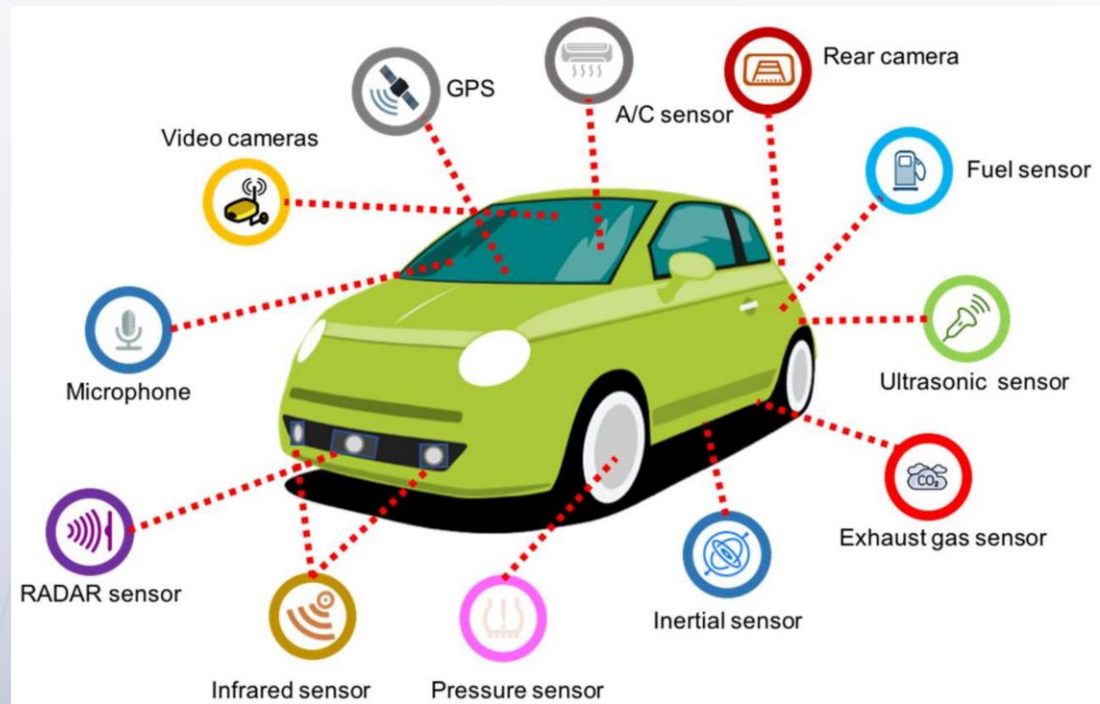


MICROCONTROLLER FOR AUTOMOTIVE INDUSTRY

4. while-loop and for-loop



อลงกรณ์ ปิระวารการณ (alongkorn.p@eat.kmutnb.ac.th)

NEXT GENERATION AUTOMOTIVE AND AUTOMATION SYSTEM TECHNOLOGY (NAAT)



Content & Schedule (Midterm)

Week	Theory	Lab.
1	- เนื้อหาและรายละเอียดการวัดผลต่างๆ ของรายวิชา - แนะนำตัวคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรม	-
2	- การเขียนโปรแกรมด้วยภาษาไพทอนขั้นพื้นฐาน ชนิดของตัวแปรการกำหนดค่าตัวแปร - ตัวดำเนินการทางคณิตศาสตร์ (+ - * / // % **) - ประเภทของข้อผิดพลาดในการทำงานของโปรแกรม (Type Error, Syntax Error, Name Error) - การรับค่าอินพุตจากผู้ใช้งาน และการแสดงผลค่าเอาต์พุต	ใบงานที่ 1
3	- การสร้างฟังก์ชันและการเรียกใช้งานฟังก์ชัน - การใช้คำสั่งป้องกันข้อผิดพลาด (Handling Exceptions) ด้วยคำสั่ง try-except	ใบงานที่ 2
4	- ตัวเปรียบเทียบ (Comparison Operators < <= > >= == != is is-not) - ตัวดำเนินการชนิดบูลีน (Boolean) สำหรับการเปรียบเทียบ (True และ False) - ตัวดำเนินการด้านลอจิก (Logical Operators) ด้วยคำสั่งต่าง (and or not) - การสร้างฟังก์ชันตรวจสอบเงื่อนไข (Statement) ด้วยคำสั่ง if, if-else, if-elif-else	ใบงานที่ 3
5	- การสร้างคำสั่งวนลูปด้วยคำสั่ง for, while	ใบงานที่ 4
6	- ประเภทและการสร้างตัวแปรชนิดอาร์เรย์ (Array : List, Tuple, Set, Dictionary)	ใบงานที่ 5
7	- การเรียกใช้ไลบรารีภายนอกด้วยคำสั่ง import เช่น math, time, random - การสร้างคำสั่งวนลูปซ้อนลูป (Nested Loops) ด้วยคำสั่ง while หรือ for	ใบงานที่ 6
8	- ระบบเลขฐานและการแปลงเลขฐาน (Number System) - หลักการเชื่อมต่อบอร์ด ESP32 กับคอมพิวเตอร์ - หลักการเขียนโปรแกรมเพื่อสั่งเปิด-ปิดหลอดไฟผ่านบอร์ด ESP32 ด้วยภาษาไพทอน	ใบงานที่ 7
9	สอบกลางภาค	- เรียนรู้ด้วยตนเอง และทบทวนเนื้อหาก่อนหน้า



Content & Schedule (Final)

Week	Theory	Lab.
10	- หลักการเขียนโปรแกรมเพื่อรับค่าจากสวิตช์ผ่านบอร์ด ESP32 ด้วยภาษาไพทอน - หลักการเขียนโปรแกรมไพกระพริบและไฟวิ่งผ่านบอร์ด ESP32 ด้วยภาษาไพทอน	ใบงานที่ 8
11	- หลักการทำงานของจอแอลซีดี (LCD) และ เขียนโปรแกรมเพื่อสั่งงานจอแอลซีดีผ่านบอร์ด ESP32 - หลักการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) และเขียนโปรแกรมเพื่อรับค่าจากเซ็นเซอร์ตรวจจับการเคลื่อนไหว	ใบงานที่ 9
12	- หลักการทำงานของคีย์สวิตช์ (Key Switch) - หลักการเขียนโปรแกรมเพื่อสั่งงานบอร์ด ESP32 ด้วยคีย์สวิตช์	ใบงานที่ 10
13	- หลักการแปลงสัญญาณระหว่าง อนาล็อก และ ดิจิตอล - หลักการทำงานของอัลตราโซนิกเซ็นเซอร์ (Ultrasonic Sensor) - หลักการเขียนโปรแกรมควบคุมระบบผ่านบอร์ด ESP32 โดยรับค่าจากอัลตราโซนิกเซ็นเซอร์	ใบงานที่ 11
14	- ความหมายและหลักการของสัญญาณคลื่น (PWM) - หลักการทำงานของมอเตอร์กระแสตรง และ หลักการสั่งงานมอเตอร์กระแสตรงด้วยสัญญาณคลื่น - หลักการเขียนโปรแกรมควบคุมหลอดไฟหรือมอเตอร์กระแสตรงผ่าน ESP32	ใบงานที่ 12
15	- หลักการทำงานของเซอร์โวมอเตอร์ (Servo Motor) และ สเต็ปป์มอเตอร์ (Stepping Motor) - หลักการควบคุม เซอร์โวมอเตอร์และสเต็ปป์มอเตอร์ และเขียนโปรแกรมควบคุมเซอร์โวมอเตอร์และสเต็ปป์มอเตอร์ ผ่าน ESP32	ใบงานที่ 13
16	- ความหมายและความสำคัญของอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things: IoT) - การประยุกต์ใช้งานอินเทอร์เน็ตของสรรพสิ่ง - หลักการเขียนโปรแกรมเพื่อควบคุมระบบผ่านเครือข่ายอินเทอร์เน็ตของสรรพสิ่งด้วย ESP32	-
17	สอบปลายภาค	สอบปฏิบัติปลายภาค



Loop-Conditions

การวนลูปมีหลายแบบ โดยมีวัตถุประสงค์เหมือนกัน นั่นก็คือการตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่? ถ้าเงื่อนไขยังคงเป็นจริงจะทำการวนลูปต่อไป แต่ถ้าเงื่อนไขมีผลเป็นเท็จก็จะหยุดการทำงานและออกนอกลูปทันที

1. while-Loop

1.1 forever-while loop (เงื่อนไขเป็นจริงเสมอ)

1.2 incremental variable-while loop (การ เพิ่ม หรือ ลด ค่าของตัวแปรที่ใช้ตรวจสอบเงื่อนไข)

1.3 Boolean comparison-while loop (การตรวจสอบเงื่อนไขบูลีน)

2. for-loop

- Start value
- Stop value
- Step value



1.1 Forever-while loop condition

- forever-while loop เป็นลูปที่มีเงื่อนไขที่เป็นจริงเสมอทำให้เป็นการรันแบบไม่รู้จบ ซึ่งจะสามารถหยุดการทำงานและออกนอกลูปได้โดยใช้คำสั่ง *break*

```
while (True):  
    print("Hi")
```

```
1 i=0  
2 while (True):  
3     print("Hi")  
4     if i>3:  
5         break  
6     i+=1
```

1.1 Forever-while loop condition

- forever-while loop เป็นลูปที่มีเงื่อนไขที่เป็นจริงเสมอ ซึ่งจะสามารถออกนอกลูปได้โดยใช้คำสั่ง *break* ซึ่งจะใช้กี่ครั้งก็ได้

```
1 while (True):
2     name = input("What is your major?: ")
3     if name == "NAAT":
4         print("I found a student from", name)
5         break
6     elif name == "IAet":
7         print("I found a student from", name)
8         break
9     else:
10        print("Sorry", name, "is not my student.")
```

```
What is your major?: CPet
Sorry CPet is not my student.
What is your major?: MAet
Sorry MAet is not my student.
What is your major?: ILet
Sorry ILet is not my student.
What is your major?: MPet
Sorry MPet is not my student.
What is your major?: NAAT
I found a student from NAAT
```



1.2 Incremental/Decremental variable-while loop

- Incrementing การเพิ่มค่าของตัวแปรที่ใช้ในการเปรียบเทียบเงื่อนไข
 - 1. กำหนดค่าเริ่มต้นของโปรแกรม
 - 2. กำหนดค่าการเพิ่มขึ้นของค่าภายในตัวแปร เช่น $a += 1$ หรือ $a = a + 1$, $a *= 10$
- Decrementing การลดค่าของตัวแปรที่ใช้ในการเปรียบเทียบเงื่อนไข
 - 1. กำหนดค่าเริ่มต้นของโปรแกรม
 - 2. กำหนดค่าการลดลงของค่าภายในตัวแปร เช่น $b -= 1$ หรือ $b = b - 1$, $b /= 20$



Incremental variable-while loop

```
1 num = 0
2 while num < 3:
3     print("NAAT")
4     num = num + 1
```

>>> %Run Test.py

NAAT
NAAT
NAAT

```
1 num = 0
2 while num < 3:
3     print("NAAT")
4     num += 1
```

>>> %Run Test.py

NAAT
NAAT
NAAT



Incremental variable-while loop

```
1 num = 0
2 while num < 3:
3     print("NAAT")
4     num += 1
```

>>> %Run Test.py

NAAT
NAAT
NAAT

```
1 num = 0
2 while num < 3:
3     print("NAAT", num)
4     num += 1
```

>>> %Run Test.py

NAAT 0
NAAT 1
NAAT 2



Incremental variable-while loop

```
1 num = 1
2 while num < 3:
3     print("NAAT", num)
4     num += 1
```

```
>>> %Run Test.py
NAAT 1
NAAT 2
```

```
1 num = 1
2 while num <= 3:
3     print("NAAT", num)
4     num += 1
```

```
>>> %Run Test.py
NAAT 1
NAAT 2
NAAT 3
```



Incremental variable-while loop

```
1 num = 0
2 while num < 3:
3     print("NAAT")
4     num += 1
```

```
>>> %Run Test.py
```

```
NAAT
NAAT
NAAT
```

```
1 num = 0
2 while num < 3:
3     print("NAAT", num)
4     num += 1
```

```
>>> %Run Test.py
```

```
NAAT 0
NAAT 1
NAAT 2
```



Decremental variable-while loop

```
1 num = 3
2 while num >= 0:
3     print("NAAT")
4     num = num - 1
```

>>> %Run Test.py

NAAT
NAAT
NAAT
NAAT

```
1 num = 3
2 while num >= 0:
3     print("NAAT")
4     num -= 1
```

>>> %Run Test.py

NAAT
NAAT
NAAT
NAAT



Decremental variable-while loop

```
1 num = 3
2 while num > 0:
3     print("NAAT", num)
4     num -= 1
```

>>> %Run Test.py

```
NAAT 3
NAAT 2
NAAT 1
```

```
1 num = 3
2 while num >= 0:
3     print("NAAT", num)
4     num -= 1
```

>>> %Run Test.py

```
NAAT 3
NAAT 2
NAAT 1
NAAT 0
```



Error-loop (Be careful !!!)

```
1 while num > 0:  
2     print("NAAT", num)  
3     num -= 1
```

```
while num > 0:
```

```
NameError: name 'num' is not defined
```

Infinity-loop (Be careful !!!)

```
1 num = 1
2 while num > 0:
3     print("NAAT", num)
4     num += 1
```

```
>>> %Run Test.py
```

```
NAAT 1
NAAT 2
NAAT 3
NAAT 4
...
NAAT 185
NAAT 186
NAAT 187
```

```
1 num = 1
2 while num > 0:
3     print("NAAT", num)
```

```
>>> %Run Test.py
```

```
NAAT 1
NAAT 1
NAAT 1
NAAT 1
NAAT 1
```



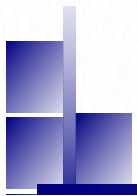
Unoperated-loop (Be careful !!!)

```
1 num = 1
2 while num < 0:
3     print("NAAT", num)
4     num += 1
```

```
>>> %Run Test.py
>>>
```

```
1 num = 1
2 while num < 0:
3     print("NAAT", num)
4     num -= 1
```

```
>>> %Run Test.py
>>>
```



1.3 Boolean comparison-while loop

and

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

or

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

not

A	not A
True	False
False	True

```
1 while True:
2     num = int(input("Number 0 - 10: "))
3     if num % 2 == 0 and num % 3 == 0:
4         print("num is 6")
5         break
6
7     elif num % 2 == 0 and num % 5 == 0:
8         print("num is 10")
9         break
10
11    else:
12        print("please continue")
```

```
>>> %Run Test.py
Number 0 - 10: 1
please continue
Number 0 - 10: 3
please continue
Number 0 - 10: 6
num is 6
>>>
```

```
>>> %Run Test.py
Number 0 - 10: 3
please continue
Number 0 - 10: 5
please continue
Number 0 - 10: 10
num is 10
>>>
```



while-loops conditions with *continue* & *break*

```
1 num = 0
2 while True:
3     if num % 2 == 0 :
4         print("Even Number", num)
5     else:
6         print("Odd Number", num)
7
8     if num >= 5:
9         print("Stop")
10        break
11    else:
12        num += 1
13
14    continue
    print("Continue")
```

continue ใช้สำหรับต้องการข้ามไปรันรอบใหม่ โดยไม่รันบรรทัดด้านล่าง
print("Continue") เมื่อมีคำสั่ง continue อยู่ด้านบน บรรทัดนี้จะไม่รัน

```
>>> %Run Test.py
```

```
Even Number 0
Odd Number 1
Even Number 2
Odd Number 3
Even Number 4
Odd Number 5
Stop
```

2. for-loop conditions : in range()

- Using function “in range” การสั่งให้ประมวลผลค่าคงที่ตามช่วงที่กำหนดไว้
- สั่งด้วยคำสั่ง `for a in range(start, stop, step)` : ยกตัวอย่างเช่น
 - `for a in range(stop)` เช่น `for a in range(10)`: → $a = 0, 1, 2, \dots, 9$
 - `for a in range(start, stop)` เช่น `for a in range(5, 10)`: → $a = 5, 6, \dots, 9$
 - `for a in range(start, stop, step)` เช่น `for a in range(5, 10, 2)`: → $a = 5, 7, 9$

```
1 for i in range(2):  
2     print("NAAT")
```

```
>>> %Run Test.py
```

```
NAAT  
NAAT
```

```
1 for i in range(0, 2):  
2     print("NAAT")
```

```
>>> %Run Test.py
```

```
NAAT  
NAAT
```

```
1 for i in range(0, 2, 1):  
2     print("NAAT")
```

```
>>> %Run Test.py
```

```
NAAT  
NAAT
```

2. for-loop conditions : in range()

```
1 for i in range(5):  
2     print("NAAT", i)
```

```
>>> %Run Test.py
```

```
NAAT 0  
NAAT 1  
NAAT 2  
NAAT 3  
NAAT 4
```

```
1 num = 5  
2 for i in range(num):  
3     print("NAAT", i)
```

```
>>> %Run Test.py
```

```
NAAT 0  
NAAT 1  
NAAT 2  
NAAT 3  
NAAT 4
```

เริ่มต้นที่ 0 และ จบการทำงานที่ คำน้อยกว่าตัวเลข ($i < 5$)

2. for-loop conditions : in range()

```
1 for i in range(0, 5):  
2     print("NAAT", i)
```

>>> %Run Test.py

```
NAAT 0  
NAAT 1  
NAAT 2  
NAAT 3  
NAAT 4
```

เริ่มต้นที่ 0 และ จบการทำงานที่ คำน้อยกว่าตัวเลข ($i < 5$)

```
1 for i in range(2, 5):  
2     print("NAAT", i)
```

>>> %Run Test.py

```
NAAT 2  
NAAT 3  
NAAT 4
```

เริ่มต้นที่ 2 และ จบการทำงานที่ คำน้อยกว่าตัวเลข ($i < 5$)

2. for-loop conditions : in range()

```
1 for i in range(0, 7):  
2     print("NAAT", i)
```

>>> %Run Test.py

```
NAAT 0  
NAAT 1  
NAAT 2  
NAAT 3  
NAAT 4  
NAAT 5  
NAAT 6
```

เริ่มต้นที่ 0 และ จบการทำงานที่ คำน้อยกว่าตัวเลข ($i < 7$)

```
1 for i in range(2, 7):  
2     print("NAAT", i)
```

>>> %Run Test.py

```
NAAT 2  
NAAT 3  
NAAT 4  
NAAT 5  
NAAT 6
```

เริ่มต้นที่ 2 และ จบการทำงานที่ คำน้อยกว่าตัวเลข ($i < 7$)

2. for-loop conditions : in range()

```
1 for i in range(0, 10, 1):  
2     print("NAAT", i)
```

```
>>> %Run Test.py
```

```
NAAT 0  
NAAT 1  
NAAT 2  
NAAT 3  
NAAT 4  
NAAT 5  
NAAT 6  
NAAT 7  
NAAT 8  
NAAT 9
```

เริ่มต้นที่ 0 และ จบการทำงานที่ 10
และเพิ่มค่าทีละ 1
0,1,2,3,4,5,6,7,8,9

```
1 for i in range(0, 10, 2):  
2     print("NAAT", i)
```

```
>>> %Run Test.py
```

```
NAAT 0  
NAAT 2  
NAAT 4  
NAAT 6  
NAAT 8
```

เริ่มต้นที่ 0 และ จบการทำงานที่ 10
และเพิ่มค่าทีละ 2
0,2,4,6,8

```
1 for i in range(0, 10, 3):  
2     print("NAAT", i)
```

```
>>> %Run Test.py
```

```
NAAT 0  
NAAT 3  
NAAT 6  
NAAT 9
```

เริ่มต้นที่ 0 และ จบการทำงานที่ 10
และเพิ่มค่าทีละ 3
0,3,6,9

Comparison of while-loop & for-loop conditions

```
k = 0
while k < 100 :
    ...
    k += 1
```

```
k = 5
while k < 100 :
    ...
    k += 1
```

```
k = 4
while k < 100 :
    ...
    k += 2
```

```
k = 100
while k > 0 :
    ...
    k += -1
```

```
for k in range(100) :
    ...
```

k = 0, 1, 2, ..., 99

```
for k in range(5,100,1) :
    ...
```

k = 5, 6, 7, ..., 99

```
for k in range(4,100,2) :
    ...
```

k = 4, 6, 8, ..., 98

```
for k in range(100, 0,-1) :
    ...
```

k = 100, 99, 98, ..., 1

range(100) → เริ่มต้นตั้งแต่ 0 ไปถึงน้อยกว่า 100

range(5,100,1) → เริ่มต้นตั้งแต่ 5 ไปถึงน้อยกว่า 100 โดยเพิ่มทีละ 1

range(4,100,2) → เริ่มต้นตั้งแต่ 4 ไปถึงน้อยกว่า 100 โดยเพิ่มทีละ 2

range(100,0,-1) → เริ่มต้นตั้งแต่ 100 ไปถึงมากกว่า 0 โดยลดทีละ 1



for-loops conditions with *continue* & *break*

```
1 for num in range(0,5,1):  
2     if num % 2 == 0 :  
3         print("Even Number", num)  
4     else:
```

continue continue ใช้สำหรับต้องการข้ามไปรันรอบใหม่ โดยไม่รันบรรทัดด้านล่าง
print("Odd Number", num) เมื่อมีคำสั่ง continue อยู่ด้านบน บรรทัดนี้จะไม่รัน

```
>>> %Run Test.py
```

```
Even Number 0  
Even Number 2  
Even Number 4
```

0 1 2 3 4

Thank You

<https://www.robotsiam.com/article/77/%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-micropython-%E0%B8%81%E0%B8%B1%E0%B8%9A-esp32-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-upycraft-ide>

