

# สถาปัตยกรรม ARDUINO

## ARDUINO คืออะไร

Arduino เริ่มต้นโครงการในปี 2005 โดยเป็นโครงการสำหรับนักศึกษาที่ The Design Institute of Ivrea, Italy. โดยพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย มีซอฟต์แวร์สำหรับการเขียนโปรแกรม แพลตฟอร์ม และดาวน์โหลดรหัสลงชิพ ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษาเกี่ยวกับระบบไมโครคอนโทรลเลอร์ อีกทั้งผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมได้

แรกเริ่มนั้น Arduino พัฒนามาบนไมโครคอนโทรลเลอร์ตระกูล AVR เมื่อได้รับความนิยมมากขึ้นจึงมีบริษัทต่างๆ ทำบอร์ดที่เทียบเท่ากับ Arduino แต่ใช้ชิพยี่ห้อต่างกัน แล้วเขียนไลบรารีของ Arduino รองรับ ทำให้สามารถใช้ IDE และใช้งานโค้ดตัวเดียวกันกับ Arduino ได้

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 1) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเปรียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

## ARDUINO IS A PLATFORM (ฐาน)

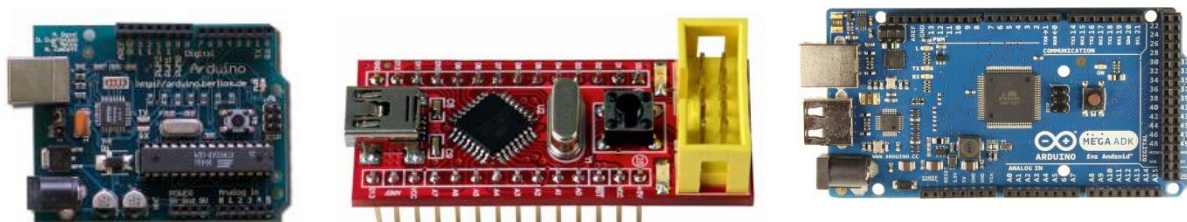
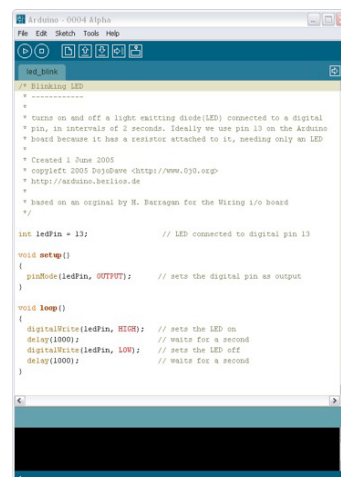
A physical Input / Output board (I/O) with a programmable Integrated Circuit (IC).

Also including an Integrated Development Environment (IDE) for programming.

The language itself is based in C

but is largely modeled upon the

[www.processing.org](http://www.processing.org)



## ใช้ทำอะไร

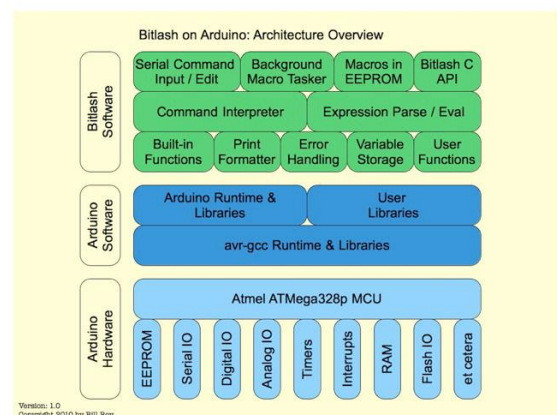
- Physical Computing projects / research
- Interactive Installations
- Rapid prototyping
- Sensors ( to sense stuff )
  - Push buttons, touch pads, tilt switches.
  - Variable resistors (eg. volume knob / sliders)
  - Photoresistors (sensing light levels)
  - Thermistors (temperature)
  - Ultrasound (proximity range finder)
- Actuators ( to do stuff )
  - Lights, LED's
  - Motors
  - Speakers
  - Displays (LCD)

## ทำไมควรใช้ ARDUINO

- เป็น Open Source ทั้ง Hardware และ Software.
- สื่อสารกับคอมพิวเตอร์ได้โดยทางพอร์ทอนุกรมผ่าน USB (รุ่น Bluetooth เร็วขึ้น).
- สามารถใช้แหล่งจ่ายได้ทั้งทาง USB และ แหล่งจ่าย DC
- สามารถทำงานได้ด้วยตัวเอง และมีหน่วยความจำขนาดเล็ก
- ทำงานได้กับสัญญาณอนาล็อกและสัญญาณดิจิทัล
- มีตัวอย่างและแหล่งข้อมูลมากมาย
- มีหลายรุ่น หลายราคา บางรุ่นราคาถูกมากๆ

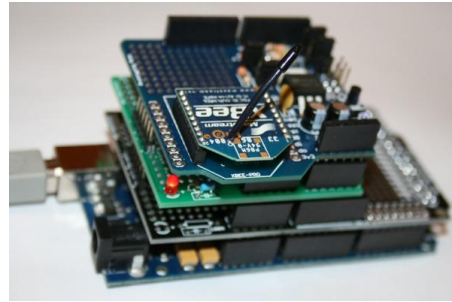
## ตัวอย่าง สถาปัตยกรรม ARDUINO

Arduino board ประกอบด้วย ไมโครคอนโทรลเลอร์ ขนาด 8 บิต ตระกูล AVR ของ Atmel มีอุปกรณ์สำหรับการ อัปโหลดโปรแกรมลงชิพ มีมาตรฐานคอนเนคเตอร์เพื่อการต่อวงจรอื่นเพิ่มเติม วงจรอื่นๆที่ต่อเพิ่มนี้เรียกว่า shields.



## ARDUINO SHIELDS

Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce.



## ATMEGA168-ARDUINO

Digital I/O 14 ขา เรียกว่า D0 – D13

Analog I/P 6 ขา เรียกว่า A0 – A5

PWM 4 ช่อง

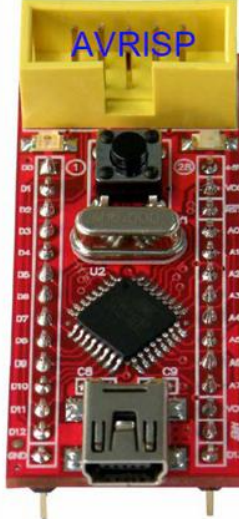
Serial port 1 ช่อง

### Atmega168 Pin Mapping

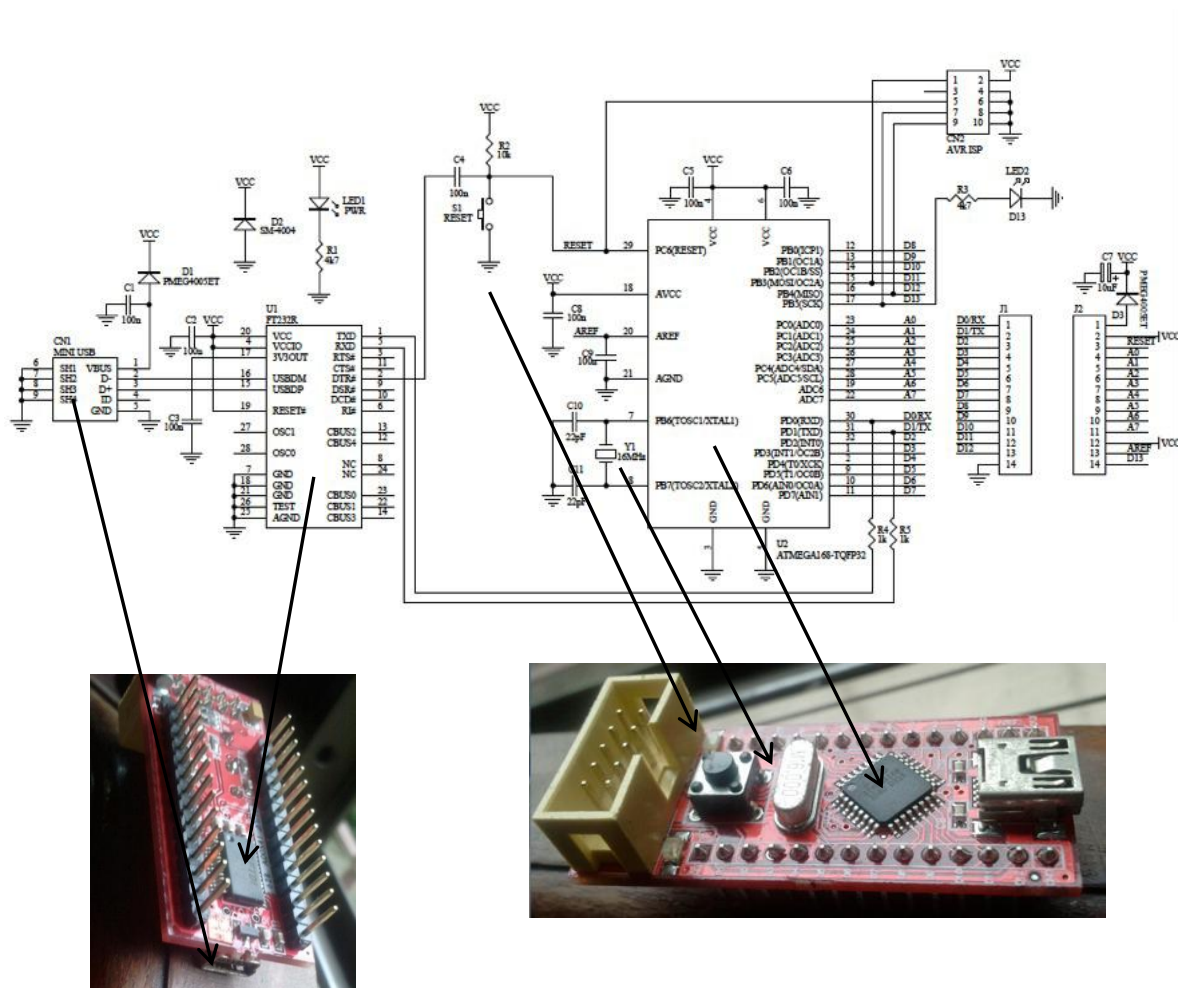
| Arduino function    |                          |    | Arduino function |                        |                      |
|---------------------|--------------------------|----|------------------|------------------------|----------------------|
| reset               | (PCINT14/RESET) PC6      | 1  | 28               | PC5 (ADC5/SCL/PCINT13) | analog input 5       |
| digital pin 0 (RX)  | (PCINT16/RXD) PD0        | 2  | 27               | PC4 (ADC4/SDA/PCINT12) | analog input 4       |
| digital pin 1 (TX)  | (PCINT17/TXD) PD1        | 3  | 26               | PC3 (ADC3/PCINT11)     | analog input 3       |
| digital pin 2       | (PCINT18/INT0) PD2       | 4  | 25               | PC2 (ADC2/PCINT10)     | analog input 2       |
| digital pin 3 (PWM) | (PCINT19/OC2B/INT1) PD3  | 5  | 24               | PC1 (ADC1/PCINT9)      | analog input 1       |
| digital pin 4       | (PCINT20/XCK/T0) PD4     | 6  | 23               | PC0 (ADC0/PCINT8)      | analog input 0       |
| VCC                 | VCC                      | 7  | 22               | GND                    | GND                  |
| GND                 | GND                      | 8  | 21               | AREF                   | analog reference     |
| crystal             | (PCINT6/XTAL1/TOSC1) PB6 | 9  | 20               | AVCC                   | VCC                  |
| crystal             | (PCINT7/XTAL2/TOSC2) PB7 | 10 | 19               | PB5 (SCK/PCINT5)       | digital pin 13       |
| digital pin 5 (PWM) | (PCINT21/OC0B/T1) PD5    | 11 | 18               | PB4 (MISO/PCINT4)      | digital pin 12       |
| digital pin 6 (PWM) | (PCINT22/OC0A/AIN0) PD6  | 12 | 17               | PB3 (MOSI/OC2A/PCINT3) | digital pin 11 (PWM) |
| digital pin 7       | (PCINT23/AIN1) PD7       | 13 | 16               | PB2 (SS/OC1B/PCINT2)   | digital pin 10 (PWM) |
| digital pin 8       | (PCINT0/CLKO/ICP1) PB0   | 14 | 15               | PB1 (OC1A/PCINT1)      | digital pin 9 (PWM)  |

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

## ET-EASY168 STAMP PIN

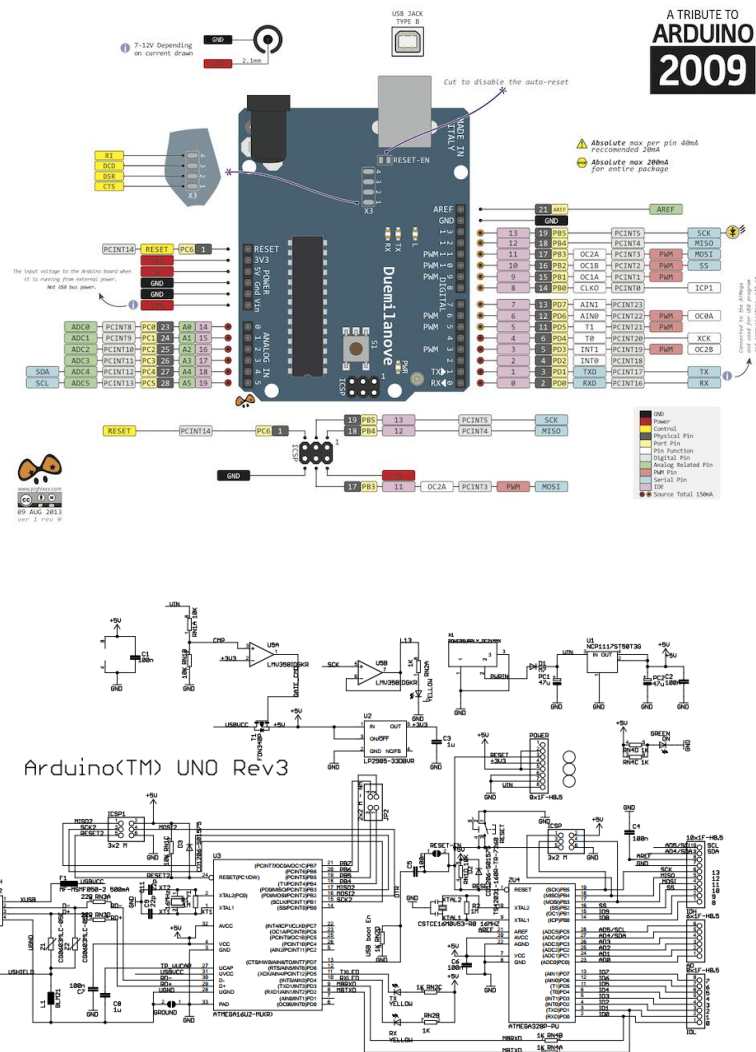
| AVR | Arduino    | Pin | ET-EASY168 STAMP                                                                  | Pin | Arduino    | AVR        |
|-----|------------|-----|-----------------------------------------------------------------------------------|-----|------------|------------|
| PD0 | Digital-0  | 1   |  | 28  | +5V(+Vin)  | +5V(+Vin)  |
| PD1 | Digital-1  | 2   |                                                                                   | 27  | +VCC(+5V)  | +VCC(+5V)  |
| PD2 | Digital-2  | 3   |                                                                                   | 26  | RESET#     | RESET(PC6) |
| PD3 | Digital-3  | 4   |                                                                                   | 25  | Analog-0   | PC0/ADC0   |
| PD4 | Digital-4  | 5   |                                                                                   | 24  | Analog-1   | PC1/ADC1   |
| PD5 | Digital-5  | 6   |                                                                                   | 23  | Analog-2   | PC2/ADC2   |
| PD6 | Digital-6  | 7   |                                                                                   | 22  | Analog-3   | PC3/ADC3   |
| PD7 | Digital-7  | 8   |                                                                                   | 21  | Analog-4   | PC4/ADC4   |
| PB0 | Digital-8  | 9   |                                                                                   | 20  | Analog-5   | PC5/ADC5   |
| PB1 | Digital-9  | 10  |                                                                                   | 19  | Analog-6   | ADC6       |
| PB2 | Digital-10 | 11  |                                                                                   | 18  | Analog-7   | ADC7       |
| PB3 | Digital-11 | 12  |                                                                                   | 17  | +VCC(+5V)  | +VCC(+5V)  |
| PB4 | Digital-12 | 13  |                                                                                   | 16  | +AREF      | +AREF      |
| GND | GND        | 14  |                                                                                   | 15  | Digital-13 | PB5        |

## ET-EASY168 STAMP CIRCUIT



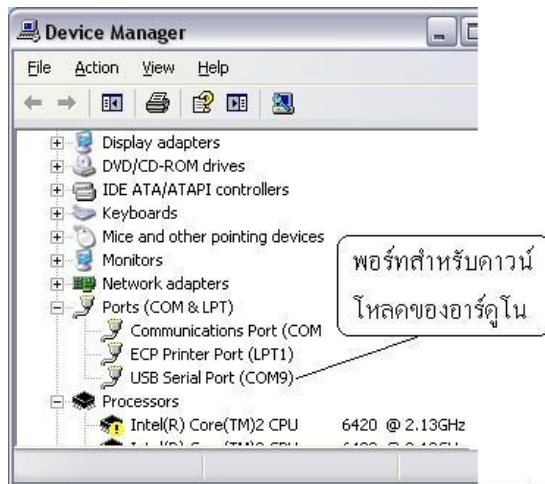
USB TO SERIAL CONVERTER

## Arduino UNO



## ติดตั้งโปรแกรม

- 1) ติดตั้งโปรแกรม คำนวณโหลดโปรแกรมอาร์ดูโน จาก <http://arduino.cc/> ในเมนู Download ซึ่งมีให้เลือกทั้ง Windows Mac OS X และ Linux: 32 bit, 64 bit ในขณะนี้ ใช้บน Windows และเป็นเวอร์ชัน arduino-1.0-windows.zip นำมาขยาย (Unzip) แล้วคัดลอกทั้งโฟลเดอร์ไปติดตั้งที่ C:\Program Files\ (สามารถติดตั้งที่ไหนก็ได้) และเพื่อความสะดวกของการเรียกใช้ให้ทำ Short cut เอาไว้จะได้ไม่ลืม
- 2) การติดตั้ง USB to Serial เมื่อต่อสายค่านวณโหลดจากบอร์ดเข้ากับพอร์ท USB เครื่องจะถามหาไดรเวอร์ ให้เลือกจากโฟลเดอร์ drivers ที่อยู่ในโฟลเดอร์โปรแกรม เช่นเมื่อติดตั้งไว้ที่ C:\Program Files\ ก็จะอยู่ที่ C:\Program Files\arduino-1.0\drivers เมื่อติดตั้งเสร็จสามารถตรวจสอบได้จาก Device Manager เช่น



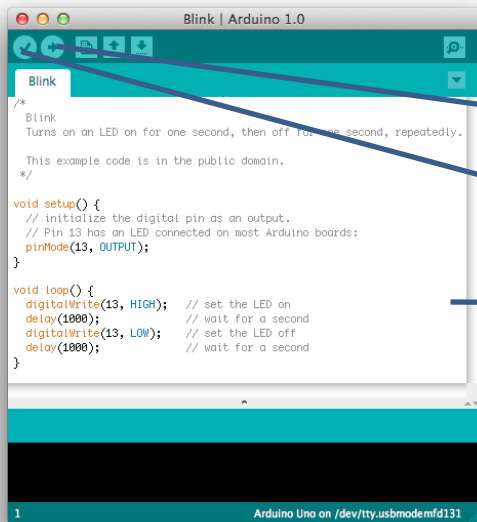
- 3) การเลือกพอร์ต รันโปรแกรม Arduno ใช้เมนู Tools เลือก Serial Port แล้วคลิกเลือกพอร์ต COM ที่ตรวจสอบได้จากข้อ 2
- 4) การเลือกบอร์ด จาก เมนู Tools ในโปรแกรม Arduino เลือก Board สำหรับบอร์ดที่ทดลองให้คลิกที่ Arduino Nano W/ ATmega168
- 5) การกำหนด Programmer ให้เลือก AVRISP mkII



- 6) การรันโปรแกรม ใช้เมนู File เลือก Upload Using Programmer หรือใช้  ขั้นตอนนี้อาร์ดูโน จะทำการแปลโปรแกรม แล้วดาวน์โหลดโปรแกรมลงบอร์ดทดลอง สุดท้าย บอร์ดทดลองก็รันโปรแกรม ถ้าไม่ทำให้ลองกดสวิทช์เซตของบอร์ดทดลอง

ข้อควรจำ เนื่องจากขา Digital 0 (PD0) เป็นขา RxD และ Digital 1 (PD1) เป็นขา TxD ของพอร์ตอนุกรม UART ซึ่ง Arduino ใช้เป็นพอร์ตดาวน์โหลดโปรแกรมลงชิพ AVR ดังนั้นขณะดาวน์โหลดโปรแกรมนี้ ห้ามต่อขา D0 และ D1 เข้ากับวงจรใดๆ

## หน้าต่าง IDE ของ Arduino



ตรวจสอบ แปลและUploadลงชิพ

ตรวจสอบและแปล

เนื้อที่เขียนโปรแกรม

หน้าต่าง console

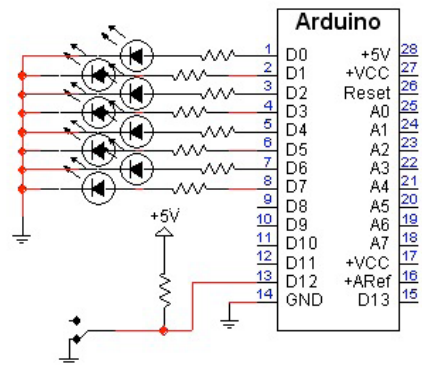
## โครงสร้างของโปรแกรม

การกำหนดค่าเบื้องต้นๆ

```
void setup() {
// put your setup code here, to run once:
}

void loop() {
// put your main code here, to run repeatedly:
}

..... Function(.....) {
}
```



### ตัวอย่างที่ 1 BLINK

```
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most
Arduino boards:
pinMode(13, OUTPUT);
}

void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
```

### ตัวอย่างที่ 2 ไฟวิ่ง

```
char flash;
void setup() {
DDRD = B11111111; //Port D as output
}

void loop() {
Flash = B00000001;
for(int i = 0;i<8;i++){
PORTD = flash;
delay(300);
flash<<=1;
}
}
```

ตัวอย่างที่ 2 มีการใช้คำสั่งของ AVR โดยตรง คือ DDRD และ PORTD