The previous code has the following hex file:

: 02 0000 04 0000 FA
: 02 0000 00 00 52 8 D 1
: 08 000 8 000 4 008 3 16 05 1 00 6 1 4 1 F
: 08 00 1 0 00 8 8 1 2 0 6 0 8 8 5 0 0 0 9 2 8 8 F
: 02 40 0 E 00 F 1 3 F 8 0
: 00 0 0 0 0 0 0 1 F F

1) #include <PIC16F84A.inc>
    #of data = 2 bytes          offset = 0000          record type = 1>
    extended linear Address Record
    Information 0000          FA=1 chick sum.

2) 02
    / 0000
    / 00
    / 05 28
    / D 1

data=2 bytes
4 hexadecimal digits
address of byte "offset"
Data record

3) 08
    / 0008
    / 00
    / 04 00 8 3 1 6 0 5 1 0 0 6 1 4 1 F
0 data
8 bytes
offset
Data record

16 hexadecimal digit
4. 08 0010 00 83 20 60 88 50 09 28 8F

- data
- offset
- record
- data

- 5 byte
- 16 hexadecimal digits

5. 02 400E 00 F1 3F 80

- data record
- offset
- 2 byte
- 4 hexadecimal digits

- hex digits

- $\frac{400E}{2} = 2007$
- address of configuration word

6. 00 0000 01 FF

- end of file record
Note: In program memory, the instruction is 14 bit.

Maximum that can be stored is 3F
Delay loops

The principle of the delay is that we count down from a previously set number, and when it reaches zero, we stop counting. The zero value indicates the end of the delay, and we continue on our way through the program. The largest number we can have is 255 (FFH).

Example 1:

Delay

\[
\text{MOVFW } \text{count} \quad 1 \rightarrow \text{will be evaluated one time}
\]

\[
\text{Goto Loop} \quad 2 \times g
\]

Loop

\[
\text{NOP} \quad 1 \times 10 \rightarrow \text{will be evaluated ten times}
\]

\[
\text{DECFSZ count} \quad 1 \times g + 2 \leftarrow \text{decrease from } 9 \text{ to } 0 \text{ (1 cycle/g)}
\]

a skip will happen (2 cycles)
Total delay = 1 + 1 + 10 + 9 + 2 + 18
= 41 cycle

1 cycle = \( \frac{4}{\text{T}_{\text{osc}}} = \frac{4}{\frac{P_{\text{osc}}}{f_{\text{osc}}}} = \frac{4}{4\text{ MHz}} = 1\text{ u/sec} \)

Total delay = 41 u/sec

**Example 2**

**Delay**

```
  movlw 255
  movwf count1

Loop1
  nop
  movlw 255
  movwf count2

Loop0
  nop
  decfsz count2
  goto loop0
  decfsz count1
  goto loop1
```

inner Loop

1 x 255 = 255
254 \( \stackrel{2}{\rightleftharpoons} 254 \times 1 \)
2

2

Skip [2]
(2 cycles)

254 \( \times 2 \)
Inner loop

Loop 2

Nop \rightarrow 1 \times 255

DecFSZ count2 \rightarrow 1 \times 254 + 2

Goto Loop 2 \rightarrow 254 \times 2

Total cycle for inner loop =

255 + 254 + 2 + 254 \times 2 = 1019 \text{ cycle}

Delay

movlw .255 \rightarrow 1

movwf count2 \rightarrow 1

Loop 1

Nop \rightarrow 1 \times 255

movlw .255 \rightarrow 1 \times 255

movwf count2 \rightarrow 1 \times 255

[Inner loop]

(1019) \times 255

DecFSZ count1 \rightarrow \text{dec. from 254 - 1 (1 \times 254)}

When dec. from 1 - 0 there will be a skip

(2 \text{ cycle})

Goto Loop 1 \rightarrow 2 \times 254

Total delay = 1 + 1 + 255 + 255 + (1019) \times 255 + 255 + 254 + 2 + 2 \times 254 = 261376 \text{ cycle}

= 0.26 \text{ sec}
Example 3

Write a PIC subroutine in assembly language to give a fixed delay of 200 uS. Assume that a 4 MHz oscillator is being used.

Delay - 200

movlw k
movwf count

D-loop
decfsz count, f

goto D-loop

return

Total cycles = 2 + 1 + 1 + 2 + (k-1) + 2*(k-1)
= (3k + 3) cycles

(3k + 3) * 5 = 200 uS

3k = 200 - 3 = 197

k = 65