

# WECON

# Programming



**WECON Technology Co., Ltd.**

Website: <http://www.we-con.com.cn/en>

Technical Support: [chengxf@we-con.com.cn](mailto:chengxf@we-con.com.cn)

Skype: Jason.chen842

Phone: 86-591-87868869

# FOR NEXT Instructions

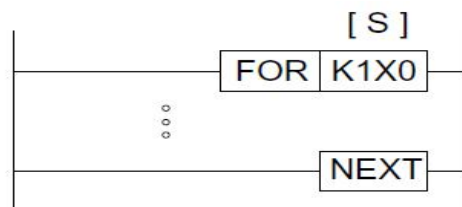
## 1. Instruction Description

Mnemonic	Function	Operands	Program steps
		S	
FOR (Start of a FOR-NEXT loop)	Identifies the start position and the number of repeats for the loop	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	FOR: 3 step
NEXT (End of a FOR-NEXT loop)	Identifies the end position for the loop	N/A Note: The FOR-NEXT loop can be nested for 5 levels, i.e. 5 FOR-NEXT loops are programmed within each other.	NEXT: 1 step

The FOR and NEXT instructions allow the specification of an area of program, i.e. the program enclosed by the instructions, which is to be repeated S number of times.

## 2.Operation:

NEXT Instruction is used to indicate the end portion of the cycle. FOR instruction designates the number of cycles of FOR~NEXT to be repeated. After the cycles have been completed, it will exit from the FOR~NEXT cycle and continue the operation.



## Points to note:

- The FOR instruction operates in a 16 bit mode hence, the value of the operand S may be within the range of 1 to 32,767. If a number between the range -32,768 and 0 (zero) is specified it is automatically replaced by the value 1, i.e. the FOR-NEXT loop would execute once.
- The NEXT instruction has NO operand.
- The FOR-NEXT instructions must be programmed as a pair e.g. for every FOR instruction there **MUST** be an associated NEXT instruction. The same applies to the NEXT instructions, there **MUST** be an associated FOR instruction. The FOR-NEXT instructions must also be programmed in the correct order. This means that programming a loop as a NEXT-FOR (the paired NEXT instruction proceeds the associated FOR instruction) is **NOT** allowed. Inserting an FEND instruction between the FOR-NEXT instructions, i.e. FOR-FEND- NEXT, is **NOT** allowed. This would have the same effect as programming a FOR without a NEXT instruction, followed by

the FEND instruction and a loop with a NEXT and no associated FOR instruction.

d) A FOR-NEXT loop operates for its set number of times **before** the main program is allowed to finish the current program scan.

e) When using FOR-NEXT loops care should be taken not to exceed the PLC's watchdog timer setting. The use of the WDT instruction and/or increasing the watchdog timer value is recommended.

### 3. Programming example

#### Nested FOR-NEXT loops:

FOR-NEXT instructions can be nested for 4 levels. This means that 4 FOR-NEXT loops can be sequentially programmed within each other.

In the example a 3 level nest has been programmed. As each new FOR-NEXT nest level is encountered the number of times that loop is repeated is increased by the multiplication of all of the surrounding/previous loops. For example, loop C operates 4 times. But within this loop there is a nested loop, B. For every completed cycle of loop C, loop B will be completely executed, i.e. it will loop 24 times. This again applies between loops B and A. The total number of times that loop A will operate for ONE scan of the program will equal;

- 1) The number of loop A operations multiplied by
- 2) The number of loop B operations multiplied by
- 3) The number of loop C operations

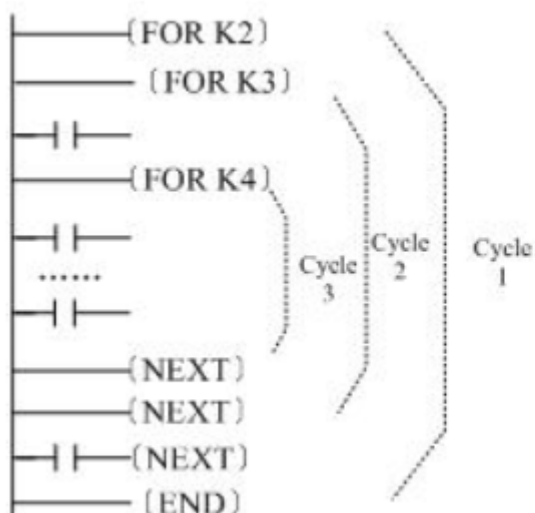
If values were associated to loops A, B and C, e.g. 7, 6 and 4 respectively, the following number of operations would take place in ONE program scan:

Number of loop C operations = 4 times

Number of loop B operations = 24 times ( $4 \times 6$ )

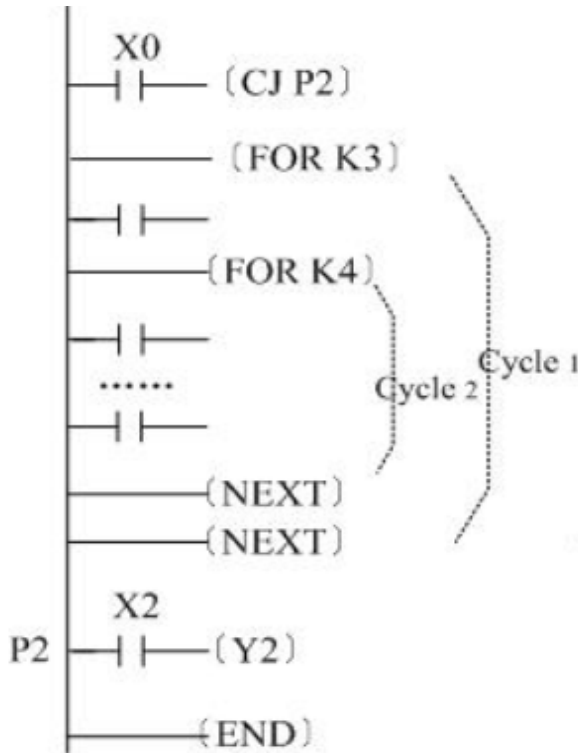
Number of loop A operations = 168 times ( $4 \times 6 \times 7$ )

#### Example 1 for instruction:



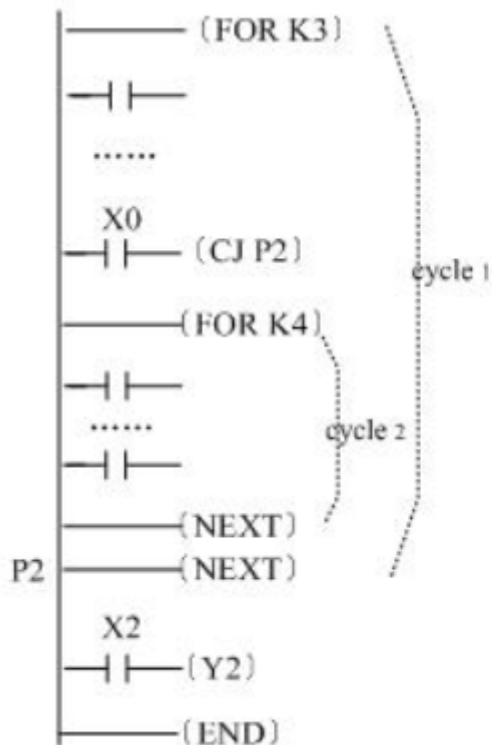
When cycle 1 is executed twice, it will continue to execute program after NEXT instruction. Every time cycle 1 is executed once, cycle 2 will be executed twice, while every time cycle is executed once, cycle 3 will be executed four times. Thus, cycle 3 will be executed for  $2 \times 3 \times 4 = 24$  times in total and cycle 2  $2 \times 3 = 6$  times.

#### Example 2 for instruction:



CJ instruction can be used to skip FOR~NEXT instruction. In the example, if X0=OFF, execute cycle 1 and 2; if X0=ON, executed cycle 1 and 2; when X0= ON, CJ instruction jumps to P2 and the program between cycle 1 and cycle 2 will not be executed.

**Example 3 for instruction:**



CJ instruction can also be used to skip inner nesting FOR~NEXT instruction. In the example, if X0=OFF, execute cycle 1 and 2; if X0=ON, CJ instruction jumps to P2 and cycle 2 nested inside cycle 1 will be skipped.

#### 4. PLC monitor

