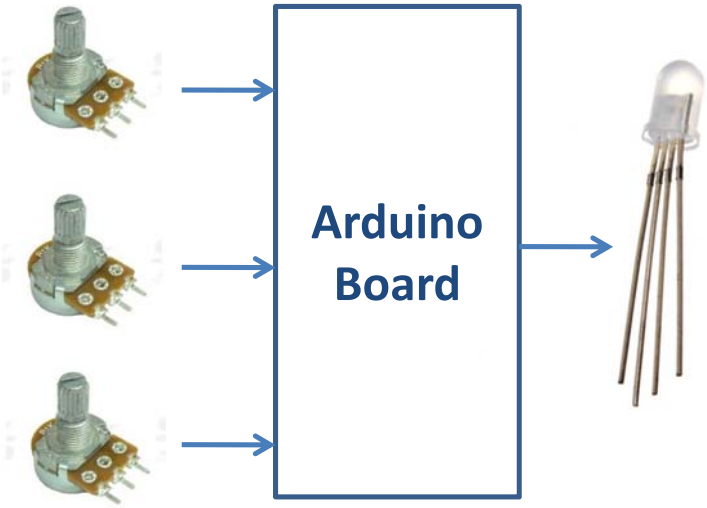
แบบฝึกหัดที่ 3.2 Tuning Color LED

จงเขียนโปรแกรมปรับสี color LED โดยรับค่าจาก ตัวต้านทานปรับค่าได้ จำนวน 3 ตัว เพื่อไปปรับค่าสี Red, Green, Blue เพื่อให้ปรับสีต่างๆ ได้ตามใจชอบ

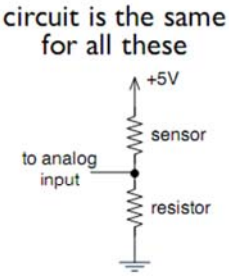


ตัวอย่าง สินค้า

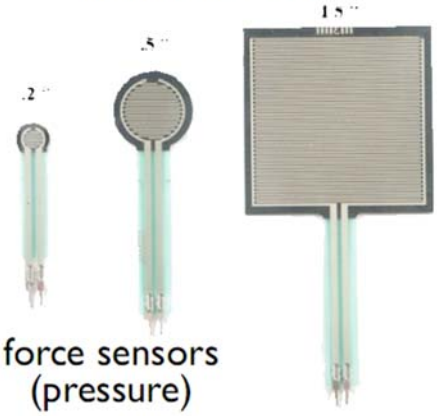
แนวทางการพัฒนา

- นำไปใช้อ่านค่าจาก Resistive Sensor ต่างๆ ได้ เช่น

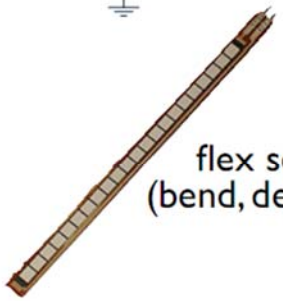
thermistor
(temperature)



photocell
(light)



force sensors
(pressure)



flex sensor
(bend, deflection)

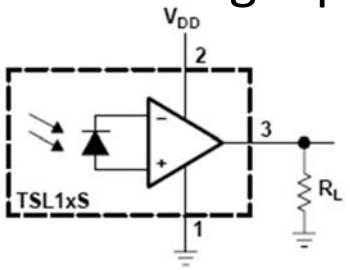
also air pressure
and others

แบบฝึกหัดที่ 3.3 Light Sensor

ต่อวงจรเซ็นเซอร์ ตรวจจับแสงดังรูป

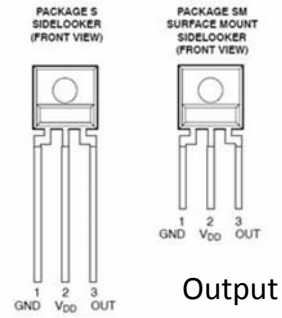
โดยให้ R_L = 10k , V_{DD} = 5V

และไปเข้าขา Analog input



ต่อเข้าขา
Analog input

รายละเอียดขา Sensor



จงเขียนโปรแกรม

Sensing the Dark

โดยที่ ให้ LED สว่าง ตามระดับความมืด

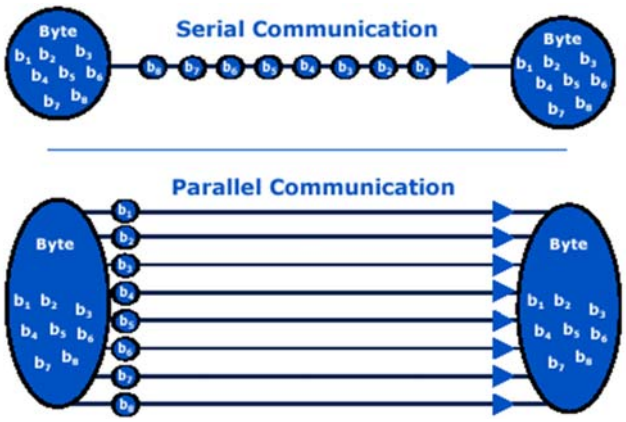
เช่น

ถ้าห้องสว่างมาก ให้ LED ดับ

ถ้าห้องมืดทึบ ให้ LED สว่างที่สุด

Output voltage is linear with light intensity

4. Serial Communication



4. Communicating with other

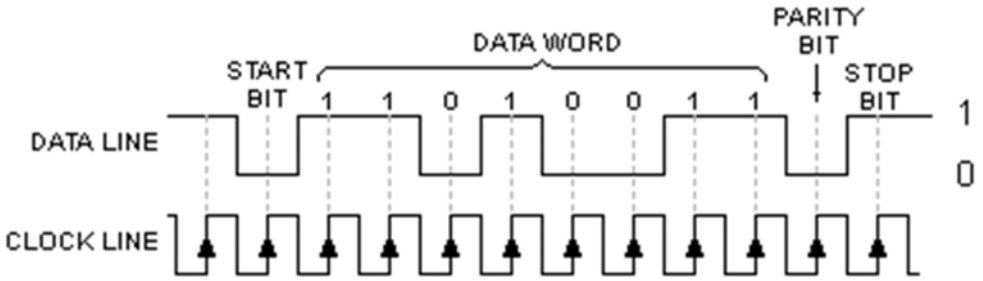
- Not just for computer-to-Arduino communications
- Many other devices speak serial
- Older keyboards & mice speak are serial (good for sensors!)
- Interface boards (graphic LCDs, servo drivers, RFID readers, Ethernet, Wi-Fi)

to Wi-Fi to Ethernet

to graphic LCD

4.1 Serial Communication

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



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 ถ้าข้อมูลที่ได้รับไม่สมบูรณ์ นำเข้าค่ามา check กับ Parity bit จะได้ค่าไม่ตรงกัน
- บิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล จะเป็นลอจิก 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
   =
```

4.2 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

4.3 Serial command

- Talking to other uses the “Serial” command

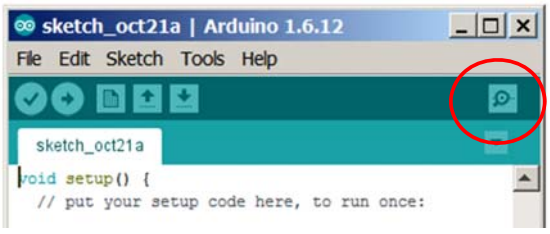
  - `Serial.begin()` : prepare to use serial
  - `Serial.print()` : send data to serial port
  - `Serial.println()` : send data and newline to serial port
  - `Serial.read()` : read data from serial port
  - `Serial.available()` : ready to read
  - `Serial.flush()` : clear buffer at incoming serial data
  - `Serial.parseInt()` : returns the first valid (long) integer number from the serial buffer

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

การทดลองที่ 4.1 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);
}
```

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

การทดลองที่ 4.2 Serial Read Basic

ทดลองรับค่าจากคอมพิวเตอร์

และส่งค่าออกกลับไปคอมพิวเตอร์

ในโปรแกรม “Serial Monitor”, ให้พิมพ์อะไร ก็ได้ จากนั้น กดปุ่ม Send


```
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);
     }
}
```

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.

การทดลองที่ 4.3 Controlling from computer

ทดลองรับค่าจากคอมพิวเตอร์

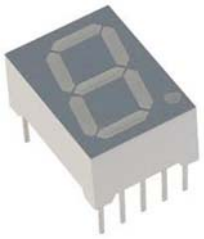
- In “Serial Monitor”, You type “H”, Press Send

- 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);
        }
   }
}
```


แบบฝึกหัดที่ 4.1

- ส่งค่าจากคอมพิวเตอร์เป็นตัวเลข 0-9 แล้วแสดงผลออก 7-segment



แบบฝึกหัดที่ 4.2

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

แบบฝึกหัดที่ 5.3

จากการทดลอง Analog input (การทดลองที่ 3.2)

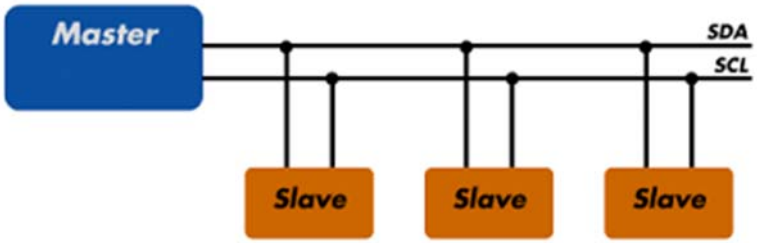
จงเขียนโปรแกรม อ่านค่าแรงดัน จากการปรับตัวต้านทานปรับค่าได้ แล้วไปแสดงผลที่จอคอมพิวเตอร์

แบบฝึกหัดที่ 5.4

จากการทดลอง Light Sensor (การทดลองที่ 3.2)

จงเขียนโปรแกรมอ่านค่าแรงดันที่ได้จาก Sensor และแสดงค่าออกจอคอมพิวเตอร์ว่า เมื่อมืดที่สุด (เอามือปิด Sensor) มีค่าเท่าใด และเมื่อสว่างสุด มีค่าเท่าใด

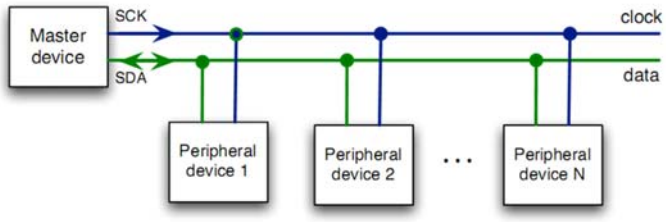
6. Inter Integrate Circuit Bus (I^2C)



I^2C , "Two Wire"

Synchronous serial bus with shared a data line

a little network for your gadgets



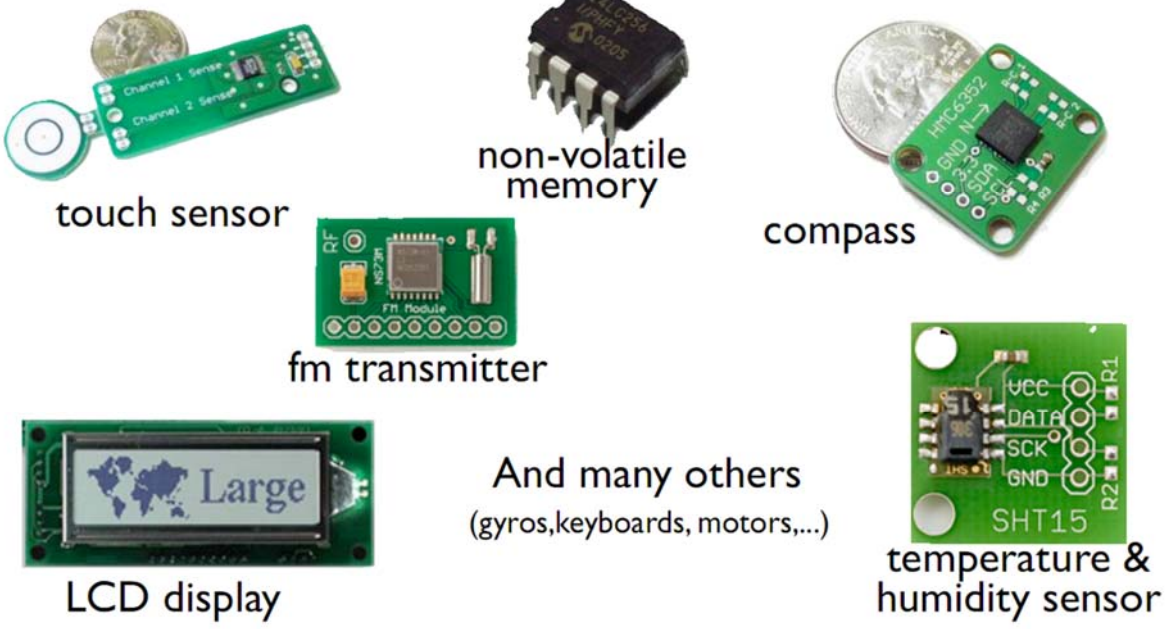
- *Up to 127 devices on one bus*
- *Up to 1Mbps data rate*
- *Really simple protocol (compared to USB,Ethernet,etc)*
- *Most microcontrollers have it built-in*

I²C , “Two Wire”

- I²C Bus ย่อมาจาก Inter Integrate Circuit Bus เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ ติดต่อสื่อสาร ระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก

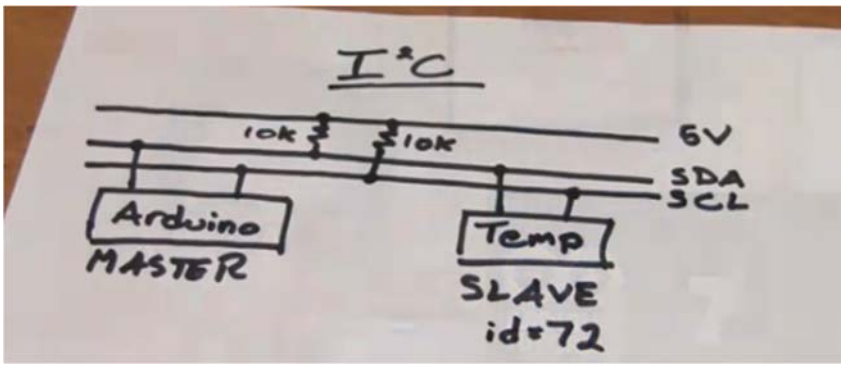
- ถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัวเข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

Many I²C devices



I²C on Arduino

- I²C built-in on Arduino's ATmega168 chip
- Use "Wire" library to access it
- **Analog In 4 is SDA signal**
- **Analog In 5 is SCK signal**



Wire library Functions

`Wire.begin()`	*join i²c bus (master)*
`Wire.begin(4)`	*join i²c bus with address #4 (Slave)*
`Wire.beginTransmission(112)`	*transmit to device #112*
`Wire.endTransmission()`	*stop transmitting*
`Wire.requestFrom(112, 2)`	*request 2 bytes from slave device #112*
`Wire.available()`	*Returns the number of bytes available for reading*
`Wire.read()`	*receive a byte*
`Wire.write("x is ")`	*sends five byte*
`Wire.write(x)`	*sends one byte*

Arduino “Wire Library”

Master Writing data to Slave

Load wire Library

Join i2c

Start sending

Send data

Stop sending

```
#include <Wire.h>
void setup()
{
  Wire.begin();                    // join i2c bus (master)
}

byte x = 0;

void loop()
{
  Wire.beginTransmission(4);       // transmit to device #4
  Wire.write("x is ");             // sends five bytes
  Wire.write(x);                   // sends one byte
  Wire.endTransmission();          // stop transmitting

  x++;
  delay(500);
}
```


Arduino “Wire Library”

Master Reading data from Slave

Join i2c

Request data

Get data

```
#include <Wire.h>

void setup()
{
 Wire.begin();                    // join i2c bus (master)
 Serial.begin(9600);              // start serial for output
}

void loop()
{
 Wire.requestFrom(2, 6);          //request 6 bytes from device #2
 while(Wire.available())          //slave may send less than requested
  {
    char c = Wire.read();         // receive a byte as character
    Serial.print(c);              // print the character
  }

 delay(500);
}
```