Grammar Notes

Function	Sample	Details
I2C Operations		
read byte	***readbyte*** 01h D1h	***readbyte*** : key word to read byte value ***readword:*** key word to read word value 01h: slave address (hexadeciaml) D1h: register address (hexadecimal) 00h: user expect value is 00h (use for verification)(hex) ***to:*** keyword means to save the readback value x: a variable named "x" to save the real result
	readbyte 01h D1h ***to*** x	
	readbyte 01h D1h 00h	
	readbyte 01h D1h 00h ***to*** x	
read word	***readword*** 01h D1h	
	readword 01h D1h ***to*** x	
	readword 01h D1h 01EFh	
	readword 01h D1h 01EFh ***to*** x	
write byte	***writebyte*** 01h D1h 01h	01h: byte value need to be written in
write word	***writeword*** 01h D1h 01EFh	01EFh: word value need to be written in
send	***send*** 01h D1h	send command
delay	***delay*** 50	delay 5000 us (5ms)
Operator		
assign value	***var*** x **=** 1	***var*** : keyword which means to define and set value to a variable x: a variable named "x" to save the result
Add	***var*** x = 1 **+** 2	
Minus	***var*** x = 2 **-** 1	
Multiply	***var*** x = 2 ***** 3	
Divide	***var*** x = 3 **/** 2	
log with e as the base	***var*** x = **ln(**4**)**	
Power	***var*** x = 3 **^** 2	
AND	***var*** x = 01h **&** Efh	
OR	***var*** x = 01h **|** Efh	
parentheses	***var*** x = **(**3-1**)***1+12/ln**(**3^2**)**	Increase internal computing priorities
Comparision		
greater than	***if*** x **>** 3	x: operand, a variable named "x"
less than	***if*** x **<** 3	3: operand (decimal)
greater than or equal	***if*** x **>=** 3	***if*** : keyword in conditional statement
less than or equal	***if*** x **<=** 3	
equal	***if*** x **==** 3	
Statement		

condition (Support nesting)	***if*** x > 3
	var x= 1
	else
	var x= 2
	endif
		if-else-endif
		or
		if-endif

looping (Support nesting)	***loop*** 3 ***readbyte*** 01h D1h A1h ***to*** x ***writebyte*** 01h D1h x ***endloop***	***loop-endloop*** 3: loop time
Others		
variable definition	***var*** x = 1	
save variables to local path	***savepath*** C:\MyFolder\save.txt ***var*** x1 = 1 ***var*** x2 = 2 ***save*** x1 x2	***savepath*** : keyword to specify the save path ***save*** : keyword to save the following variables
value format	***var*** x = 01h	01h: ends with "h" means hexadecimal
	var x = 10	10: decimal
comment	***#*** this is a comment	***#*** : comments needs to start with ''#"

Sample 1

```
readword 01h D1h 00FFh
savepath C:\MyFolder\save.txt
var x = 1
var value = 12h
var test = 5.1

# this is a test
loop 3
   readbyte 01h D1h A1h to value
   var value = value & 7Fh | 80h        #D[7] = 1
   writebyte 01h D1h value
   send 01h D1h                         #this is a test

   loop 2
      writebyte 01h D1h x
      save value x
      var x = x + 1
   endloop

   delay 100
endloop
```

Sample 2

```
var x = 1
var a = (3-1)*1+12/6     # a = 4
var value = 0

loop a
   if x > 3
      var value= 1
   else
      if x < 2
         var value = 2
      else
         var value = 3
      endif
   endif
endloop
```