



2022/07/15

Arduino語法(C++語言)

YZU ROBOT CLUB in Banqiao Senior High School

目錄

Part 01

程式基本架構

setup() & loop()函式

Part 02

資料型態

變數~

Part 03

運算子

算術 關係 邏輯

Part 04

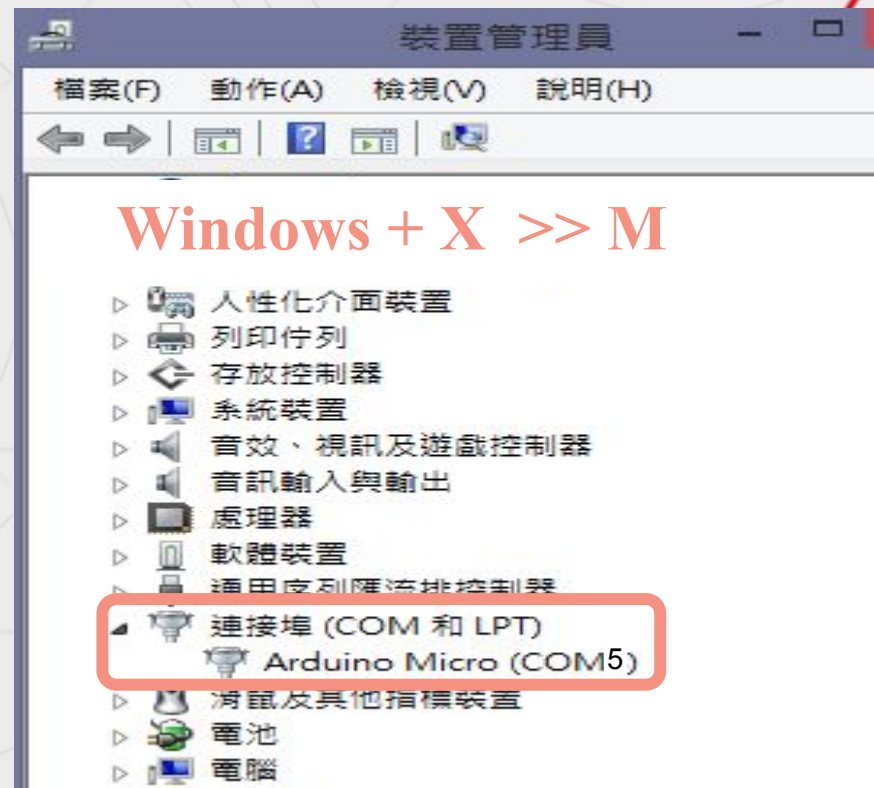
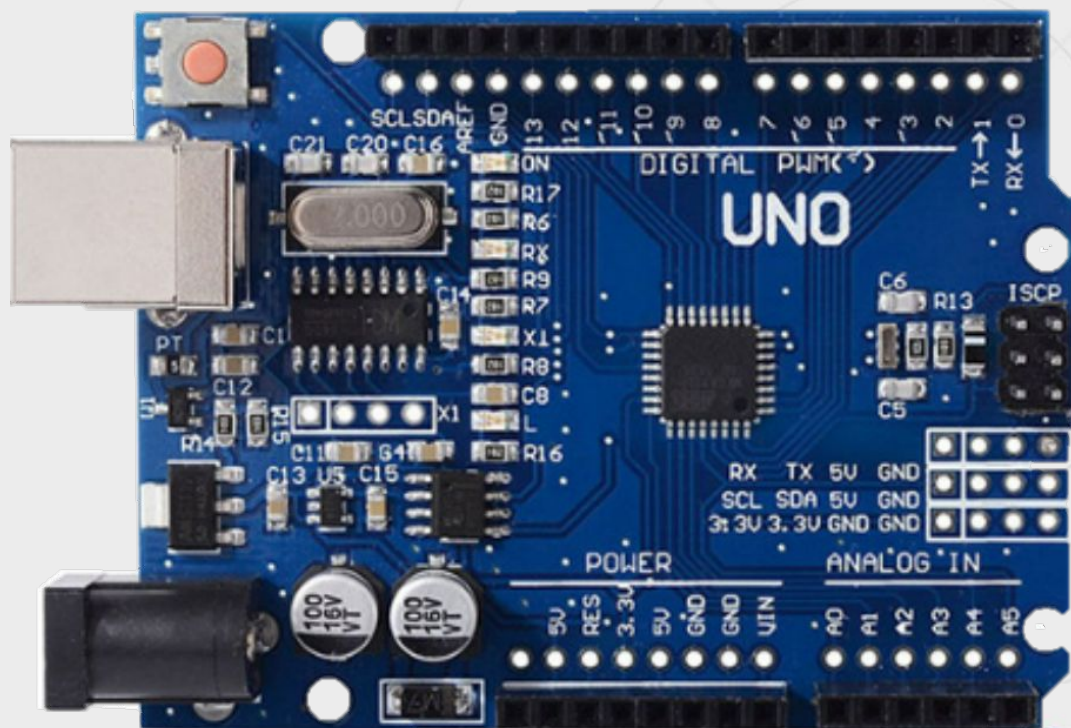
指令

條件判斷 迴圈

Arduino使用

控制板 & 序列埠

電腦接上控制板



Windows + X >> M

Arduino使用 控制板 & 序列埠

開啟路徑:

\桌面\板橋高中\軟體\Arduino



開發板>>Arduino Uno
處理器>>ATmega328P(Old Bootloader)
序列埠
燒錄器>>ArduinoISP

程式基本架構

```
1 //1. 匯入函式庫與定義 (可有可無)
2 #include <stdio.h>
3 #define LEDPIN 13;
4
5 //2. 宣告常數與全域變數 (可有可無)
6 const float pi=3.14159;
7 int r;
8
9 //3. 設定函式(必要)
10 void setup() {
11     //在這裡輸入程式碼，只會執行一次
12     //{}，用於界定程式區
13 }
14 //4. 無限迴圈(必要)
15 void loop() {
16     //在這裡輸入程式碼，會一直重複執行
17 }
```

控制板每次重新啟動時，都會先執行一次 `setup()`，然後再不斷執行 `loop()`。

Arduino 程式會一行一行的執行，每一行會以分號隔開。

補充

#define

define語法會使編譯器在讀到指定名稱時將他替換為指定的值

基本指令：

#define 指定名稱 指定的值

```
1 #define mine 5
2 int a = mine
```

▲我們寫的



```
1 #define mine 5
2 int a = 5
```

▲編譯器看到的

從Arduino傳遞序列埠訊息給電腦

Serial(序列埠)基本指令：

Serial.begin (鮑率);
設定序列埠通訊格式, ex.資料傳輸速率。

Serial.print (輸出值 *or* 參數名稱);
從序列埠輸出字串資料。


Serial.println (輸出值 *or* 參數名稱);
為 **print** + **/n**, 後有回車符和換行符(/n)。


Serial.write (位元組 *or* 參數名稱);
從序列埠輸出字元組資料。

字元(ASCII)

DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII
1	☺	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
2	☻	33	!	65	A	97	a	129	Û	161	í	193	ł	225	ß
3	♥	34	"	66	B	98	b	130	è	162	ó	194	Ť	226	Ô
4	♦	35	#	67	C	99	c	131	â	163	ú	195	Ŧ	227	Ò
5	♣	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ō
6	♠	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	Õ
7	•	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
8	◻	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
9	○	40	(72	H	104	h	136	ê	168	¿	200	ℒ	232	ƒ
10	◼	41)	73	I	105	i	137	ë	169	®	201	℔	233	Ú
11	🔗	42	*	74	J	106	j	138	è	170	¬	202	℥	234	Û
12	🔢	43	+	75	K	107	k	139	ï	171	½	203	℥	235	Ü
13	🎵	44	,	76	L	108	l	140	î	172	¼	204	℥	236	ý
14	🎶	45	-	77	M	109	m	141	ì	173	¡	205	=	237	Ý
15	☀	46	.	78	N	110	n	142	Ä	174	«	206	≡	238	~
16	▶	47	/	79	O	111	o	143	Å	175	»	207	□	239	'
17	▲	48	0	80	P	112	p	144	È	176	⋯	208	ð	240	-
18	↕	49	1	81	Q	113	q	145	æ	177	⋯	209	Ð	241	±
19	≡	50	2	82	R	114	r	146	Æ	178	■	210	Ê	242	=
20	℥	51	3	83	S	115	s	147	ô	179		211	Ë	243	¾
21	§	52	4	84	T	116	t	148	ö	180	└	212	È	244	¶
22		53	5	85	U	117	u	149	ò	181	Á	213	Ì	245	§
23	↕	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
24	↔	55	7	87	W	119	w	151	ù	183	À	215	Î	247	ˆ
25	↕	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	˚
26	↕	57	9	89	Y	121	y	153	Ö	185	℥	217	┘	249	˝
27	↑	58	:	90	Z	122	z	154	Ü	186		218	┐	250	˙
28	└	59	;	91	[123	{	155	ø	187	┐	219	■	251	₁
29	↕	60	<	92	\	124		156	£	188	┘	220	■	252	₃
30	▲	61	=	93]	125	}	157	Ø	189	¢	221	:	253	₂
31	▼	62	>	94	^	126	~	158	×	190	¥	222	ì	254	■
		63	?	95	_	127	␣	159	f	191	└	223	■	255	space

第一個程式 Hello World

 編譯、上傳

序列埠監控視窗

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
    Serial.println("Hello world");  
    delay(100);  
}
```

1. 記得上傳
2. 開啟序列埠監控視窗

將輸入的字串打印到序列埠監控視窗

從Arduino接收序列埠資料

基本指令：

Serial.available () ;

傳回目前序列埠收到的資料。

Serial.readString () ;

讀取並傳回第一個資料(字串), 如果回傳-1(無資料)。

Serial.readBytesUntil () ;

從序列埠讀入一段資料。

程式範例(複讀機)



編譯、上傳

序列埠監控視窗



```
1 void setup() {  
2     // put your setup code here, to run once:  
3     Serial.begin(9600);  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8     String s;  
9     s=Serial.readString();  
10    Serial.print(s);  
11    delay(100);  
12 }
```

1. 記得上傳
2. 開啟序列埠監控視窗

變數

宣告變數的不同方法:

資料型態 變數名稱 (自訂);

ex. *int* x;

資料型態 變數名稱 (自訂) = 儲存值;

ex. *int* x = 10;

***變數名稱不可以是保留字, 像是int、string、return...*

資料型態

類型	中文名稱	佔用記憶體大小	數值範圍
bool	布林	1byte	True或False(1或0)
char	字元	1byte	-128 ~ 127
int	整數	1byte	-32768 ~ 32767
32位元微控制器		4byte	-2,147,483,648 ~ 2,147,483,647
long	長整數	4byte	-2,147,483,648 ~ 2,147,483,647
float	浮點數(小數)	4byte	$\pm 3.4E+38$

資料型態

程式範例

```
int pi = 3.14159;
```

```
int x = 3 / 2 ;
```

```
float y = 3 / 2 ;
```

問會印出什麼?

要如何更改才對??

如何轉換資料型態~~

運算子

算數

算數符號	意義	例子	結果	優先順序
pow(a,b)	指數(次方)	a=2,b=3	8	1
*	乘法	2 * 3	6	3
/	除法	8 / 6	1	3
%	餘數	8 % 6	2	5
+	加法	1+2	3	6
-	減法	2 - 1	1	6
	負數	-6	-6	2

運算子

比較 特殊運算

比較運算子	說明
<code>==</code>	(是否)等於
<code>!=</code>	不等於
<code><</code>	小於
<code>></code>	大於
<code><=</code>	小於等於
<code>>=</code>	大於等於

運算子	意義	例子
<code>++</code>	累加	<code>x ++</code>
<code>--</code>	遞減	<code>x --</code>
<code>+=</code>	指定增加	<code>x += 2</code> == <code>x = x + 2</code>
<code>-=</code>	指定減少	<code>x -= 2</code> == <code>x = x - 2</code>
<code>*=</code>	指定相乘	<code>x *= 2</code> == <code>x = x * 2</code>
<code>/=</code>	指定相除	<code>x /= 2</code> == <code>x = x / 2</code>

**** (x /= 2) 等同 (x 寫入 x / 2)**

補充

邏輯 特殊符號 基本指令

基本指令：

延遲 **delay**(毫秒數);

特殊符號：

單行註解 // 註解文字

多行註解 /* 註解文字...
...註解文字 */

換行 **/n**

更多跳脫字元請自行查詢~~

運算子	說明
&&	且(AND)
	或(OR)
!	相反(NOT)

指令

迴圈

while 迴圈:

- 1. 檢查條件式是否成立
- 2. 執行指令

重複

否
迴圈結束

```
while (條件式){  
    指令1;  
    指令2;  
    :  
}
```

** 不斷重複, 直到條件式不符合時跳出
** 若不更改條件參數將無限輪迴

指令

迴圈

do...while 迴圈:

- 1. 執行指令
 - 2. 檢查條件式是否成立
- 重複
- 否
- 迴圈結束
-

```
do {  
    指令1;  
    指令2;  
    :  
} while (條件式);
```

** 不斷重複, 直到條件式不符合時跳出

** 指令至少會執行一次

程式範例(計算機)

將輸入的兩數字相加



編譯、上傳

序列埠監控視窗



```
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8     String s;  
9     int a;  
10    int b;  
11    Serial.print("Please input int a:");  
12    while(Serial.available()==0) {  
13    }  
14    s=Serial.readString();  
15    a=s.toInt();  
16    Serial.println(a);  
17    Serial.print("Please input int b:");  
18    while(Serial.available()==0) {  
19    }  
20    s=Serial.readString();  
21    b=s.toInt();  
22    Serial.println(b);  
23    Serial.print("a+b=");  
24    Serial.println(a+b);  
25    delay(100);  
26 }
```


指令

迴圈

for 迴圈:

1. 設定初始變數
2. 檢查條件式是否成立
3. 執行指令
4. 執行計數

重複

否

迴圈結束

```
for (初始變數; 條件式; 計數) {  
    指令1;  
    指令2;  
    :  
}
```

** 不斷重複, 直到條件式不符合時跳出
** 初始變數只能再回圈內使用

指令

迴圈程式範例

for 迴圈:

1. 設定初始變數
2. 檢查條件式是否成立
3. 執行指令
4. 執行計數

重複

否

迴圈結束

範例：

```
for ( int i = 0 ; i <= 10 ; i++ ) {  
    Serial.println(i) ;  
}
```

Question :

問會印出什麼？

那如果條件式改成 $i < 10$ 呢？

for 迴圈

程式範例

完成以下圖形：

```
*  
* *  
* * *  
* * * *  
* * * * *
```

** 用 `Serial.readString()` ; 輸入任意行數

運算子

程式範例

▶ 計算BMI：

$\text{bmi} = \text{weight} / (\text{height公尺})^2$

提示：定義bmi, weight, height 三個參數

指令

條件判斷

If

指令	說明	一定要有
if	只有一個條件	o
else if	有一個以上的條件	x
else	不符合條件時執行	x

** 若所有條件皆不符合也沒有else, 此程式將不執行。

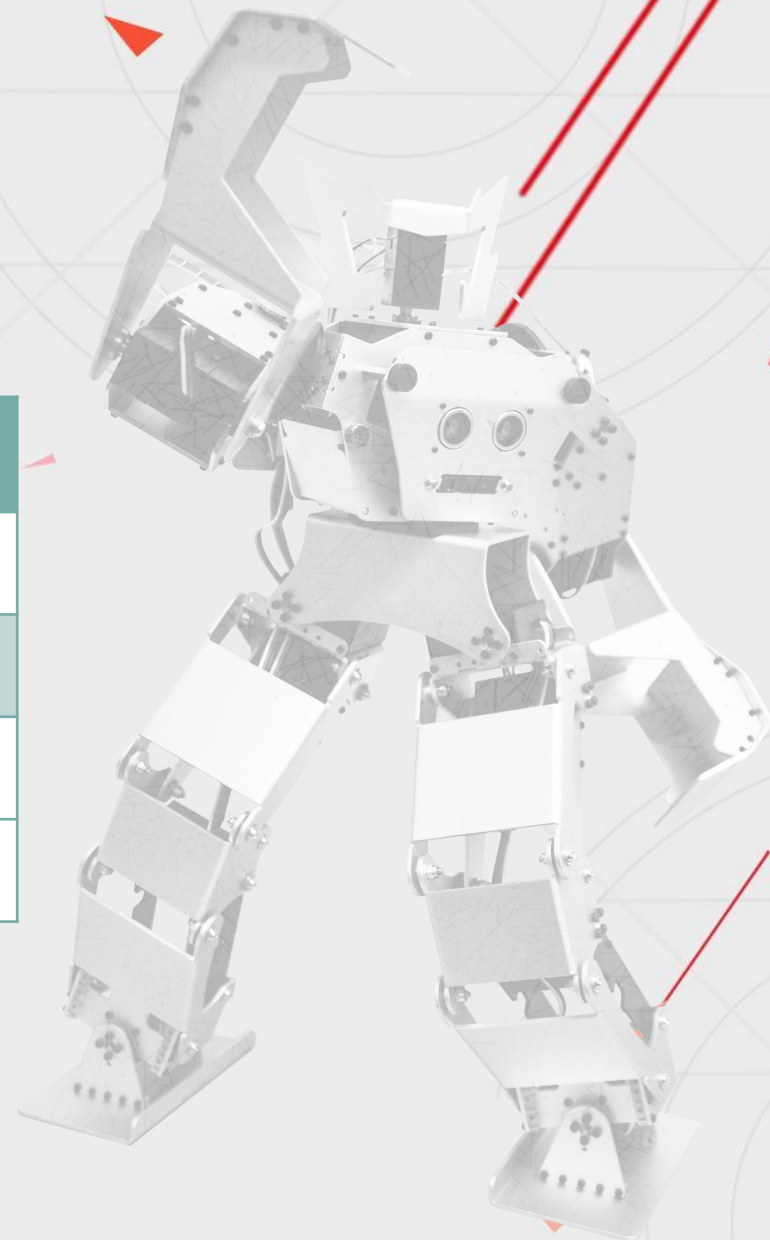
```
If ( 條件式A ){  
    指令A;  
}  
else if ( 條件式B ){  
    指令B;  
}  
else {  
    指令C;  
}
```

** else if & else 不能單獨出現
** ;(分號)請記得打

請將BMI程式完成

顯示	BMI
體重過低	<18.5
體重正常	$18.5\sim23.9$
超重	$24\sim27.9$
肥胖	≥ 28

請使用 If 完成



函式

1. 設定資料型態
2. 設定變數
3. 編寫指令
4. 回傳結果

範例：

```
int add(int a, int b){  
    int c = a + b;  
    return c;  
}
```

Question :

把剛才的BMI計算用函式改寫

函式

程式範例

▶ 計算階乘：

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

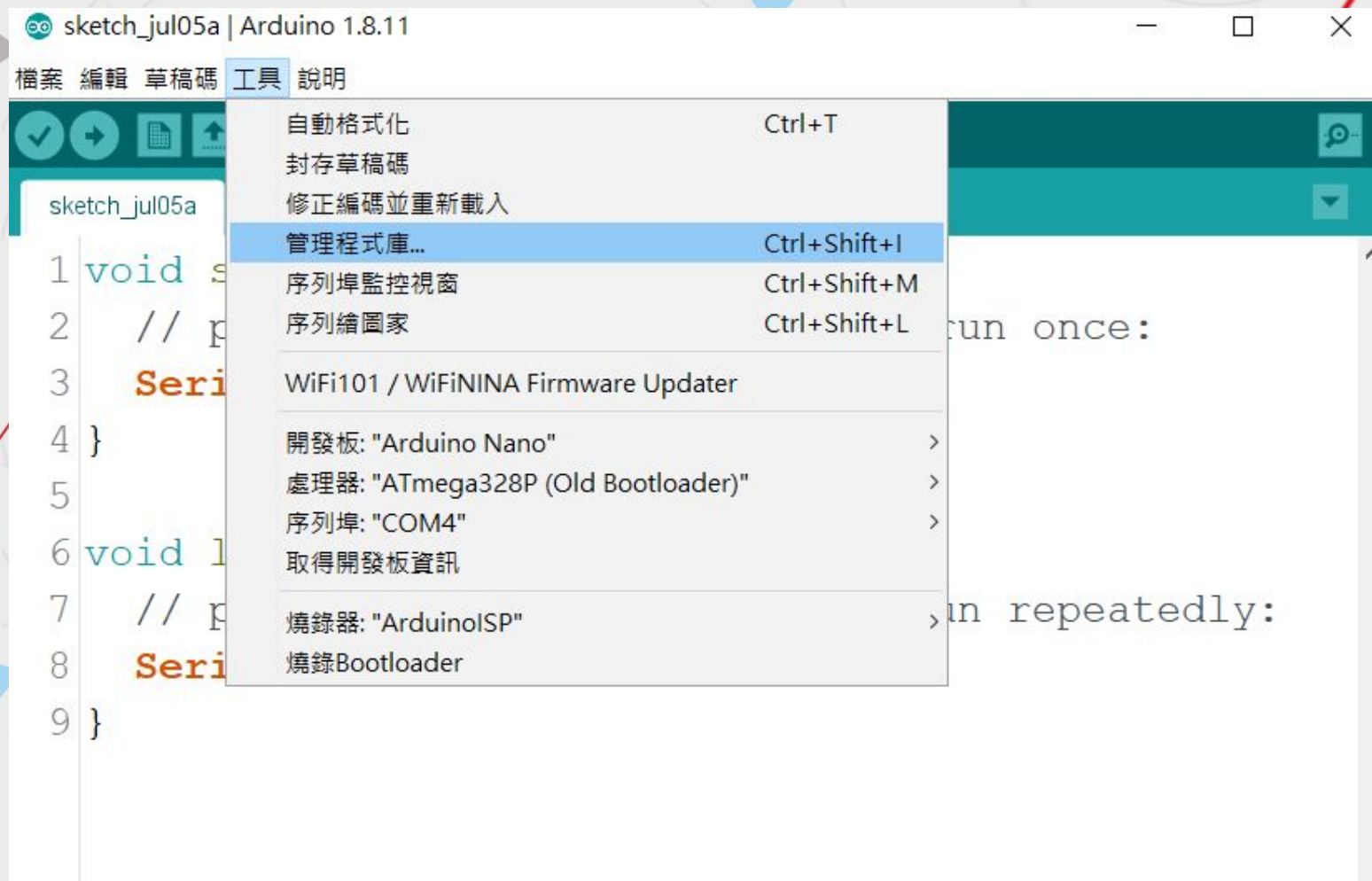
輸入數字並回傳階乘的值

提示：先定義回傳值的資料型態

Arduino使用

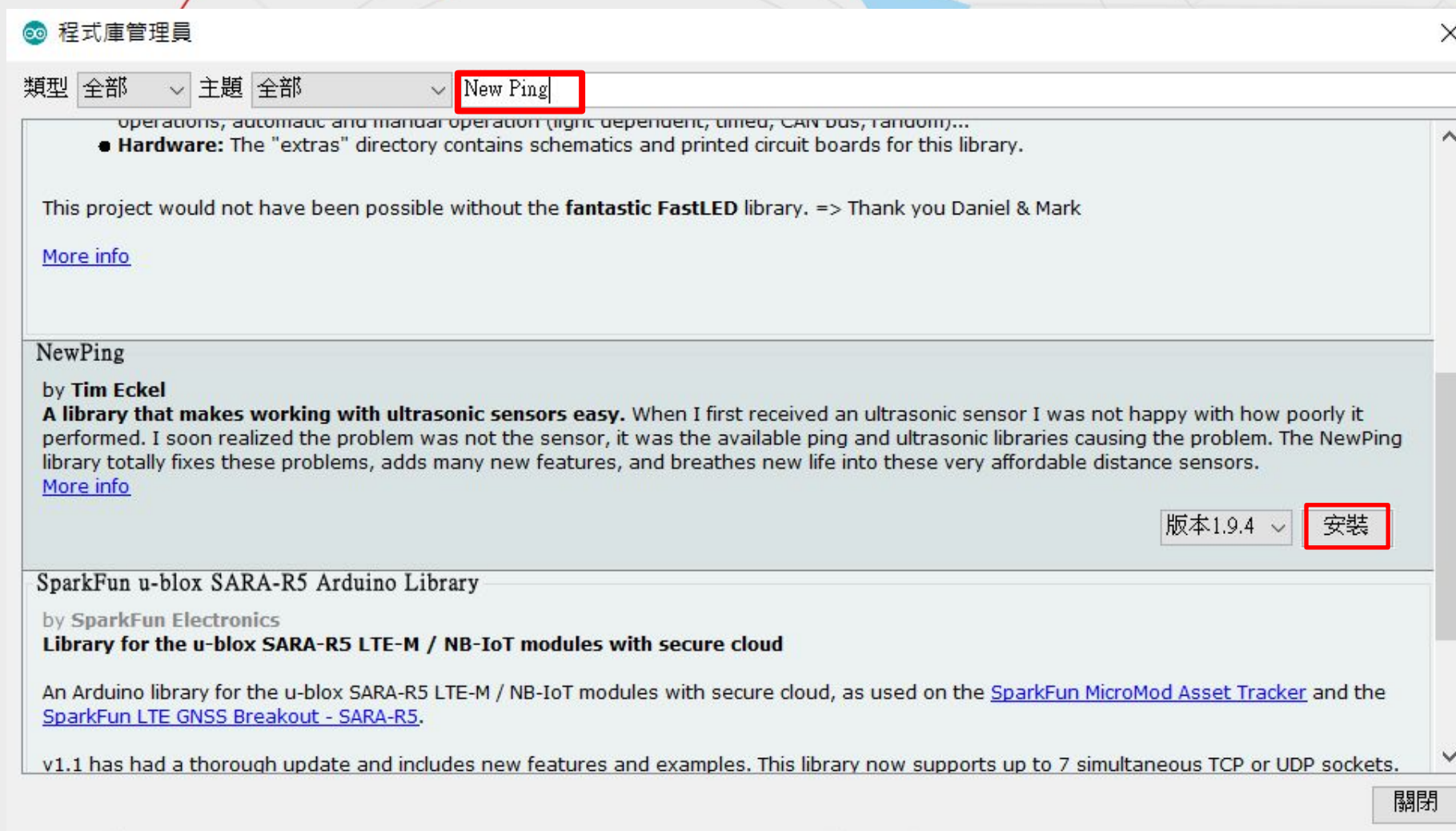
匯入草稿碼

開啟路徑：
工具
>>管理程式庫



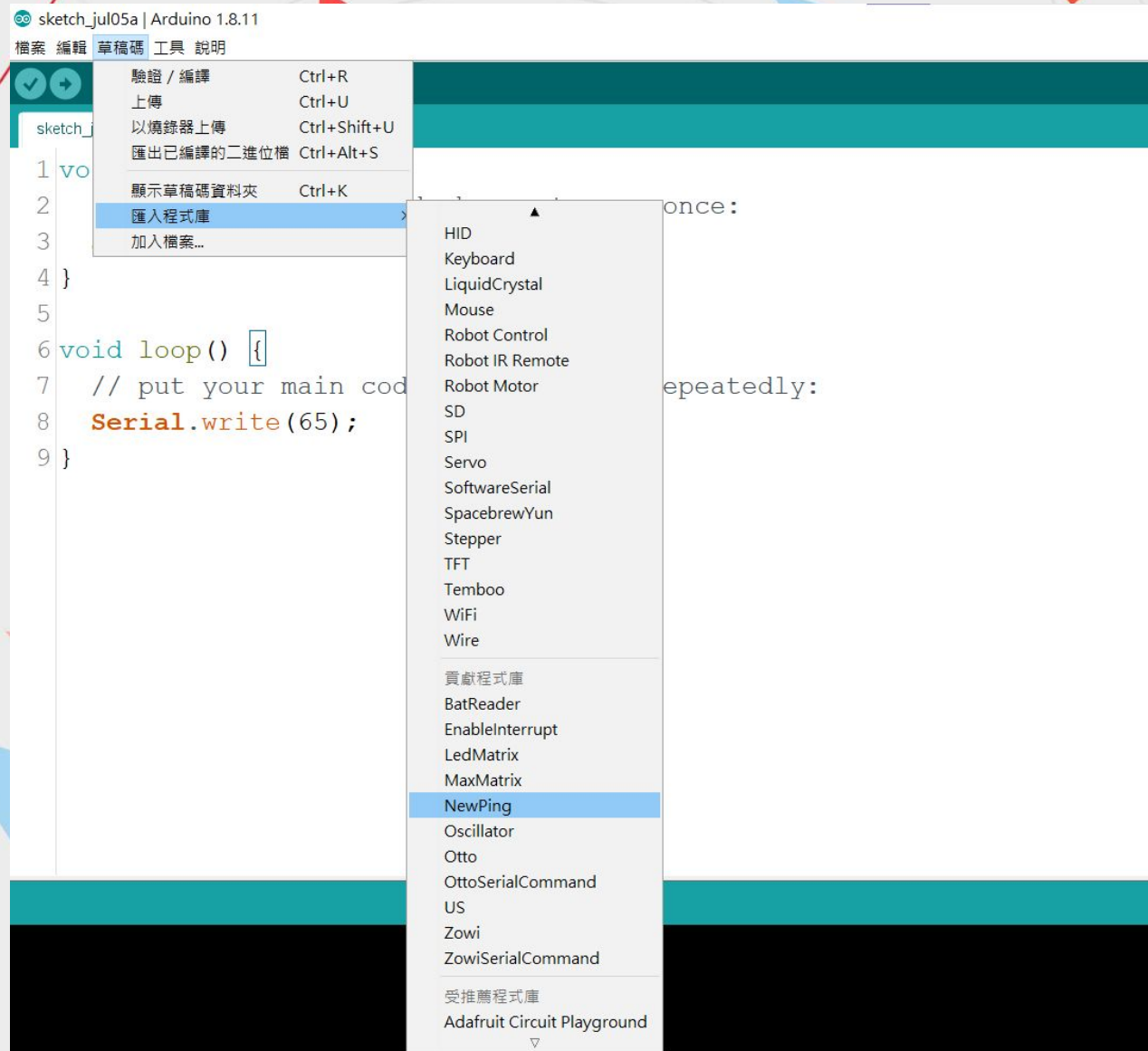
Arduino使用

下載草稿碼清單



Arduino使用

已匯入草稿碼清單



開啟路徑:
草稿碼
>>匯入程式庫
>>貢獻程式庫

Arduino語法介紹完

基本架構 資料型態 運算子 語法

