Chương 2
Các hàm cơ bản

Nội dung chương 2

- Các hàm vào/ra số
- Các hàm vào/ra tương tự
- Các hàm vào/ra nâng cao
- Các hàm timer
- Các hàm truyền thông
- Các hàm ngắt
- Các hàm toán học

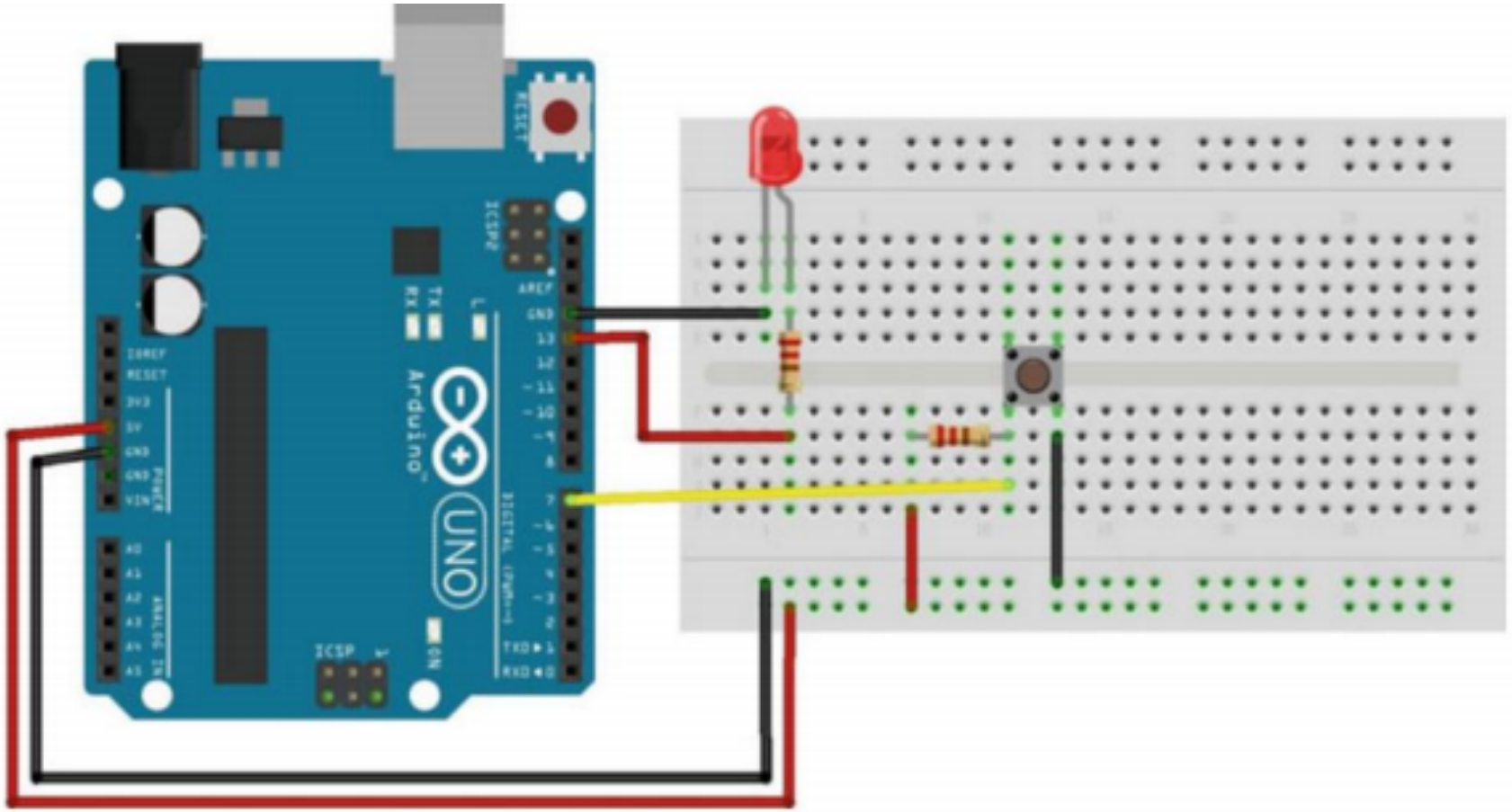
Các hàm vào ra số

```
pinMode(pin,mode)
Parameters
    pin: the number of the pin whose mode you
         wish to set
    mode: INPUT, OUTPUT, or INPUT_PULLUP
Returns
    None
```

```
digitalWrite(pin,value)
Parameters
    Pin: the number of the pin you want to write
    value: HIGH or LOW
Returns
    None
```

```
digitalRead(pin)
Parameters
    pin: the number of the pin you want to read (int)
Returns
    HIGH or LOW
```

Ví dụ: Bật/tắt LED khi nhả/nhấn nút



Ví dụ: Bật LED khi nhấn nút

```
int led = 13;  // connect LED to pin 13
int pin = 7;   // connect pushbutton to pin 7
int value = 0; // variable to store the read value

void setup() {
  pinMode(led, OUTPUT);  // set pin 13 as output
  pinMode(pin, INPUT);   // set pin 7 as input
}
void loop() {
  value = digitalRead(pin); // set value equal to the pin 7 input
  digitalWrite(led, value); // set LED to the pushbutton value
}
```

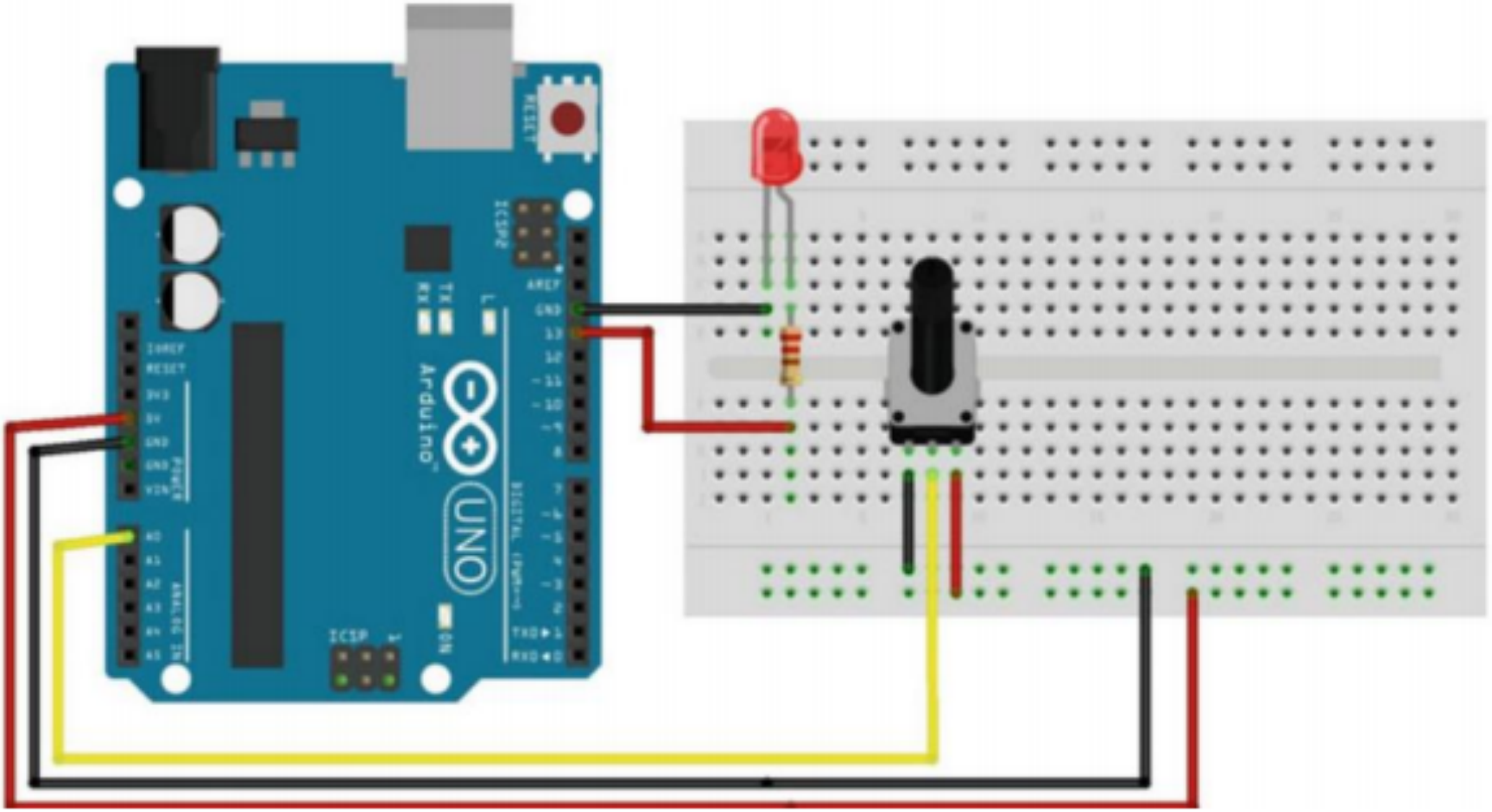
Các hàm vào ra tương tự

```
analogReference(type)
Parameters
    type: which type of reference to use (DEFAULT,
          INTERNAL, INTERNAL1V1, INTERNAL2V56, or
          EXTERNAL)
Returns
    None
```

```
analogRead(pin)
Parameters
    pin: the number of the analog input pin to read from
         (0-5)
Returns
    int(0 to 1023)
```

```
analogWrite(pin,value)
Parameters
    pin: the number of the pin you want to write
    value: the duty cycle between 0 (always off, 0%) and
           255 (always on, 100%)
Returns
    None
```

Ví dụ: Độ sáng đèn LED thay đổi



Ví dụ: Độ sáng đèn LED thay đổi

```
int led = 13;  // connect LED to pin 13
int pin = 0;   // potentiometer on analogy pin 0
int value = 0; // variable to store the read value

void setup() {
}
void loop() {
  value = analogRead(pin); // set value equal to the pin 0's input
  value /= 4;              // converts 0-1023 to 0-255
  analogWrite(led, value); // output PWM signal to LED
}
```

Các hàm vào/ra nâng cao

```
shiftOut(dataPin,clockPin,bitOrder,value)
Parameters
   dataPin: the pin on which to output each bit (int)
   clockPin: the pin to toggle once the dataPin has been
             set to the correct value (int)
   bitOrder: which order to shift out the bits; either
             MSBFIRST or LSBFIRST. (Most Significant Bit
             First, or, Least Significant Bit First)
   value: the data to shift out (byte)
Returns
   None
```

Các hàm vào/ra nâng cao

```
pulseIn(pin,value,timeout)
Parameters
   pin: the number of the pin on which you want to read
        the pulse (int)
   value: type type of pulse to read: either HIGH or LOW
        (int)
   timeout (optional): the number of microseconds to wait
        for the pulse to start; default is one second
        (unsigned long)
Returns
   the length of the pulse (in microseconds) or 0 if no
        pulse started before the timeout
```

Các hàm Timer

```
delay(ms)
Parameters
    ms: the number of milliseconds to pause (unsigned
        long)
Returns
    None
```

```
delayMicroseconds(us)
Parameters
    us: the number of microseconds to pause (unsigned int)
Returns
    None
```

Các hàm Timer (tt)

```
millis()
Parameters
    None
Returns
    Number of milliseconds since the program started
        (unsigned long)
```

```
micros()
Parameters
    None
Returns
    Number of microseconds since the program started
        (unsigned long)
```

Các hàm truyền thông

```
Serial.begin(speed)
Parameters
    speed: set the baud rate
Returns
    None
```

```
Serial.available()
Parameters
    None
Returns
    the number of bytes available to read
```

```
Serial.read()
Parameters
    None
Returns
    the first byte of incoming serial data available (or
    -1 if no data is available) -int
```

Các hàm truyền thông (tt)

```
Serial.print(val)
Parameters
    val: the value to print - any data type
Returns
    None
```

```
Serial.println(val,format)
Parameters
    val: the value to print - any data type
    format: specifies the number base (for integral data
            types) or number of decimal places (for
            floating point types)
Returns
    the number of bytes available to read
```

Ví dụ: Bật/tắt LED từ máy tính

```
int ledpin =13;
void setup() {
 Serial.begin(9600);
pinMode(ledpin,OUTPUT);
}

void loop() {
 if( Serial.available()>0)
 char setupled = Serial.read();}
  switch(setupled)
  {
  case '1' :    {
     digitalWrite(ledpin,HIGH); break;
     }
  case '0':
  {
  digitalWrite(ledpin,LOW); break; }}}
```

Các hàm ngắt

```
attachInterrupt(digitalPinToInterrupt(pin),ISR,mode)
Parameters
    interrupt: the number of the interrupt (int)
    pin: the pin number
    ISR: the interrupt service routine (ISR) to call when the
         interrupt occurs; this function must take no
         parameters and return nothing. This function is
         sometimes referred to as an interrupt service routine.
    mode: defines when the interrupt should be triggered. Four
         constants are predefined as valid values:
        LOW to trigger the interrupt whenever the pin is low,
        CHANGE to trigger the interrupt whenever the pin changes
         value
        RISING to trigger when the pin goes from low to high,
        FALLING for when the pin goes from high to low.
Returns
    None
```

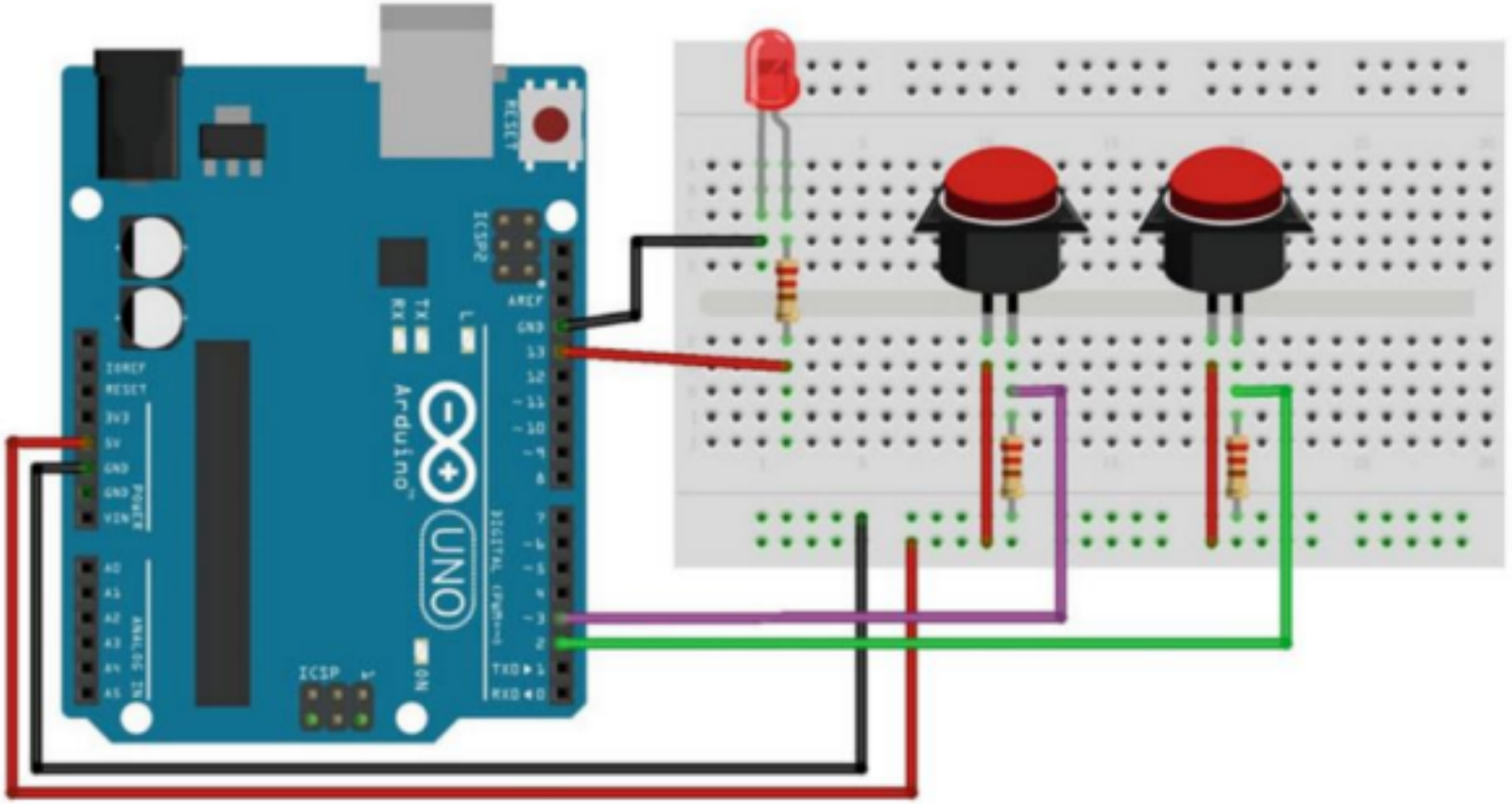
Các hàm ngắt (tt)

```
detachInterrupt(interrupt)
Parameters
    interrupt: the number of the interrupt to disable
Returns
    None
```

```
Interrupts()
Parameters
    None
Returns
    None
```

```
noInterrupts()
Parameters
    None
Returns
    None
```

Ví dụ: Ngắt từ nút nhấn



Ví dụ: Ngắt từ nút nhấn

```
#define LED 13
void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  attachInterrupt(0, displayMicros, RISING);
  attachInterrupt(1, displayMillis, RISING);
}
void loop() {
  digitalWrite(LED, HIGH);
  delay(500);
  digitalWrite(LED, LOW);
  delay(500);
}
void displayMicros() {
  Serial.write("micros()=");
  Serial.println(micros());
}
void displayMillis() {
  Serial.write("millis()=");
  Serial.println(millis());
}
```

Các hàm toán học

```
min(x,y)
Parameters
    x: the first number, any data type
    y: the second number, any data type
Returns
    The smaller of the two numbers
```

```
max(x,y)
Parameters
    x: the first number, any data type
    y: the second number, any data type
Returns
    The larger of the two numbers
```

```
random(min,max)
Parameters
    min: lower bound of the random value, inclusive (optional)
    max: upper bound of the random value, exclusive
Returns
    a random number between min and max
```

Các hàm toán học (tt)

```
abs(x)
Parameters
    x: the number
Returns
    x: if x is greater than or equal to 0.
    -x: if x is less than 0.
```

```
pow(base,exponent)
Parameters
    base: the number (float)
    exponent: the power to which the base is raised (float)
Returns
    The result of the exponentiation (double)
```

```
sqrt(x)
Parameters
    x: the number, any data type
Returns
    double, the number's square root
```

Các hàm toán học (tt)

```
constrain(x,a,b)
Parameters
    x: the number to constrain, all data types
    a: the lower end of the range, all data types
    b: the upper end of the range, all data types
Returns
    x: if x is between a and b
    a: if x is less than a
    b: if x is greater than b
```

```
map(value,fromLow,fromHigh,toLow,toHigh)
Parameters
    value: the number to map
    fromLow: the lower bound of the value's current range
    fromHigh: the upper bound of the value's current range
    toLow: the lower bound of the value's target range
    toHigh: the upper bound of the value's target range
Returns
    The mapped value
```