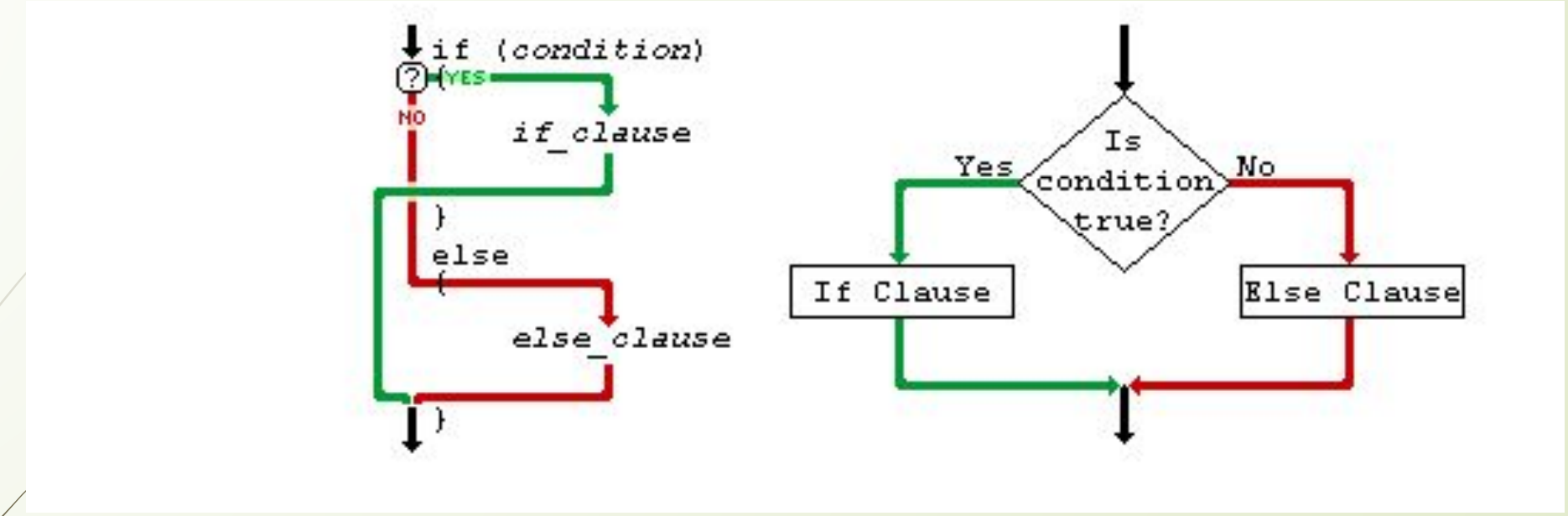
TEE3101
Microcontroller and Smart Control

Control Sructure of Arduino

Tadchanon Chuman
Department of Electrical Technology, SSRU



Control structures are used to define execution conditions, such as:

- Executing a set of instructions when a specified condition is satisfied; otherwise, no action is performed.
- Selecting and executing one operation from multiple alternatives based on the satisfied condition.
- Repeating a set of instructions for a specified number of times or while a given condition remains satisfied.

Outline

- **Condition Structures**
  - if
  - if and else
  - if and else if
- **Loop Condition Structures**
  - for
  - while
- **Other Condition Structures**
  - break
  - switch

Blink Example

The Blink example is used as a reference to study various types of control structures.

```
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**: Configures the specified pin to operate as an input or output.
- **digitalWrite**: Sets the output voltage of a pin to either HIGH or LOW.
- **delay**: Pauses the program execution for a specified period of time (in milliseconds).

if Sructure

The if structure executes a statement only when the specified condition is true.

```
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
```

Text following // is a comment, not an executable command, and does not affect the operation of the Arduino board.

Statements inside the braces { } after if(...) are executed when the condition in ( ) is true.

if Sructure

The comparison operators that can be used with if() are shown in the following table.

code	Symbol	Meaning
x==y	x=y	x is equal to y
x!=y	x≠y	x is not equal to y
x<y	x<y	x is less than y
x>y	x>y	x is grater than y
x<=y	x≤y	x is not grater than y
x>=y	x≥y	x is not less than y

if Sructure: Exercise 1

Modify the program shown on slide 5 to meet the following conditions.

Group	Condition	
1	LED blinks when x ≤ 1	ON/OFF interval of 0.3 sec
2	LED blinks when x ≠ 2	ON/OFF interval of 0.6 sec
3	LED blinks when x < 0	ON/OFF interval of 0.4 sec
4	LED blinks when x ≤ 0	ON/OFF interval of 0.5 sec
5	LED blinks when x > 0	ON/OFF interval of 0.7 sec

if and else Sructure

The if–else structure executes one action when the condition is true and another action when the condition is false.

```
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  }
```

if and else Sructure: Exercise 2

Modify the program shown on slide 8 to meet the following conditions.

Group	Condition
1	LED blinks when x ≠ 7; otherwise, it remains ON continuously.
2	LED blinks when x ≤ 16; otherwise, it remains ON continuously.
3	LED blinks when x ≤ 0; otherwise, it remains ON continuously.
4	LED blinks when x < 0; otherwise, it remains ON continuously.
5	LED blinks when x > 40; otherwise, it remains ON continuously.

if and else if Sructure

The if–else if structure supports two or more conditions evaluated sequentially; once an if() condition is true, the subsequent else if() conditions are not evaluated.

```
int x = 6;
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  if (x>5)
  {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
 }
  else if (x>=3)
  {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
  }
  else
  {
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off
  }
}

```

if and else if Sructure : Exercise 3

Modify the program in slide 10 to satisfy the following conditions.

Group	Condition
1	Blink when x = 3; ON when x > 0; otherwise OFF
2	Blink when x ≤ 6; ON when x ≤ 10; otherwise OFF
3	Blink when x < 0; ON when x = 0; otherwise OFF
4	Blink when x < 0; ON when x > 0; otherwise OFF
5	Blink when x > 0; ON when x < 0; otherwise OFF

for Sructure

The for() structure is used to repeatedly execute the statements inside { } as long as the condition in ( ) is true.

Its general format is e.g.,

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

- int i = 0 initializes the variable i with a starting value of 0
- i < 3 is the condition that allows the statements inside { } to execute
- i++ increments i by 1 after each loop iteration

for Sructure

When a for() structure is placed inside the loop() function, it results in nested repetition, where one loop operates inside another. (Upload the code below and observe the output.)

```
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  for (int i=0;i<3;i++)
  {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(200);                       // wait for 0.2 sec
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(200);                       // wait for 0.2 sec
 }
  delay(2000); // wait for 2 sec
}
```

for Sructure: Exercise 4

Modify the program in slide 13 to satisfy the following conditions.

Group	Condition
1	Blink (1 s ON/OFF) × 4, OFF 5 s, repeat
2	Blink (2 s ON/OFF) × 3, OFF 5 s, repeat
3	Blink (3 s ON/OFF) × 2, OFF 5 s, repeat
4	Blink (4 s ON/OFF) × 2, OFF 5 s, repeat
5	Blink (3 s ON/OFF) × 3, OFF 5 s, repeat

while Sructure

The while() structure works similarly to for(), but the initial value is defined before while(), and the increment is specified at the last line inside { }.

```
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
```

while Sructure: Exercise 5

Modify the program in slide 15 to satisfy the following conditions.

Group	Condition
1	Blink (3 s ON/OFF) × 3, OFF 5 s, repeat
2	Blink (1 s ON/OFF) × 4, OFF 5 s, repeat
3	Blink (2 s ON/OFF) × 3, OFF 5 s, repeat
4	Blink (3 s ON/OFF) × 2, OFF 5 s, repeat
5	Blink (4 s ON/OFF) × 2, OFF 5 s, repeat

break Sructure

The break statement is used to exit a for or while loop before the specified condition is completed. It is commonly used with if() to define when the loop should terminate.

```
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
```

switch-case Structure

The switch() structure is similar to if and else if, but each condition is evaluated using fixed numeric cases such as 1, 2, 3, and so on.

```
switch(i)
{
    case 1:   // Executed when i = 1
        // Statements for case 1
        break;

    case 2:   // Executed when i = 2
        // Statements for case 2
        break;

    case 3:   // Executed when i = 3
        // Statements for case 3
        break;
}
```

A break; statement must always be included after the statements of each case.

switch-case Structure

```
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 }
```

switch-case Structure: Exercise 6

Each group is required to modify the program shown on slide 18 by defining the following three cases:

- Case 1: The LED blinks continuously with 0.3 s ON and 0.1 s OFF.
- Case 2: The LED blinks continuously with 0.1 s ON and 0.3 s OFF.
- Case 3: The LED alternates between 0.1 s ON/OFF and 0.5 s ON/OFF continuously.

Mixed Structure: Exercise 7

Let x be an integer variable.

Using control structures (such as if, else, else if, for, while, and switch), write a program to control the LED according to the following conditions:

- If x = 0: Blink the LED with 0.1 s ON/OFF for 5 cycles, then OFF for 1 s, and repeat continuously.
- If x = 1: Blink the LED with 0.2 s ON/OFF for 4 cycles, then OFF for 2 s, and repeat continuously.
- If x < 0: The LED remains OFF continuously.
- If x > 1: The LED remains ON continuously.