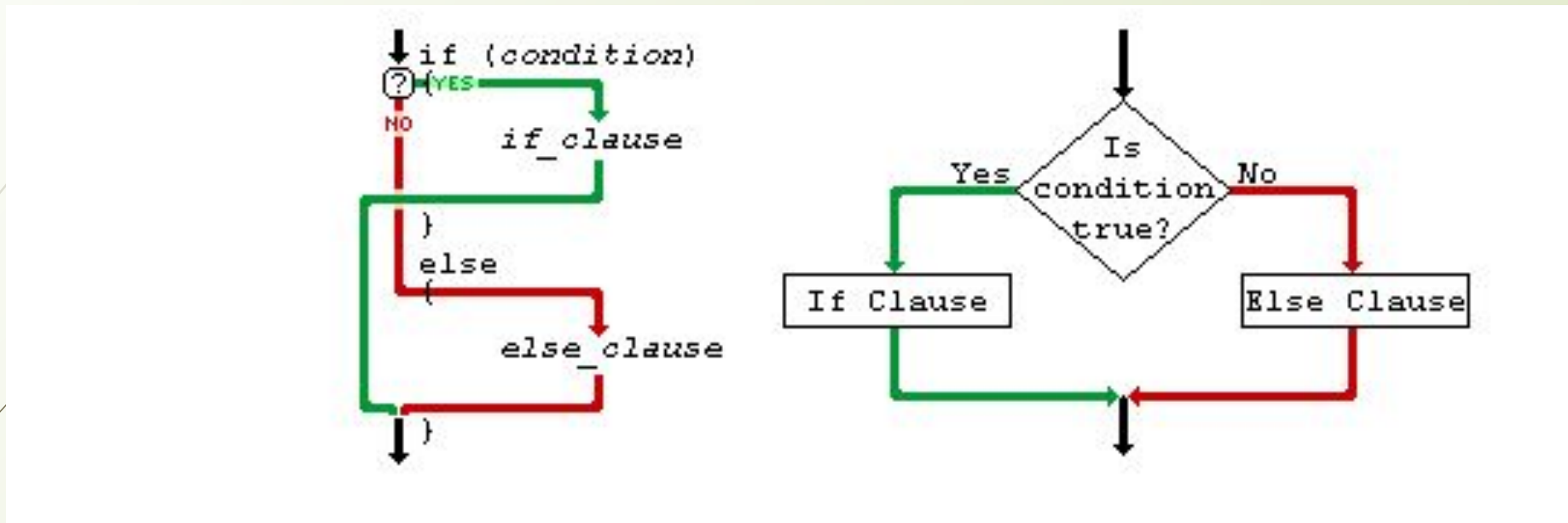


TEE3101
ไมโครคอนโทรลเลอร์และการควบคุมอัจฉริยะ
โครงสร้างการควบคุมของโปรแกรม Arduino

Tadchanon Chuman
Department of Electrical Technology, SSRU



โครงสร้างการควบคุมใช้เพื่อสร้างเงื่อนไขการดำเนินการ เช่น

- ให้ดำเนินการเมื่อเงื่อนไขที่กำหนดสอดคล้อง ไม่เช่นนั้นให้ไม่ดำเนินการใดๆ
- ให้เลือกดำเนินการอย่างใดอย่างหนึ่งที่อยู่ภายใต้เงื่อนไขที่สอดคล้อง
- ให้ดำเนินการซ้ำในจำนวนครั้งที่ต้องการเมื่อเงื่อนไขที่กำหนดสอดคล้อง

Outline

- โครงสร้างเงื่อนไข
 - if
 - if และ else
 - if และ else if
- โครงสร้างเงื่อนไขแบบเวียนซ้ำ
 - for
 - while
- โครงสร้างอื่น ๆ
 - break
 - switch

ตัวอย่าง Blink

เราจะใช้โครงสร้างตัวอย่าง Blink เพื่อศึกษาโครงสร้างการควบคุมต่างๆ

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3 }  
4 void loop() {  
5   digitalWrite(LED_BUILTIN, HIGH); // LED on  
6   delay(1000); // wait for a second  
7   digitalWrite(LED_BUILTIN, LOW); // LED off  
8   delay(1000); // wait for a second  
9 }
```

- **PinMode:** กำหนดค่า pin ที่ระบุให้ทำงานเป็น I/O
- **DigitalWrite:** ป้อนค่าแรงดัน HIGH/LOW ให้กับ pin เอาต์พุต
- **Delay:** ค้างสถานะของโปรแกรมไว้ตามเวลาที่กำหนด (หน่วยเป็น ms)

โครงสร้าง if

โครงสร้าง if เป็นโครงสร้างที่ให้ดำเนินการเมื่อเงื่อนไขที่กำหนดสอดคล้อง

```
1  int x = 3;
2  void setup() {
3      pinMode(LED_BUILTIN, OUTPUT);
4  }
5
6  void loop() {
7      if (x>5)
8      {
9          digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
10         delay(500); // wait for a second
11         digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
12         delay(500); // wait for a second
13     }
14 }
15
```

ตัวอักษรหลัง // คือ comment ไม่ใช่ code คำสั่ง ไม่มีผลกับการทำงานของบอร์ด Arduino

คำสั่งที่อยู่ใน { } หลัง if(...) จะทำงานเมื่อเงื่อนไขใน (...) เป็นจริง

โครงสร้าง if

เครื่องหมายเปรียบเทียบจำนวนที่สามารถใช้กับ if() มีดังตาราง

code	สัญลักษณ์ คณิตศาสตร์	ความหมาย
$x==y$	$x=y$	x มีค่าเท่ากับ y
$x!=y$	$x\neq y$	x มีค่าไม่เท่ากับ y
$x<y$	$x<y$	x มีค่าน้อยกว่า y
$x>y$	$x>y$	x มีค่ามากกว่า y
$x\leq y$	$x\leq y$	x มีค่าน้อยกว่าหรือเท่ากับ y
$x\geq y$	$x\geq y$	x มีค่ามากกว่าหรือเท่ากับ y

โครงสร้าง if

ให้ตัดแปลงโปรแกรมในสไลด์ที่ 4 ให้ได้เงื่อนไขดังนี้

group	เงื่อนไข
1	ไฟกระพริบเมื่อ x มีค่าไม่เกิน 1 เวลาติด/ดับ 0.1 วินาที
2	ไฟกระพริบเมื่อ x มีค่าไม่เท่ากับ 2 เวลาติด/ดับ 0.2 วินาที
3	ไฟกระพริบเมื่อ x มีค่าติดลบ เวลาติด/ดับ 0.2 วินาที
4	ไฟกระพริบเมื่อ x มีค่าไม่เป็นบวก เวลาติด/ดับ 0.2 วินาที

โครงสร้าง if และ else

โครงสร้าง if และ else เป็นโครงสร้างที่ให้ดำเนินการอย่างหนึ่งเมื่อเงื่อนไขสอดคล้อง และดำเนินการอีกอย่างเมื่อเงื่อนไขไม่สอดคล้อง

```
1  int x = 3;
2  void setup() {
3      pinMode(LED_BUILTIN, OUTPUT);
4  }
5  void loop() {
6      if (x>5)
7      {
8          digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
9          delay(500); // wait for a second
10         digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
11         delay(500); // wait for a second
12     }
13     else
14     {
15         digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
16     }
17 }
```

โครงสร้าง if และ else

ให้ตัดแปลงโปรแกรมในสไลด์ที่ 7 ให้ได้เงื่อนไขดังนี้

group	เงื่อนไข
1	ไฟกระพริบเมื่อ x มีค่าไม่เท่ากับ 3 ไม่เช่นนั้นให้ติดตลอด
2	ไฟกระพริบเมื่อ x มีค่าไม่เกิน 6 ไม่เช่นนั้นให้ติดตลอด
3	ไฟกระพริบเมื่อ x มีค่าไม่เป็นบวก ไม่เช่นนั้นให้ติดตลอด
4	ไฟกระพริบเมื่อ x มีค่าติดลบ ไม่เช่นนั้นให้ติดตลอด

โครงสร้าง if และ else if

โครงสร้าง if และ else if เป็นโครงสร้างแบบ 2 เงื่อนไขขึ้นไป โดยพิจารณาตามลำดับ เช่น ถ้า if() สอดคล้อง จะไม่สนใจเงื่อนไข else if()

```
1  int x = 6;
2  void setup() {
3      pinMode(LED_BUILTIN, OUTPUT);
4  }
5  void loop() {
6      if (x>5)
7      {
8          digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
9          delay(500); // wait for a second
10         digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
11         delay(500); // wait for a second
12     }
13     else if (x>=3)
14     {
15         digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
16     }
17     else
18     {
19         digitalWrite(LED_BUILTIN, LOW); // turn the LED off
20     }
21 }
22
```

โครงสร้าง if และ else

ให้ตัดแปลงโปรแกรมในสไลด์ที่ 9 ให้ได้เงื่อนไขดังนี้

group	เงื่อนไข
1	ไฟกระพริบเมื่อ $x=3$ ไม่เช่นนั้นให้ไฟติดตลอดเมื่อ $x>0$ ไม่เช่นนั้นให้ไฟดับตลอด
2	ไฟกระพริบเมื่อ $x\leq 6$ ไม่เช่นนั้นให้ไฟติดตลอดถ้า $x\leq 10$ ไม่เช่นนั้นให้ไฟดับตลอด
3	ไฟกระพริบเมื่อ $x\leq 0$ ไม่เช่นนั้นให้ไฟติดตลอดถ้า $x=0$ ไม่เช่นนั้นให้ไฟดับตลอด
4	ไฟกระพริบเมื่อ $x<0$ ไม่เช่นนั้นให้ไฟติดตลอดถ้า $x>0$ ไม่เช่นนั้นให้ไฟดับตลอด

โครงสร้าง for

โครงสร้าง for() จะเป็นคำสั่งให้ทำซ้ำคำสั่งใน { } หลัง for() ตราบเท่าที่เงื่อนไขใน () เป็นจริง มีรูปแบบการใช้คำสั่งดังนี้

for(ค่าเริ่มต้น, เงื่อนไข, การเพิ่มค่าต่อรอบ) เช่น

```
for(int i=0;i<3;i++)
```

```
{...}
```

- ❑ int i=0 หมายถึงกำหนดตัวแปร i เริ่มต้นเท่ากับ 0
- ❑ i<3 เงื่อนไขที่จะให้ดำเนินการคำสั่งใน {...}
- ❑ i++ คือเมื่อดำเนินการคำสั่งใน {...} 1 ครั้งแล้วให้ i เพิ่ม 1 ค่า

โครงสร้าง for

โครงสร้าง for() อยู่ใน loop() จะเป็นลักษณะของการทำซ้ำ ซ้อนกับคำสั่งทำซ้ำอีกที (ลอง upload code ด้านล่างแล้วสังเกตผล)

```
1 void setup() {
2   | pinMode(LED_BUILTIN, OUTPUT);
3 }
4 void loop() {
5   for (int i=0;i<3;i++)
6   {
7     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
8     delay(200); // wait for 0.2 sec
9     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
10    delay(200); // wait for 0.2 sec
11  }
12  delay(2000); // wait for 2 sec
13 }
```

โครงสร้าง for

ให้ตัดแปลงโปรแกรมในสไลด์ที่ 12 ให้ได้เงื่อนไขดังนี้

group	เงื่อนไข
1	ไฟกระพริบด้วยเวลาติด/ดับ 0.3 วินาที จำนวน 3 รอบ แล้วดับ 5 วินาที วนซ้ำไปเรื่อย ๆ
2	ไฟกระพริบด้วยเวลาติด/ดับ 0.1 วินาที จำนวน 6 รอบ แล้วดับ 3 วินาที วนซ้ำไปเรื่อย ๆ
3	ไฟกระพริบด้วยเวลาติด/ดับ 0.2 วินาที จำนวน 5 รอบ แล้วดับ 1 วินาที วนซ้ำไปเรื่อย ๆ
4	ไฟกระพริบด้วยเวลาติด/ดับ 0.4 วินาที จำนวน 4 รอบ แล้วดับ 4 วินาที วนซ้ำไปเรื่อย ๆ

โครงสร้าง while

โครงสร้าง while() จะมีลักษณะการใช้งานคล้าย for() แต่ต่างกันที่กำหนดค่าเริ่มต้นก่อน while() และกำหนดการเพิ่มค่าที่บรรทัดสุดท้ายใน { }

```
1 void setup() {
2   |   pinMode(LED_BUILTIN, OUTPUT);
3 }
4 void loop() {
5   int i=0;
6   while (i<3)
7   {
8     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
9     delay(200); // wait for 0.2 sec
10    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
11    delay(200); // wait for 0.2 sec
12    i++;
13  }
14  delay(2000); // wait for 2 sec
15 }
16
```

โครงสร้าง while

ให้ตัดแปลงโปรแกรมในสไลด์ที่ 14 ให้ได้เงื่อนไขดังนี้

group	เงื่อนไข
1	ไฟกระพริบด้วยเวลาติด/ดับ 0.1 วินาที จำนวน 6 รอบ แล้วดับ 3 วินาที วนซ้ำไปเรื่อย ๆ
2	ไฟกระพริบด้วยเวลาติด/ดับ 0.2 วินาที จำนวน 5 รอบ แล้วดับ 1 วินาที วนซ้ำไปเรื่อย ๆ
3	ไฟกระพริบด้วยเวลาติด/ดับ 0.4 วินาที จำนวน 4 รอบ แล้วดับ 4 วินาที วนซ้ำไปเรื่อย ๆ
4	ไฟกระพริบด้วยเวลาติด/ดับ 0.3 วินาที จำนวน 3 รอบ แล้วดับ 5 วินาที วนซ้ำไปเรื่อย ๆ

โครงสร้าง break

โครงสร้าง break ใช้สำหรับออกจาก for หรือ while ก่อนเงื่อนไขที่กำหนดไว้ มักจะใช้คู่กับ if() เพื่อกำหนดเงื่อนไข

ว่าตอนไหนจะ break

```
1 void setup() {
2   |   pinMode(LED_BUILTIN, OUTPUT);
3 }
4 void loop() {
5   for(int i=0;i<10;i++)
6   {
7     digitalWrite(LED_BUILTIN, HIGH);
8     delay(200);
9     digitalWrite(LED_BUILTIN, LOW);
10    delay(200);
11  }
12  delay(2000);
13 }
14
```

```
1 void setup() {
2   |   pinMode(LED_BUILTIN, OUTPUT);
3 }
4 void loop() {
5   for(int i=0;i<10;i++)
6   {
7     if(i==5) break;
8     digitalWrite(LED_BUILTIN, HIGH);
9     delay(200);
10    digitalWrite(LED_BUILTIN, LOW);
11    delay(200);
12  }
13  delay(2000);
14 }
15
```

โครงสร้าง switch

โครงสร้าง switch() จะมีลักษณะคล้าย if และ else if แต่เงื่อนไขแต่ละกรณีจะเป็นตัวเลข 1,2,3... เท่านั้น

```
switch(i)
{
  case 1: (เงื่อนไขนี้ทำงานเมื่อค่า i=1)
    คำสั่งต่าง ๆ สำหรับ case 1
    break;
  case 2: (เงื่อนไขนี้ทำงานเมื่อค่า i=2)
    คำสั่งต่าง ๆ สำหรับ case 2
    break;
  case 3: (เงื่อนไขนี้ทำงานเมื่อค่า i=3)
    คำสั่งต่าง ๆ สำหรับ case 3
    break;
}
```

หลังจากชุดคำสั่งของแต่ละ case จะต้องใส่ break; ด้วยเสมอ

โครงสร้าง switch-case

```
1 void setup() {
2   |   pinMode(LED_BUILTIN, OUTPUT);
3 }
4 void loop() {
5   int i=3;
6   switch(i)
7   {
8     case 1:
9       digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
10      delay(200); // wait for 0.2 sec
11      digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
12      delay(200); // wait for 0.2 sec
13      break;
14     case 2:
15      digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
16      delay(1000); // wait for 1 sec
17      digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
18      delay(1000); // wait for 1 sec
19      break;
20     case 3:
21      digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
22      break;
23   }
24 }
```

โครงสร้าง switch-case

ให้แต่ละกลุ่มดัดแปลงโปรแกรมในสไลด์ที่ 18 โดยให้แบ่งเป็น 3 กรณีดังนี้

- case 1 ให้ไฟกระพริบไปเรื่อยๆ โดยให้ไฟติด 0.3 วินาที และให้ไฟดับ 0.1 วินาที
- case 2 ให้ไฟกระพริบไปเรื่อยๆ โดยให้ไฟติด 0.1 วินาที และให้ไฟดับ 0.3 วินาที
- case 3 ให้ไฟกระพริบติด/ดับ 0.1 วินาที แล้วให้ติด/ดับ 0.5 วินาที สลับกันไปเรื่อยๆ

โครงสร้างผสม

กำหนดให้ x เป็นตัวเลขจำนวนเต็ม

ให้ใช้โครงสร้างการควบคุมต่าง ๆ (เช่น if, else, else if, for, while, switch) เขียนโปรแกรมให้ไฟกะพริบตามเงื่อนไขต่อไปนี้

- ถ้า $x=0$ ให้ไฟกะพริบด้วยเวลาติด/ดับ 0.1 วินาที จำนวน 5 รอบ แล้วดับ 1 วินาที วนซ้ำไปเรื่อย ๆ
- ถ้า $x=1$ ให้ไฟกะพริบด้วยเวลาติด/ดับ 0.2 วินาที จำนวน 4 รอบ แล้วดับ 2 วินาที วนซ้ำไปเรื่อย ๆ
- ถ้า $x<0$ ให้ไฟดับตลอด
- ถ้า $x>1$ ให้ไฟติดตลอด