Ethernet w/ MicroSD add-on for Arduino Nano

Item# ARSH-0053

Overview :

This add-on module allows you to connect the Arduino Nano to the internet. You get all of the benefits of using the Arduino Nano. This includes breadboard friendliness for quick prototyping. It is fully assembled with long pin headers so you don't need to purchase and solder the headers separately. You're ready to go out of the box. Yes, you get the same stackable benefits as the Arduino shields and can run the same code. Its small size of 3.7"x1.0" makes easy to hide or fit in a small project box.

It is based on the Wiznet W5100 Ethernet chip which provids a network (IP) stack capable of both TCP and UDP. The ETHERNET-4NANO supports up to four simultaneous socket connections. Use theEthernet library to write sketches which connect to the internet using the add-on module.

The ETHERNET-4NANO connects to an Arduino Nano using female long wire-wrap headers which extend through the board for the breadboard connections. This keeps the pin layout intact and allows another add-on module to be stacked on top or bottom.

The Micro-SD card slot can be used to store files for serving over a network. It is compatible with the Ethernet library for the Arduino 0019 and later. The SD card library is not yet included in the standard Arduino distribution, but the sdfatlib by Bill Greiman works well. See the tutorial from Adafruit Industries for instructions on the Ethernet shield which is compatible with the ETHERNET-4NANO module (thanks Limor!).

Also included a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up.

The Arduino Nano communicates with both the W5100 and MicroSD card using the SPI bus. This is on digital pins 11, 12, and 13 on the Nano. Pin 10 is used to select the W5100 and pin 4 is used for the MicroSD card. These pins cannot be used for General Purpose Input/Output(GPIO).

> *Note : that because the W5100 and MicroSD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you\'re not using one of the peripherals in your program, however, you\'ll need to explicitly deselect it. To do this with the MicroSD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.*
>
> **The ETHERNET-4NANO provides a standard RJ45 ethernet jack.**
>
> **The reset button on the Nano resets both the W5100 and the Arduino Nano.**

Electronics Source Co.,Ltd
7/129 Central Pinklao Bldg., 17FL., Unit 1702
Baromrachonnee Rd., Bangkok-noi, Bangkok 10700

Website : http://www.es.co.th
Email : info@es.co.th
Tel : (662) 884-9210 (6 line)
Fax : (662) 884-9213-4

Specifications :

- Embedded Server/Client
- Under voltage reset
- MicroSD for storage
- 4 mounting holes for standalone application
- Breadboard friendly, small size 3.7”x1.0”
- Many status LEDs for troubleshooting
- Long header pins for stackability

Electronics Source Co.,Ltd
7/129 Central Pinklao Bldg., 17FL., Unit 1702
Baromrachonnee Rd., Bangkok-noi, Bangkok 10700

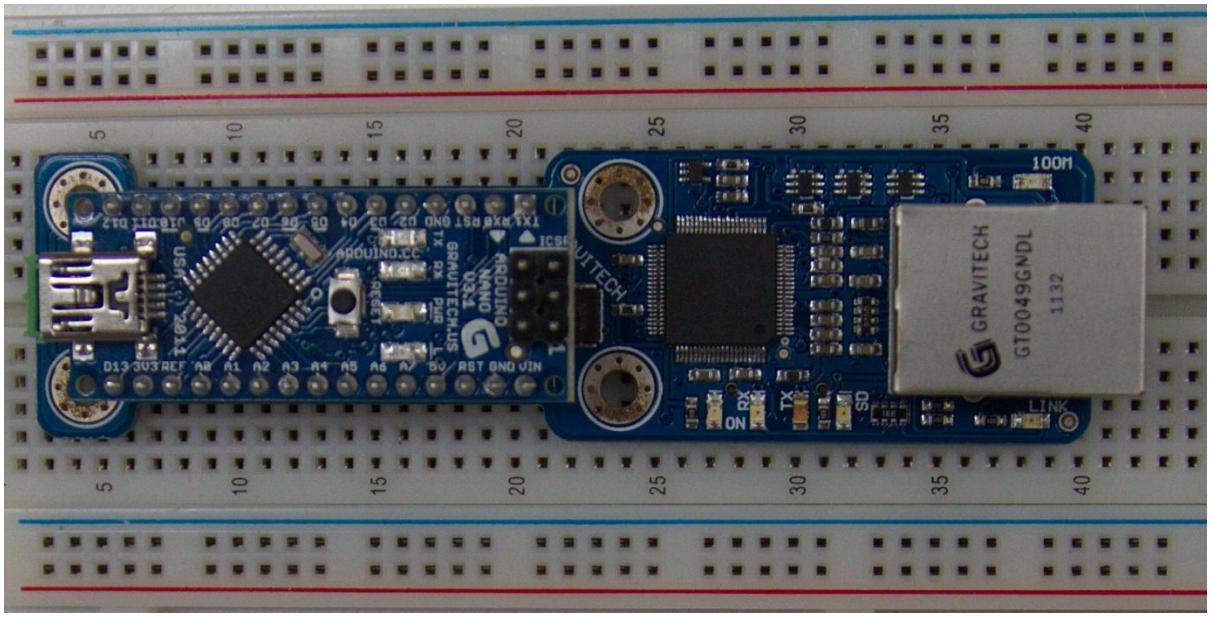
Website : http://www.es.co.th
Email : info@es.co.th
Tel : (662) 884-9210 (6 line)
Fax : (662) 884-9213-4

Ethernet.h & SD.h Library

Library สองตัวนี้ แนะนำให้ใช้คู่กับ Ethernet with MicroSD add-on for Arduino Nano โดยเราสามารถเชื่อมต่อ internet และ สามารถเปิดไฟล์ใน microSD card ได้ ใช้การเชื่อมต่อแบบ SPI ทั้ง 2 โมดูล

***หมายเหตุ: ตัวนี้เป็น Arduino nano shield หากต้องการใช้กับ Arduino อื่นๆ ควรดูการต่อสายให้ดี*

1. MOSI สำหรับ Nano D11
2. MISO สำหรับ Nano D12
3. SCK สำหรับ Nano D13
4. SS ของ Ethernet D10
5. SS ของ microSD D4



รูปที่ 1 ตัวอย่างการวางอุปกรณ์

Ethernet class

- begin()
- localIP()
- maintain()

IPAddress class

- IPAddress()

Server class

- Server
- EthernetServer()
- begin()
- available()
- write()
- print()
- println()

Client class

- Client
- EthernetClient()
- if (EthernetClient)
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()
- stop()

EthernetUDP class

- begin()
- read()
- write()
- beginPacket()
- endPacket()
- parsePacket()
- available()
- stop()
- remoteIP()
- remotePort()

SD class

- begin()
- exists()
- mkdir()
- open()
- remove()
- rmdir()

File class

- available()
- close()
- flush()
- peek()
- position()
- print()
- println()
- seek()
- size()
- read()
- write()
- isDirectory()
- openNextFile()
- rewindDirectory()

Ethernet.begin()

Description

ตั้งค่าเน็ตเวิร์คเบื้องต้น

Syntax

Ethernet.begin(mac)

Ethernet.begin(mac, ip)

Ethernet.begin(mac, ip, dns)

Ethernet.begin(mac, ip, dns, gateway)

Ethernet.begin(mac, ip, dns, gateway, subnet)

Parameters

Mac : array of 6 bytes -> MAC address from Ethernet shield

***หมายเหตุ: shield ใหม่ๆจะมี mac address sticker มาให้ถ้า shield เก่าให้สร้างเอง*

Ip : array of 4 bytes -> the IP address of the device

Dns : array of 4 byte -> the IP address of the DNS server

Gateway : array of 4 bytes -> the IP address of the network

Subnet : array of 4 bytes -> the subnet mask of the network

Returns

The DHCP version of this function

- true : เชื่อมต่อ DHCP สำเร็จ
- false : เชื่อมต่อ DHCP สำเร็จ

Example

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,0,177);
Ethernet.begin(mac, ip);
```

Average time Usage

296965 microseconds

Ethernet.localIP()

Description

รับค่า IP address จาก router

Syntax

Ethernet.localIP()

Parameters

None

Returns

IPAddress : IPAddress Object

Example

```
//รับค่า ใส่ไว้ในตัวแปรขนาด IPAddress
IPAddress isShow;
isShow = Ethernet.localIP();
Serial.print(isShow);
```

Average time Usage

53 microseconds

Ethernet.maintain()

Description

ปกติ อุปกรณ์จะได้รับ IP address มาจาก DHCP ช่วงเวลาหนึ่ง หากได้รับ IP address จาก DHCP เมื่อต้องการใช้ address ต่อ ด้วย Ethernet.maintain() จะให้อุปกรณ์ร้องขอ IP address ได้อีกครั้ง แต่จะได้รับ IP address ใดขึ้นอยู่กับการ config server ของคุณ อาจจะได้ IP เดิม หรือ IP ใหม่ หรือ ไม่ได้เลย

Syntax

Ethernet.maintain()

Parameters

None

Returns

- 0 : nothing happened
- 1 : renew failed
- 2 : renew success
- 3 : rebind fail
- 4 : rebind success

Example

Byte error = Ethernet.maintain();

Average time Usage

8 microseconds

IPAddress()

Description

ประกาศค่า IP address สามารถประกาศได้ทั้ง local และ remote address

Syntax

IPAddress(address)

Parameters

address : 4 bytes -> ค่า IP address

Returns

None

Example

- IPAddress subcat(192,168,1,64);
- IPAddress littlePS(0xC0,0xA8,0x01,0x70);

Average time Usage

5 microseconds

EthernetServer()

Description

สร้าง server ที่จะคอยรับค่าจาก port ที่กำหนด

Syntax

Server(port)

Parameters

Port : int -> ค่าพอร์ตที่เปิดไว้รอรับค่า

Returns

None

Example

- EthernetServer server = EthernetServer(23);
- EthernetServer server(23);

Average time Usage

4 microseconds

begin()

Description

สั่งให้ server เริ่มรับค่าที่ถูกส่งมา

Syntax

ชื่อserver.begin()

Parameters

None

Returns

None

Example

```
EthernetServer server1 = EthernetServer(23);
server1.begin();
```

Average time Usage

138 microseconds

available()

Description

เช็ค server ว่ามี data พร้อมจะอ่านหรือไม่ ถ้ามีจะใช้ Client Object กลับมา

Syntax

ชื่อserver.available()

Parameters

None

Returns

Client Object : client ที่ได้รับจะถูกใช้โดย Client.read() อีกต่อนึง

Example

```
EthernetClient client = server1.available();
```

Average time Usage

232 microseconds

write()

Description

ให้เขียนข้อมูลลงใน client ที่เชื่อมต่อกับเน็ตเวิร์คนี้ ข้อมูลในที่นี้ จะเป็นการส่งของ series of bytes

Syntax

ชื่อserver.write(val)

ชื่อserver.write(buf, len)

Parameters

Val : byte or char -> ส่งค่า 1 byte

Buf : array of (byte or char) -> array ที่เก็บข้อมูลที่จะส่ง

Len : int -> ความยาวของ array

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

```
server.write(client.read());
```

Average time Usage

220 microseconds

print()

Description

Print ตัวเลขใส่ client ทั้งหมดที่เชื่อมต่อกับ server

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อserver.print(data)

ชื่อserver.print(data, BASE)

Parameters

Data : char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 1

ตารางที่ 1

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

```
EthernetServer server1(23);
Server1.print(thisChar);
```

Average time Usage

221 microseconds

println()

Description

Print ตัวเลขและขึ้นบรรทัดใหม่ใส่ client ทั้งหมดที่เชื่อมต่อกับ server

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อserver.println()

ชื่อserver.println(data)

ชื่อserver.println(data, BASE)

Parameters

Data : char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 2

ตารางที่ 2

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

```
EthernetServer server1(23);
Server1.println();
```

Average time Usage

658 microseconds

EthernetClient()

Description

ประกาศตัวแปร สำหรับ client

Syntax

EthernetClient()

Parameters

None

Returns

None

Example

EthernetClient client;

Average time Usage

7 microseconds

If(client)

Description

เช็คว่า client พร้อมหรือไม่

Syntax

If(*ชื่อclient*)

While(*ชื่อclient*)

Parameters

Client : EthernetClient Object -> ชื่อ clientที่จะเช็ค

Returns

- Bool : returns true if the specified client is available.

Example

```
while(!client){
  ; // wait until there is a client connected to proceed
}
```

Average time Usage

4 microseconds

Connected()

Description

ตรวจสอบว่า client connect อยู่รึเปล่า

***หากยังมีข้อมูลที่ยังไม่ได้อ่านอยู่ใน client จะถือว่า clientนั้น ยังเชื่อมต่ออยู่*

Syntax

ชื่อclient.connected()

Parameters

None

Returns

- Bool : true if the client is connected, false if not.

Example

```
if (!client.connected())
{ Serial.print”yeah”);
```

Average time Usage

4 microseconds

Connect()

Description

เริ่มการเชื่อมต่อไปที่ IP address และ port

***หมายเหตุเลือกเป็น URL ก็ได้*

Syntax

ชื่อclient.connect()

ชื่อclient.connect(ip, port)

ชื่อclient.connect(URL, port)

Parameters

IP : array of 4 bytes -> IP ที่ client พยายามจะเชื่อมต่อ

URL : String -> URL ของ website

Port : int -> port ที่ client จะเชื่อมไปหา

Returns

- True : if the connection success
- False : if not

Example

client.connect(server, 80);

Average time Usage

1254 microseconds

Write()

Description

เขียนข้อมูลไปให้ server ที่ clientนี้ติดต่ออยู่

****หมายเหตุส่งเป็น series of bytes*

Syntax

ชื่อclient.write(val)

ชื่อclient.write(buf, len)

Parameters

val: byte or char	-> ส่งค่า 1 byte
buf: array of (byte or char)	-> array ที่เก็บข้อมูลที่จะส่ง
len: int	-> ความยาวของ array

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

Client1.write('a');

Average time Usage

8 microseconds

print()

Description

Print ตัวเลขใส่server ที่ clientนี้เชื่อมต่ออยู่

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อclient.print(data)

ชื่อclient.print(data, BASE)

Parameters

Data : char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 3

ตารางที่ 3

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

client1.print(thisChar);

Average time Usage

144 microseconds

println()

Description

Print ตัวเลขและขึ้นบรรทัดใหม่ใส่ server ที่ client ตัวนี้เชื่อมต่ออยู่

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อclient.println()

ชื่อclient.println(data)

ชื่อclient.println(data, BASE)

Parameters

data: char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 4

ตารางที่ 4

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

EthernetServer server1(23);

*Client*1.println();

Average time Usage

157 microseconds

Available()

Description

หาค่าจำนวน bytes ที่สามารถอ่านได้

***หมายเหตุ : เป็นค่าจำนวนข้อมูลที่เขียนใส่ client โดย server ที่มันเชื่อมต่ออยู่*

Syntax

ชื่อclient.available()

Parameters

None

Returns

Int : จำนวนของ bytes ที่พร้อมอ่าน

Example

```
Int j = client1.available();
```

Average time Usage

4 microseconds

Read()

Description

อ่านค่า byte ถัดไปที่รับมาจาก server ที่ client เชื่อมต่ออยู่

Syntax

ชื่อclient.read()

Parameters

None

Returns

- Byte : ถ้ามีค่า

Example

char c = client.read();

Average time Usage

48 microseconds

Flush()

Description

เคลียร์ค่าที่ถูกเขียนใส่ client ไว้แต่ยังไม่ถูกอ่าน

Syntax

ชื่อclient.flush()

Parameters

None

Returns

None

Example

Client1.flush();

Average time Usage

4 microseconds

Stop()

Description

หยุดการเชื่อมต่อไปที่ server

Syntax

ชื่อclient.stop()

Parameters

None

Returns

None

Example

Client1.stop();

Average time Usage

4 microseconds

Begin()

Description

เริ่มการใช้งาน Ethernet UDP และ network

Syntax

ชื่อUDP.begin(localPort);

Parameters

localPort : int -> เป็นพอร์ตให้ ethernetUDP คอยดู

Returns

None

Example

```
#include <EthernetUdp.h>
EthernetUDP Udp;
Udp.begin(localPort);
```

Average time Usage

6 microseconds

Read()

Description

หาค่าจำนวน bytes ที่สามารถอ่านได้

***หมายเหตุ : เป็นค่าจำนวนข้อมูลที่เขียนใส่ client โดย server ที่มันเชื่อมต่ออยู่*

Syntax

ชื่อUDP.read();

ชื่อUDP.read(packetBuffer, MaxSize);

Parameters

packetBuffer : char -> ตัวเก็บสำรอง ข้อมูลเข้า

Maxsize : int -> ขนาดที่ใหญ่ที่สุดของ buffer

Returns

- Char : character ใน buffer

Example

Char c = Udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);

Average time Usage

7 microseconds

Write()

Description

หาค่าจำนวน bytes ที่สามารถอ่านได้

**หมายเหตุ : เป็นค่าจำนวนข้อมูลที่เขียนใส่ client โดย server ที่มันเชื่อมต่ออยู่

Syntax

ชื่อUDP.write(message);

ชื่อUDP.write(buffer, size);

Parameters

Message : char -> ตัวอักษรที่จะออกไป

Buffer : array of (byte or char) -> array ที่จะส่งออกไป ขนาด 8บิท

Size : int -> ขนาดของ array

Returns

- Int : จำนวนของ bytes ที่ส่งไปเรียบร้อย

Example

Udp.write("hello");

Average time Usage

126 microseconds

beginPacket()

Description

เริ่มการเชื่อมต่อเพื่อเขียน UDP data ให้ remote connection

Syntax

ชื่อUDP.beginPacket(remoteIP, remotePort);

Parameters

remoteIP : 4 bytes -> IP address of the remote connection

remotePort : int -> port of the remote connection

Returns

None

Example

```
Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
```

Average time Usage

80 microseconds

endPacket()

Description

ต้องเรียกใช้หลังการเขียนข้อมูลให้ remote connection

Syntax

ชื่อUDP.endPacket()

Parameters

None

Returns

None

Example

Udp.endPacket();

Average time Usage

4 microseconds

parsePacket()

Description

ตรวจสอบว่ามี UDP packet หรือไม่และบอก ขนาดของ packet

Syntax

*ชื่อUDP.*parsePacket()

Parameters

None

Returns

- Int : ขนาดของ UDP packet

Example

int packetSize = Udp.parsePacket();

Average time Usage

39 microseconds

Available()

Description

รับค่าจำนวน byte ที่สามารถอ่านได้จาก buffer (ข้อมูลมาถึงแล้ว)

Syntax

ชื่อUDP.available()

Parameters

None

Returns

- Int : จำนวน byte ที่พร้อมจะถูกอ่าน

Example

```
if(Udp.available())
{Serial.print(“oh yeah”);}
```

Average time Usage

4 microseconds

Stop()

Description

ยุติการเชื่อมต่อกับ server คืนทรัพยากรที่ UDP นำมาใช้

Syntax

ชื่อUDP.stop()

Parameters

None

Returns

None

Example

ชื่อUDP.stop();

Average time Usage

4 microseconds

remoteIP()

Description

รับค่า IP address ของ remote connection

Syntax

ชื่อUDP.remoteIP()

Parameters

None

Returns

4 bytes : IP address of the remote connection

Example

IPAddress ipremote = Udp.remoteIP();

Average time Usage

4 microseconds

remotePort()

Description

รับค่า port ของ remote connection

Syntax

ชื่อUDP.remotePort()

Parameters

None

Returns

- Int : Port ของ UDP connection ถึง remote host

Example

Int x = Udp.remotePort();

Average time Usage

4 microseconds

begin()

Description

เริ่มต้นการใช้งาน Library และ card โดยเปิดการทำงานของ SPI default จะเป็นขา 10 แต่ในที่นี้เราต้องใส่เป็น ขา 4

Syntax

SD.begin()

SD.begin(cspin)

Parameters

cspin : int -> pin ที่เชื่อมต่อกับ CS ของ SD card (สำหรับบอร์ดนี้ คือ 4)

Returns

- true : เชื่อมต่อสำเร็จ
- false: เชื่อมต่อล้มเหลว

Example

```
//เป็นขา สำหรับบอร์ดเรา
SD.begin(4);
```

Average time Usage

10776 microseconds

exists()

Description

ทดสอบว่ามีไฟล์นี้หรือโฟลเดอร์นี้ในระบบหรือไม่

Syntax

SD.exists(filename)

Parameters

Filename : string -> ชื่อไฟล์ที่จะตรวจสอบ

Returns

- true : เมื่อไฟล์มีอยู่จริง
- false : ไม่พบไฟล์

Example

SD.exist(/var/www/song1.mp3)

Average time Usage

5832 microseconds

mkdir()

Description

สร้างโฟลเดอร์ บน SDcard สามารถสร้างทีละหลายไฟล์ได้

เช่น SD.mkdir("a/b/c") will create a, b, and c.

Syntax

SD.mkdir(filename)

Parameters

Filename : string -> ชื่อโฟลเดอร์ที่จะสร้าง

Returns

- true : เมื่อสร้างโฟลเดอร์สำเร็จ
- false : เมื่อล้มเหลว

Example

SD.mkdir(“abah”);
// ได้โฟลเดอร์ชื่อ abah

Average time Usage

8240 microseconds

open()

Description

เปิดไฟล์จาก SD card ถ้าเปิดไฟล์เพื่อเขียนข้อมูล ไฟล์จะถูกสร้างให้โดยอัติโนมัติ

Syntax

SD.open(filepath)

SD.open(filepath, mode)

Parameters

Filepath : string -> ชื่อไฟล์ที่จะเปิด

mode : string -> โหมดของไฟล์ที่จะเปิด มี 2แบบ

- FILE_READ : open the file for reading, starting at the beginning of the file.
- FILE_WRITE : open the file for reading and writing, starting at the end of the file.

Returns

- File object : ได้ตัวไฟล์ที่จะเปิด

Example

//ลองเปิดไฟล์เพื่อเขียน

File mm = SD.open("exam.txt",FILE_WRITE);

Average time Usage

5956 microseconds

remove()

Description

ลบไฟล์ออกจาก SD card

Syntax

SD.remove(filename)

Parameters

Filename : string -> ชื่อไฟล์ที่จะลบ

Returns

- true : เมื่อลบไฟล์สำเร็จ
- false : เมื่อล้มเหลว

Example

SD.remove(“exam.txt”);

Average time Usage

5924 microseconds

rmdir()

Description

ลบโฟลเดอร์ใดๆออกจาก card แต่ โฟลเดอร์นั้นต้องว่างก่อน

Syntax

SD.rmdir(filename)

Parameters

Filename : string -> ชื่อโฟลเดอร์ที่ต้องการจะลบ

Returns

- true : เมื่อลบโฟลเดอร์สำเร็จ
- false : เมื่อล้มเหลว

Example

SD.rmdir(“doujin”);

Average time Usage

5864 microseconds

available()

Description

ตรวจสอบว่ามี byte พร้อมให้อ่านจากไฟล์หรือไม่

Syntax

ชื่อfile.available()

Parameters

None

Returns

- Int : จำนวน byte ที่พร้อมให้อ่าน

Example

Int bytes = file1.available();

Average time Usage

4 microseconds

close()

Description

ปิดไฟล์ แต่ต้องมั่นใจก่อนว่า save ข้อมูลลง SD card แล้วนะจ๊ะ

Syntax

ชื่อfile.close()

Parameters

None

Returns

None

Example

File1.close();

Average time Usage

4 microseconds

flush()

Description

ล้างข้อมูลใน buffer ทั้งหมด

***หมายเหตุ ฟังก์ชั่นนี้จะทำงานทุกครั้งเมื่อปิดไฟล์*

Syntax

ชื่อfile.flush()

Parameters

None

Returns

None

Example

File1.flush();

Average time Usage

4 microseconds

peek()

Description

แอบดูข้อมูลโดยไม่เลื่อน pointer ไปข้างหน้า

Syntax

ชื่อfile.peek()

Parameters

None

Returns

- Byte : ได้ข้อมูลถัดไปในไฟล์

Example

Byte s = file1.peek();

Average time Usage

8 microseconds

position()

Description

รับค่า pointer ที่อยู่ในไฟล์ ณ ปัจจุบัน

Syntax

ชื่อfile.position()

Parameters

None

Returns

- Unsigned long : ค่าตำแหน่ง pointer ที่ชี้ในไฟล์

Example

Unsigned long pt = file1.position();

Average time Usage

4 microseconds

print()

Description

เขียนข้อมูลลงไฟล์ ซึ่งไฟล์ต้องมีสถานะเป็น write

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อfile.print(data)

ชื่อfile.print(data, BASE)

Parameters

Data : char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 5

ตารางที่ 5

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

File1.print(thisChar);

Average time Usage

144 microseconds

println()

Description

เขียนข้อมูลลงไฟล์และขึ้นบรรทัดใหม่ ซึ่งไฟล์ต้องมีสถานะเป็น write

***โดยจะแปลงตัวเลขเป็น ASCII character*

Syntax

ชื่อfile.println()

ชื่อfile.println(data)

ชื่อfile.println(data, BASE)

Parameters

Data : char, byte, int, long, or string -> ข้อมูลที่จะ print

BASE : string -> ฐานของเลขที่จะ print ดังตารางที่ 6

ตารางที่ 6

Input	Description
BIN	Print เลขฐาน 2
OCT	Print เลขฐาน 8
DEC	Print เลขฐาน 10
HEX	Print เลขฐาน 16

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

File1.println(thisChar);

Average time Usage

162 microseconds

seek()

Description

เปลี่ยนตำแหน่ง pointer ไปอยู่ที่ใดๆ ในไฟล์ ได้ตั้งแต่ 0 ถึง size of file

Syntax

ชื่อfile.seek(pos)

Parameters

Pos : unsigned long -> ตำแหน่งที่ต้องการ

Returns

- true : เมื่อย้ายตำแหน่งสำเร็จ
- false : เมื่อล้มเหลว

Example

File1.seek(28);

Average time Usage

4 microseconds

size()

Description

เรียกค่าขนาดของไฟล์

Syntax

ชื่อfile.size()

Parameters

None

Returns

- Unsigned long : ขนาดของไฟล์ หน่วย bytes

Example

Unsigned long size = file1.size();

Average time Usage

4 microseconds

read()

Description

อ่านข้อมูลจากไฟล์

Syntax

ชื่อfile.read()

Parameters

None

Returns

- Byte : ข้อมูลในไฟล์

Example

Byte mm = file1.read();

Average time Usage

8 microseconds

write()

Description

เขียนข้อมูลลงไฟล์

Syntax

ชื่อfile.write(data)

ชื่อfile.write(buf, len)

Parameters

data : byte or char or string(char*) -> ข้อมูล

Buf : array of (byte or char) -> array ที่เก็บข้อมูลที่จะส่ง

Len : int -> ความยาวของ array

Returns

- Byte : จำนวนของ byte ที่ถูกเขียน

Example

File1.write(“omg”);

Average time Usage

8 microseconds

isDirectory()

Description

ฟังก์ชั่นบอกว่าไฟล์ในตอนนี้ เป็น โฟลเดอร์หรือไม่

Syntax

ชื่อfile.isDirectory()

Parameters

None

Returns

- true : ไฟล์ที่ชี้อยู่เป็นโฟลเดอร์
- false : เมื่อไม่ใช่

Example

```
if (file1.isDirectory())
{Serial.print(“oh yeah”);}
```

Average time Usage

4 microseconds

openNextFile()

Description

เรียกค่า file หรือ folder ที่อยู่ข้างใน directory นี้อีกที

Syntax

ชื่อfile.openNextFile()

Parameters

None

Returns

File object : file ที่อยู่ใน folderนี้

Example

File entry = dir.openNextFile() ;

Average time Usage

16 microseconds

rewindDirectory()

Description

กลับไปที่ไฟล์แรกใน folder นี้

Syntax

ชื่อfile.rewindDirectory()

Parameters

None

Returns

None

Example

File1.rewindDirectory();

Average time Usage

8 microseconds