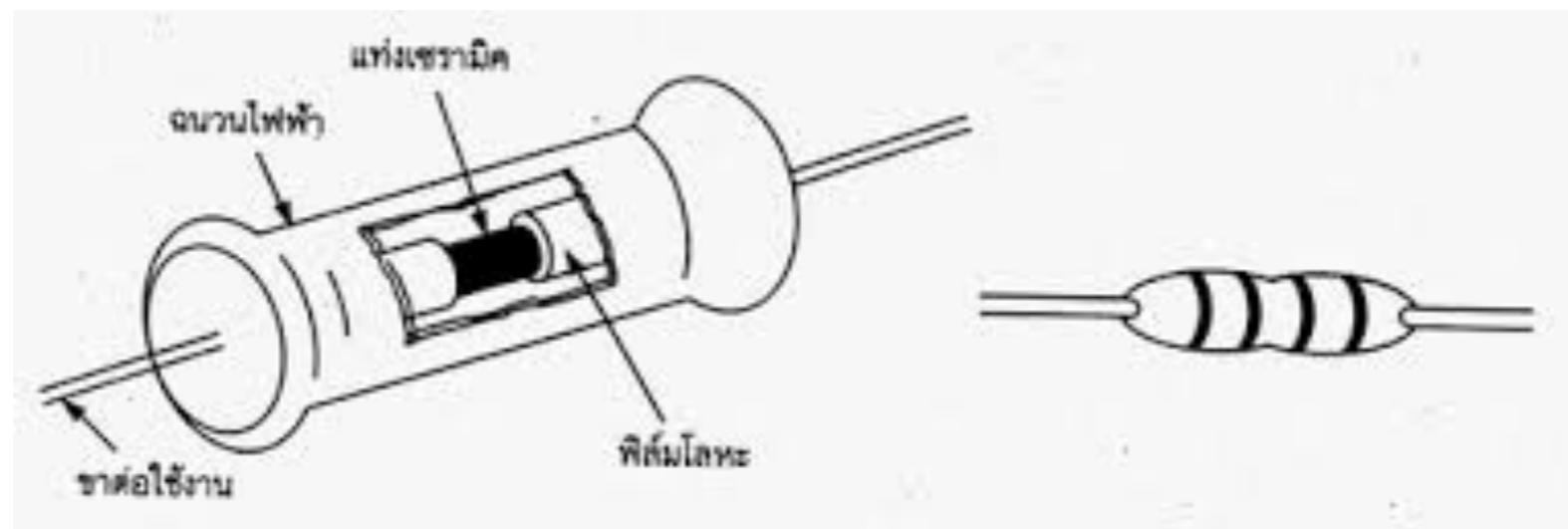
- เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่งที่มีคุณสมบัติในการต้านการไหลผ่านของกระแสไฟฟ้าสร้างขึ้นด้วยลวดต้านทานหรือถ่านคาร์บอน
- ถ้าอุปกรณ์นี้มีความต้านทานมาก กระแสไฟฟ้าที่ไหลผ่านจะน้อยลง เป็นอุปกรณ์ไฟฟ้าชนิดพาลซีฟสองขั้วสร้างความต่างศักย์ไฟฟ้าคร่อมขั้วทั้งสอง (V) โดยมีสัดส่วนมากน้อยตามปริมาณกระแสไฟฟ้าที่ไหลผ่าน (I) อัตราส่วนระหว่างความต่างศักย์ และปริมาณกระแสไฟฟ้าคือ ค่าความต้านทานทางไฟฟ้า หรือค่าความต้านทานของตัวนำมีหน่วยเป็นโอห์ม (สัญลักษณ์ : Ω) เขียนเป็นสมการตามกฎของโอห์ม ดังนี้



$$R = \frac{V}{I}$$

หลักการทำงานของ Resistors

- เมื่อทำการป้อนแรงดันให้ LED ต้องมีแรงดันที่พอเหมาะสมและข้าว
แรงดันที่ถูกต้อง LED จึงทำงานได้
- ถ้าป้อนแรงดันมากไปจะทำให้ชำรุดเสียหาย โดยแก้ไขด้วยการ
คำนวณค่าตัวต้านทานจากสมการตัวต้านทานไฟฟ้า
- ตัวต้านทานทำหน้าที่ลดการไหลของกระแสและในเวลาเดียวกันก็
ทำหน้าที่ลดระดับแรงดันไฟฟ้าภายในวงจรทั่วไป Resistors อาจ
เป็นแบบค่าความต้านทานคงที่ หรือค่าความต้านทานแปรได้



$$R = \frac{E - E_{LED}}{I_{LED}}$$

จากสูตรเมื่อ

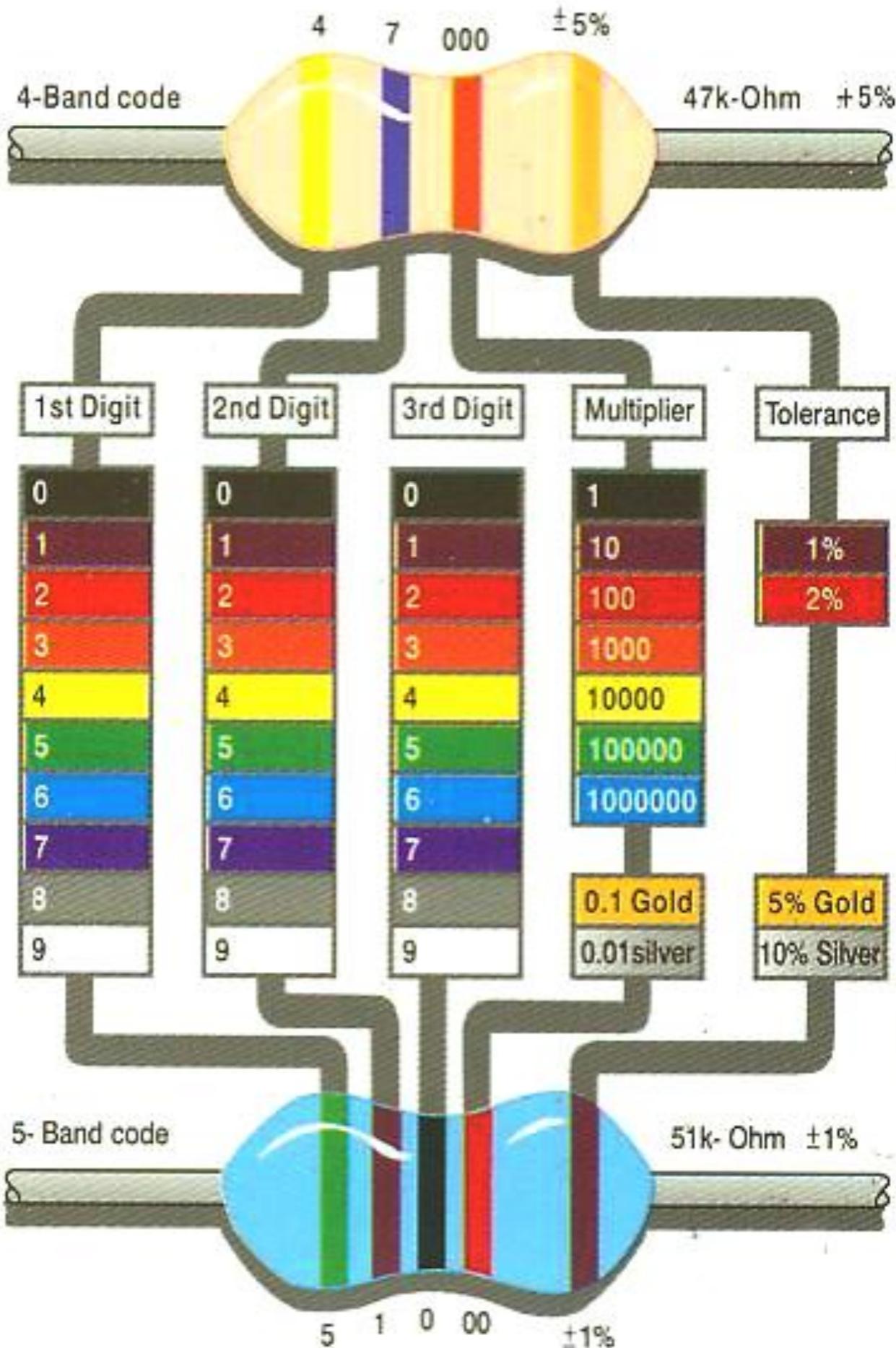
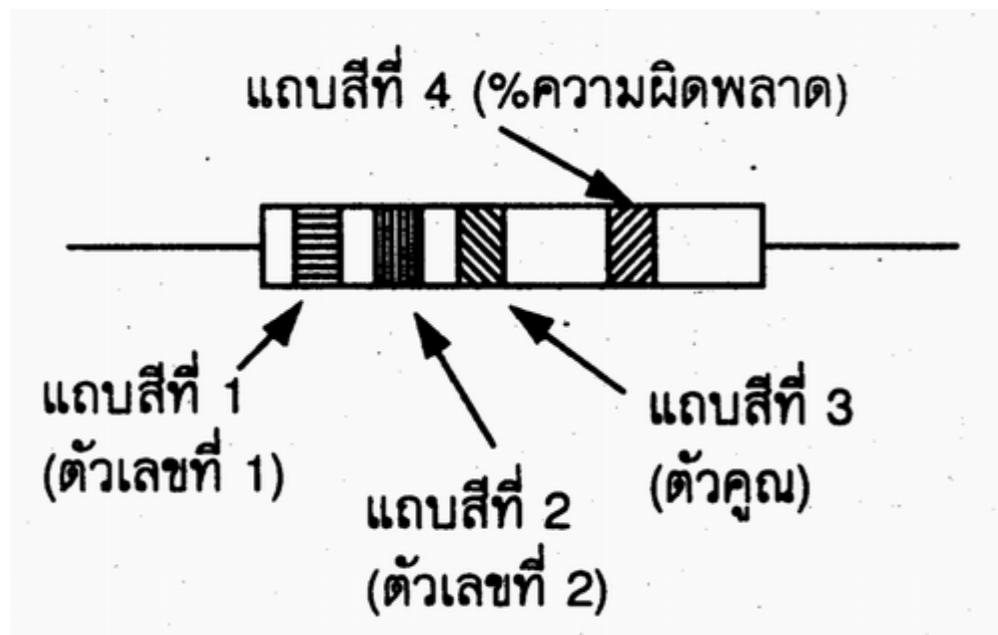
R = ตัวต้านทานจำกัดกระแสที่ต่อในวงจร(โอห์ม: Ω)

E = แรงดันที่จ่ายให้วงจร(โวลต์:V)

E_{LED} = แรงดันที่ต้องการให้ต่อกรุ่ม LED(โวลต์:V)

I_{LED} = กระแสที่ต้องการจ่ายให้ LED(แอมป์:A)

วิธีอ่าน ค่าตัวต้านทาน



อ้างอิงจาก <http://www.oknation.net/blog/gantsumbrella/2007/12/09/entry-3>

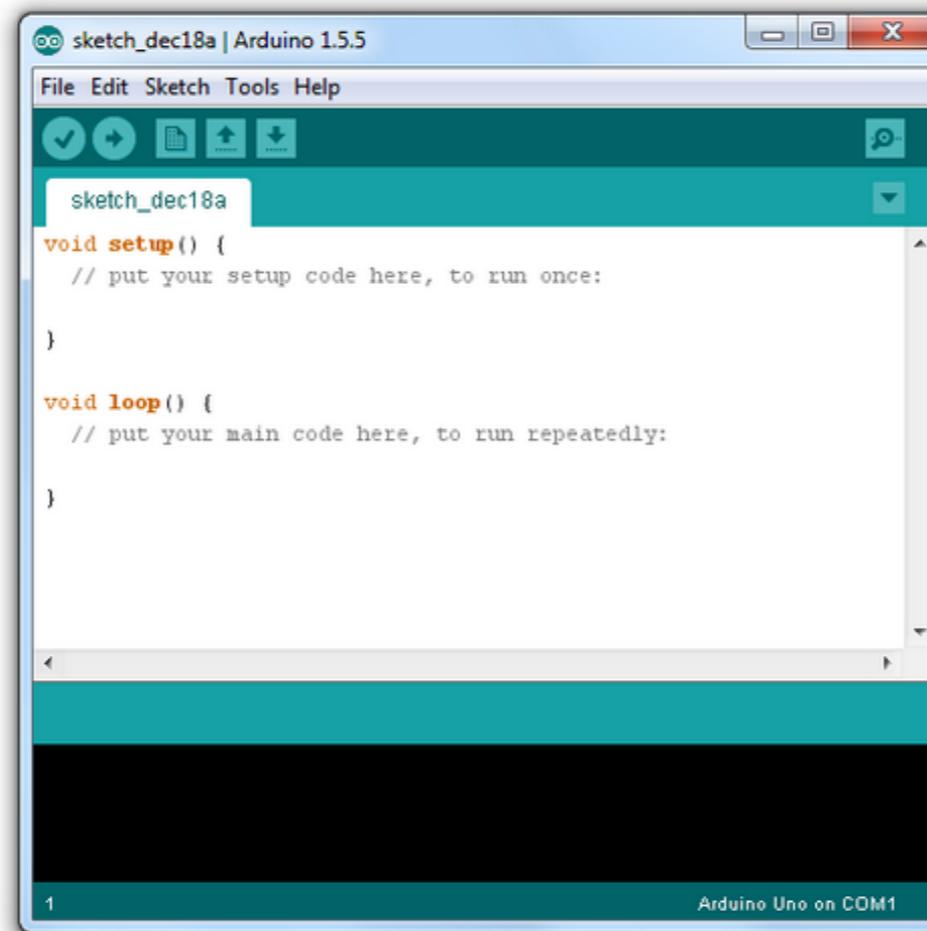
HOW TO ARDUINO IDE

HOW TO ARDUINO IDE

1. เมื่อดาวน์โหลด Arduino IDE มาติดตั้งเรียบร้อยจะมี Icon นี้



2. เปิดโปรแกรม Arduino IDE จะพบหน้าจอดังกล่าว



HOW TO ARDUINO IDE

3. โปรแกรม Arduino IDE จะมีการทำงานดังนี้

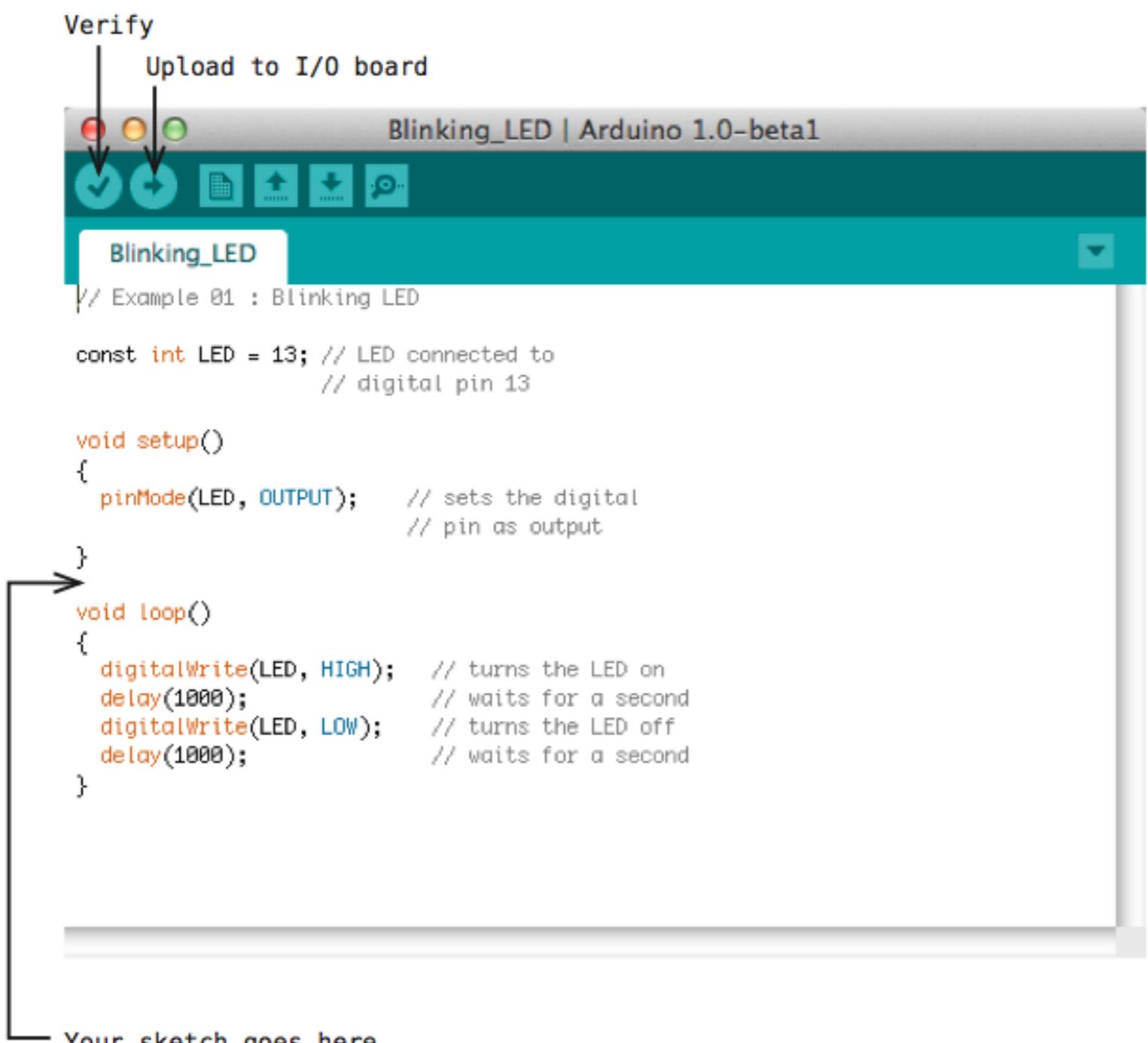


Figure 4-3.

The Arduino IDE with your first sketch loaded

- **Verify** เช็คโค้ดว่าถูกต้องหรือไม่โดยการcompile
- **Upload Burn**ลงArduino Board
- **New**สร้างไฟล์ sketch ขึ้นใหม่
- **Open**เปิดไฟล์งานเดิมมาแก้ไข
- **Save**บันทึกในรูปแบบ sketch (.INO)
- **Serial Monitor**แสดงผลและสามารถส่งข้อมูลโดยผ่าน serial port(com port)

HOW TO ARDUINO IDE

4. โปรแกรม Arduino IDE จะมี Icon menu การทำงานดังนี้

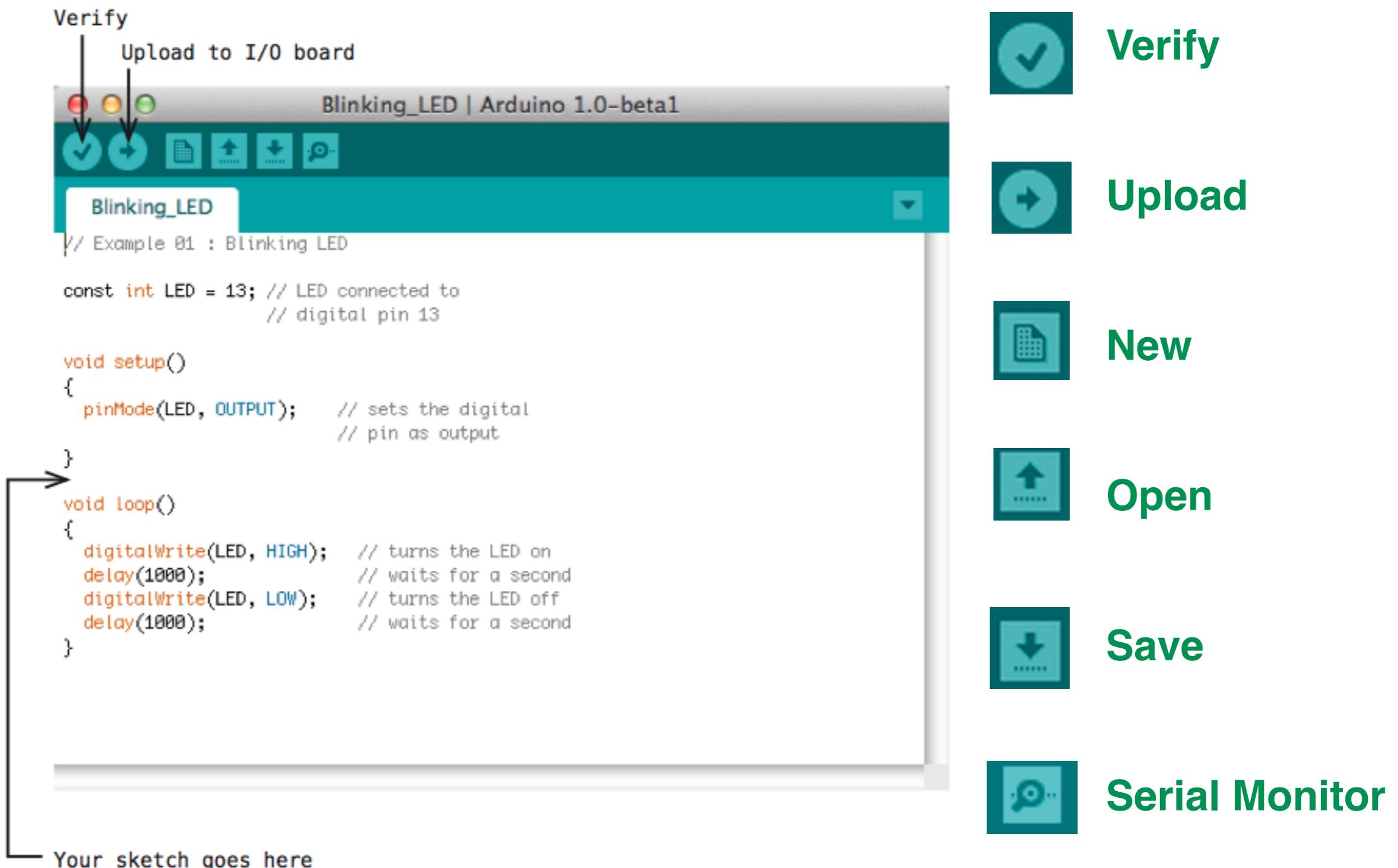


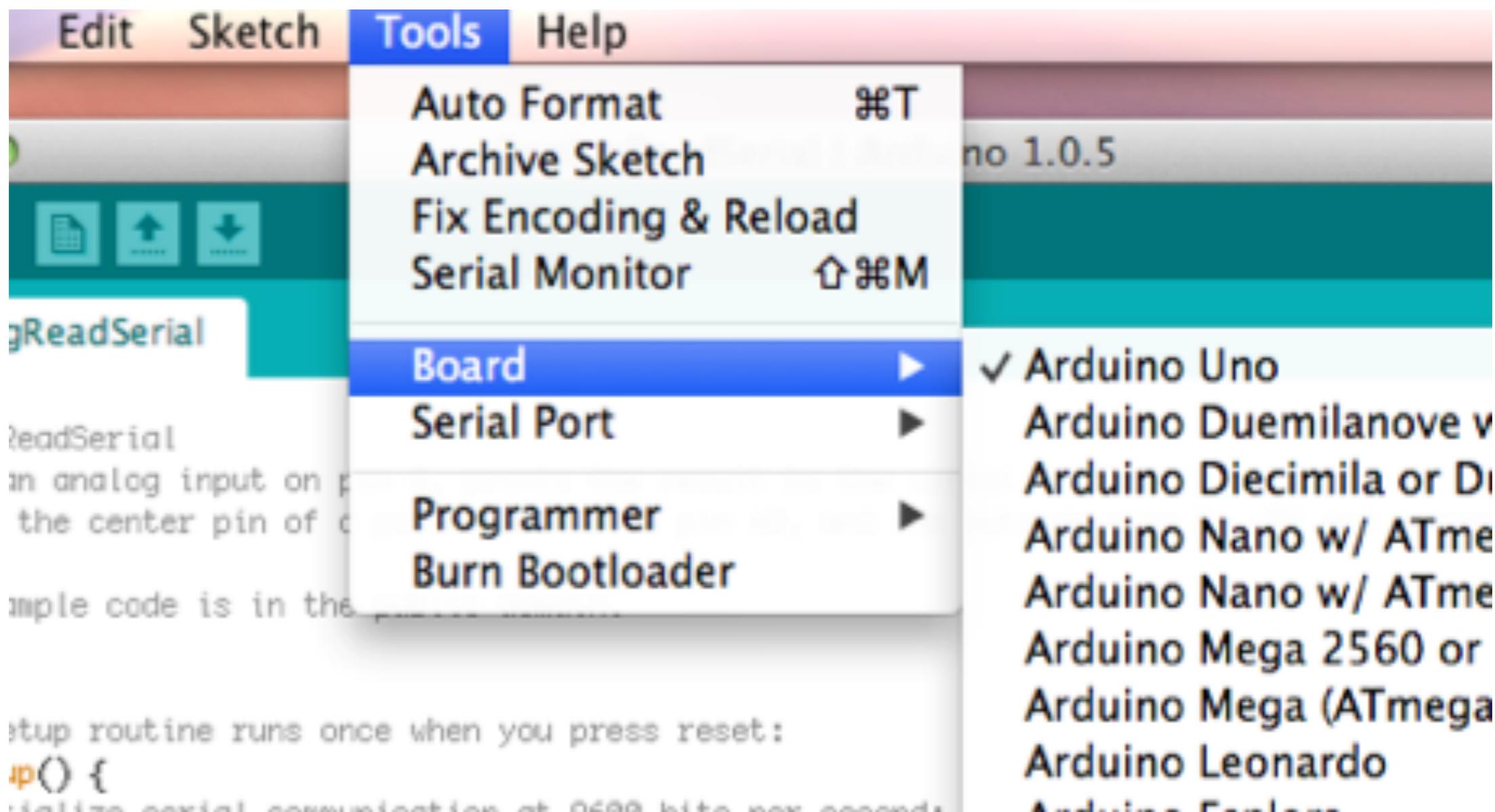
Figure 4-3.

The Arduino IDE with your first sketch loaded

HOW TO ARDUINO IDE

5. เมื่อทำการเชื่อมต่อ Arduino บอร์ดเข้ากับคอมพิวเตอร์แล้ว ให้ทำการเชื่อมโดยดังนี้

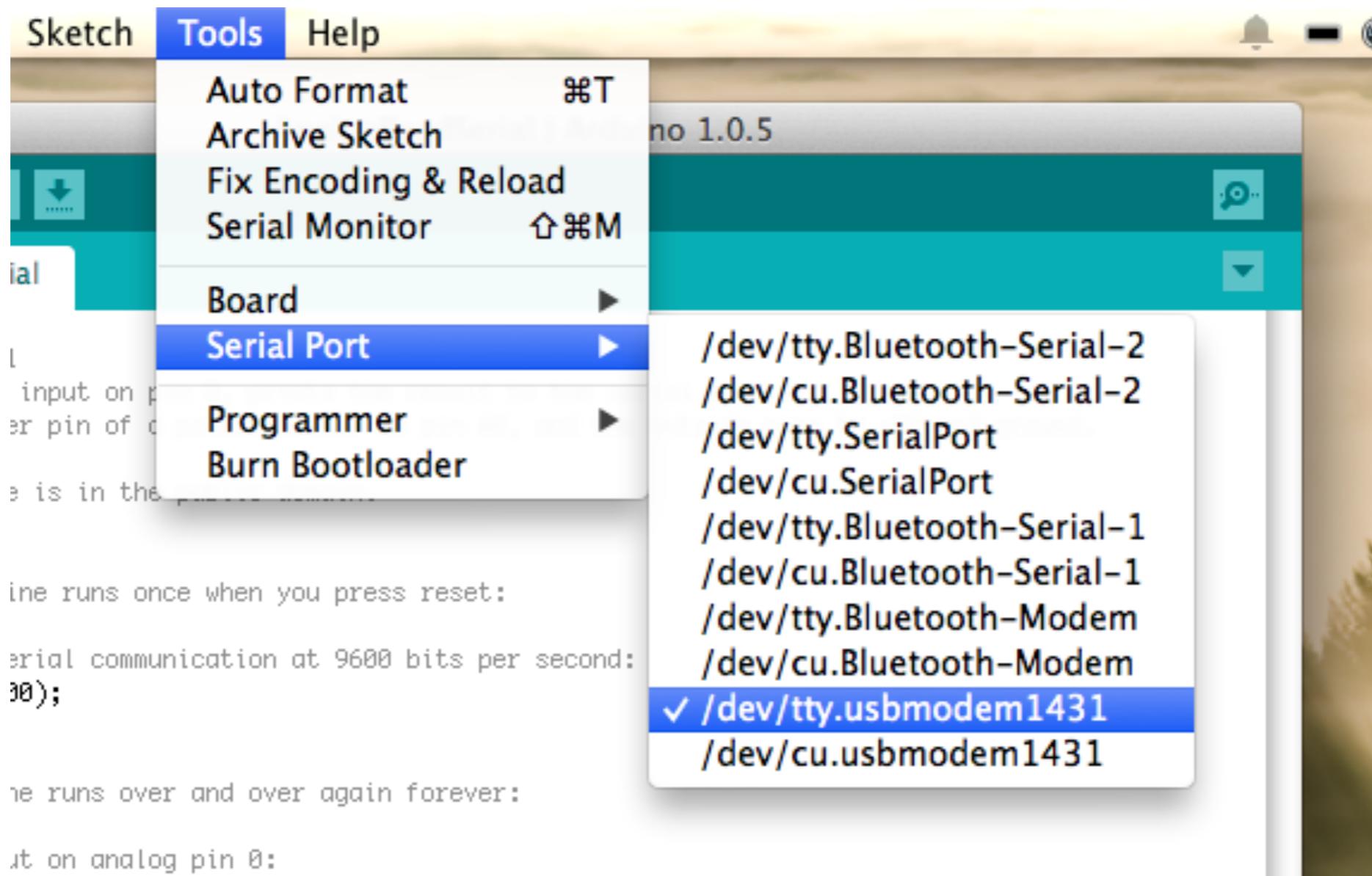
5.1 menu->Tools->Board เลือกบอร์ดรุ่นที่เราใช้



HOW TO ARDUINO IDE

5. เมื่อทำการเชื่อมต่อ Arduino บอร์ดเข้ากับคอมพิวเตอร์แล้ว ให้ทำการเชื่อมโดยดังนี้

5.2 menu—>Tools—>Serial Port เลือกPort ที่เชื่อมการสื่อสารของเรา



HOW TO ARDUINO IDE

5. เมื่อทำการเชื่อมต่อ Arduino บอร์ดเข้ากับคอมพิวเตอร์แล้ว ให้ทำการเชื่อมโดยดังนี้

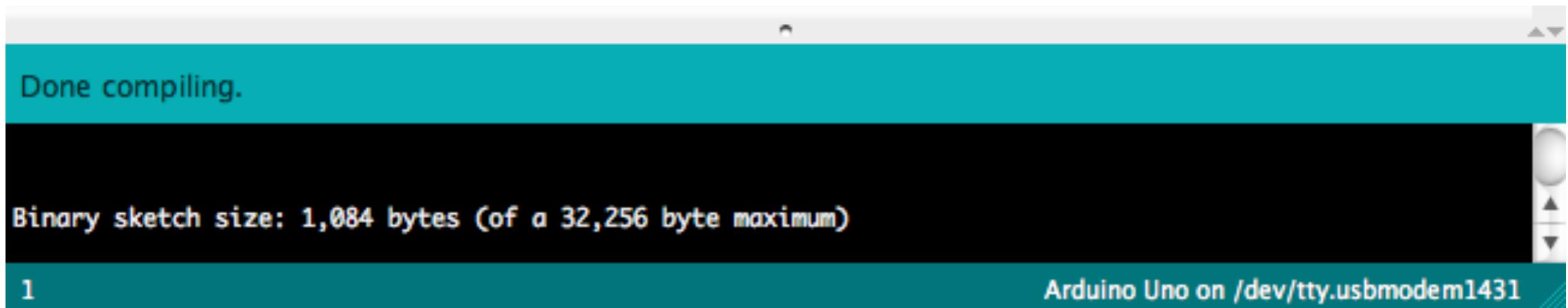
5.2 menu—>Tools—>Serial Port เลือกPort ที่เชื่อมการสื่อสารของเรา



HOW TO ARDUINO IDE

6. เมื่อเขียนโปรแกรมแล้วเสร็จ

6.1 ทำการ verify code เมื่อไฟล์ compile ผ่านจะแสดงผลดังภาพ

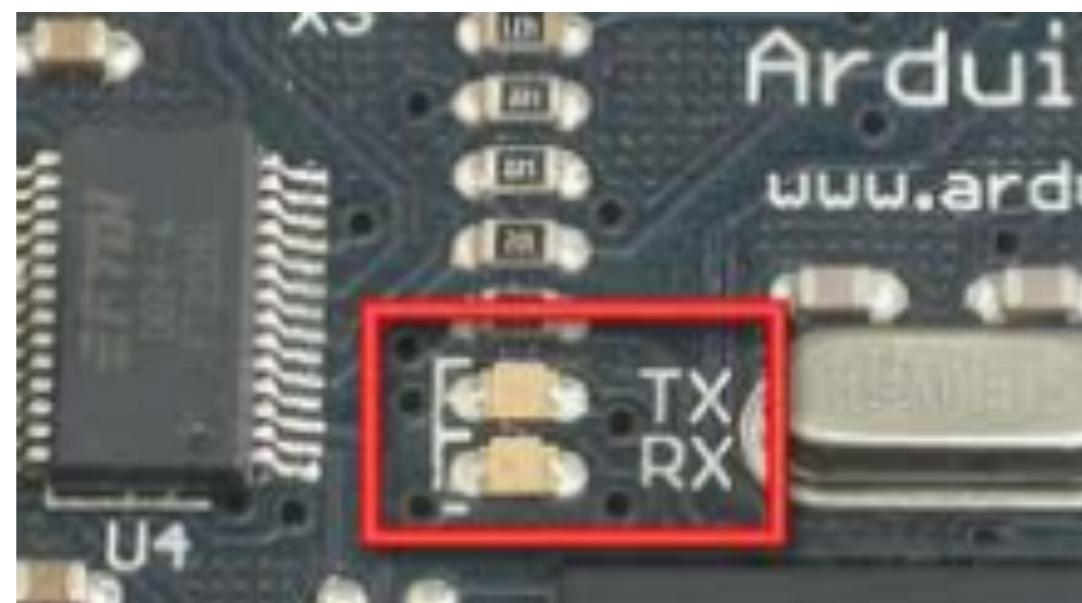
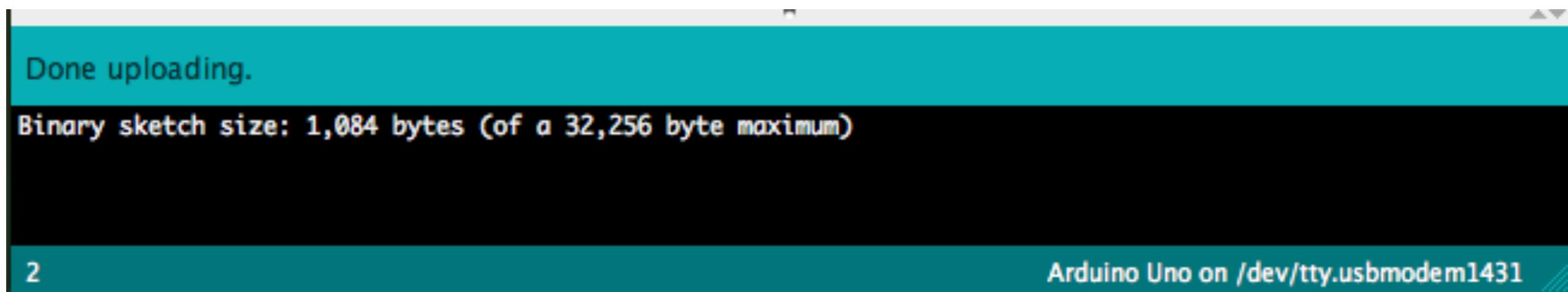


HOW TO ARDUINO IDE

6. เมื่อเขียนโปรแกรมแล้วเสร็จ

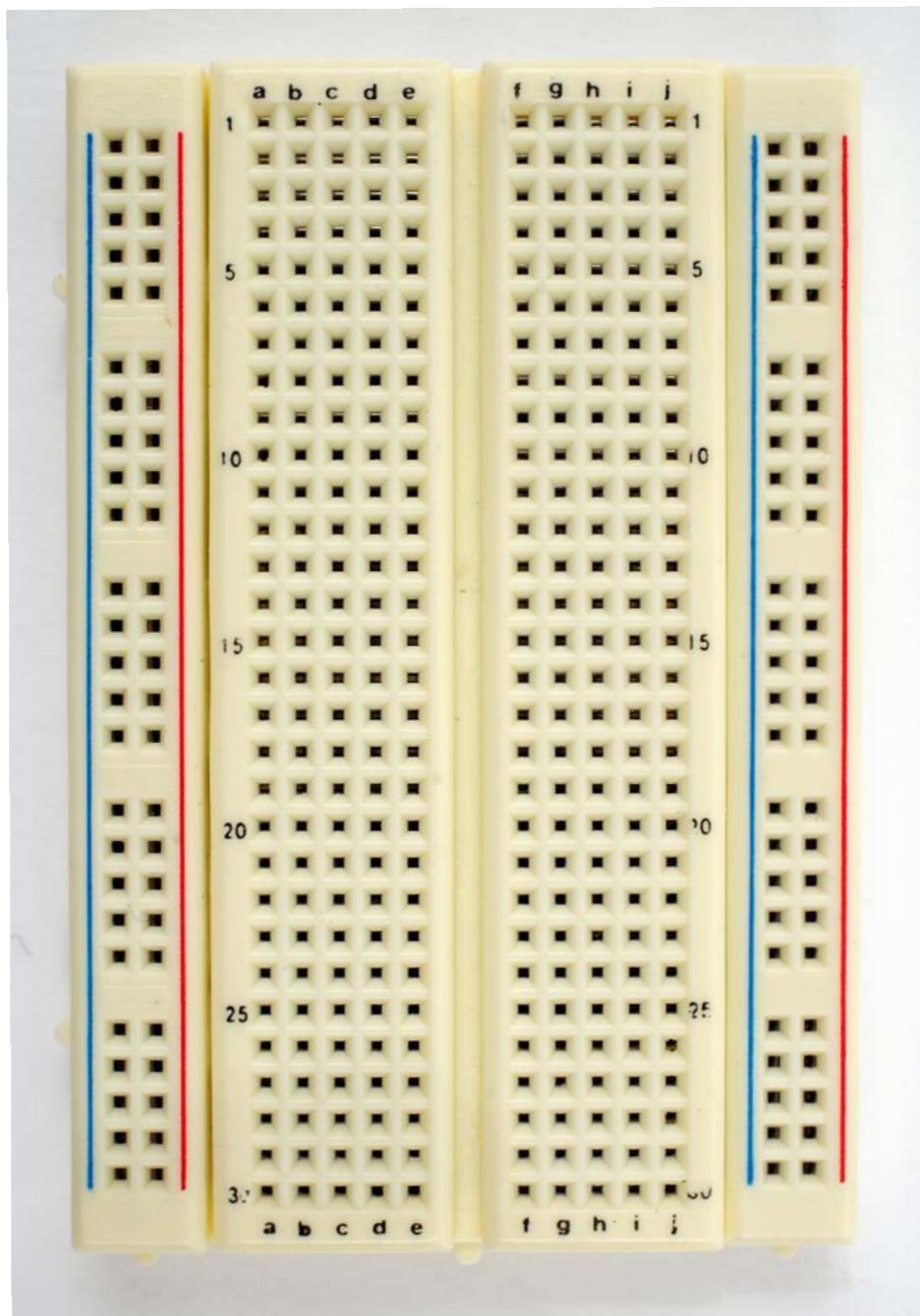
6.2 ทำการอัพโหลด  ไฟล์ burn เข้า Arduino บอร์ด เมื่อไฟล์ upload ผ่านจะแสดงผลดังภาพ

ไฟ LED TX/RX จะกระพริบไปมา ** จึงไม่ควรต่อวงจรโดยใช้ Di 0 และ 1



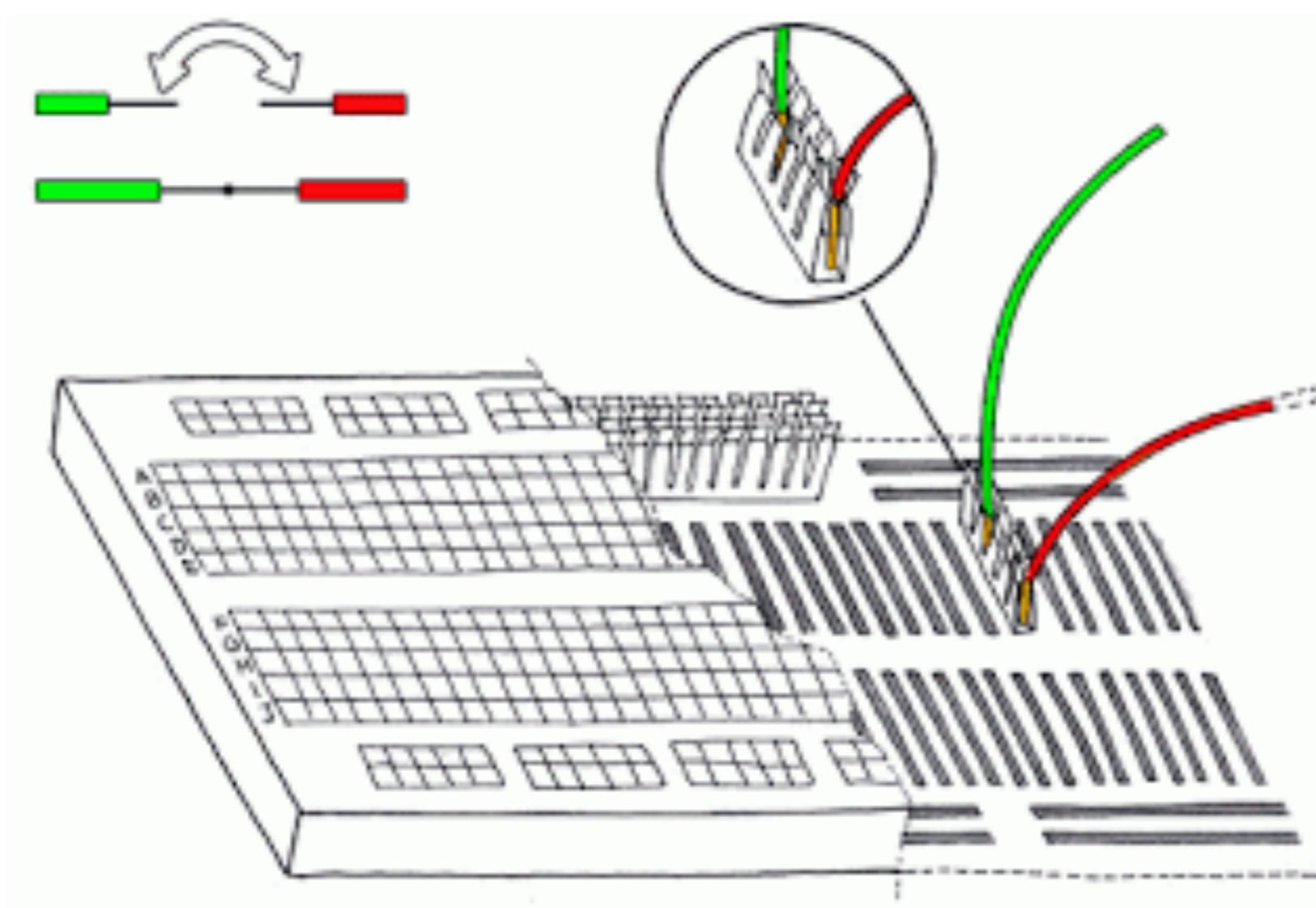
หลักการใช้งานบอร์ดทดลอง

ໂພຣໂທບອർດ (ອັງກຸນ: protoboard) ອີ່ວ ເບຣດບອർດ (ອັງກຸນ: breadboard)



ໂພຣໂທບອርດ

គីអូអុបរណ៍ទេសចរណ៍ត័ណបេបាំដោយផ្សាហ៍តិក ស៊វនខ៉ាង ឯង
បើនរួលើលេខពីរដែលមានចំណេះចំណេះ ដែលមានចំណេះចំណេះ

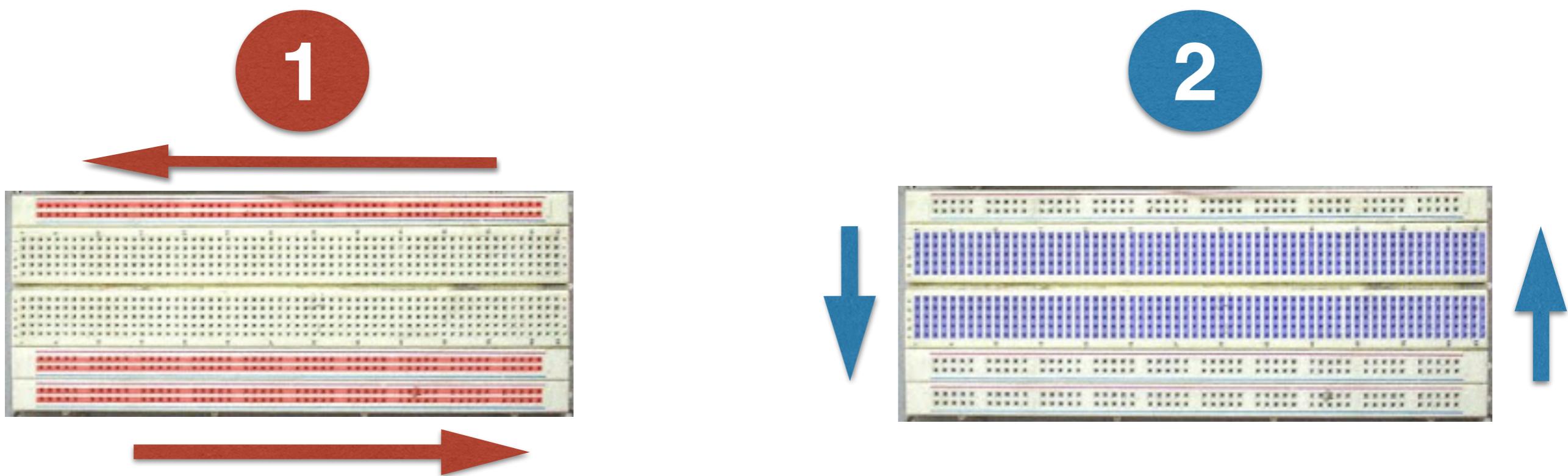


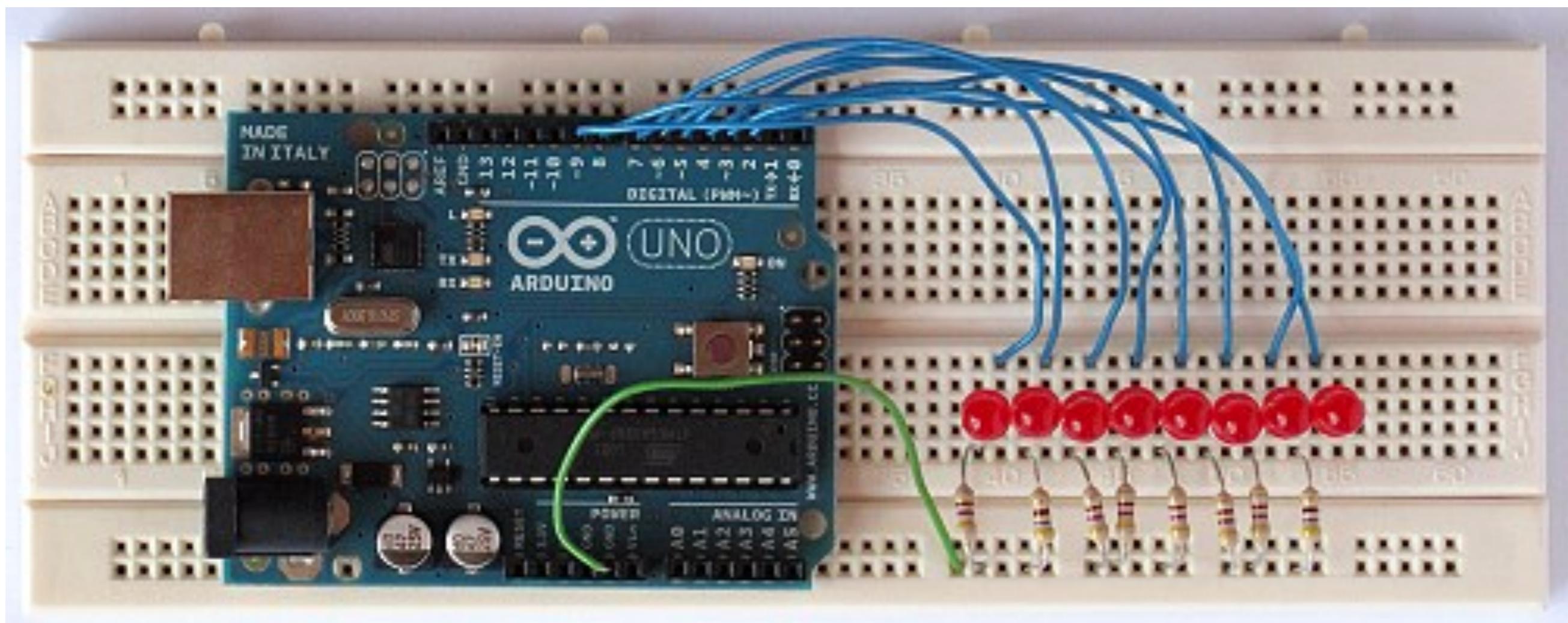
ลักษณะการวางแผนลวดของ พอร์ตบอร์ด

ลักษณะเส้นลวดจะวางตามลูกศรดังภาพ

1. เส้นสีแดง เป็นจุดที่ทำการต่อการเลี้ยงไฟฟ้า ให้แก่ช่วงจุดคือ V+ และ Gnd(Ground)
2. เส้นสีน้ำเงิน เป็นจุดที่ลงอุปกรณ์ที่ใช้ทดลอง เช่น หลอดไฟ ตัวต้านทาน

**เคล็ดลับ ควรใช้ตัวต้านทานต่อจุด Gnd แทนการใช้สายไฟจะลดจำนวนสายไฟและทำให้เป็นระเบียบมากขึ้น





จากภาพ จะเห็นการต่อวงจรที่ไม่สีนเปลืองสายไฟมาก
เนื่องจาก ใช้ตัวต้านทานเชื่อมที่ Gnd แล้ว

Lab#1



เตรียมอุปกรณ์

1. Arduino Uno R3	1
2. สาย USB	1
3. หลอด LED	4
4. ตัวต้านทาน $220\ \Omega$	4
5. photo board	1
6. สาย M to M	5

1. ดาวน์โหลด โปรแกรมจากเว็บ arduino.cc
2. ติดตั้ง โปรแกรมลง ในเครื่องของท่าน
3. ทดลองนำไฟล์ตัวอย่างที่ File—>Examples
—>01.Basics—>Blink
4. verify และ upload
5. ทดลองทำไฟกระพริบหลายดวง และกระพริบ
จากต้นไปท้าย และจากท้ายไปต้น

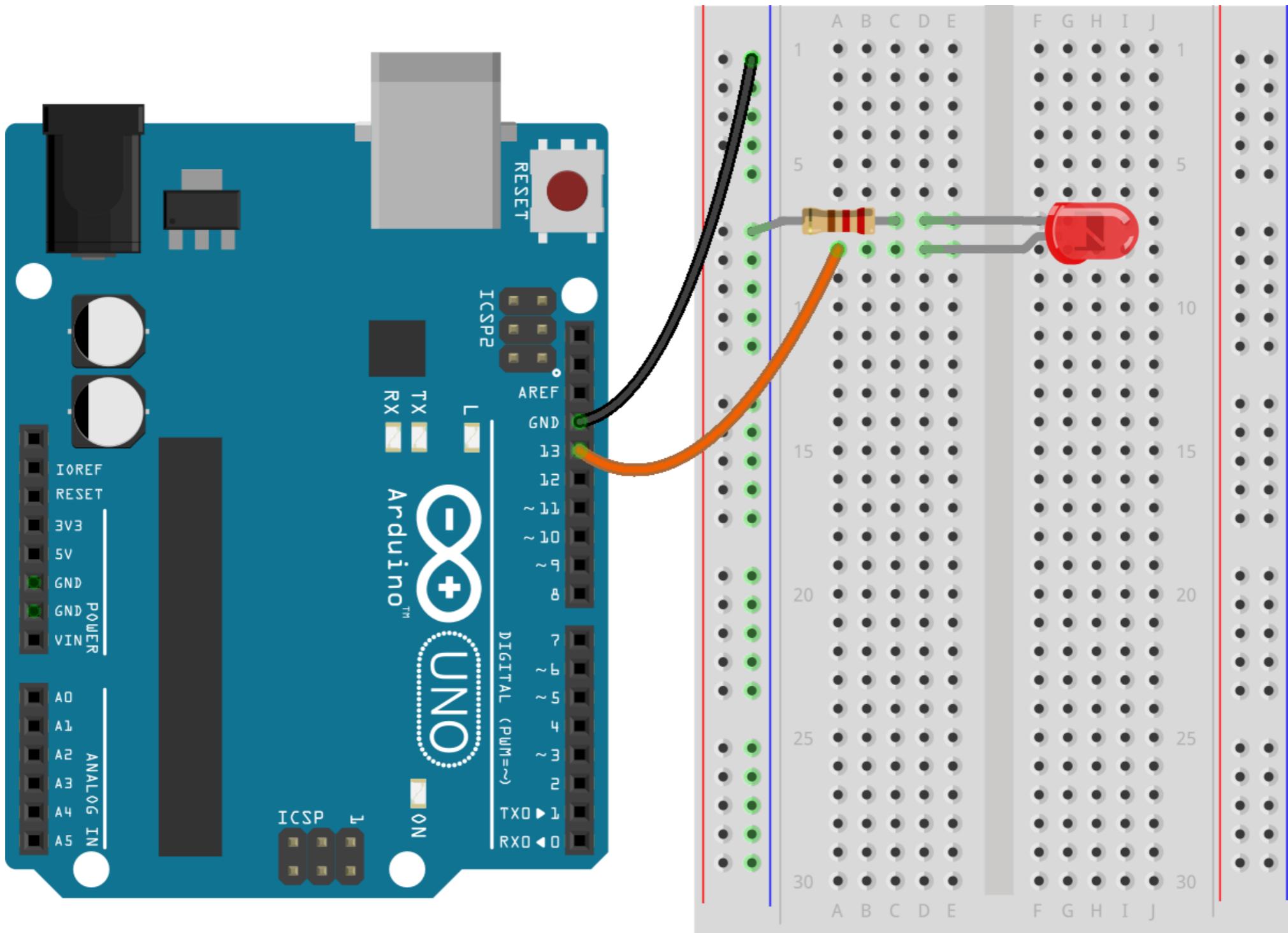
LAB1

Blinking LED

ไฟกระพริบ แบบเริ่มต้น (Blinking LED)



ແຜນຜັງຈອງ ລົດໄຟກະພຣິບໄປມາລົດເດືອວ



fritzing

1. ให้หลอดไฟกระพริบไปมาหลอดเดียว

```
1  /*
2   * Blink
3   * Turns on an LED on for one second, then off for one second, repeatedly.
4   *
5   * This example code is in the public domain.
6   */
7
8 // Pin 13 has an LED connected on most Arduino boards.
9 // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14     // initialize the digital pin as an output.
15     pinMode(led, OUTPUT);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20     digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
21     delay(1000);                // wait for a second
22     digitalWrite(led, LOW);      // turn the LED off by making the voltage LOW
23     delay(1000);                // wait for a second
24 }
```

25

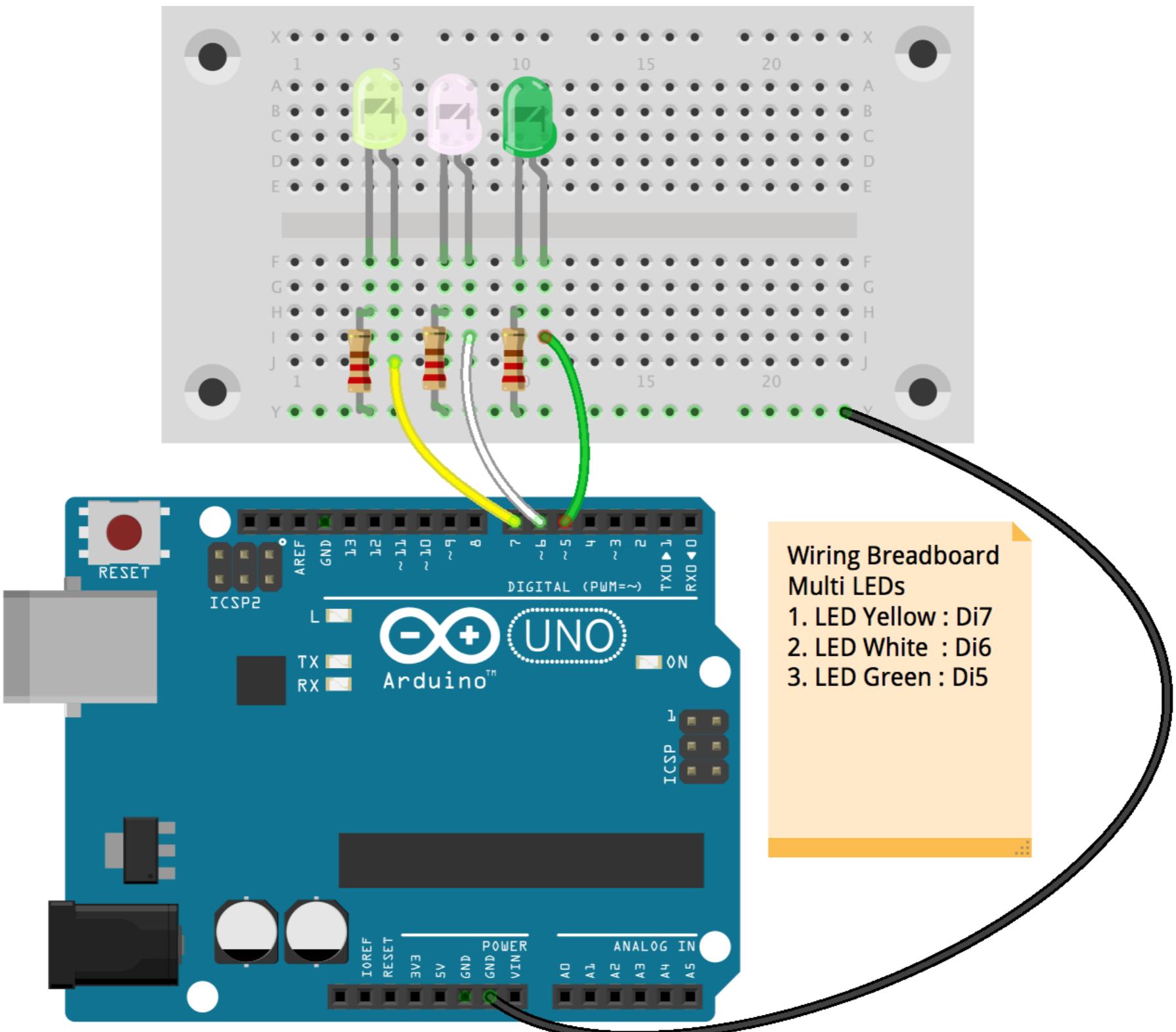
LAB2

Multiple Blinky

สร้างไฟกระพริบมากกว่า 1 ดวง (Multiple Blinky)



ແຜນຜົ່ງລອດໄຟກະພົບໄປມາຫລາຍລອດ



2. หลอดไฟกระพริบไปมาหลายหลอด

```
2 int timer = 100;           // The higher the number, the slower the timing.  
3  
4 void setup() {  
5     // use a for loop to initialize each pin as an output:  
6     for (int thisPin = 5; thisPin < 8; thisPin++) {  
7         pinMode(thisPin, OUTPUT);  
8     }  
9 }
```



```
11 void loop0 {  
12     // loop from the lowest pin to the highest:  
13     for (int thisPin = 5; thisPin < 8; thisPin++) {  
14         // turn the pin on:  
15         digitalWrite(thisPin, HIGH);  
16         delay(timer);  
17         // turn the pin off:  
18         digitalWrite(thisPin, LOW);  
19     }  
}
```

2

```
21 // loop from the highest pin to the lowest:  
22 for (int thisPin = 7; thisPin >= 5; thisPin--) {  
23     // turn the pin on:  
24     digitalWrite(thisPin, HIGH);  
25     delay(timer);  
26     // turn the pin off:  
27     digitalWrite(thisPin, LOW);  
28 }  
29 }
```

3

LAB 3

Fade LED

<Lab 3> การหร์ไฟ (Fade LED)



<Lab 3> การหรไฟ (Fade LED) 1

```
1 // สำหรับบอร์ด Arduino ซึ่ง PWM บนบอร์ดสามารถใช้ซึ่ง 3,5,6,9,10,11  
2 // บน Funbasic I/O Shiled ใช้ได้ซึ่ง 6,9,10,11  
3 int ledPin = 6;  
4  
5 void setup() {  
6     // put your setup code here, to run once:  
7  
8     pinMode(ledPin, OUTPUT); // Set ledPin to an output  
9 }  
10  
11 void loop() {  
12     // put your main code here, to run repeatedly:  
13  
14     // ตั้งค่าให้ Value = 0 ไฟจะออกประมาณ 0 โวลต์ ทำให้ LED ดับ  
15     analogWrite(ledPin, 0); // Sets voltage to 0V (0/255 * 5V). LED is off.  
16     delay(1000);           // Wait a second  
17  
18     // ตั้งค่าให้ Value = 64 ไฟจะออกประมาณ 1.25 โวลต์ ทำให้ LED สว่างเล็กน้อย  
19     analogWrite(ledPin, 64); // Sets voltage to ~1.25V (64/255 * 5V). Pretty dim.
```

<Lab 3> การหร์ไฟ (Fade LED) 2

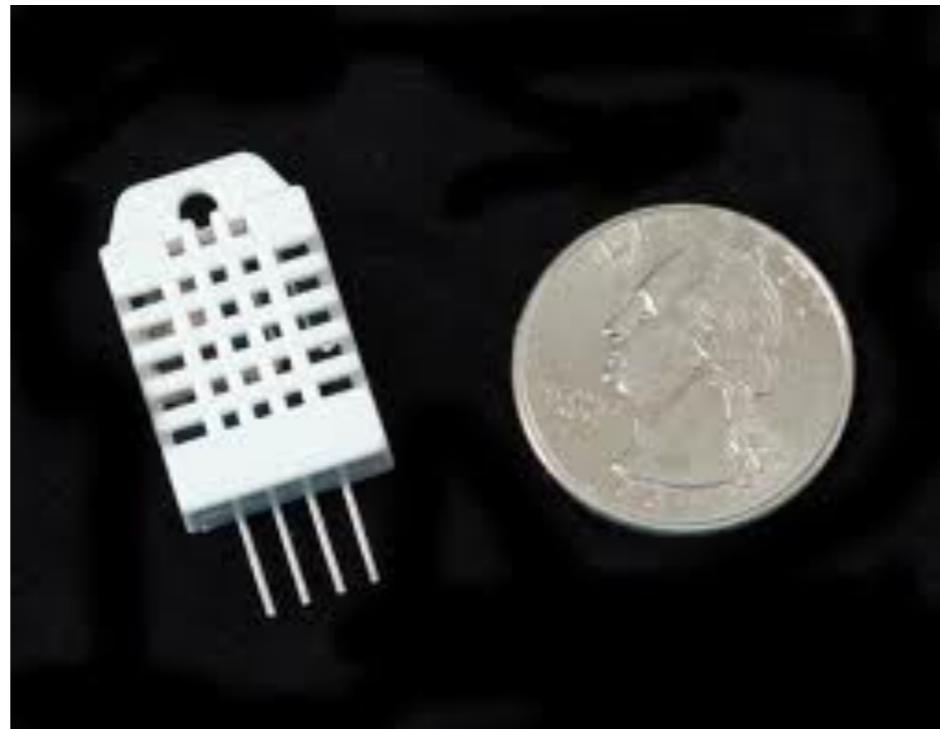
```
20 delay(1000); // Wait a second
21
22 // ตั้งค่าให้ Value = 128 ไฟจะออกประมาณ 2.5 โวลต์ ทำให้ LED สว่างปานกลาง
23 analogWrite(ledPin, 128); // Set voltage to ~2.5V. Half-bright.
24 delay(1000); // Wait a second
25
26 // ตั้งค่าให้ Value = 192 ไฟจะออกประมาณ 3.75 โวลต์ ทำให้ LED สว่างเกือบจะสุด
27 analogWrite(ledPin, 192); // Set voltage to ~3.75V. Getting brighter!
28 delay(1000); // Wait a second
29
30 // ตั้งค่าให้ Value = 255 ไฟจะออกประมาณ 5 โวลต์ ทำให้ LED สว่างสุด
31 analogWrite(ledPin, 255); // Set voltage to 5V. Fully on (turn down for what!?).
32 delay(1000); // Wait a second
33 }
```

Digital Temperature Humidity Sensor

วัดอุณหภูมิและความชื้นสัมพัทธ์แบบดิจิทัล
(Digital Temperature Humidity Sensor)

Digital Temperature Humidity Sensor

- DHT22 (AM2302) High Accuracy Digital Temperature and Humidity Sensor
- โมดูลวัดอุณหภูมิและความชื้นที่มีความแม่นยำสูง



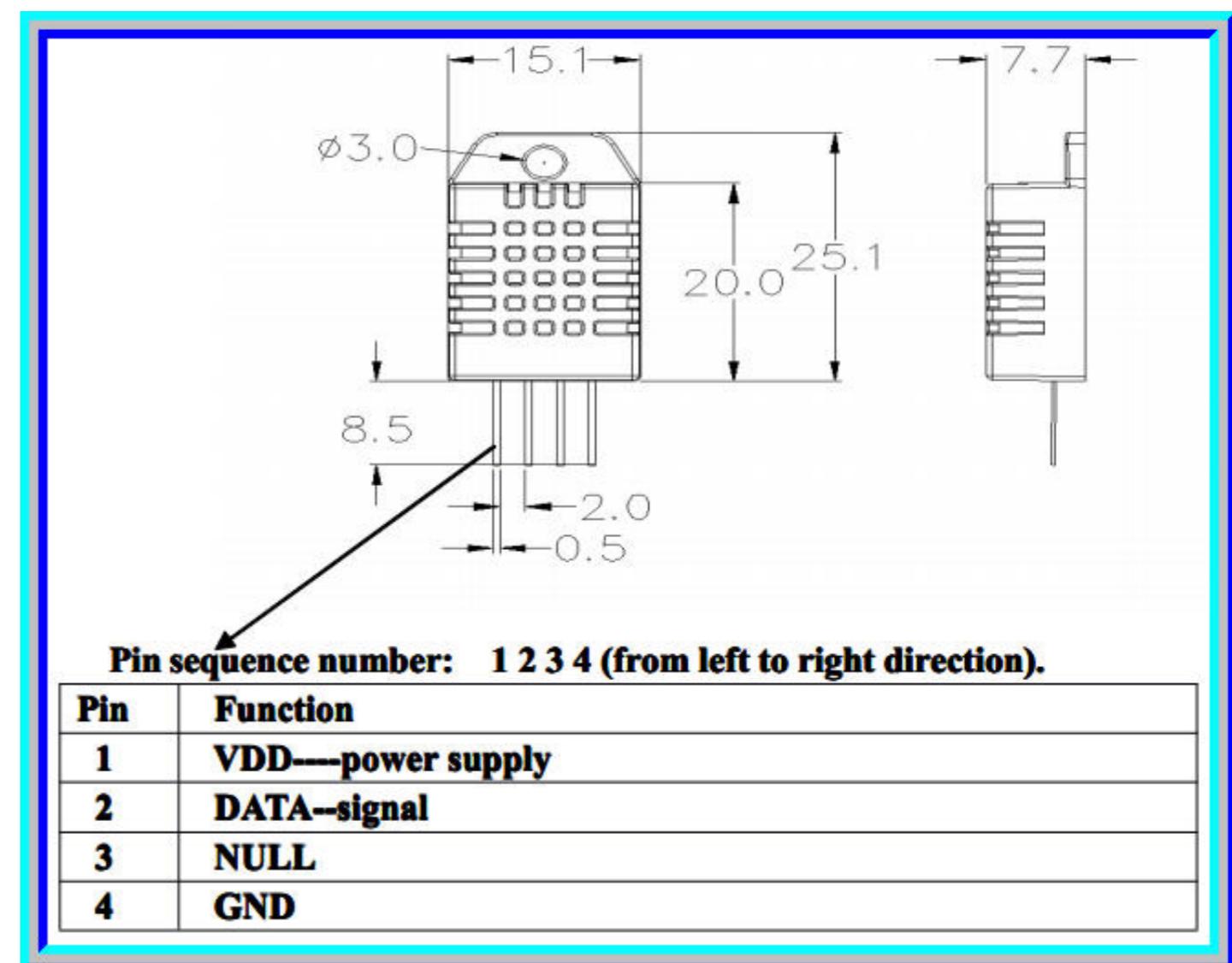
Digital Temperature Humidity Sensor

Product Description

- Accuracy humidity +-2%RH(Max +-5%RH); temperature +-0.2Celsius
- Resolution or sensitivity humidity 0.1%RH; temperature 0.1 Celsius
- Repeatability humidity +-1%RH; temperature +-0.2Celsius
- Humidity hysteresis +-0.3%RH
- Long-term Stability +-0.5%RH/year
- Sensing period Average: 2s
- Interchangeability fully interchangeable

Features:

- 3.3-6V Input
- 1-1.5mA measuring current
- 40-50 uA standby current
- Humidity from 0-100% RH
- -40 - 80 degrees C temperature range
- +-2% RH accuracy
- +-0.5 degrees C



การเพิ่ม Library ของอุปกรณ์



การเพิ่ม Library ของอุปกรณ์(I)

I. Download Library จาก web

This is a screenshot of a GitHub repository page for the **DHT-sensor-library**. The repository is owned by **adafruit**.

Repository Statistics:

- 21 commits
- 1 branch
- 0 releases
- 6 contributors

Recent Activity:

- tdicola authored on Jul 27: Update example to note cycle threshold parameter for Arduino Due. (latest commit fcd865b3bc)
- examples/DHTtester: Update example to note cycle threshold parameter for Arduino Due. (2 months ago)
- DHT.cpp: Resolve #24 by adding convert F to C function. (2 months ago)
- DHT.h: Resolve #24 by adding convert F to C function. (2 months ago)
- README.txt: added mini readme (3 years ago)

README.txt Content:

```
This is an Arduino library for the DHT series of low cost temperature/humidity sensors.

To download, click the DOWNLOADS button in the top right corner, rename the uncompressed folder DHT. Check that the DHT folder contains DHT.cpp and DHT.h. Place the DHT library folder your <arduinostetchfolder>/libraries/ folder. You may need to create the libraries subfolder if its your first library. Restart the IDE.
```

Actions:

- Watch (67)
- Star (248)
- Fork (254)
- Code
- Issues (0)
- Pull Requests (1)
- Wiki
- Pulse
- Graphs
- HTTPS clone URL: <https://github.com>
- You can clone with HTTPS, SSH, or Subversion.
- Clone in Desktop
- Download ZIP

A large black arrow points to the "Download ZIP" button at the bottom right of the page.

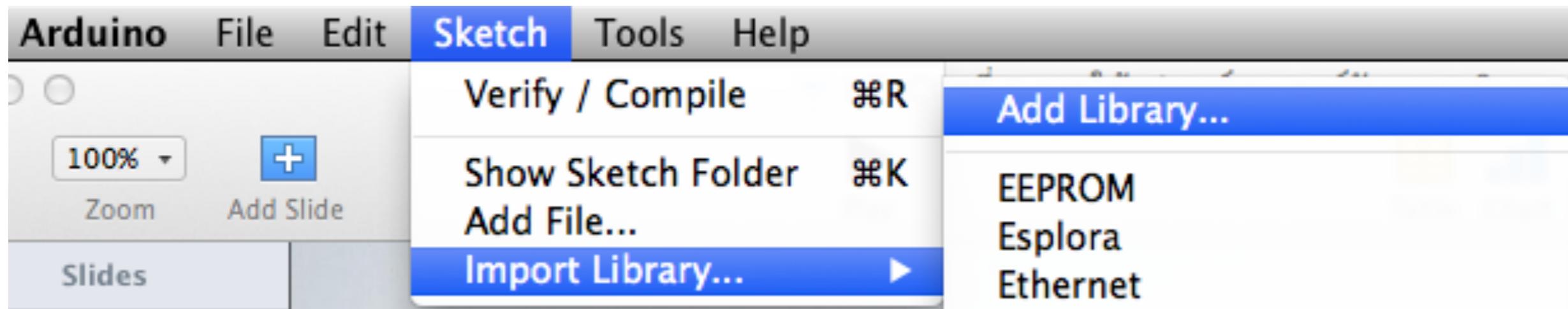
การเพิ่ม Library ของอุปกรณ์(2)

2. ทำการแตก Zip จะได้ Folder ชื่อ DHT-sensor-library-master
3. ทำการ rename Folder จากชื่อ DHT-sensor-library-master เป็น DHT_sensor

การเพิ่ม Library ของอุปกรณ์(3)

4. ทำการเพิ่ม Library โดยเปิดเมนู

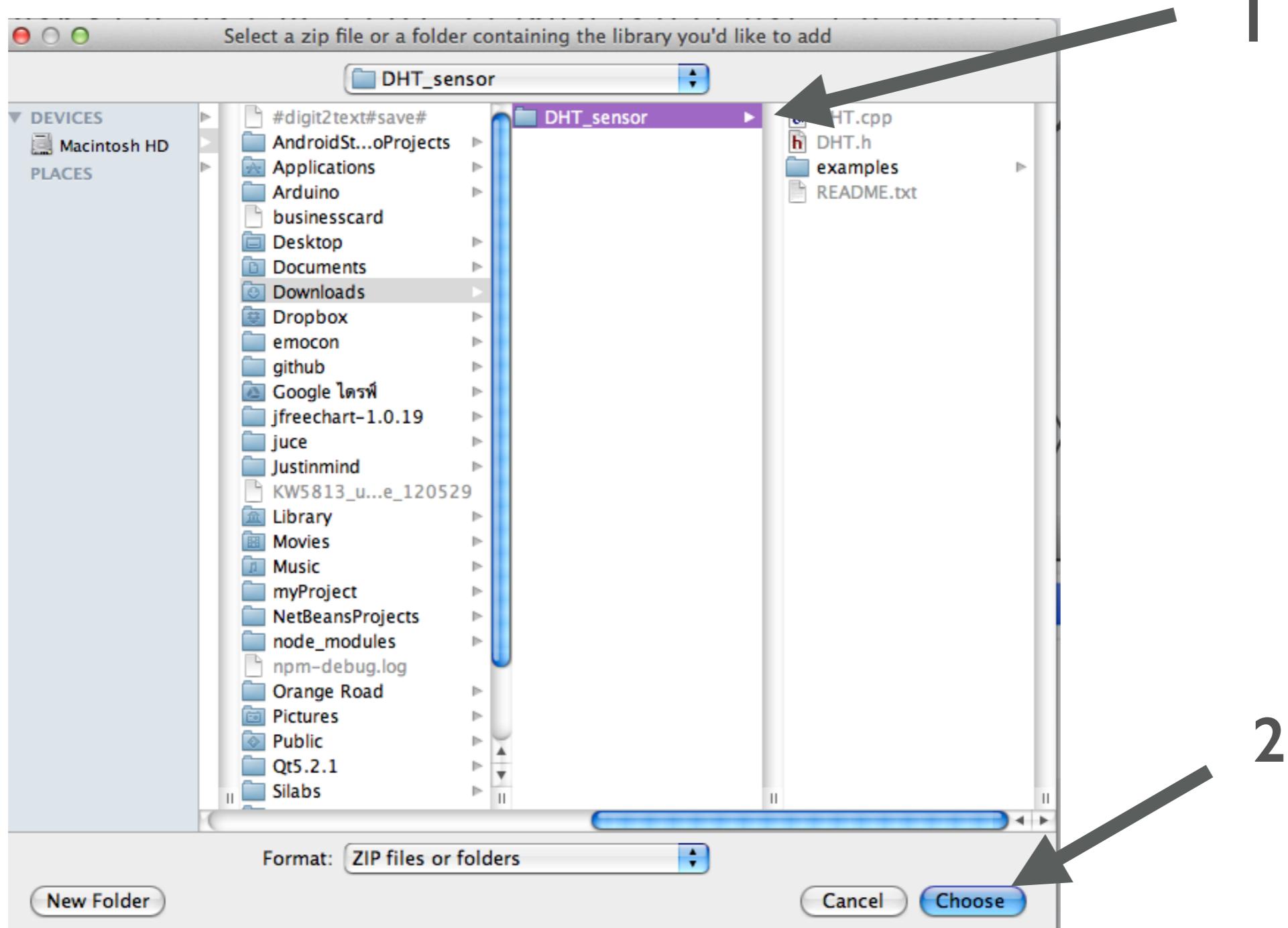
sketch -> Import Library.. ->add Library



การเพิ่ม Library ของอุปกรณ์(3)

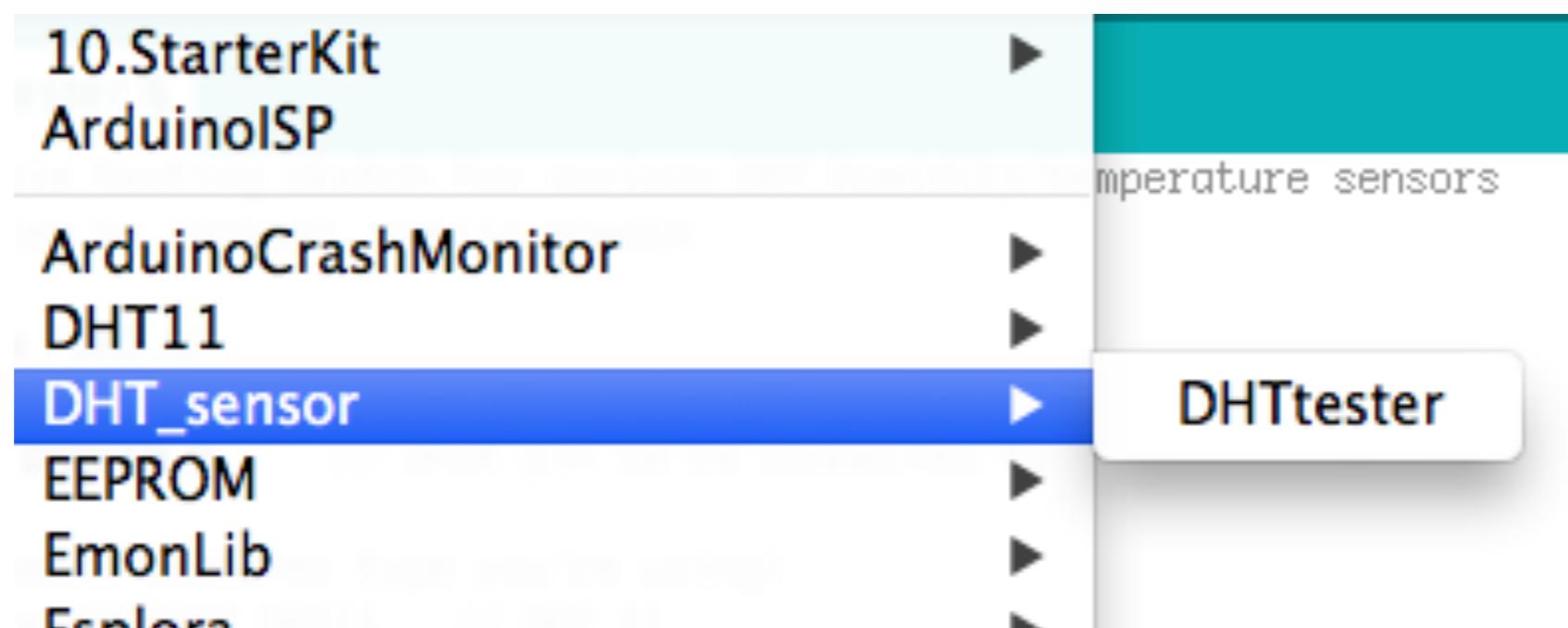
4. ทำการเพิ่ม Library โดยเปิดเมนู

sketch -> Import Library.. ->add Library



การเพิ่ม Library ของอุปกรณ์(4)

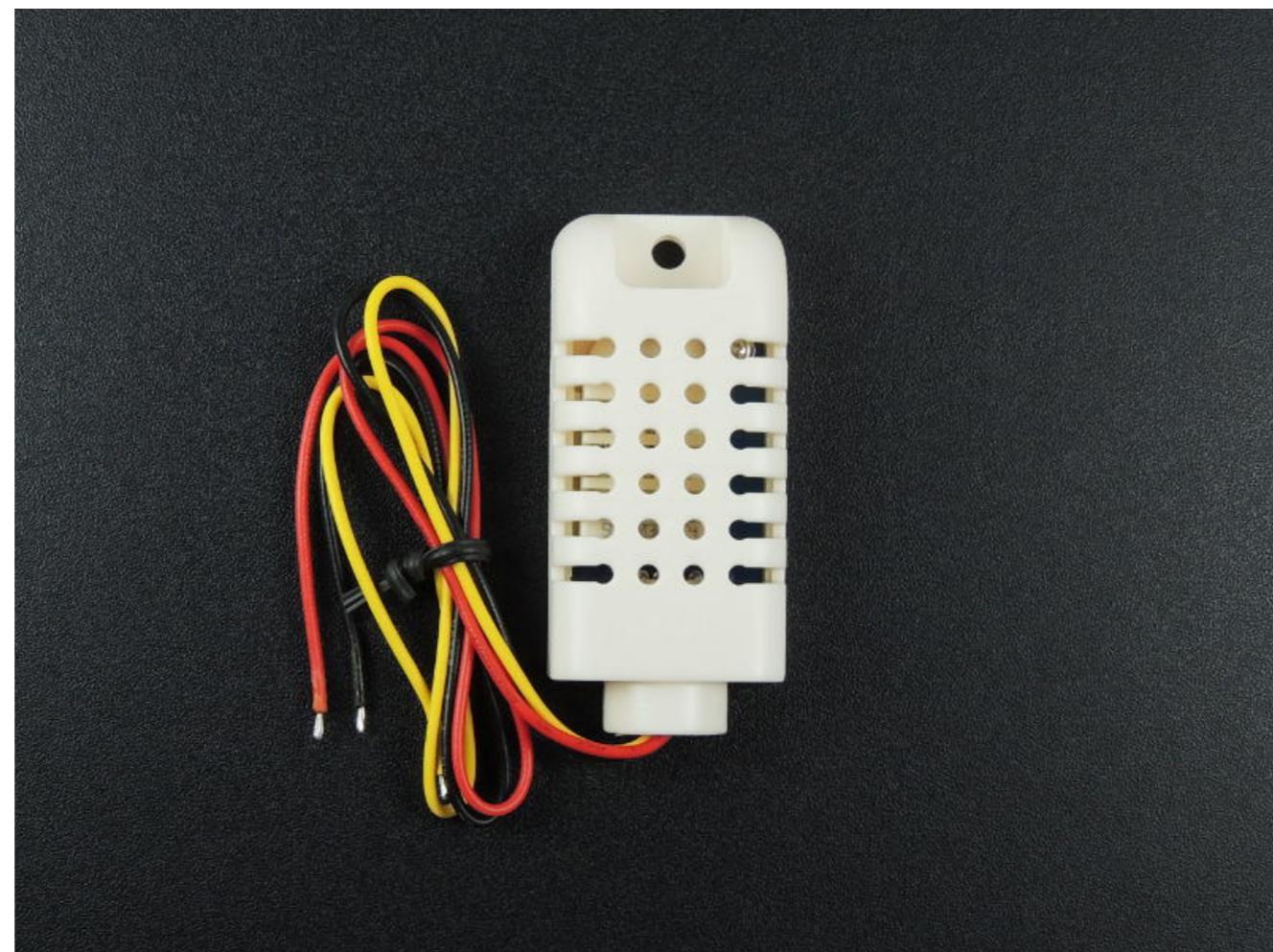
4. ทำการ restart โปรแกรม Arduino IDE และเปิด File -> Examples -> dht_sensor ถ้าว่าเพิ่ม Library อุปกรณ์สำเร็จ



LAB4

Digital Temperature Humidity Sensor

วัดอุณหภูมิและความชื้นสัมพัทธ์แบบดิจิทัล
(Digital Temperature Humidity Sensor)

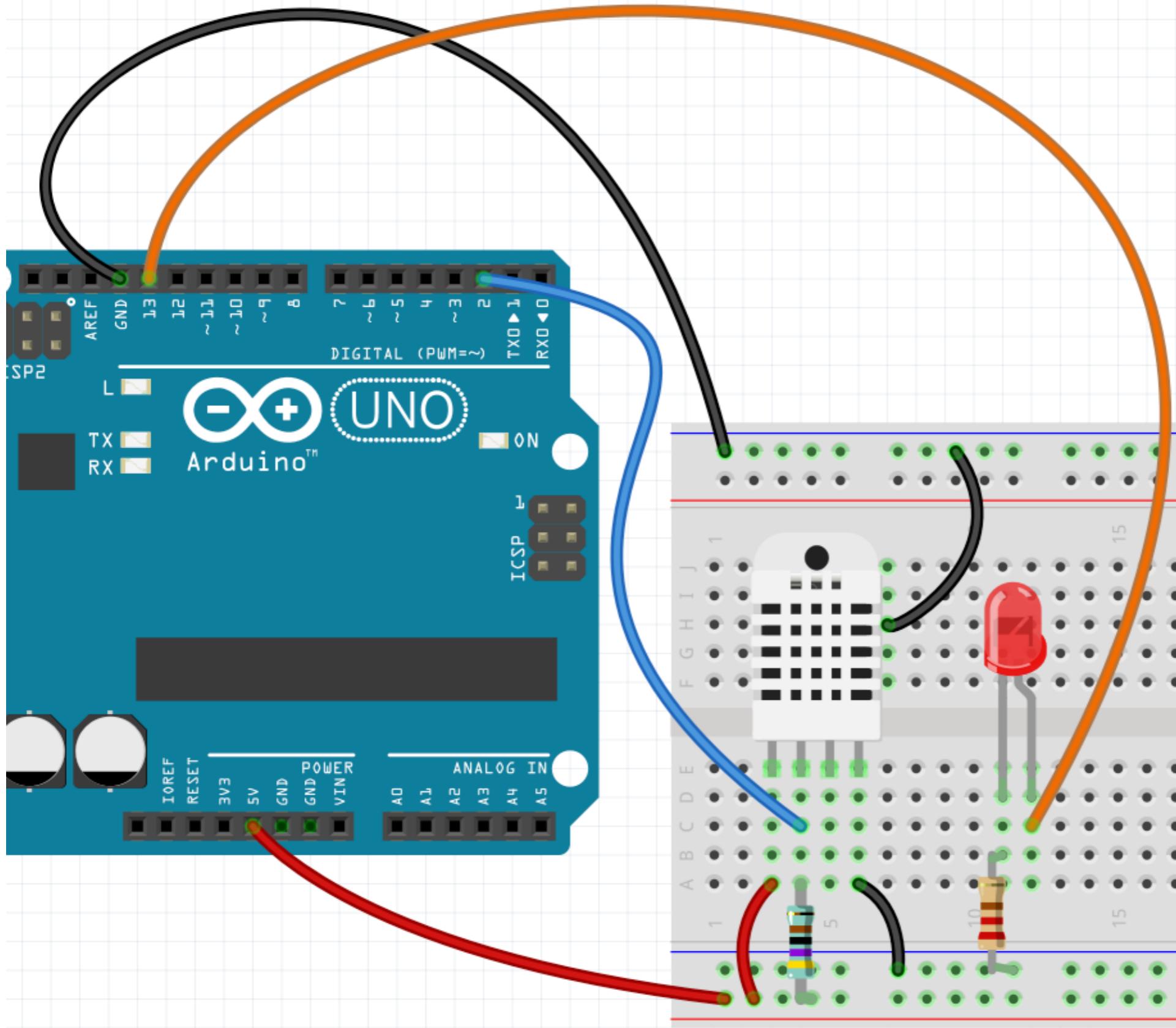


WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR

เตรียมอุปกรณ์

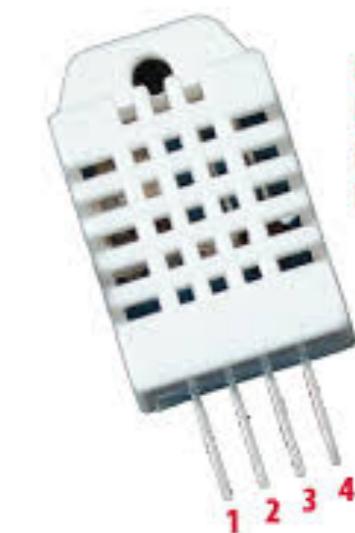
- | | |
|---------------------|---|
| 1. Arduino Uno R3 | 1 |
| 2. สาย USB | 1 |
| 3. DHT22 | 1 |
| 4. หลอด LED | 1 |
| 5. ตัวต้านทาน 330 Ω | 1 |
| 6. ตัวต้านทาน 47k Ω | 1 |
| 7. photo board | 1 |
| 8. สาย M to M | 7 |

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR



DHT22 Pinout

- Pin 1: VCC (3V to 5.5V)
- Pin 2: Data
- Pin 3: Not Connected
- Pin 4: Ground



รูปการต่อวงจร Arduino Digital Temperature Humidity Sensor Ex I

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EXI

TEMPERATURE HUMIDITY SENSOR EXI(I)

```
#include "DHT.h" //เรียกใช้ library DHT.h  
#define DHTPIN 2 // เลือกขา Di 2  
#define DHTTYPE DHT22 // DHT 22 (AM2302)  
DHT dht(DHTPIN, DHTTYPE); //ใช้ function dht  
const int led = 13; //ระบุหลอด LED pin 13  
  
void setup() {  
    pinMode(led, OUTPUT); //ให้ค่า led เป็น OUTPUT  
    Serial.begin(9600);  
    Serial.println("DHTxx test!");  
    dht.begin(); //เริ่มเรียกการใช้งาน dht จาก library  
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EXI TEMPERATURE HUMIDITY SENSOR EXI(2)

```
void loop() {  
  
    // Reading temperature or humidity takes about 250 milliseconds!  
  
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)  
  
    float h = dht.readHumidity();  
  
    float t = dht.readTemperature();  
  
    // check if returns are valid, if they are NaN (not a number) then something went wrong!  
  
    if (isnan(t) || isnan(h)) {  
  
        Serial.println("Failed to read from DHT");  
  
    }  
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EXI

TEMPERATURE HUMIDITY SENSOR EXI(3)

```
else {
```

```
    Serial.print("Humidity: "); //พิมพ์ประโยค Humidity:
```

```
    Serial.print(h); //พิมพ์ค่าความชื้นสัมพัทธ์ในอากาศ
```

```
    Serial.print(" % \t"); //พิมพ์ประโยค % และทำการ tab
```

```
    Serial.print("Temperature: "); //พิมพ์ประโยค Temperature:
```

```
    Serial.print(t); //พิมพ์ค่าอุณหภูมิ
```

```
    Serial.println(" *C"); ///พิมพ์ประโยค *C และทำการขึ้นบรรทัดใหม่
```

```
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EXI

TEMPERATURE HUMIDITY SENSOR EXI(4)

```
if (t>=32) //ถ้าอุณหภูมิมากกว่าหรือเท่ากับ 32
```

```
{
```

```
    digitalWrite(led, HIGH); //หลอด LED ติด
```

```
}
```

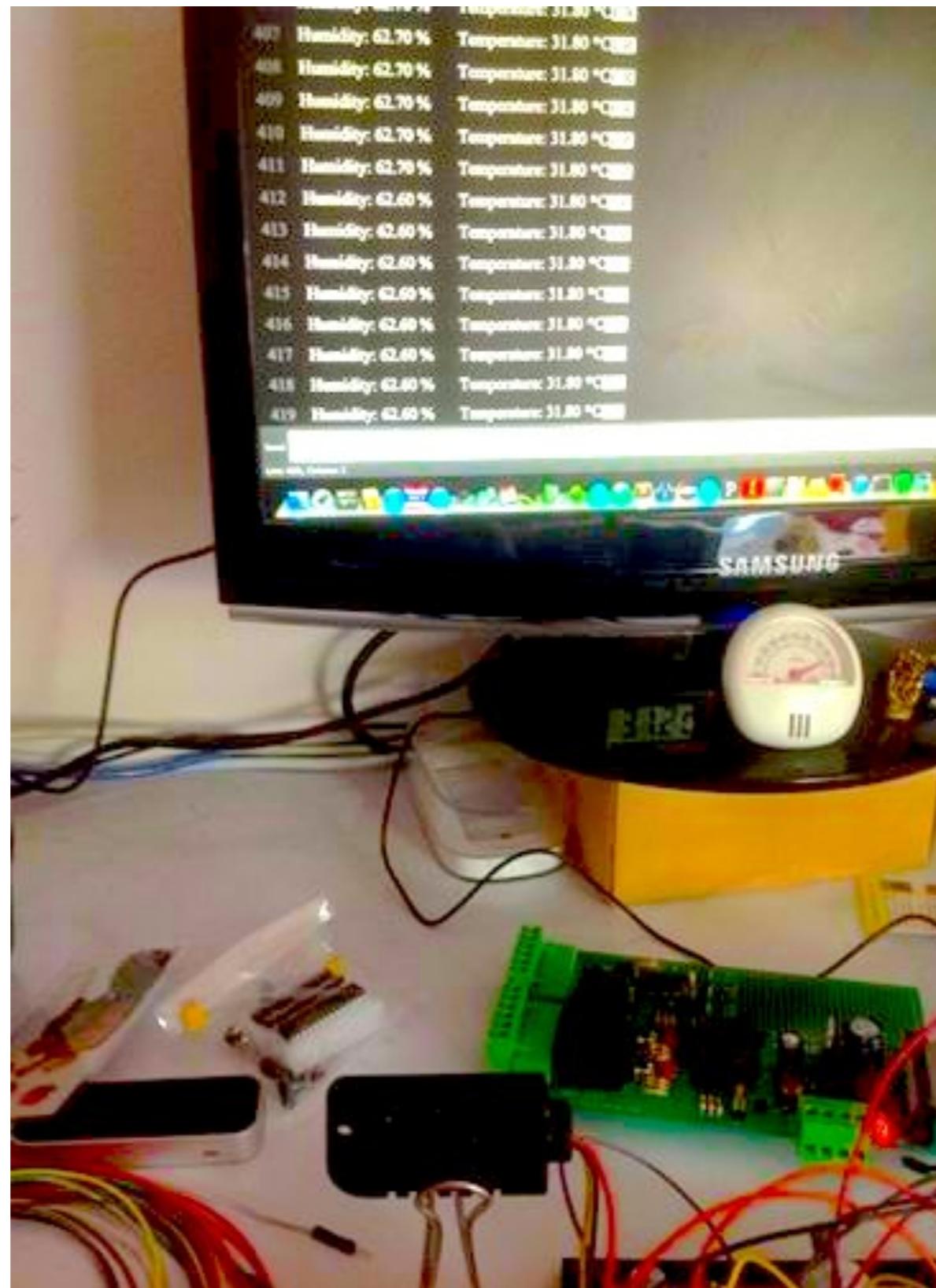
```
else{
```

```
    digitalWrite(led, LOW); //หลอด LED ดับ
```

```
}
```

```
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR



ผลการทดลอง Arduino Digital Temperature Humidity Sensor Ex I

WORKSHOP ARDUINO DIGITAL TEMPERATURE HUMIDITY SENSOR EXI

1. ต่อวงจรตามในภาพ
2. เขียนโค้ดของ Arduino IDE code Digital Temperature Humidity Sensor
3. verify,upload
4. ทดลองเปิด Serial Monitor และสังเกตุค่า ว่าถ้าอุณหภูมิมากกว่าหรือเท่ากับ 32 C หลอดไฟ LED ติดหรือไม่
5. ทดลองแก้ code ใหม่ ให้รับค่าความชื้นสัมพันธ์น้อยกว่า หรือเท่ากับ 60 ให้หลอดไฟหลอดไฟ LED ติดถ้าไม่ใช่ให้หลอดดับ

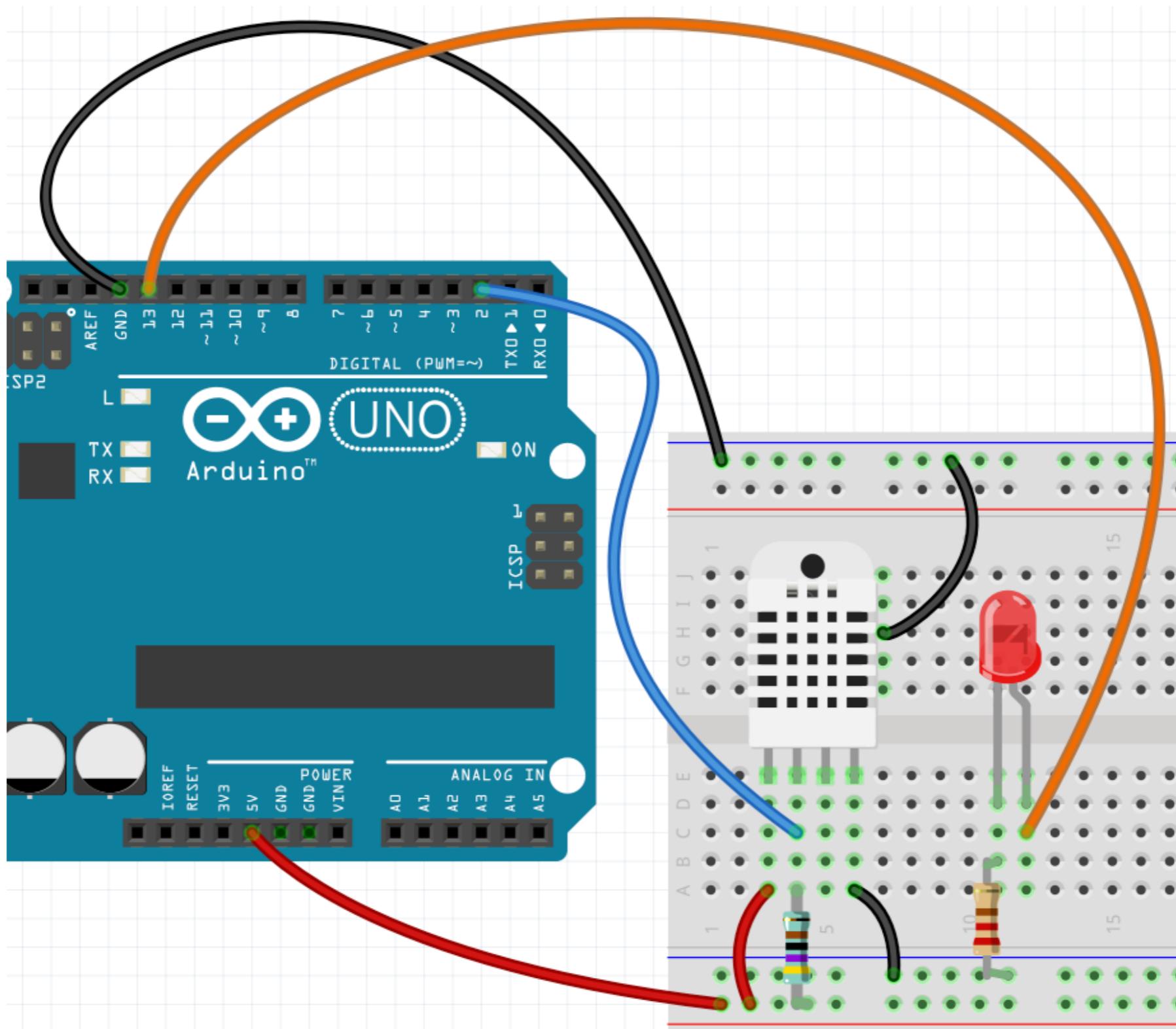
LAB5

Data Logger

Data Logger (เก็บข้อมูลจากอุปกรณ์)



WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EX2



รูปการต่อวงจร Arduino Digital Temperature Humidity Sensor Ex2

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EX2

TEMPERATURE HUMIDITY SENSOR EX2(I)

```
#include "DHT.h" //เรียกใช้ library DHT.h  
#define DHTPIN 2 // เลือกขา Di 2  
#define DHTTYPE DHT22 // DHT 22 (AM2302)  
DHT dht(DHTPIN, DHTTYPE); //ใช้ function dht  
const int led = 13; //ระบุหลอด LED pin 13  
  
void setup() {  
    pinMode(led, OUTPUT); //ให้ค่า led เป็น OUTPUT  
    Serial.begin(9600);  
    Serial.println("DHTxx test!");  
    dht.begin(); //เริ่มเรียกการใช้งาน dht จาก library  
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EX2

TEMPERATURE HUMIDITY SENSOR EX2(2)

```
void loop() {  
  
    // Reading temperature or humidity takes about 250 milliseconds!  
  
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)  
  
    float h = dht.readHumidity();  
  
    float t = dht.readTemperature();  
  
    // check if returns are valid, if they are NaN (not a number) then something went wrong!  
  
    if (isnan(t) || isnan(h)) {  
  
        Serial.println("Failed to read from DHT");  
  
    }  
}
```

WORKSHOP DIGITAL TEMPERATURE HUMIDITY SENSOR EX2

TEMPERATURE HUMIDITY SENSOR EX2(3)

```
else {
```

```
    Serial.print(h); //พิมพ์ค่าความชื้นสัมพัทธ์ในอากาศ
```

```
    Serial.print(" % \t"); //พิมพ์ประโยชน์ % แล้วทำการ tab
```

```
    Serial.print(t); //พิมพ์ค่าอุณหภูมิ
```

```
}
```

```
}
```

WORKSHOP ARDUINO DIGITAL TEMPERATURE HUMIDITY SENSOR EXI

1. ต่อวงจรตามในภาพ
2. เขียนโค้ดของ Arduino IDE code Digital Temperature Humidity Sensor
3. verify,upload
4. ทดลองเปิด Serial Monitor และสังเกตุค่า อุณหภูมิ
5. ทำWork Shop การอธิบายงาน

TEMPERATURE HUMIDITY SENSOR REPORT(I)

I. คัดลอกค่าจาก serial port ในการนี้เป็นแบบด้านขวาคือ sensor ทำงานผิดพลาด

1	DHTxx test!
2	33.90 29.70
3	33.90 29.70
4	33.90 29.70
5	33.90 29.70
6	33.90 29.70
7	34.00 29.70
8	34.00 29.70
9	33.90 29.70
0	33.90 29.70
1	33.90 29.70
2	33.90 29.70
3	33.90 29.70
4	33.90 29.70
5	33.90 29.70
6	33.80 29.70
7	33.80 29.70
8	33.80 29.70

33.50 29.70
33.40 29.70
33.40 29.70
33.40 29.70
33.40 29.70
Read failFailed to read from DHT
0.00 0.00
0.00 0.00
0.00 0.00

TEMPERATURE HUMIDITY SENSOR REPORT(2)

2. วางแผนในโปรแกรม spread sheet

The screenshot shows a spreadsheet application window with the following details:

- Toolbar:** Includes icons for View (125% zoom), Zoom, Function, Table, Chart, Text, and other standard spreadsheet tools.
- Sheet:** Labeled "Sheet 1".
- Columns:** A, B, C, D, E.
- Rows:** 1 through 21.
- Data:** The data is organized into two columns: Temperature (B) and Humidity (C). The first 19 rows show values of 33.90 and 29.70 respectively. Rows 20 and 21 show values of 33.80 and 29.70 respectively. Row 10 has a yellow dot at the end of the cell in column C, indicating it is the current active cell.
- Caption:** "Table 1" is displayed above the grid.

	A	B	C	D	E
1					
2		33.90	29.70		
3		33.90	29.70		
4		33.90	29.70		
5		33.90	29.70		
6		33.90	29.70		
7		34.00	29.70		
8		34.00	29.70		
9		33.90	29.70		
10		33.90	29.70		
11		33.90	29.70		
12		33.90	29.70		
13		33.90	29.70		
14		33.90	29.70		
15		33.90	29.70		
16		33.80	29.70		
17		33.80	29.70		
18		33.80	29.70		
19		33.80	29.70		
20					
21					

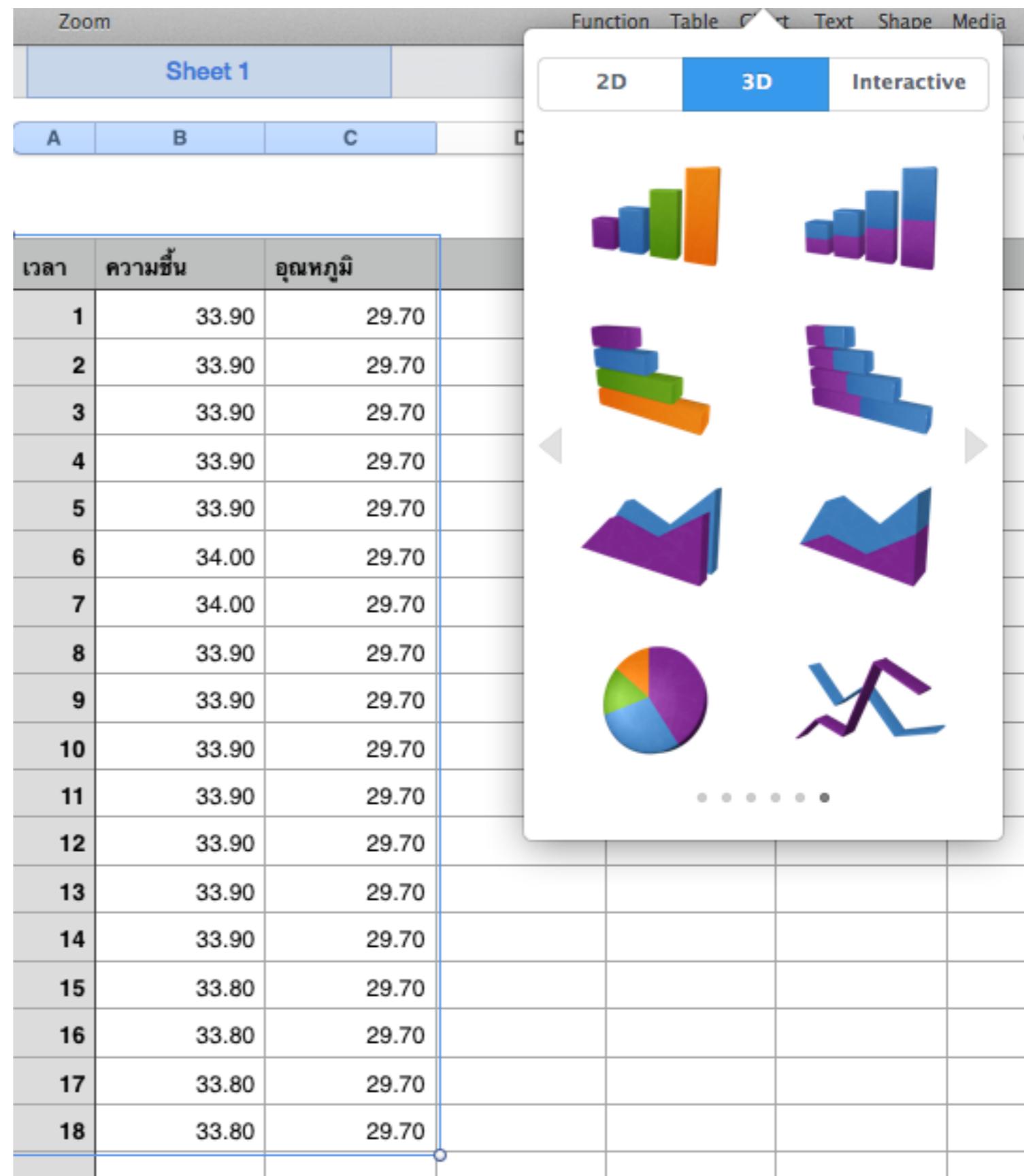
TEMPERATURE HUMIDITY SENSOR REPORT(3)

3. ทำการสร้างตัวเลขของแต่ละช่อง
และชนิดค่าจาก sensor

เวลา	ความชื้น	อุณหภูมิ
1	33.90	29.70
2	33.90	29.70
3	33.90	29.70
4	33.90	29.70
5	33.90	29.70
6	34.00	29.70
7	34.00	29.70
8	33.90	29.70

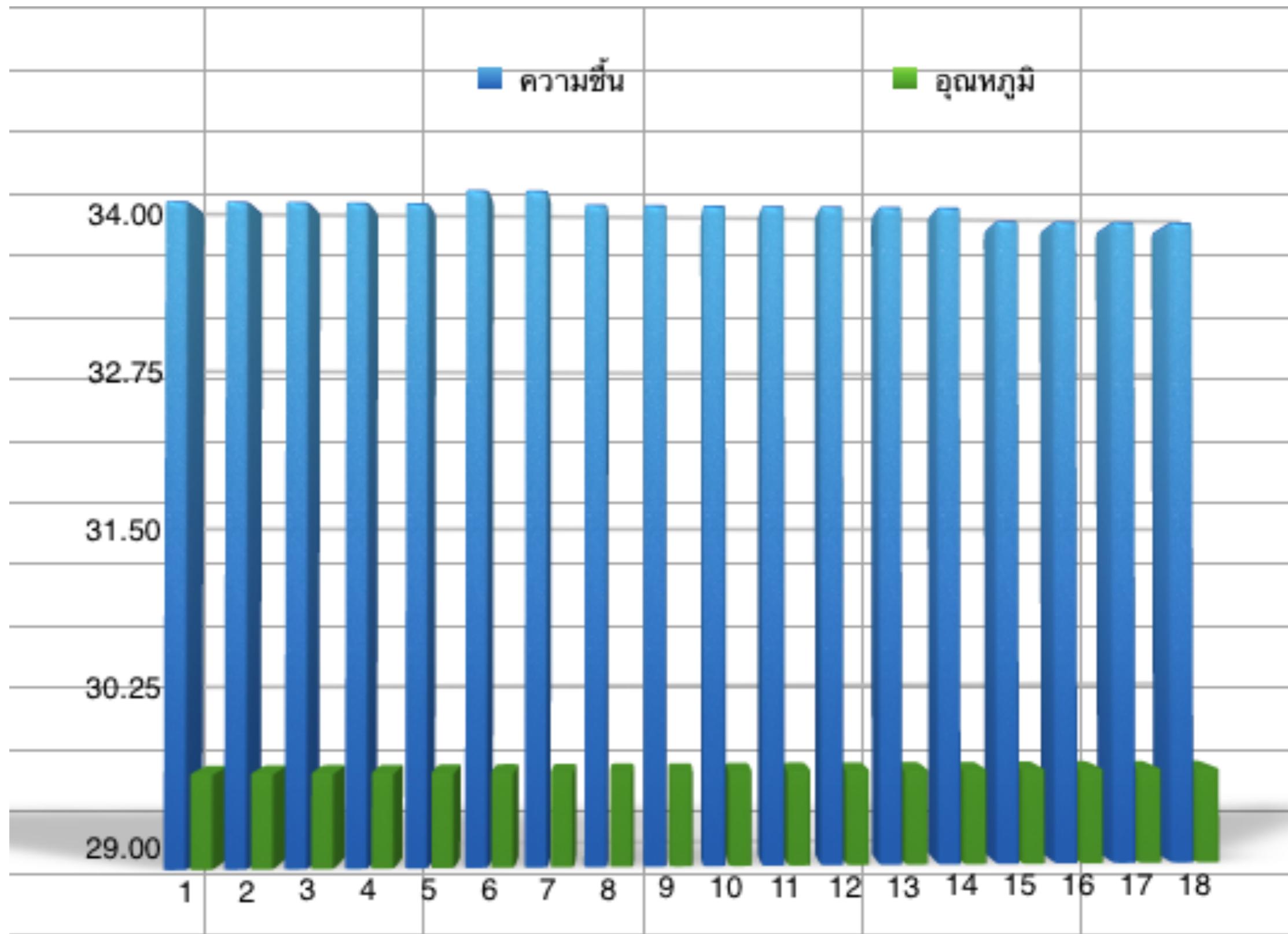
TEMPERATURE HUMIDITY SENSOR REPORT(4)

4. ทำการเลือกข้อมูลทั้งหมด
และเลือก chart



TEMPERATURE HUMIDITY SENSOR REPORT(4)

5. ผลที่ได้จากการทำ chart



Q & A