```
'PICMB876 microcomputer boad Servo sample Program.(For Servo motor PWM Control)

'Program Name  SERVO.BAS
'CopyRights    (C)2003,Japan Tech.Hanzougane Yoshiaki Morohashi All Rights Reserved.
'Date          2003.07.25
'Language      microEngineering Labs,Inc. PICBasic Pro Compiler
'Target Device PIC16F876 on PICMB876(PIC microcomputer boad)
'Input         Differential Input(Servo Sens.) instrument OP-AMP(AD624 Analog Devices)
'Output        PWM Output(Servo Motor) PORTC.2 (0-5V @1KHz PWM)


        Include "modedefs.bas"


'LCD DEFINE
        DEFINE LCD_DREG     PORTB
        DEFINE LCD_DBIT     0
        DEFINE LCD_RSREG    PORTB
        DEFINE LCD_RSBIT    4
        DEFINE LCD_EREG     PORTB
        DEFINE LCD_EBIT     5
        DEFINE LCD_BITS     4
        DEFINE LCD_LINES    2

'HPWM DEFINE
        DEFINE CCP1_REG PORTC       'HPWM 1 Pin PORT
        DEFINE CCP1_BIT 2           'HPWM 1 Pin bit
        DEFINE CCP2_REG PORTC       'HPWM 2 Pin PORT
        DEFINE CCP2_BIT 1           'HPWM 2 Pin bit

        DEFINE HPWM1_TIMER 1        'HPWM 1 timer select

'A/D  DFINE
        DEFINE   ADC_BITS 10
        DEFINE   ADC_CLOCK 2
        DEFINE   ADC_SAMPLEUS 50


'Variable define

        PWMC    Var  byte
        PWMC    =    0

        ADRH   Var  byte
        ADRL   Var  byte
        w0     Var  word
        wh     Var  word
        AD8    Var  byte
        ADRH   =    0
        ADRL   =    0
        w0     =    0
        wh     =    0
        AD8    =    0


        GOTO       main


PWMsub: HPWM 1, PWMC,1000                   'Output PWM on PORTC.2 at 1KHz
        RETURN
```

```
main:   ADRH=0:ADRL=0:wh=0:w0=0
'Differential Input(Servo Sens.) : AD624 AMP ×1, CON2(AMP-IN 23pin , AMP+IN 25pin)
'                                  on PICMB876 microcomputer boad
'
        TRISA  = %00011111                  '%0001 1111(a/d input AN0)
        ADCON1 = %10000101                  '%1000 0101
        ADCIN  0,wh                         'ADC conversion START

        ADRH = ADRESH                       'A/D result move ADRH
        ADRL = ADRESL                       'A/D result move ADRL

        w0   = ADRH * 256 + ADRL
        IF w0 < 0 Then
                   LCDOUT $FE, 1, "**A/D Error**"
                   PAUSE 2000
                   GOTO main
                Else
                   GOTO cnt
                Endif

cnt:    AD8  = w0 >> 2                  '10bits --->8bits
        PWMC = AD8                      'PWMC(0 - 255 ,Duty cycle0% - 100%)
        IF PWMC <= 10 Then GOTO servos
        GOSUB PWMsub                    'PWM subroutine:Output PORTC.2(Servo Mortor)
        LCDOUT $FE, 1, "**Servo    **"
        PAUSE 200
        LCDOUT $FE, 1, "**Servo.   **"
        PAUSE 200
        LCDOUT $FE, 1, "**Servo..  **"
        PAUSE 200
        LCDOUT $FE, 1, "**Servo... **"
        PAUSE 200
        LCDOUT $FE, 1, "**Servo....**"
        PAUSE 200
        GOTO  main

servos: LCDOUT $FE, 1, "**Servo end**"
        PAUSE 500
        LCDOUT $FE, 1
        PAUSE 500
        LCDOUT $FE, 1, "**Servo end**"
        PAUSE 500

        GOTO main
```