

## 4. Liquid Crystal Display



### 1. Liquid Crystal Display : LCD



รายละเอียดต่างๆ

1. GND ต่อลงกราวด์
2. VDD ต่อไฟเลี้ยง 5 Volt
3. Contrast ทำหน้าที่ปรับความเข้มของจอ
4. RS (Register select)
5. R/W ควบคุม Read/Write
6. EN ควบคุม Enable
7. Data ควบคุมข้อมูลที่ส่งไปแสดงผล Data bit 0 - Data bit 7

## 2. Function ที่ใช้ในการควบคุมจอ LCD

Arduino มี Library มาตรฐานสำหรับการเชื่อมต่อจอ LCD ที่ใช้ chip ของ Hitachi HD44780 ( หรือ chip อื่นๆ ที่ compatible)

โดยมี function หลักๆ ที่อยู่ใน Library **LiquidCrystal.h** ดังนี้

- **LiquidCrystal()** ใช้ในการกำหนดขา LCD ที่ต่อ กับขา Arduino

Syntax

- LiquidCrystal(rs, enable, d4, d5, d6, d7)
- LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)

- **begin()** ใช้ในการกำหนดคอลัมน์ และ แถว ของ LCD
- **clear()** ใช้ในการ clear หน้าจอ LCD ทั้งหมด และ cursor มาอยู่ตำแหน่งเริ่มต้น ตรงแถวบนสุดซ้ายมือ

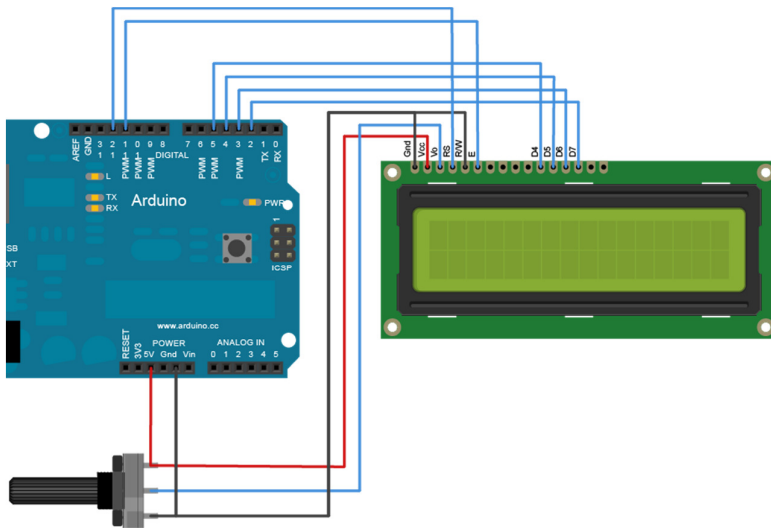
3

- |                           |   |
|---------------------------|---|
| • home()                  | ใช้ในการย้าย cursor มายังแถวบนสุดซ้ายมือ        |
| • setCursor()             | ใช้ในการระบุตำแหน่งของ cursor                   |
| • cursor() , noCursor()   | ใช้กำหนดการแสดง cursor                          |
| • blink(), noBlink()      | ใช้กำหนดการกะพริบของ cursor                     |
| • print()                 | ใช้ในการเขียน <u>ข้อความ</u> ลง LCD             |
| • write()                 | ใช้ในการเขียน <u>ตัวอักษร</u> ลง LCD            |
| • display() , noDisplay() | ใช้ในการควบคุมการปิด-เปิด หน้าจอ                |
| • scrollDisplayLeft()     | ใช้ในการเลื่อนข้อความไปทางซ้าย                  |
| • scrollDisplayRight()    | ใช้ในการเลื่อนข้อความไปทางขวา                   |
| • leftToRight()           | ใช้ในการกำหนดทิศทางการเขียนข้อความ จากซ้ายไปขวา |
| • rightToLeft()           | ใช้ในการกำหนดทิศทางการเขียนข้อความ จากขวาไปซ้าย |
| • createChar()            | ใช้ในการสร้างตัวอักษรตัวใหม่                    |

4

# การทดลองที่ 1. Hello world LCD

- ต่อ LCD เข้ากับบอร์ดทดลอง ดังนี้



**Note!!!**

lcd.setCursor(0, 1);  
set the cursor to column 0, line 1  
line 1 is the second row, since counting begins with 0

## Syntax

LiquidCrystal(rs, enable, d4, d5, d6, d7)

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

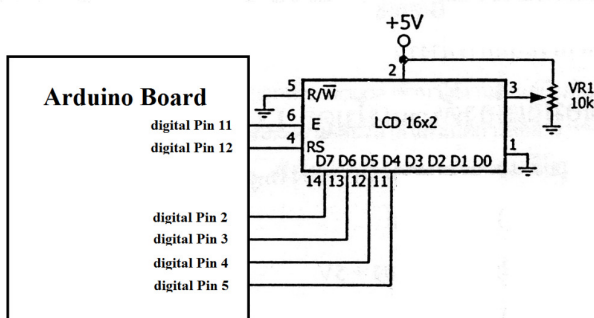
void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis()/1000);
}
```



5

# การทดลองที่ 2. Text Direction

- ต่อ LCD เข้ากับบอร์ดทดลองดังนี้



This program prints  
**a** through **l** right to left,  
then **m** through **r** left to right,  
then **s** through **z** right to left again.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int thisChar = 'a';

void setup() {
  lcd.begin(16, 2);
  lcd.cursor();
}

void loop() {
  if (thisChar == 'm') {
    lcd.rightToLeft();
  }
  if (thisChar == 's') {
    lcd.leftToRight();
  }
  if (thisChar > 'z') {
    lcd.home();
    thisChar = 'a';
  }
  lcd.write(thisChar);
  delay(1000);
  thisChar++;
}
```

6

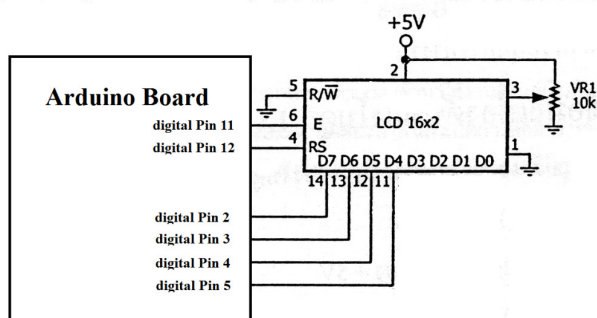
# Create a custom character

- Create a custom character for use on the LCD. Up to eight characters of 5x8 pixels are supported (numbered 0 to 7).
- To display a custom character on the screen, write() its number.

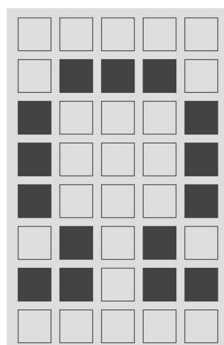


## การทดลองที่ 3. Create custom character

- ต่อ LCD เข้ากับบอร์ดทดลองดังนี้



- โปรแกรมนี้เป็นการ  
สร้างตัวอักษร Ω



```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

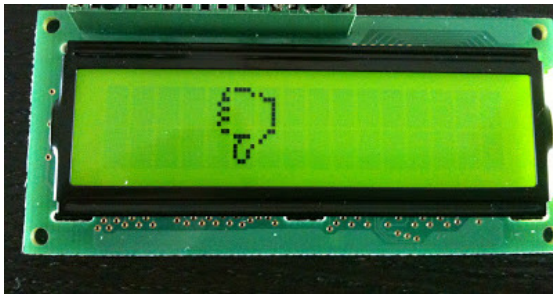
byte newChar[8] = {
    B00000,
    B01110,
    B10001,
    B10001,
    B10001,
    B01010,
    B11011,
    B00000 };

void setup() {
    lcd.createChar(1, newChar);
    lcd.begin(16, 2);
    lcd.write(1); }

void loop() {}
```

## แบบฝึกหัดที่ 1 Custom Font

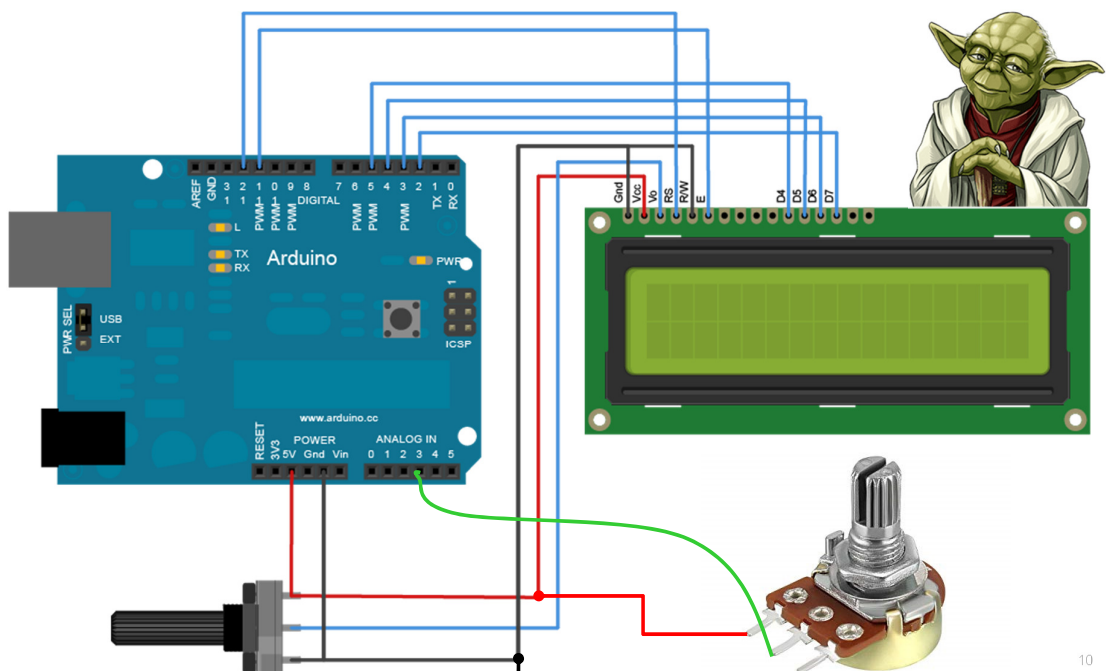
- จงสร้างตัวอักษรรูป ♥ จากนั้นให้แสดงออกจอ LCD โดยที่
  - จอแถวแรก แสดงว่า I ♥ IT และ แสดงตรงกลางแถว
  - จอแถวที่สอง ให้ custom font ใดๆ ก็ได้ ตามที่ออกแบบเอง อีก 1 ตัว



9

## แบบฝึกหัดที่ 2. Arduino Volt Meter

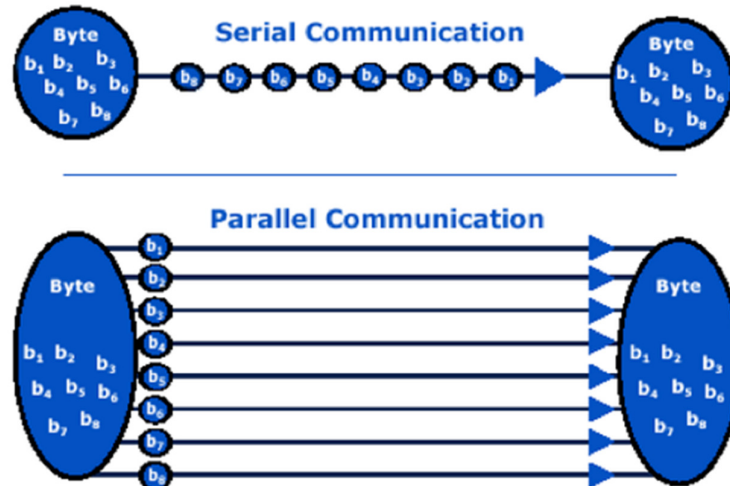
จงเขียนโปรแกรมอ่านค่า แรงดัน จากตัวต้านปรับค่าได้ ไปแสดงผลออกที่จอ LCD แสดงค่าแรงดัน ที่อ่านได้ (0-5V โดยมีทศนิยม 2 ตำแหน่ง)



10



# 5. Serial Communication



## 1. Communicating with other

- Not just for computer-to-Arduino communications
- Many other devices speak serial
- Older keyboards & mice speak serial (good for sensors!)
- Interface boards (graphic LCDs, servo drivers, RFID readers, Ethernet, Wi-Fi)



to Wi-Fi

to Ethernet



to graphic LCD



GPS Module

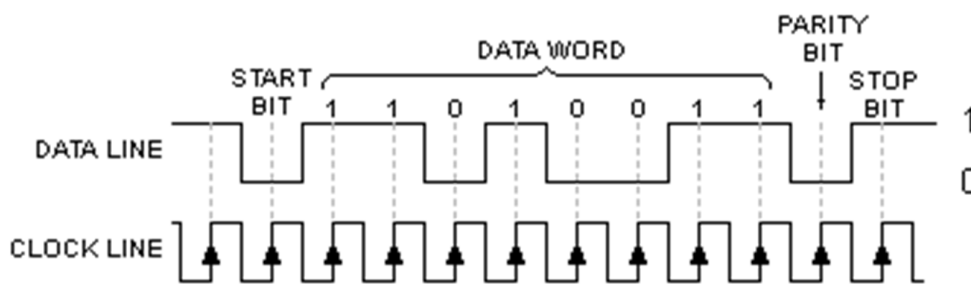
RFID Reader

Serial LCD

Serial Servo Control

## 2. Serial Communication

- การสื่อสารแบบอนุกรม หรือ **Serial** เป็นส่งข้อมูล โดยใช้เทคนิคการเลื่อนข้อมูล (**Shift Bit**) ส่งไปที่ละบิต บนสายสัญญาณเส้นเดียว โดยการส่งข้อมูลแบบ **Serial** จะไม่มีการ **sync** สัญญาณนาฬิการะหว่างตัวรับและตัวส่ง แต่จะอาศัยวิธี**ตั้งค่าความเร็วในการรับส่งสัญญาณให้เท่ากัน** หรือเรียกว่าตั้งค่า **baud rate** และส่งสัญญาณ **start** และ **stop** เพื่อบอกว่า เป็นส่วนต้นของข้อมูล (**start bit**) หรือ ส่วนท้ายของข้อมูล (**stop bit**) ดังรูป



13

รูปแบบของ ข้อมูลจากที่ส่งผ่าน **Serial** จะมีการเพิ่ม **Start bit** และ **Stop bit** เข้าไปเพิ่มจากข้อมูลเดิม

- บิตเริ่มต้น (**Start bit**) จะมีขนาด 1 บิต จะเป็นลอจิก **LOW**
- บิตข้อมูล (**Data bit**) 8 บิต ข้อมูลที่จะส่ง
- บิตภาวะคู่หรือคี่ (**Parity bit**) มีขนาด 1 บิต ใช้ตรวจสอบข้อมูล ถ้าข้อมูลที่  
ได้รับไม่สมบูรณ์ นำค่ามา **check** กับ **Parity bit** จะได้ค่าไม่ตรงกัน
- บิตหยุด (**Stop bit**) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล จะเป็น  
ลอจิก **HIGH**

“Serial” because data is broken down into bits, each sent one-by-one on a single wire:

‘H’

= 0 1 0 0 1 0 0 0

= L H L L H L L L

= HIGH  
LOW

14

## 3. Arduino Communications

---

- Psst, Arduino doesn't really do USB
- It really is "serial", like old RS-232 serial
- All microcontrollers can do serial
- Not many can do USB
- Serial is easy, USB is hard

15

## 4. Serial command

---

- Talking to other uses the "Serial" command

– <code>Serial.begin()</code>	: prepare to use serial
– <code>Serial.print()</code>	: send data to serial port
– <code>Serial.println()</code>	: send data and newline to serial port
– <code>Serial.read()</code>	: read data from serial port
– <code>Serial.available()</code>	: ready to read
– <code>Serial.flush()</code>	: clear buffer at incoming serial data

- Can talk to not just computers.
- Most things more complex than simple sensors/actuators speak serial.

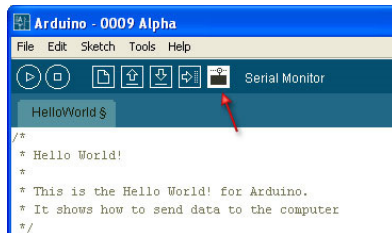
16



# การทดลองที่ 1 Serial Hello world !

ทดลองส่งค่าไปยังคอมพิวเตอร์

- Send "Hello World" to computer and Blink LED
- Click on "Serial Monitor" button to see output



- Watch LED at DigitalPin13

```
int ledPin = 13;
int i = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(19200);
}

void loop() {
  Serial.print(i++);
  Serial.println("Hello world");
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
}
```

17

## Note!! : Serial.print()

- Prints data to the serial port as **human-readable ASCII text**. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is.

For example:

- |                                |                                     |
|--------------------------------|-------------------------------------|
| – Serial.print(78)             | gives "78"                          |
| – Serial.print(1.23456)        | gives "1.23"                        |
| – Serial.print(byte(78))       | gives "N" (whose ASCII value is 78) |
| – Serial.print('N')            | gives "N"                           |
| – Serial.print("Hello world.") | gives "Hello world."                |

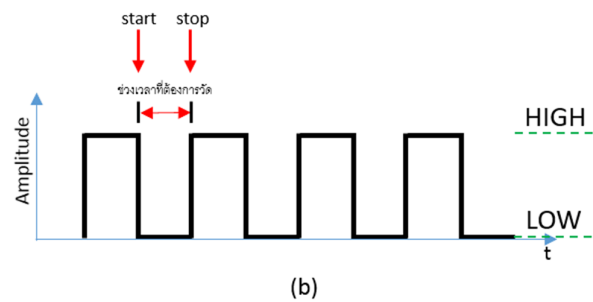
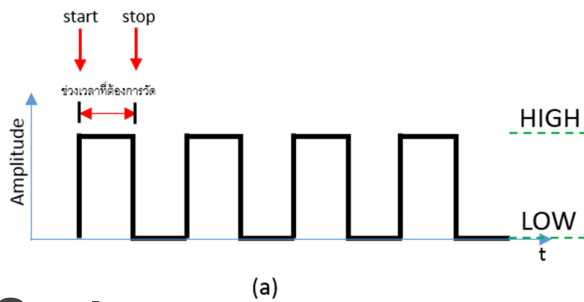
An optional second parameter specifies the base (format) to use

- |                         |                 |                   |
|-------------------------|-----------------|-------------------|
| – Serial.print(78, BIN) | gives "1001110" | BIN : เลขฐานสอง   |
| – Serial.print(78, OCT) | gives "116"     | OCT : เลขฐานแปด   |
| – Serial.print(78, DEC) | gives "78"      | DEC : เลขฐานสิบ   |
| – Serial.print(78, HEX) | gives "4E"      | HEX : เลขฐานสิบหก |

18

# การใช้งานคำสั่ง `pulseIn`

- ใช้วัดความกว้าง Pulse ของสัญญาณ



## Syntax

```
pulseIn(pin, value)  
pulseIn(pin, value, timeout)
```

รูป (a) value = HIGH    รูป (b) value = LOW

timeout คือช่วงเวลาสูงสุดที่ฟังก์ชันนี้ยังทำงานอยู่ ค่า default คือ 1 วินาที หรือ 1,000,000 ไมโครวินาที

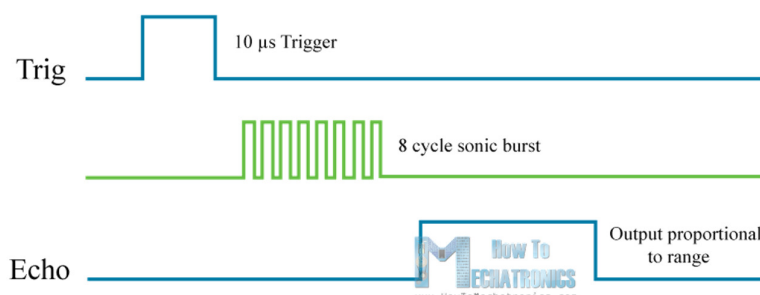
19

## Ultrasonic Module



โมดูล HC-SR04 เป็นอุปกรณ์อิเล็กทรอนิกส์ราคาถูก สำหรับวัดระยะห่างด้วยคลื่นอัลตราโซนิก (ใช้คลื่นเสียงความถี่ ประมาณ 40kHz)

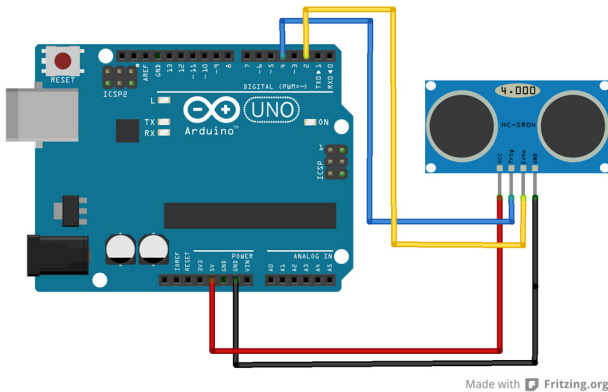
ในการวัดระยะห่างแต่ละครั้ง จะต้องสร้างสัญญาณแบบ Pulse ที่มีความกว้าง (Pulse Width) อย่างน้อย 10  $\mu\text{s}$  ป้อนให้ขา TRIG และหลังจากนั้นให้วัดความกว้างของสัญญาณช่วง HIGH จากขา ECHO ถ้าวัดถูกอยู่ใกล้ ความกว้างของสัญญาณ Pulse ที่ได้ก็จะน้อย แต่ถ้าวัดถูกอยู่ไกลออกไป ก็จะได้ค่าความกว้างของสัญญาณ Pulse ที่มากขึ้น



20

# ตัวอย่าง การใช้งาน pulseIn

โปรแกรมวัดระยะทาง แสดงผลออก Serial monitor



The speed of sound is 340 m/s  
or 29 microseconds per  
centimeter

```
void setup()
{
    Serial.begin(9600);
    pinMode(4, OUTPUT); // 4 = Trig
    pinMode(2, INPUT);  // 2 = Echo
}

void loop()
{
    digitalWrite(4, HIGH);
    delayMicroseconds(10);
    digitalWrite(4, LOW);

    int pulseWidth = pulseIn(2, HIGH);

    Serial.print("Pulse Width: ");
    Serial.println(pulseWidth);
    long distance = pulseWidth/29/2;
    Serial.print("Distance: ");
    Serial.println(distance);
    delay(1000);
}
```

21

## แบบฝึกหัดที่ 1

จากการทดลอง Analog input จงเขียนโปรแกรม อ่านค่าแรงดัน จาก  
การปรับตัวต้านทานปรับค่าได้ แล้วไปแสดงผลที่จอคอมพิวเตอร์



## แบบฝึกหัดที่ 2

จากการทดลอง Temp Sensor จงเขียนโปรแกรมอ่านค่าอุณหภูมิ ที่  
ได้จาก Sensor และแสดงค่าออกจอคอมพิวเตอร์

## การทดลองที่ 2 Serial Read Basic

ทดลองรับค่าจากคอมพิวเตอร์

และส่งค่าออกกลับไปคอมพิวเตอร์

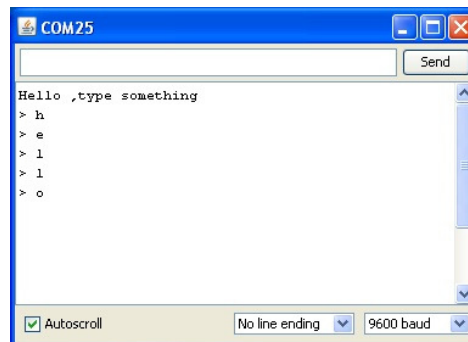
ในโปรแกรม “Serial Monitor”, ให้พิมพ์อะไรก็ได้  
จากนั้น กดปุ่ม Send



### Note !!!

- `Serial.available()` tells you if data present to read.
- Always check `Serial.available()` or if `Serial.read() != -1` to determine if there's actual data to read.

```
char inByte = 0;
void setup() {
  Serial.begin(19200);
  Serial.println("Hello,type something");
}
void loop() {
  if (Serial.available()) {
    inByte = Serial.read();
    Serial.print("> ");
    Serial.println(inByte);
  }
}
```



23

## การทดลองที่ 3 Controlling from computer

ทดลองรับค่าจากคอมพิวเตอร์

- In “Serial Monitor”, You type “H”, Press Send



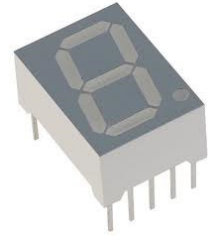
- When you type “H”, LED Blink.

```
int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(19200);
}
void loop(){
  if( Serial.available() ){
    int val = Serial.read();
    if (val == 'H') {
      digitalWrite(ledPin,HIGH);
      delay(500);
      digitalWrite(ledPin,LOW);
      delay(500);
    }
  }
}
```

24

## แบบฝึกหัดที่ 3

- ส่งค่าจากคอมพิวเตอร์เป็นตัวเลข 0-9 แล้วแสดงผลออก 7-segment



## แบบฝึกหัดที่ 4 \*\*

- ส่งค่าจากคอมพิวเตอร์เป็นตัวเลข 0-255 ไปควบคุมความสว่างของ LED ที่ต่อที่ Digital pin 9. ( ใช้คำสั่ง `analogWrite` ในการควบคุมความสว่าง )

ข้อนี้ ยากนิดหน่อย ว่างๆ ก็ลองทำดู

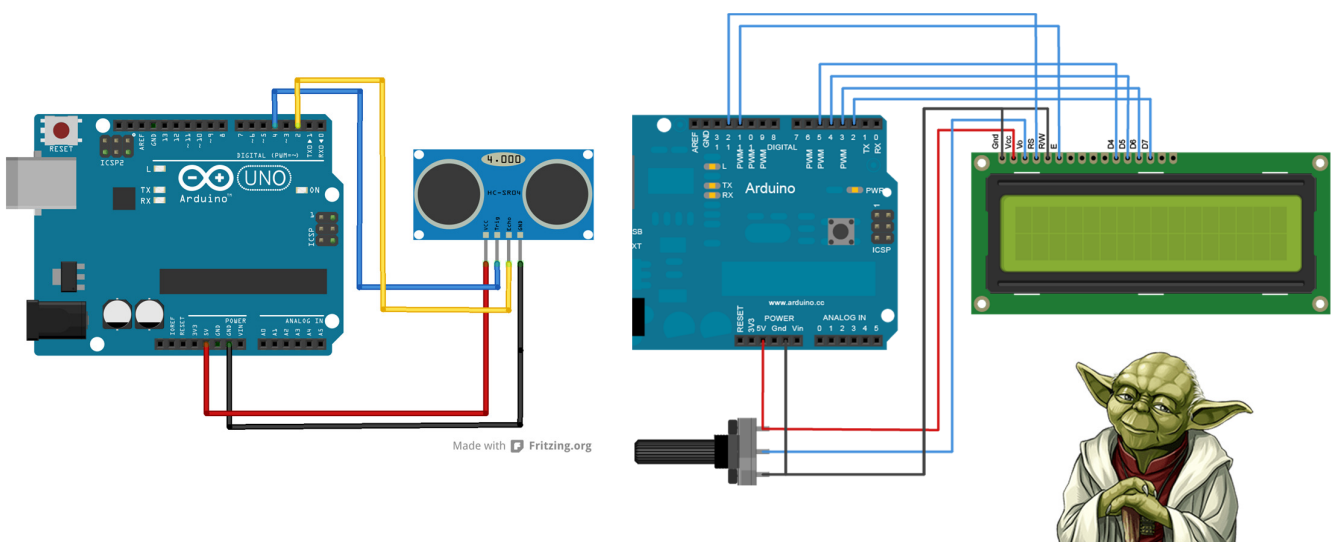
การส่งค่า ใช้ Serial monitor ส่งไปยังบอร์ด



25

## แบบฝึกหัดที่ 5 ตลับเมตรไร้สาย

อ่านค่าระยะทาง โดยใช้ Ultrasonic แล้วแสดงผลออกที่จอ LCD



26